To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

Notice

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

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

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

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

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

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

User's Manual

32

H8SX/1655 Group, H8SX/1655M Group

Hardware Manual

Renesas 32-Bit CISC Microcomputer

H8SX Family / H8SX/1600 Series

H8SX/1655	R5F61655
H8SX/1652	R5F61652
H8SX/1655M	R5F61655M
H8SX/1652M	R5F61652M

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 2009.10

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 v
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/1655 Group and the H8
Group. Before using any of the documents, please visit our web site to verify that yo
most up-to-date available 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/1655 Group, H8SX/1655M Group Hardware Manual	This r
Software Manual	Detailed descriptions of the CPU and instruction set	H8SX Family Software Manual	REJ0
Application Note	Examples of applications and sample programs	The latest versions are available fr web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

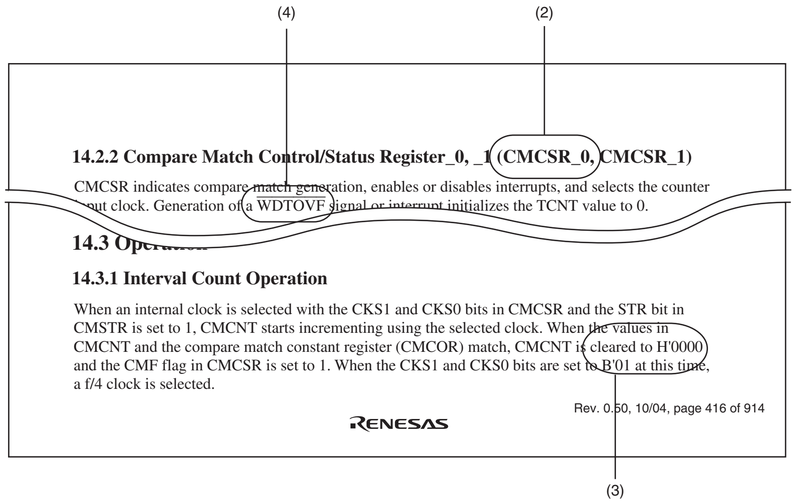
Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.

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



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 communications interface
TMR	8-bit timer
TPU	16-bit timer pulse unit
WDT	Watchdog timer

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communications 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

Section 14 16-Bit Timer Pulse Unit (TPU) ....

RENESAS

RENESAS

controller, which enable high-speed data transfer, and a bus-state controller, which enab connection to different kinds of memory. The LSI of the Group also includes serial communication interfaces, A/D and D/A converters, and a multi-function timer that mak control easy. Together, the modules realize low-cost configurations for end systems. The consumption of these modules is kept down dynamically by an on-chip power-managem function. The on-chip ROM is a flash memory (F-ZTAT™*) with a capacity of 512 Kby (H8SX/1655 and H8SX/1655M) or 384 Kbytes (H8SX/1652 and H8SX/1652M).

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

1.1.1 Applications

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

CPU	CPU	• 32-bit high-speed H8SX CPU (CISC type) Upwardly compatible for H8/300, H8/300H, and H8S CP object level • General-register architecture (sixteen 16-bit general reg • 11 addressing modes • 4-Gbyte address space Program: 4 Gbytes available Data: 4 Gbytes available • 87 basic instructions, classifiable as bit arithmetic and lo instructions, multiply and divide instructions, bit manipul instructions, multiply-and-accumulate instructions, and o • Minimum instruction execution time: 20.0 ns (for an ADD instruction while system clock Iϕ = 50 MHz and V_{cc} = 3.0 to 3.6 V) • On-chip multiplier ($16 \times 16 \rightarrow 32$ bits) • Supports multiply-and-accumulate instructions ($16 \times 16 + 42 \rightarrow 42$ bits)
	Operating mode	• Advanced mode Normal, middle, or maximum mode is not supported.

		Mode 4: On-chip ROM disabled external extended mode, bus (selected by driving the MD1 and MD0 pins lo driving the MD2 pin high) Mode 5: On-chip ROM disabled external extended mode, (selected by driving the MD1 pin low and driving t and MD0 pins high) Mode 6: On-chip ROM enabled external extended mode (selected by driving the MD0 pin low and driving t and MD1 pins high) Mode 7: Single-chip mode (can be externally extended) (selected by driving the MD2, MD1, and MD0 pins • Low power consumption state (transition driven by the instruction)
Power on reset (POR)*		• At power-on or low power supply voltage, an internal re signal is generated
Voltage detection circuit (LVD)*		• At low power supply voltage, an internal reset signal ar interrupt are generated
Interrupt (source)	Interrupt controller (INTC)	• 13 external interrupt pins (NMI, and $\overline{\text{IRQ11}}$ to $\overline{\text{IRQ0}}$) • Internal interrupt sources H8SX/1655 Group: 119 pins H8SX/1655M Group: 120 pins • 2 interrupt control modes (specified by the interrupt con register) • 8 priority orders specifiable (by setting the interrupt prio register) • Independent vector addresses

		• Extended repeat-area function
	DMA controller (DMAC)	• 4-channel DMA transfer available • 3 activation methods (auto-request, on-chip module inte external request) • 3 transfer modes (normal transfer, repeat transfer, block transfer) • Dual or single address mode selectable • Extended repeat-area function
	Data transfer controller (DTC)	• Allows DMA transfer over 78 channels (number of DTC activation sources) • Activated by interrupt sources (chain transfer enabled) • 3 transfer modes (normal transfer, repeat transfer, block transfer mode) • 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 8 areas of which is independently controllable — Chip-select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output — Access in 2 or 3 states can be selected for each are — 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 amor internal CPU, DMAC, EXDMAC, and DTC, and external 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 • 5 low-power-consumption modes: Sleep mode, all-moc clock-stop mode, software standby mode, deep softwa standby mode, and hardware standby mode

		Unit 0: Software, timer (TPU (unit 0) /TMR (units 0 and 1 trigger, and external trigger Unit 1: Software, TMR (units 2 and 3) trigger, and extern trigger • Activation of DTC and DMAC by ADI interrupt: Unit 0: DTC and DMAC can be activated by an ADI inte Unit 1: DMAC can be activated by an ADI1 interrupt.
D/A converter	D/A converter (DAC)	• 10-bit resolution × 2 output channels • Output voltage: 0 V to Vref, maximum conversion time: (with 20-pF load)
Timer	8-bit timer (TMR)	• 8 bits × 8 channels (can be used as 16 bits × four chann • Select from among 7 clock sources (6 internal clocks an external clock) • Allows the output of pulse trains with a desired duty cyc PWM signals
	16-bit timer pulse unit (TPU)	• 16 bits × 12 channels (unit 0, unit 1*) • Select from among 8 counter-input clocks for each chan • Up to 24 pulse inputs and outputs • Counter clear operation, simultaneous writing to multiple counters (TCNT), simultaneous clearing by compare ma input capture possible, simultaneous input/output for reg possible by counter synchronous operation, and up to 1 PWM output possible by combination with synchronous operation • Buffered operation, cascaded operation (32 bits × two channels), and phase counting mode (two-phase encod input) settable for each channel • Input capture function supported • Output compare function (by the output of compare mat waveform) supported Note: * Pin function of unit 1 cannot be used in the exte extended mode.

Watchdog timer	Watchdog timer (WDT)	• 8 bits × one channel (selectable from eight counter inp clocks) • Switchable between watchdog timer mode and interva mode
Serial interface	Serial communi-cations interface (SCI)	• 6 channels (select asynchronous or clock synchronous communications mode) • Full-duplex communications capability • Select the desired bit rate and LSB-first or MSB-first tra • Average transfer rate clock input from TMR (SCI_5, SC • IrDA transmission and reception conformant with the Ir Specifications version 1.0 • On-chip cyclic redundancy check (CRC) calculator for improved reliability in data transfer
Smart card/SIM		• The SCI module supports a smart card (SIM) interface
I²C bus interface	I²C bus interface 2 (IIC2)	• 2 channels • Bus can be directly driven (the SCL and SDA pins are open drains).
Universal serial bus interface	Universal serial bus interface (USB)	• On-chip UDC (USB Device Controller) supporting USB transceiver • Transfer speed: full-speed (12 Mbps) • Bulk transfer by DMA • Self-power mode and bus power mode selectable
I/O ports		• 9 CMOS input-only pins • 75 CMOS input/output pins • 8 large-current drive pins (port 3) • 40 pull-up resistors • 16 open drains
Package		• LQFP-120 package • LGA-145 package

	• Flash programming/erasure voltage: 3.0 to 3.6 V • Supply current: — 50 mA (typ.) (V_{cc} = $PLLV_{cc}$ = DrV_{cc} = 3.0 V, AV_{cc} = 3 = Pϕ = Bϕ = 35 MHz)
Operating peripheral temperature (°C)	• –20 to +75°C (regular specifications) • –40 to +85°C (wide-range specifications)

Note: * Supported only by the H8SX/1655M Group.

Table 1.2 Comparison of Support Functions in the H8SX/1655 and H8SX/1655M

Function		H8SX/1655 Group	H8SX/1655M Group
DMAC		O	O
DTC		O	O
PPG		O	O
UBC		O	O
SCI		O	O
IIC2		O	O
TMR		O	O
WDT		O	O
10-bit ADC		O	O
10-bit DAC		O	O
EXDMAC		O	O
POR/LVD		—	O
Package	LQFP-120	O	O
	LGA-145	O	O

	R5F61652N50LGV	384 Kbytes	40 Kbytes	LGA-145	
	R5F61655D50FPV	512 Kbytes	40 Kbytes	LQFP-120	Wi sp
	R5F61652D50FPV	384 Kbytes	40 Kbytes	LQFP-120	
	R5F61655D50LGV	512 Kbytes	40 Kbytes	LGA-145	
	R5F61652D50LGV	384 Kbytes	40 Kbytes	LGA-145	
H8SX/1655M	R5F61655MN50FPV	512 Kbytes	40 Kbytes	LQFP-120	Re sp
	R5F61652MN50FPV	384 Kbytes	40 Kbytes	LQFP-120	
	R5F61655MN50LGV	512 Kbytes	40 Kbytes	LGA-145	
	R5F61652MN50LGV	384 Kbytes	40 Kbytes	LGA-145	
	R5F61655MD50FPV	512 Kbytes	40 Kbytes	LQFP-120	Wi sp
	R5F61652MD50FPV	384 Kbytes	40 Kbytes	LQFP-120	
	R5F61655MD50LGV	512 Kbytes	40 Kbytes	LGA-145	
	R5F61652MD50LGV	384 Kbytes	40 Kbytes	LGA-145	

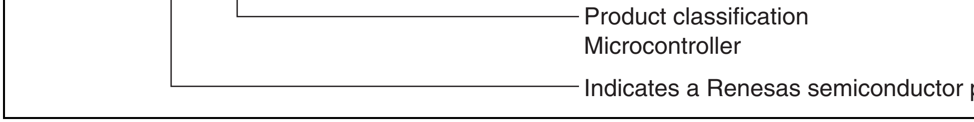
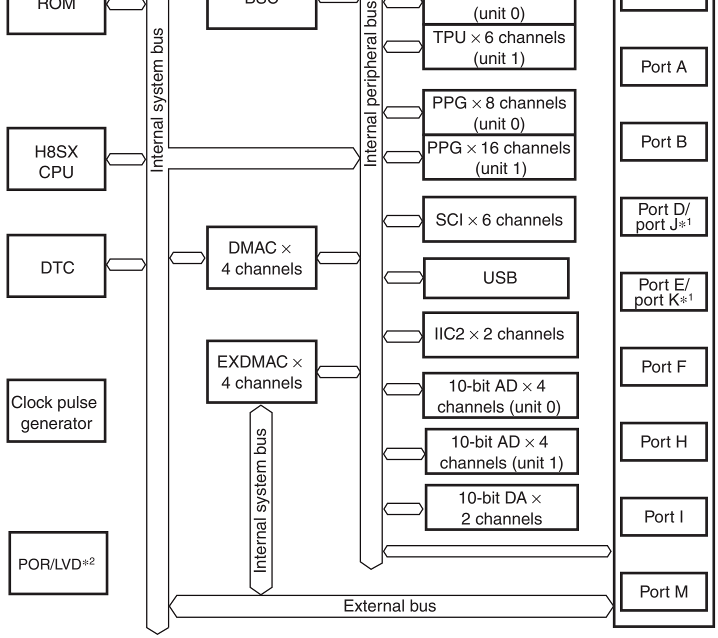


Figure 1.1 How to Read the Product Name Code



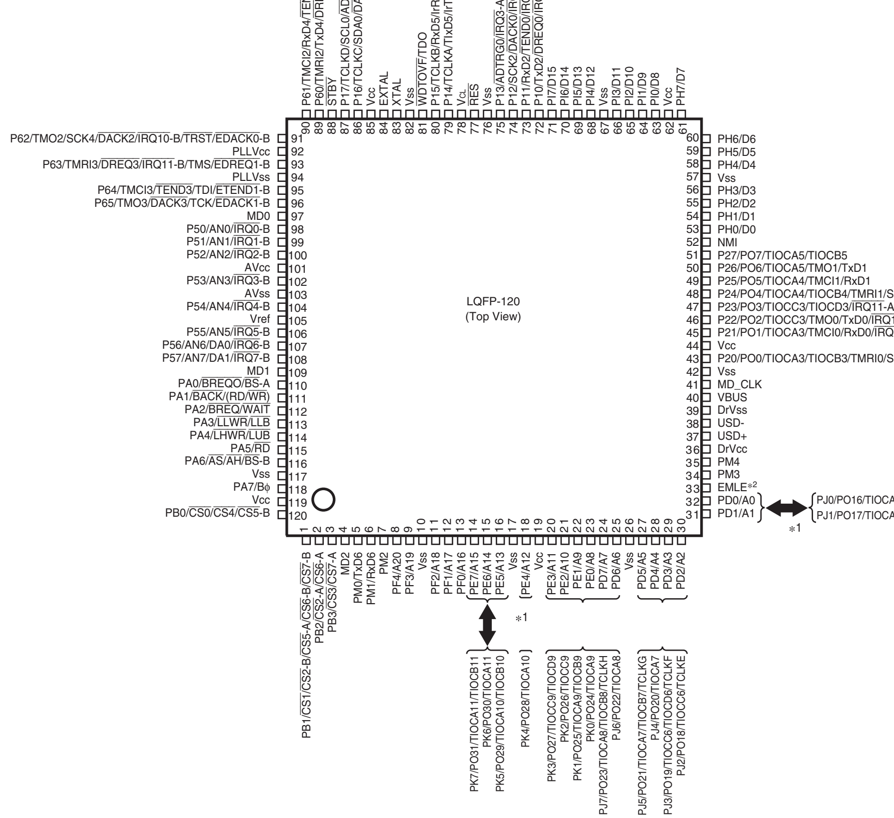
[Legend]

CPU:	Central processing unit	TMR:	8-bit timer
DTC:	Data transfer controller	TPU:	16-bit timer pulse unit
BSC:	Bus controller	PPG:	Programmable pulse generator
DMAC:	DMA controller	SCI:	Serial communications interface
EXDMAC:	EXDMA controller	USB:	Universal serial bus interface
WDT:	Watchdog timer	IIC2:	IIC bus interface 2
		POR/LVD*2:	Power-on reset / Low voltage detection circuit

Notes: 1. In single-chip mode, the port D and port E functions can be used in the initial state. Pin functions are selectable by setting the PCJKE bit in PFCRD. Ports D and E are enabled when PCJKE = 0 (initial value) and ports J and K are enabled when PCJKE = 1. In external extended mode, only ports D and E can be used.

2. Supported only by the H8SX/1655M Group.

Figure 1.2 Block Diagram



Notes: 1. In single-chip mode ports D and E can be used (initial state).
Pin functions are selectable by setting the PCJKE bit in PFCRD. Ports D and E are enabled when PCJKE = 0 (initial value) and ports J and K are enabled when PCJKE = 1. In external extended mode, only ports D and E can be used.
2. This pin is an on-chip emulator enable pin. Drive this pin low for the connection in normal operating mode.
The on-chip emulator function is enabled by driving this pin high. When the on-chip emulator is in use, the P62, P63, P64, P65, and WDTOVF pins are dedicated pins for the on-chip emulator. For details on a connection example with the E10A, see E10A Emulator User's Manual.

Figure 1.3 Pin Assignments (LQFP-120: H8SX/1655 Group and H8SX/1655M

	1	2	3	4	5	6	7	8	9	10	11	12
G	PF1	PE6 /PK6*1	PE7 /PK7*1	Vss	LGA-145 (Perspective top view)					P14	P13	Vss
H	PE5 /PK5*1	PE4 /PK4*1	PE3 /PK3*1	PF0						V_{CL}	P11	NC*3
J	Vcc	PE1 /PK1*1	PD7 /PJ7*1	PE2 /PK2*1						PI7	P10	P12
K	PE0 /PK0*1	PD5 /PJ5*1	PD4 /PJ4*1	PD6 /PJ6*1	Vcc	P21	P25	P26	NC*3	PH2	PI4	PI5
L	PD3 /PJ3*1	Vss	EMLE*2	VBUS	MD_CLK	P24	NC*3	NC*3	NMI	PH1	PI2	PI3
M	PD2 /PJ2*1	PD1 /PJ1*1	DrVcc	USD-	Vss	P20	P23	NC*3	PH0	Vss	PH4	PH7
N	PD0 /PJ0*1	PM3	PM4	USD+	DrVss	P22	NC*3	P27	Vcc	PH3	PH5	PH6

Notes: 1. In single-chip mode ports D and E can be used (initial state).
Pin functions are selectable by setting the PCJKE bit in PFCRD.
Ports D and E are enabled when PCJKE = 0 (initial value)
and ports J and K are enabled when PCJKE = 1.

In external extended mode, only ports D and E can be used.

2. This pin is an on-chip emulator enable pin. Drive this pin low for the connection in normal operati
The on-chip emulator function is enabled by driving this pin high. When the on-chip emulator is in
the P62, P63, P64, P65, and $\overline{WDTOVF}$ pins are dedicated pins for the on-chip emulator.
For details on a connection example with the E10A, see E10A Emulator User's Manual.

3. Leave NC pin open.

Figure 1.4 Pin Assignments (LGA-145: H8SX/1655 Group and H8SX/1655M G

4	D1	MD2	MD2	MD2
5	D3	PM0/TxD6	PM0/TxD6	PM0/TxD6
6	D2	PM1/RxD6	PM1/RxD6	PM1/RxD6
7	E1	PM2	PM2	PM2
8	F1	PF4/A20	PF4/A20	PF4/A20
9	F4	PF3/A19	PF3/A19	PF3/A19
10	F3	Vss	Vss	Vss
11	F2	PF2/A18	PF2/A18	PF2/A18
12	G1	PF1/A17	PF1/A17	PF1/A17
13	H4	PF0/A16	PF0/A16	PF0/A16
14	G3	PE7/A15	• PE7/A15 • PK7/PO31/TIOCA11/TIOCB11*[1]	A15
15	G2	PE6/A14	• PE6/A14 • PK6/PO30/TIOCA11*[1]	A14
16	H1	PE5/A13	• PE5/A13 • PK5/PO29/TIOCA10/TIOCB10*[1]	A13
17	G4	Vss	Vss	Vss
18	H2	PE4/A12	• PE4/A12 • PK4/PO28/TIOCA10*[1]	A12
19	J1	Vcc	Vcc	Vcc
20	H3	PE3/A11	• PE3/A11 • PK3/PO27/TIOCC9/TIOCD9*[1]	A11
21	J4	PE2/A10	• PE2/A10 • PK2/PO26/TIOCC9*[1]	A10

25	K4	PD6/A6	• PD6/A6 • PJ6/PO22/TIOCA8*[1]	A6
26	L2	Vss	Vss	Vss
27	K2	PD5/A5	• PD5/A5 • PJ5/PO21/TIOCA7/TIOCB7/TCLKG*[1]	A5
28	K3	PD4/A4	• PD4/A4 • PJ4/PO20/TIOCA7*[1]	A4
29	L1	PD3/A3	• PD3/A3 • PJ3/PO19/TIOCC6/TIOCD6/TCLKF*[1]	A3
30	M1	PD2/A2	• PD2/A2 • PJ2/PO18/TIOCC6/TCLKE*[1]	A2
31	M2	PD1/A1	• PD1/A1 • PJ1/PO17/TIOCA6/TIOCB6*[1]	A1
32	N1	PD0/A0	• PD0/A0 • PJ0/PO16/TIOCA6*[1]	A0
33	L3	EMLE	EMLE	EMLE
34	N2	PM3	PM3	PM3
35	N3	PM4	PM4	PM4
36	M3	DrVcc	DrVcc	DrVcc
37	N4	USD+	USD+	USD+
38	M4	USD–	USD–	USD–
39	N5	DrVss	DrVss	DrVss
40	L4	VBUS	VBUS	VBUS

		TxD0/$\overline{\text{IRQ10}}$-A	$\overline{\text{IRQ10}}$-A	$\overline{\text{IRQ10}}$-A
47	M7	P23/PO3/TIOCC3/TIOCD3/ $\overline{\text{IRQ11}}$-A	P23/PO3/TIOCC3/TIOCD3/ $\overline{\text{IRQ11}}$-A	P23/PO3/TIOCC3/TI $\overline{\text{IRQ11}}$-A
48	L6	P24/PO4/TIOCA4/TIOCB4/ TMRI1/SCK1	P24/PO4/TIOCA4/TIOCB4/ TMRI1/SCK1	P24/PO4/TIOCA4/TI TMRI1/SCK1
49	K7	P25/PO5/TIOCA4/TMCI1/ RxD1	P25/PO5/TIOCA4/TMCI1/ RxD1	P25/PO5/TIOCA4/TM RxD1
50	K8	P26/PO6/TIOCA5/TMO1/ TxD1	P26/PO6/TIOCA5/TMO1/ TxD1	P26/PO6/TIOCA5/TM TxD1
51	N8	P27/PO7/TIOCA5/TIOCB5	P27/PO7/TIOCA5/TIOCB5	P27/PO7/TIOCA5/TI
52	L9	NMI	NMI	NMI
53	M9	PH0/D0	PH0/D0	D0
54	L10	PH1/D1	PH1/D1	D1
55	K10	PH2/D2	PH2/D2	D2
56	N10	PH3/D3	PH3/D3	D3
57	M10	Vss	Vss	Vss
58	M11	PH4/D4	PH4/D4	D4
59	N11	PH5/D5	PH5/D5	D5
60	N12	PH6/D6	PH6/D6	D6
61	M12	PH7/D7	PH7/D7	D7
62	N9	Vcc	Vcc	Vcc
63	M13	PI0/D8	PI0/D8	PI0/D8
64	L13	PI1/D9	PI1/D9	PI1/D9
65	L11	PI2/D10	PI2/D10	PI2/D10
66	L12	PI3/D11	PI3/D11	PI3/D11

73	H11	P11/RxD2/TEND0/IRQ1-A/ETEND0-A	P11/RxD2/TEND0/IRQ1-A/ETEND0-A	P11/RxD2/TEND0/IRQ ETEND0-A
74	J12	P12/SCK2/DACK0/IRQ2-A/EDACK0-A	P12/SCK2/DACK0/IRQ2-A/EDACK0-A	P12/SCK2/DACK0/IRQ EDACK0-A
75	G11	P13/ADTRG0/IRQ3-A/EDRAK0	P13/ADTRG0/IRQ3-A/EDRAK0	P13/ADTRG0/IRQ3-A/
76	G12	Vss	Vss	Vss
77	G13	RES	RES	RES
78	H10	V_{CL}	V_{CL}	V_{CL}
79	G10	P14/TCLKA/TxD5/IrTxD/SDA1/DREQ1/IRQ4-A/EDREQ1-A	P14/TCLKA/TxD5/IrTxD/SDA1/DREQ1/IRQ4-A/EDREQ1-A	P14/TCLKA/TxD5/IrTx SDA1/DREQ1/IRQ4-A EDREQ1-A
80	F11	P15/TCLKB/RxD5/IrRxD/SCL1/TEND1/IRQ5-A/ETEND1-A	P15/TCLKB/RxD5/IrRxD/SCL1/TEND1/IRQ5-A/ETEND1-A	P15/TCLKB/RxD5/IrRx SCL1/TEND1/IRQ5-A/ ETEND1-A
81	F13	WDTOVF	WDTOVF/TDO*2	WDTOVF
82	F12	Vss	Vss	Vss
83	E12	XTAL	XTAL	XTAL
84	E13	EXTAL	EXTAL	EXTAL
85	F10	Vcc	Vcc	Vcc
86	E11	P16/TCLKC/SDA0/DACK1/IRQ6-A/EDACK1-A	P16/TCLKC/SDA0/DACK1/IRQ6-A/EDACK1-A	P16/TCLKC/SDA0/ DACK1/IRQ6-A/EDAC
87	E10	P17/TCLKD/SCL0/ADTRG1/IRQ7-A/EDRAK1	P17/TCLKD/SCL0/ADTRG1/IRQ7-A/EDRAK1	P17/TCLKD/SCL0/ ADTRG1/IRQ7-A/EDR
88	D13	STBY	STBY	STBY
89	B13	P60/TMRI2/TxD4/DREQ2/IRQ8-B/EDREQ0-B	P60/TMRI2/TxD4/DREQ2/IRQ8-B/EDREQ0-B	P60/TMRI2/TxD4/DRE IRQ8-B/EDREQ0-B

95	B11	P64/TMCI3/TEND3/ ETEND1-B	P64/TMCI3/TEND3/TDI*2/ ETEND1-B	P64/TMCI3/TEND3/E
96	A10	P65/TMO3/DACK3/ EDACK1-B	P65/TMO3/DACK3/TCK*2/ EDACK1-B	P65/TMO3/DACK3/E
97	D10	MD0	MD0	MD0
98	A9	P50/AN0/IRQ0-B	P50/AN0/IRQ0-B	P50/AN0/IRQ0-B
99	C9	P51/AN1/IRQ1-B	P51/AN1/IRQ1-B	P51/AN1/IRQ1-B
100	B9	P52/AN2/IRQ2-B	P52/AN2/IRQ2-B	P52/AN2/IRQ2-B
101	B8	AVcc	AVcc	AVcc
102	D9	P53/AN3/IRQ3-B	P53/AN3/IRQ3-B	P53/AN3/IRQ3-B
103	C8	AVss	AVss	AVss
104	B7	P54/AN4/IRQ4-B	P54/AN4/IRQ4-B	P54/AN4/IRQ4-B
105	A8	Vref	Vref	Vref
106	D8	P55/AN5/IRQ5-B	P55/AN5/IRQ5-B	P55/AN5/IRQ5-B
107	D7	P56/AN6/DA0/IRQ6-B	P56/AN6/DA0/IRQ6-B	P56/AN6/DA0/IRQ6-
108	A7	P57/AN7/DA1/IRQ7-B	P57/AN7/DA1/IRQ7-B	P57/AN7/DA1/IRQ7-
109	B6	MD1	MD1	MD1
110	D6	PA0/BREQO/BS-A	PA0/BREQO/BS-A	PA0/BREQO/BS-A
111	A5	PA1/BACK/(RD/WR)	PA1/BACK/(RD/WR)	PA1/BACK/(RD/WR)
112	B4	PA2/BREQ/WAIT	PA2/BREQ/WAIT	PA2/BREQ/WAIT
113	D5	PA3/LLWR/LLB	PA3/LLWR/LLB	LLWR/LLB
114	A4	PA4/LHWR/LUB	PA4/LHWR/LUB	PA4/LHWR/LUB
115	C5	PA5/RD	PA5/RD	RD
116	C4	PA6/AS/AH/BS-B	PA6/AS/AH/BS-B	PA6/AS/AH/BS-B
117	C3	Vss	Vss	Vss

Pin No. LGA-145: D4, N13		
Vcc	Vcc	Vcc

Notes: 1. These pins can be used when the PCJKE bit in PFCRD is set to 1 in single-ch

2. Pins TDO, $\overline{TRST}$, TMS, TDI, and TCK are enabled in mode 3.

	$PLLV_{ss}$	Input	Ground pin for the PLL circuit.
	DrV_{cc}	Input	Power supply pin for the transceiver with on-chip USB. C the system power supply.
	DrV_{ss}	Input	Ground pin for the transceiver with on-chip USB.
Clock	XTAL	Input	Pins for a crystal resonator. An external clock signal can through the EXTAL pin. For an example of this connectio section 26, Clock Pulse Generator.
	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 levels on must not be changed during operation.
	MD_CLK	Input	Pins for switching the multiplication ratio of the clock puls generator. The signal levels on these pins must not be ch during operation.
System control	$\overline{\text{RES}}$	Input	Reset signal input pin. This LSI enters the reset state wh signal goes low.
	$\overline{\text{STBY}}$	Input	This LSI enters hardware standby mode when this signal
	EMLE	Input	Input pin for the on-chip emulator enable signal. If the on emulator is used, the signal level should be fixed high. If emulator is not used, the signal level should be fixed low
On-chip emulator	$\overline{\text{TRST}}$	Input	On-chip emulator pins or boundary scan pins. When the is driven high, these pins are dedicated for the on-chip er When the EMLE pin is driven low and to mode 3, these p dedicated for the boundary scan.
	TMS	Input	
	TDI	Input	
	TCK	Input	
	TDO	Output	
Address bus	A20 to A0	Output	Output pins for the address bits.

BS-A/BS-B	Output	Indicates the start of a bus cycle.
AS	Output	Strobe signal which indicates that the output addr the address bus is valid in access to the basic bus or byte control SRAM interface space.
AH	Output	This signal is used to hold the address when acce address-data multiplexed I/O interface space.
RD	Output	Strobe signal which indicates that reading from th bus interface space is in progress.
RD/WR	Output	Indicates the direction (input or output) of the data
LHWR	Output	Strobe signal which indicates that the higher-orde (D15 to D8) is valid in access to the basic bus inte space.
LLWR	Output	Strobe signal which indicates that the lower-order to D0) is valid in access to the basic bus interface
LUB	Output	Strobe signal which indicates that the higher-orde (D15 to D8) is valid in access to the byte control S interface space.
LLB	Output	Strobe signal which indicates that the lower-order to D0) is valid in access to the byte control SRAM space.
CS0 CS1 CS2-A/CS2-B CS3 CS4 CS5-A/CS5-B CS6-A/CS6-B CS7-A/CS7-B	Output	Select signals for areas 0 to 7.
WAIT	Input	Requests wait cycles in access to the external spa

	IRQ2-A/IRQ2-B IRQ1-A/IRQ1-B IRQ0-A/IRQ0-B		
DMA controller (DMAC)	DREQ0-A/DREQ0-B DREQ1-A/DREQ1-B DREQ2 DREQ3	Input	Requests DMAC activation.
	DACK0-A/DACK0-B DACK1-A/DACK1-B DACK2 DACK3	Output	DMAC single address-transfer acknowledge sign
	TEND0-A/TEND0-B TEND1-A/TEND1-B TEND2 TEND3	Output	Indicates end of data transfer by the DMAC.
EXDMA controller (EXDMAC)	EDREQ0-A/EDREQ0-B EDREQ1-A/EDREQ1-B	Input	Requests EXDMAC activation.
	EDACK0-A/EDACK0-B EDACK1-A/EDACK1-B	Output	EXDMAC single address-transfer acknowledge s
	ETEND0-A/ETEND0-B ETEND1-A/ETEND1-B	Output	Indicates end of data transfer by the EXDMAC.
	EDRAK0 EDRAK1	Output	Notification to external device of EXDMAC reque acceptance and start of execution

			outputs.
	TIOCA5 TIOCB5	Input/ output	Signals for TGRA_5 and TGRB_5. These pins are input capture inputs, output compare outputs, or P outputs.
	TCLKE TCLKF TCLKG TCLKH	Input	Input pins for external clock signals.
	TIOCA6 TIOCB6 TIOCC6 TIOCD6	Input/ output	Signals for TGRA_6 to TGRD_6. These pins are u input capture inputs, output compare outputs, or P outputs.
	TIOCA7 TIOCB7	Input/ output	Signals for TGRA_7 and TGRB_7. These pins are input capture inputs, output compare outputs, or P outputs.
	TIOCA8 TIOCB8	Input/ output	Signals for TGRA_8 and TGRB_8. These pins are input capture inputs, output compare outputs, or P outputs.
	TIOCA9 TIOCB9 TIOCC9 TIOCD9	Input/ output	Signals for TGRA_9 to TGRD_9. These pins are u input capture inputs, output compare outputs, or P outputs.
	TIOCA10 TIOCB10	Input/ output	Signals for TGRA_10 and TGRB_10. These pins a as input capture inputs, output compare outputs, o outputs.
	TIOCA11 TIOCB11	Input/ output	Signals for TGRA_11 and TGRB_11. These pins a as input capture inputs, output compare outputs, o outputs.
Programmable pulse generator (PPG)	PO31 to PO16, PO7 to PO0	Output	Output pins for the pulse signals.

	TxD5 TxD6		
	RxD0 RxD1 RxD2 RxD4 RxD5 RxD6	Input	Input pins for data reception.
	SCK0 SCK1 SCK2 SCK4	Input/ output	Input/output pins for clock signals.
SCI with IrDA (SCI)	IrTxD	Output	Output pin that outputs encoded data for IrDA.
	IrRxD	Input	Input pin that inputs encoded data for IrDA.
I^2C bus interface 2 (IIC2)	SCL0, SCL1	Input/ output	Input/output pin for IIC clock. Bus can be directly the NMOS open drain output.
	SDA0, SDA1	Input/ output	Input/output pin for IIC data. Bus can be directly the NMOS open drain output.
Universal serial bus interface (USB)	USD+	Input/ output	Input/output pin for USB data.
	USD−		
	VBUS	Input	Input/output pin to connect/disconnect USB cable
A/D converter	AN7 to AN0	Input	Input pins for the analog signals to be processed converter.
	$\overline{\text{ADTRG0}}$, $\overline{\text{ADTRG1}}$	Input	Input pins for the external trigger signal that starts conversion.
D/A converter	DA1, DA0	Output	Output pins for the analog signals from the D/A c

P27 to P20	Input/ output	8-bit input/output pins.
P57 to P50	Input	8-bit input/output pins.
P65 to P60	Input/ output	6-bit input/output pins.
PA7	Input	Input-only pin
PA6 to PA0	Input/ output	7-bit input/output pins.
PB3 to PB0	Input/ output	4-bit input/output pins.
PD7 to PD0	Input/ output	8-bit input/output pins.
PE7 to PE0	Input/ output	8-bit input/output pins.
PF4 to PF0	Input/ output	5-bit input/output pins.
PH7 to PH0	Input/ output	8-bit input/output pins.
PI7 to PI0	Input/ output	8-bit input/output pins.
PM4 to PM0	Input/ output	5-bit input/output pins.
PJ7 to PJ0*	Input/ output	8-bit input/output pins.
PK7 to PK0*	Input/ output	8-bit input/output pins.

Note: * These pins can be used when the PCJKE bit in PFCRD is set to 1 in single-ch

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

— 8 × 8-bit register-register multiply: 1 state (when the multiplier is available.

— 16 ÷ 8-bit register-register divide: 10 states (when the divider is available.)

— 16 × 16-bit register-register multiply: 1 state (when the multiplier is available.

— 32 ÷ 16-bit register-register divide: 18 states (when the divider is available.)

— 32 × 32-bit register-register multiply: 5 states (when the multiplier is available

— 32 ÷ 32-bit register-register divide: 18 states (when the divider is available.)

- 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 and H8SX/1655M Group. Normal, middle, and maximum modes are n supported.

2. The multiplier and divider are supported by the H8SX/1655 Group and H8SX Group.

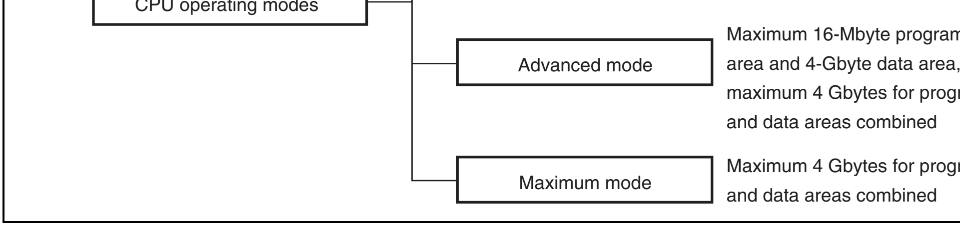


Figure 2.1 CPU Operating Modes

2.2.1 Normal Mode

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

- Address Space

  The maximum address space of 64 Kbytes can be accessed.
- Extended Registers (En)

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

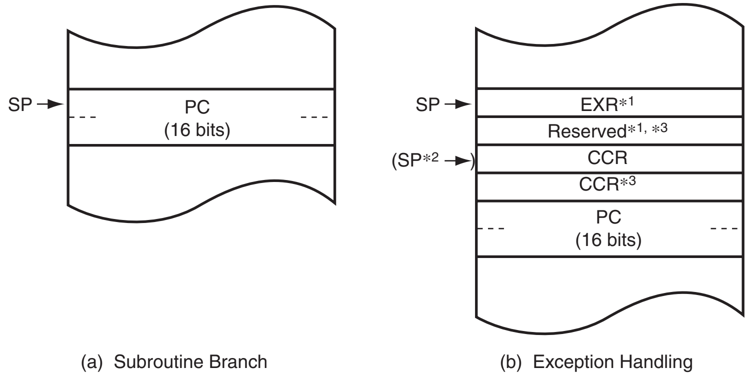
  All instructions and addressing modes can be used. Only the lower 16 bits of effectiv addresses (EA) are valid.

Figure 2.2 Exception Vector Table (Normal Mode)

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

- Stack Structure

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



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

Figure 2.3 Stack Structure (Normal Mode)

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

- Instruction Set

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

- Exception Vector Table and Memory Indirect Branch Addresses

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

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

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

- Stack Structure

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

- Instruction Set

  All instructions and addressing modes can be used.

- Exception Vector Table and Memory Indirect Branch Addresses

  In advanced mode, the top area starting at H'00000000 is allocated to the exception ve table. One branch address is stored per 32 bits. The upper eight bits are ignored and th 24 bits are stored. The structure of the exception vector table is shown in figure 2.4.

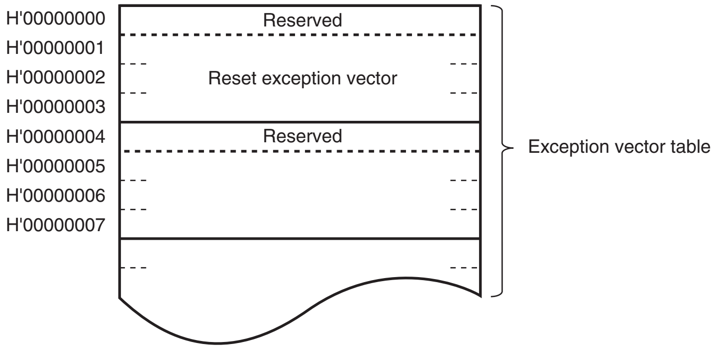


Figure 2.4 Exception Vector Table (Middle and Advanced Modes)

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

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

(a) Subroutine Branch (b) Exception Handling

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

Figure 2.5 Stack Structure (Middle and Advanced Modes)

2.2.4 Maximum Mode

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

- Address Space

  The maximum address space of 4 Gbytes can be linearly accessed.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers or as the upper 16-b segments of 32-bit registers or address registers.
- Instruction Set

  All instructions and addressing modes can be used.
- Exception Vector Table and Memory Indirect Branch Addresses

  In maximum mode, the top area starting at H'00000000 is allocated to the exception table. One branch address is stored per 32 bits. The structure of the exception vector shown in figure 2.6.

Figure 2.6 Exception Vector Table (Maximum Modes)

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

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

- Stack Structure

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

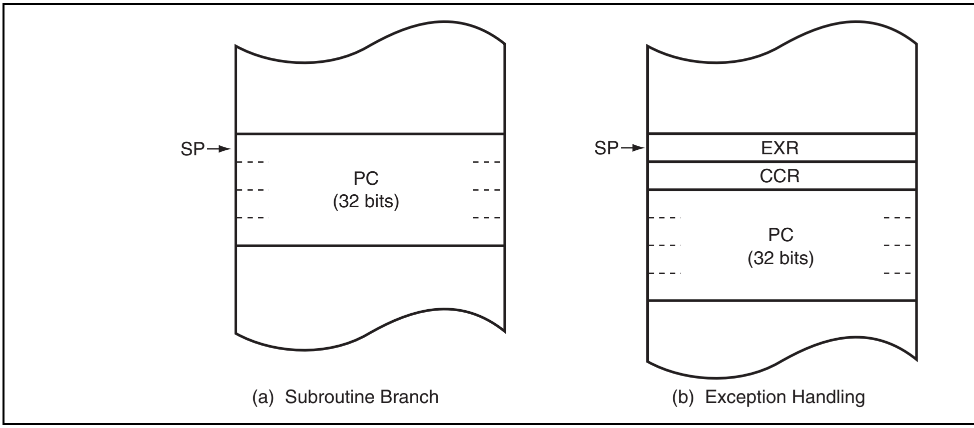


Figure 2.7 Stack Structure (Maximum Mode)

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

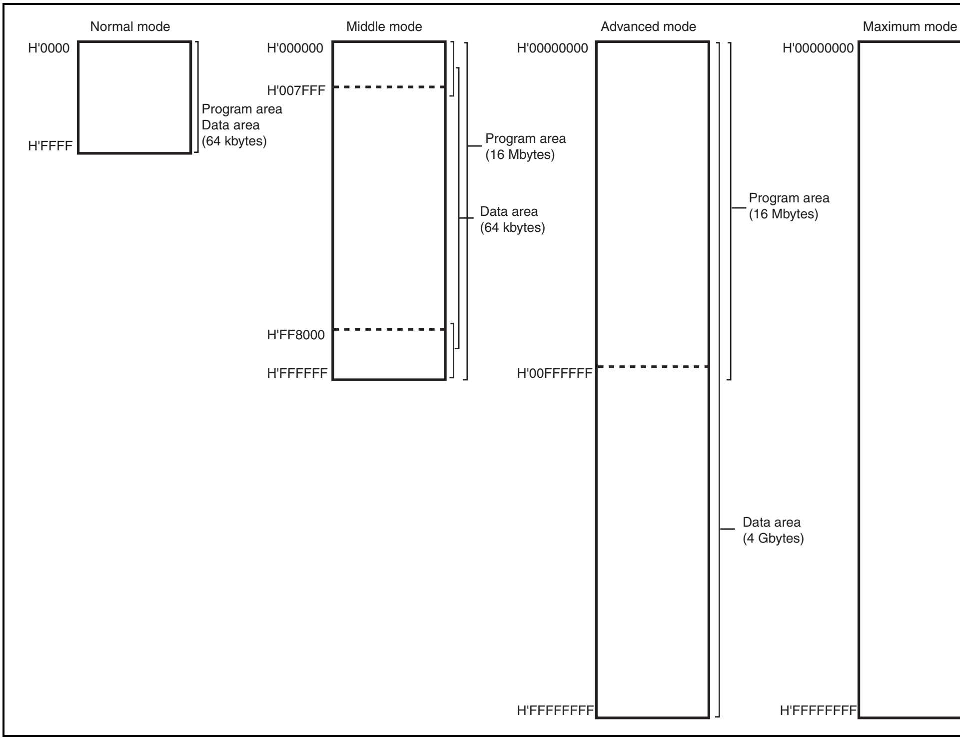


Figure 2.8 Memory Map

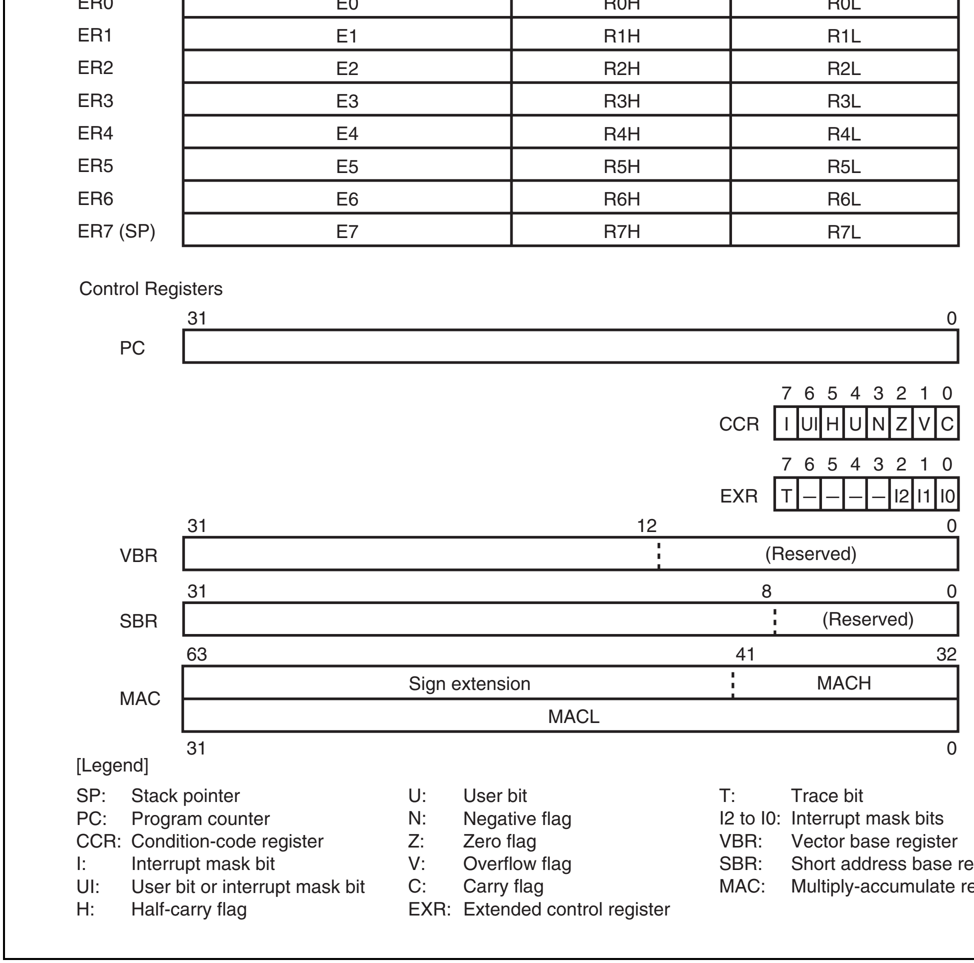


Figure 2.9 CPU Registers

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

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

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

The usage of each register can be selected independently.

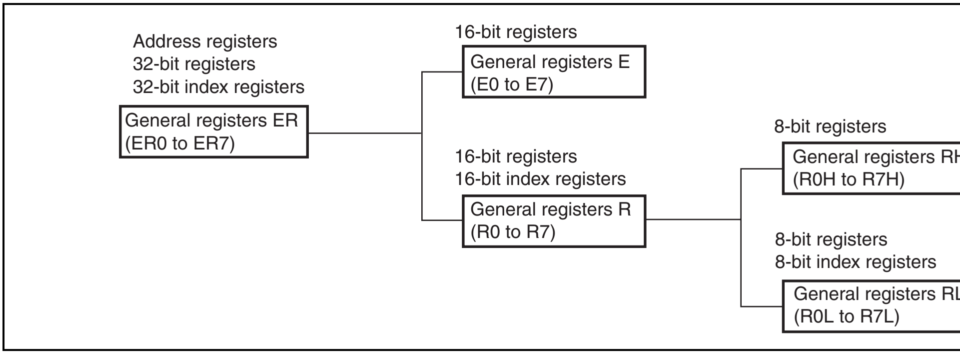


Figure 2.10 Usage of General Registers

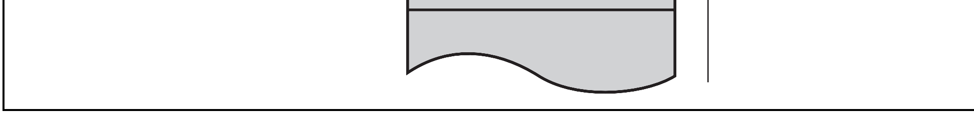


Figure 2.11 Stack

2.5.2 Program Counter (PC)

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

Bit	Bit Name	Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts when set to 1. This bit is set the start of an exception handling.
6	UI	Undefined	R/W	User Bit/Interrupt Mask Bit Can be written to and read from by software LDC, STC, ANDC, ORC, and XORC instructi
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, or NEG.B instruction is executed, this flag is there is a carry or borrow at bit 3, and cleared otherwise. When the ADD.W, SUB.W, CMP.W NEG.W instruction is executed, this flag is se there is a carry or borrow at bit 11, and cleare otherwise. When the ADD.L, SUB.L, CMP.L, instruction is executed, this flag is set to 1 if t carry or borrow at bit 27, and cleared to 0 oth
4	U	Undefined	R/W	User Bit Can be written to and read from by software LDC, STC, ANDC, ORC, and XORC instructi
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit (re sign bit) of data.

otherwise. A carry has the following types:

- Carry from the result of addition
- Borrow from the result of subtraction
- Carry from the result of shift or rotation

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

2.5.4 Extended Control Register (EXR)

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

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

For details, see the corresponding section.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception is g each time an instruction is executed. When thi cleared to 0, instructions are executed in sequ
6 to 3	—	All 1	R/W	Reserved These bits are always read as 1.
2	I2	1	R/W	Interrupt Mask Bits
1	I1	1	R/W	These bits designate the interrupt mask level (
0	I0	1	R/W	

8-bit absolute address addressing mode (@aa:8), this register is used as the upper addres
initial value is H'FFFFFF00. The SBR contents are changed with the LDC and STC inst

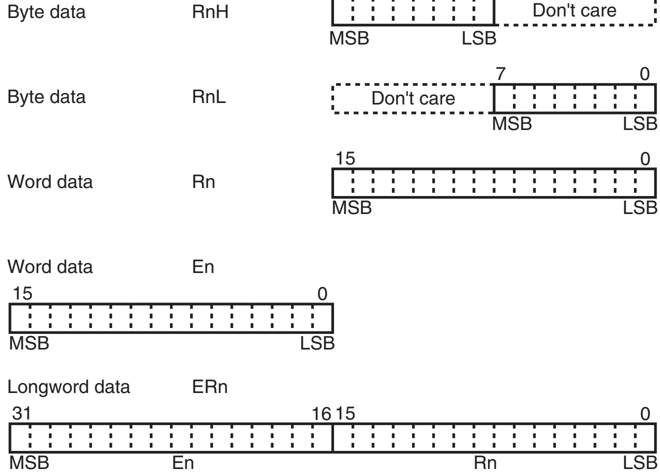
2.5.7 Multiply-Accumulate Register (MAC)

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

2.5.8 Initial Values of CPU Registers

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

Figure 2.12 shows the data formats in general registers.



[Legend]

ERn: General register ER
En: General register E
Rn: General register R
RnH: General register RH
RnL: General register RL
MSB: Most significant bit
LSB: Least significant bit

Figure 2.12 General Register Data Formats

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

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

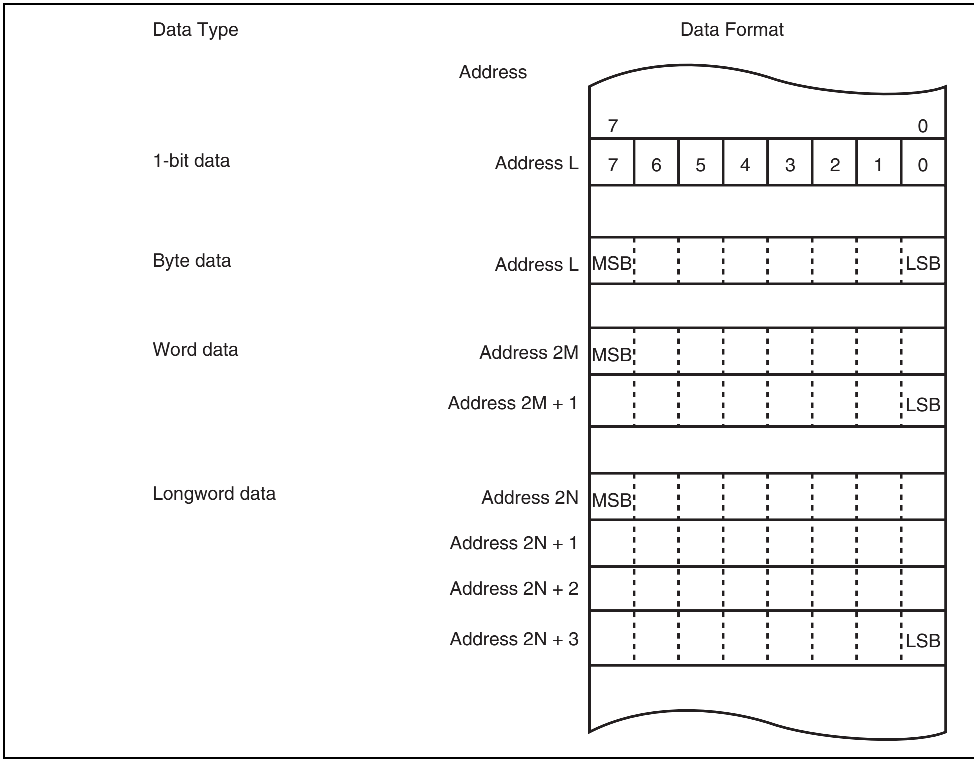


Figure 2.13 Memory Data Formats

	POP, PUSH*[1]	W/L
	LDM, STM	L
	MOVA	B/W*
Block transfer	EEPMOV	B
	MOVMD	B/W/
	MOVSD	B
Arithmetic operations	ADD, ADDX, SUB, SUBX, CMP, NEG, INC, DEC	B/W/
	DAA, DAS	B
	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	B/W
	MULU, DIVU, MULS, DIVS	W/L
	MULU/U*[6], MULS/U*[6]	L
	EXTU, EXTS	W/L
	TAS	B
	MAC*[6]	—
	LDMAC*[6], STMAC*[6]	—
	CLRMAC*[6]	—
Logic operations	AND, OR, XOR, NOT	B/W/
Shift	SHLL, SHLR, SHAL, SHAR, ROTL, ROTR, ROTXL, ROTXR	B/W/
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	B
	BSET/EQ, BSET/NE, BCLR/EQ, BCLR/NE, BSTZ, BISTZ	B
	BFLD, BFST	B

[Legend]

B: Byte size

W: Word size

L: Longword size

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

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

2. Size of data to be added with a displacement
3. Size of data to specify a branch condition
4. Bcc is the generic designation of a conditional branch instruction.
5. Size of general register to be restored
6. Only when the multiplier is available.

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

		W/L		D	D	D	D	D		D
	EXTU, EXTS	W/L		D	D	D	D	D		D
	TAS	B			D					
	MAC*[12]	—								
	CLRMAC*[12]	—								
	LDMAC*[12]	—		S						
	STMAC*[12]	—		D						
Logic operations	AND, OR, XOR	B		S	D	D	D	D	D	D
		B		D	S	S	S	S	S	S
		B			SD	SD	SD	SD		SD
		W/L	S	SD	SD	SD	SD	SD		SD
	NOT	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D
Shift	SHLL, SHLR	B		D	D	D	D	D	D	D
		W/L*[6]		D	D	D	D	D		D
		B/W/L*[7]		D						
	SHAL, SHAR ROTL, ROTR ROTXL, ROTXR	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D
Bit manipu-lation	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

(VBR, SBR)									
STC (CCR, EXR)	B/W*9		D	D	D		D*11		D
STC (VBR, SBR)	L		D						
ANDC, ORC, XORC	B	S							
SLEEP	—								
NOP	—								

[Legend]

d: d:16 or d:32

S: Can be specified as a source operand.

D: Can be specified as a destination operand.

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

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

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

Notes:
1. Only @aa:16 is available.
2. @ERn+ as a source operand and @−ERn as a destination operand
3. Specified by ER5 as a source address and ER6 as a destination address for transfer.
4. Size of data to be added with a displacement
5. Only @ERn− is 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 when immediate or register direct, otherwise, word
10. Only @ERn+ is available
11. Only @−ERn is available
12. Only when the multiplier is available.

	Bcc	—		O					
	BRA	—		O	O				
	BRA/S	—		O*					
	JMP	—	O			O	O	O	O
	BSR	—		O					
	JSR	—	O			O	O	O	O
	RTS, RTS/L	—							
System control	TRAPA	—							
	RTE, RTE/L	—							

[Legend]

d: d:8 or d:16

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

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

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit 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 multiple general registers. Tw or four general registers which have serial register numbers can specified.
STM	L	Rn (register list) → @−SP Saves the contents of multiple general registers on the stack. Tw or four general registers which have serial register numbers can specified.
MOVA	B/W	EA → Rd Zero-extends and shifts the contents of a specified general regist memory data and adds them with a displacement. The result is s general register.

MOVMD.W	W	Transfers a data block. Transfers word data which begins at a memory location specifie to a memory location specified by ER6. The number of word da transferred is specified by R4.
MOVMD.L	L	Transfers a data block. Transfers longword data which begins at a memory location spe ER5 to a memory location specified by ER6. The number of lon data to be transferred is specified by R4.
MOVSD.B	B	Transfers a data block with zero data detection. Transfers byte data which begins at a memory location specifie to a memory location specified by ER6. The number of byte dat transferred is specified by R4. When zero data is detected durin the transfer stops and execution branches to a specified addres

INC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd
DEC		Increments or decrements a general register by 1 or 2. (Byte ope can be incremented or decremented by 1 only.)
ADDS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd
SUBS		Adds or subtracts the value 1, 2, or 4 to or from data in a general
DAA	B	Rd (decimal adjust) → Rd
DAS		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 bits × 8 bits → 16 bits, or 16 bits × 16 bits → 32 bits.
MULU/U*	L	Rd × Rs → Rd
		Performs unsigned multiplication on data in two general 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.
MULS	W/L	Rd × Rs → Rd
		Performs signed multiplication on data in two general registers: e bits × 16 bits → 16 bits, or 32 bits × 32 bits → 32 bits.
MULS/U*	L	Rd × Rs → Rd
		Performs signed multiplication on data in two general registers (3 32 bits → upper 32 bits).
DIVXU	B/W	Rd ÷ Rs → Rd
		Performs unsigned division on data in two general registers: eith ÷ 8 bits → 8-bit quotient and 8-bit remainder, or 32 bits ÷ 16 bits quotient and 16-bit remainder.

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

Note: Only when the multiplier is available.

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

Table 2.8 Shift Operation Instructions

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

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

		Exclusive-ORs 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 d
BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in the contents of a general register or a location to the carry flag. The bit number is specified by 3-bit imm data.
BILD	B	~ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in the contents of a gener register or a memory location to the carry flag. The bit number is by 3-bit immediate data.
BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in the contents of general register or a memory location. The bit number is specifie immediate data.
BSTZ	B	Z → (<bit-No.> of <EAd>) Transfers the zero flag value to a specified bit in the contents of a memory location. The bit number is specified by 3-bit immediate
BIST	B	~ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in th contents of a general register or a memory location. The bit numb specified by 3-bit immediate data.

field in memory location contents.

Table 2.10 Branch Instructions

Instruction	Size	Function
BRA/BS BRA/BC	B	Tests a specified bit in memory location contents. If the specifie condition is satisfied, execution branches to a specified address
BSR/BS BSR/BC	B	Tests a specified bit in memory location contents. If the specifie condition is satisfied, execution branches to a subroutine at a sp address.
Bcc	—	Branches to a specified address if the specified condition is sati
BRA/S	—	Branches unconditionally to a specified address after executing instruction. The next instruction should be a 1-word instruction e 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 mult general registers.

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

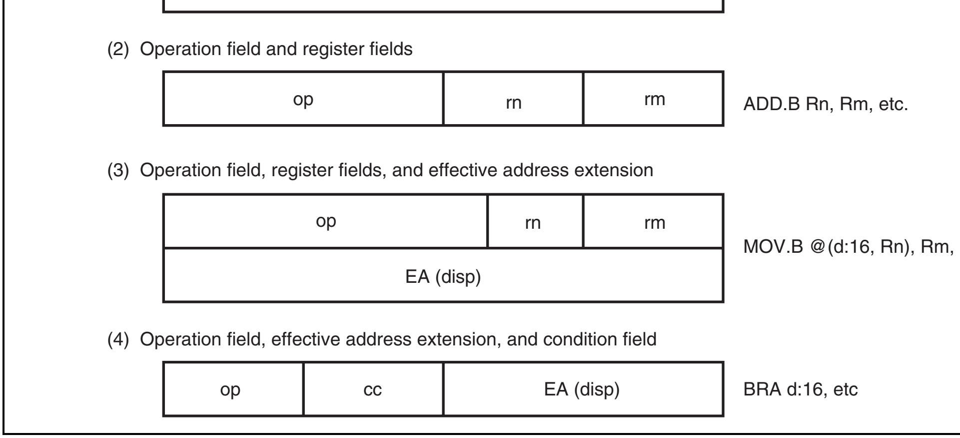


Figure 2.14 Instruction Formats

- Operation Field

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

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

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

  Specifies the branch condition of Bcc instructions.

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:2,ERn)/@(d:16,ERn)/@(d:32,ER
4	Index register indirect with displacement	@(d:16, RnL.B)/@(d:16,Rn.W)/@(d:1 @(d:32, RnL.B)/@(d:32,Rn.W)/@(d:3
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.L
10	Memory indirect	@@aa:8
11	Extended memory indirect	@@vec:7

2.8.1 Register Direct—Rn

The operand value is the contents of an 8-, 16-, or 32-bit general register which is specifie register field in the instruction code.

R0H to R7H and R0L to R7L can be specified as 8-bit registers.

R0 to R7 and E0 to E7 can be specified as 16-bit registers.

ER0 to ER7 can be specified as 32-bit registers.

The operand value is the contents of a memory location which is pointed to by the sum o
contents of an address register (ERn) and a 16- or 32-bit displacement. ERn is specified
register field of the instruction code. The displacement is included in the instruction cod
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 the displacement is 2, 4, or
operand is word data, or when the displacement is 4, 8, or 12 and the operand is longwo

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

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

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

- Register indirect with post-decrement—@ERn–

  The operand value is the contents of a memory location which is pointed to by the con
  an address register (ERn). ERn is specified by the register field of the instruction code
  the memory location is accessed, 1, 2, or 4 is subtracted from the address register cont
  the remainder is stored in the address register. The value subtracted is 1 for byte acces
  word access, or 4 for longword access.

using this addressing mode, data to be written is the contents of the general register after
calculating an effective address. If the same general register is specified in an instruction
effective addresses are calculated, the contents of the general register after the first calcul
an effective address is used in the second calculation of an effective address.

Example 1:

MOV.W R0, @ER0+
When ER0 before execution is H'12345678, H'567A is written at H'12345678.

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

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

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

Table 2.13 shows the accessible absolute address ranges.

Table 2.13 Absolute Address Access Ranges

Absolute Address		Normal Mode	Middle Mode	Advanced Mode	Maxim Mode
Data area	8 bits (@aa:8)	A consecutive 256-byte area (the upper address is set in SB			
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF	H'00000000 to H'00007F H'FFFF8000 to H'FFFFF	
	32 bits (@aa:32)			H'00000000 to H'FFFFFI	
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'FFFI

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

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

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

2.8.9 Program-Counter Relative with Index Register—@(RnL.B, PC), @(Rn.W or @(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 conte contents of an address register specified by the register field in the instruction code (RnL ERn) is zero-extended and multiplied by 2. The PC contents to which the displacement is the address of the first byte of the next instruction. In advanced mode, only the lower 24 this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00).

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

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

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

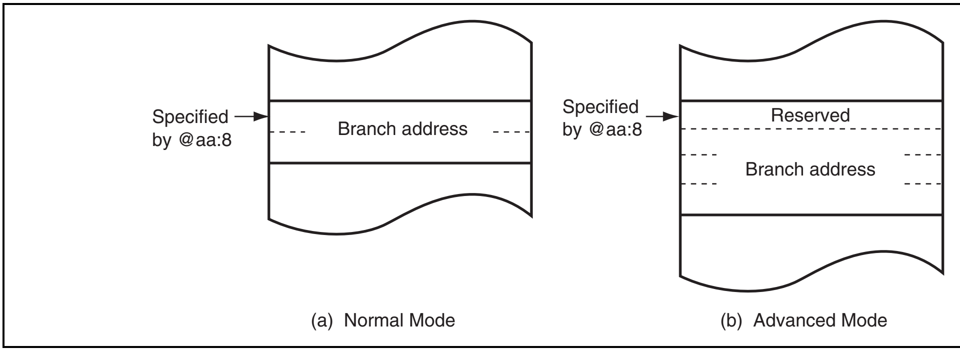


Figure 2.15 Branch Address Specification in Memory Indirect Mode

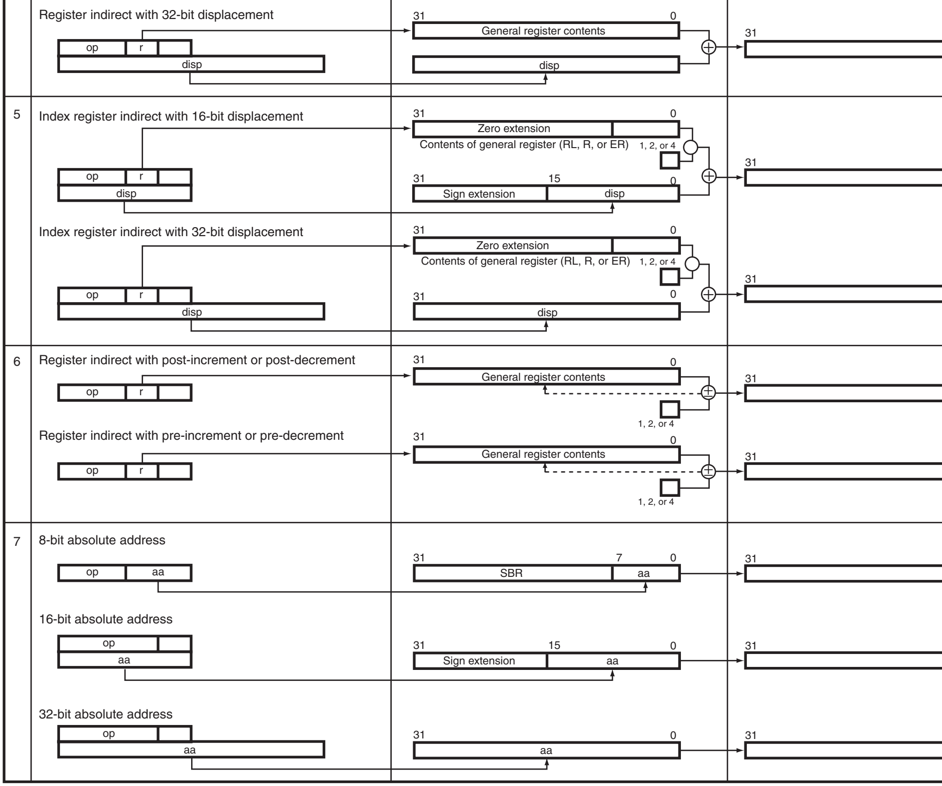
advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

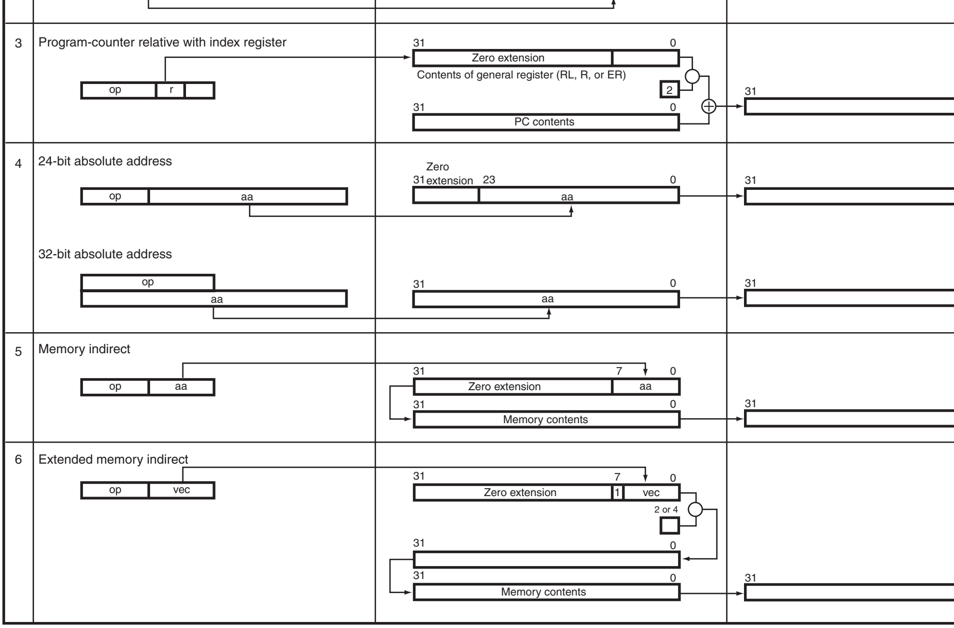
2.8.12 Effective Address Calculation

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

The valid bits in middle mode are as follows:

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


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



2.8.13 MOVA Instruction

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

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

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

- Exception-handling state

  The exception-handling state is a transient state that occurs when the CPU alters the processing flow due to activation of an exception source, such as, a reset, trace, inter trap instruction. The CPU fetches a start address (vector) from the exception handlin table and branches to that address. For further details, see section 6, Exception Hand

- Program execution state

  In this state the CPU executes program instructions in sequence.

- Bus-released state

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

- Program stop state

  This is a power-down state in which the CPU stops operating. The program stop stat when a SLEEP instruction is executed or the CPU enters hardware standby mode. F see section 27, Power-Down Modes.

* A transition to the reset state occurs when the RES signal goes low in all states except hardware standby mode. A transition can also be made to the reset state when the watchdog timer overflows.

Figure 2.16 State Transitions

Mode	MD2	MD1	MD0	Mode	Space	Mode	ROM	Defau
1	0	0	1	Advanced mode	16 Mbytes	User boot mode	Enabled	—
2	0	1	0			Boot mode	Enabled	—
3	0	1	1			Boundary scan enabled single-chip mode	Enabled	—
4	1	0	0			On-chip ROM disabled extended mode	Disabled	16 bi
5	1	0	1				Disabled	8 bits
6	1	1	0			On-chip ROM enabled extended mode	Enabled	8 bits
7	1	1	1			Single-chip mode	Enabled	—

In this LSI, an advanced mode as the CPU operating mode and a 16-Mbyte address spac available. The initial external bus widths are 8 bits or 16 bits. As the LSI initiation mode external extended mode, on-chip ROM initiation mode, or single-chip initiation mode ca selected.

Modes 1 and 2 are the user boot mode and the boot mode, respectively, in which the flas can be programmed and erased. For details on the user boot mode and boot mode, see se Flash Memory.

Modes 4 to 6 are external extended modes, in which the external memory and devices can be accessed. In the external extended modes, the external address space can be designated as 8-bit or 16-bit address space for each area by the bus controller after starting program execution.

If 16-bit address space is designated for any one area, it is called the 16-bit bus widths mode. If 8-bit address space is designated for all areas, it is called the 8-bit bus width mode.

MD3 to MD0 are latched. Latching is released by a reset.

Bit	15	14	13	12	11	10	9	
Bit Name	—	—	—	—	MDS3	MDS2	MDS1	
Initial Value	0	1	0	1	Undefined*	Undefined*	Undefined*	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	Undefined*	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
14	—	1	R	These are read-only bits and cannot be mod
13	—	0	R	
12	—	1	R	
11	MDS3	Undefined*	R	Mode Select 3 to 0
10	MDS2	Undefined*	R	These bits indicate the operating mode sele the mode pins (MD2 to MD0) (see table 3.2
9	MDS1	Undefined*	R	When MDCR is read, the signal levels input MD2 to MD0 are latched into these bits. The latches are released by a reset.
8	MDS0	Undefined*	R	

Note: Determined by pins MD2 to MD0.

Table 3.2 Settings of Bits MDS3 to MDS0

MCU Operating Mode	Mode Pins			MDCR			
	MD2	MD1	MD0	MDS3	MDS2	MDS1	M
1	0	0	1	1	1	0	1
2	0	1	0	1	1	0	0
3	0	1	1	0	1	0	0
4	1	0	0	0	0	1	0
5	1	0	1	0	0	0	1
6	1	1	0	0	1	0	1
7	1	1	1	0	1	0	0

Bit	7	6	5	4	3	2	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	—	1	R/W	Reserved
14	—	1	R/W	These bits are always read as 1. The write va always be 1.
13	MACS	0	R/W	MAC Saturation Operation Control Selects either saturation operation or non-sat operation for the MAC instruction. 0: MAC instruction is non-saturation operatior 1: MAC instruction is saturation operation
12	—	1	R/W	Reserved This bit is always read as 1. The write value s always be 1.
11	FETCHMD	0	R/W	Instruction Fetch Mode Select This LSI can prefetch an instruction in units o 32 bits. Select the bus width for instruction fet depending on the used memory for the storag programs. 0: 32-bit mode 1: 16-bit mode

				bit when PCJKE = 1*². When writing 0 to this bit after reading EXPE = external bus cycle should not be executed. The external bus cycle may be carried out in pa with the internal bus cycle depending on the se the write data buffer function and the state the EXDMAC releases the bus mastership. 0: External bus disabled 1: External bus enabled
8	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. This bit 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 valu always be 0.
1	DTCMD	1	R/W	DTC Mode Select Selects DTC operating mode. 0: DTC is in full-address mode 1: DTC is in short address mode
0	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.

Notes: 1. The initial value depends on the LSI initiation mode.

2. For details on the settings of the EXPE and PCJKE bits when the external add space is in use, see section 13.3.12, Port Function Control Register D (PFCRD

This is the boot mode for the flash memory. The LSI operates in the same way as in mo
except for programming and erasing of the flash memory. For details, see section 24, Fl
Memory.

3.3.3 Mode 3

This is the boundary scan function enabled single-chip activation mode. The operation i
as mode 7 except for the boundary scan function. For details on the boundary scan funct
section 25, Boundary Scan.

3.3.4 Mode 4

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

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

3.3.6 Mode 6

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

The initial bus width mode immediately after a reset is eight bits, with 8-bit access to all a
Ports D, E, and F function as input ports, but they can be used as an address bus by specif
data direction register (DDR) for each port. For details, see section 13, I/O Ports. Port H f
as a data bus, and parts of ports A and B function as bus control signals. However, if any
designated as a 16-bit access space by the bus controller, the bus width mode switches to
and ports H and I function as a data bus.

3.3.7 Mode 7

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

All I/O ports can be used as general input/output ports. The external address space cannot
accessed in the initial state, but setting the EXPE bit in the system control register (SYSC
enables the external address space. After the external address space is enabled, ports H an
be used as a data bus and ports D, E, and F as an address output bus by specifying the dat
direction register (DDR) for each port. When the external address space is not in use, por
K can be used by setting the PCJKE bit in the port function control register D (PFCRD) t
details, see section 13, I/O Ports.

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

[Legend]

P: I/O port

A: Address bus output

D: Data bus input/output

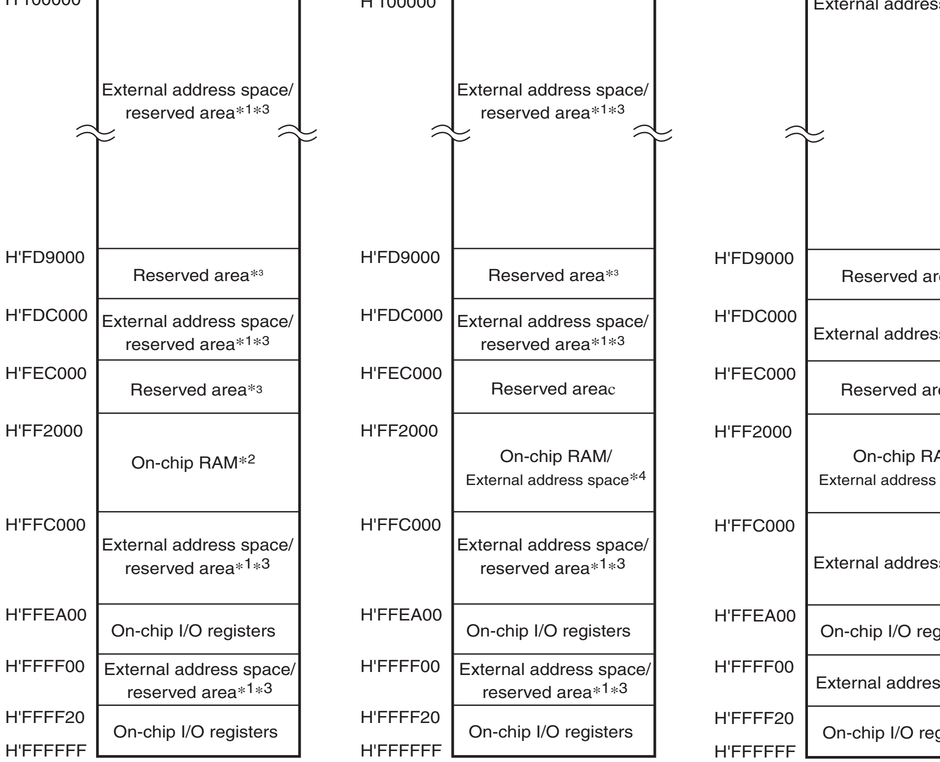
C: Control signals, clock input/output

*: Immediately after a reset

3.4 Address Map

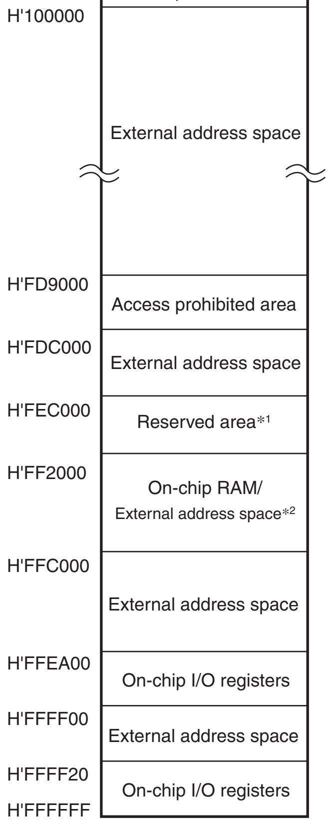
3.4.1 Address Map

Figures 3.1 and 3.2 show the address map in each operating mode.



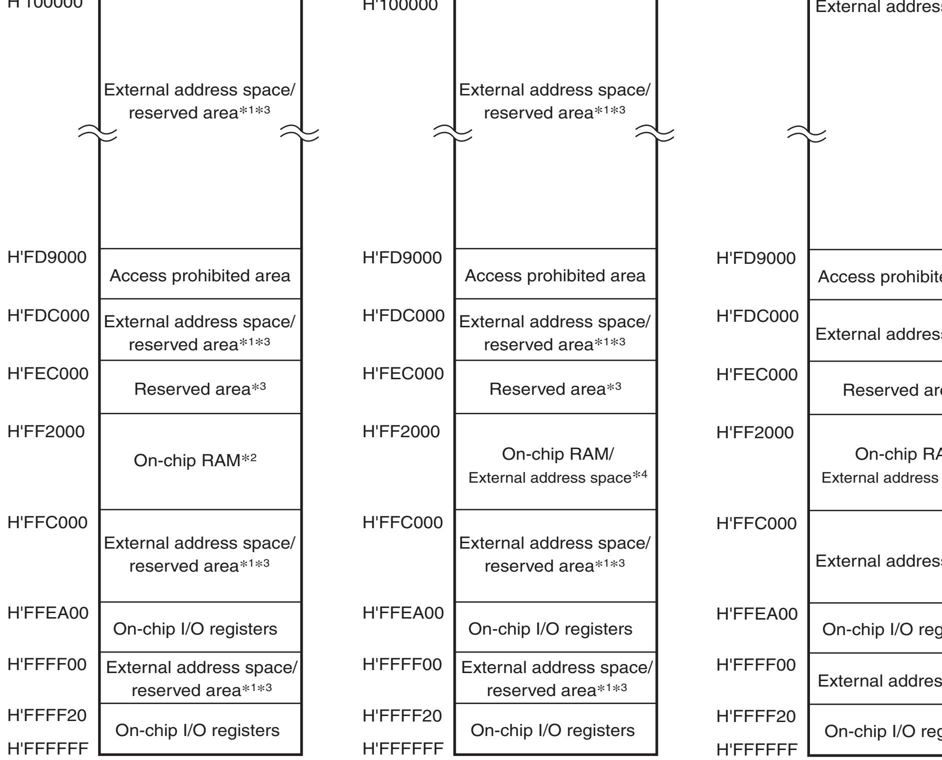
Notes:1. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE = 0.
2. The on-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR to 0.
3. Do not access the reserved areas.
4. This area is specified as the external address space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

Figure 3.1 Address Map in Each Operating Mode of H8SX/1655 and H8SX/165



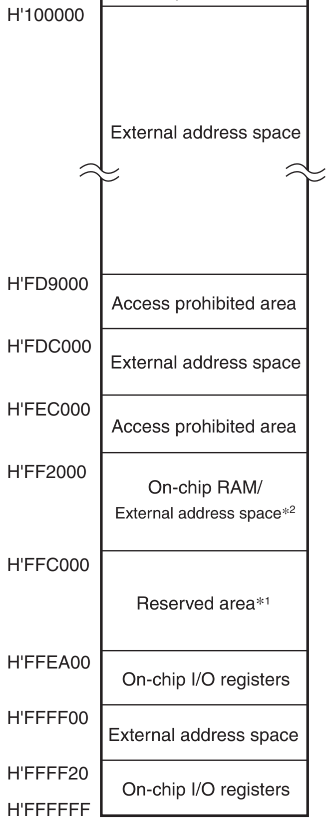
Notes: 1. Do not access the reserved area.
2. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0

Figure 3.1 Address Map in Each Operating Mode of H8SX/1655 and H8SX/165



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

Figure 3.2 Address Map in Each Operating Mode of H8SX/1652 and H8SX/165



Notes: 1. Do not access the reserved area.
2. This area is specified as the external address space by clearing the RAME bit in SYSCR to (

Figure 3.2 Address Map in Each Operating Mode of H8SX/1652 and H8SX/165

When using power-on reset* and voltage monitoring reset*, $\overline{RES}$ pin must be fixed high

Table 4.1 Reset Names And Sources

Reset Name	Source
Pin reset	Voltage input to the $\overline{RES}$ pin is driven low.
Power-on reset*	Vcc rises or lowers
Voltage-monitoring reset*	Vcc falls (voltage-detection: Vdet)
Deep software standby reset	Deep software standby mode is canceled b interrupt.
Watchdog timer reset	The watchdog timer overflows.

Note: * Supported only by the H8SX/1655M Group.

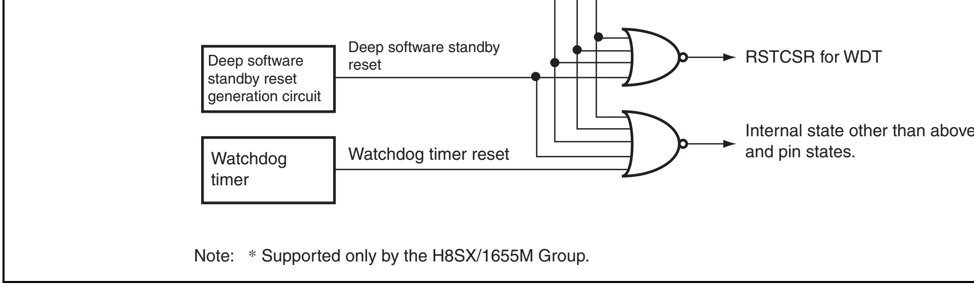


Figure 4.1 Block Diagram of Reset Circuit

When a reset is canceled, the reset exception handling is started. For the reset exception
see section 6.3, Reset.

4.2 Input/Output Pin

Table 4.2 shows the pin related to reset.

Table 4.2 Pin Configuration

Pin Name	Symbol	I/O	Function
Reset	$\overline{RES}$	Input	Reset input

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

Notes: 1. Only 0 can be written to clear the flag.
2. Supported only by the H8SX/1655M Group.
3. Initial value is undefined in the H8SX/1655M Group.
4. Only 0 can be written to clear the flag in the H8SX/1655M Group.
5. Only read is possible in the H8SX/1655M Group.

Bit	Bit Name	Initial Value	R/W	Description
7	DPSRSTF	0	R/(W)*1	Deep Software Standby Reset Flag Indicates that deep software standby mode is ca by an interrupt source specified with DPSIER or DPSIEGR and an internal reset is generated. [Setting condition] When deep software standby mode is canceled interrupt source. [Clearing conditions] • When this bit is read as 1 and then written b • When a pin reset, power-on reset*2 and volta monitoring reset*2 is generated.
6 to 3	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.

Bit	Bit Name	Value	R/W	Description
2	LVDF	Undefined	R/(W)*[1]	LVD Flag This bit indicates that the voltage detection circ detected a low voltage (Vcc at or below Vdet). [Setting condition] Vcc falling to or below Vdet. [Clearing condition] • After Vcc has exceeded Vdet and the spec stabilization period has elapsed, writing 0 to after reading it as 1. • Generation of a pin reset or power-on rese
1	—	Undefined	R/W	Reserved The write value should always be 0.
0	PORF	Undefined	R	Power-on Reset Flag This bit indicates that a power-on reset has bee generated. [Setting condition] Generation of a power-on reset [Clearing condition] Generation of a pin reset

Notes: 1. Only 0 can be written to clear the flag.
2. Supported only by the H8SX/1655M Group.

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

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Timer Overflow Flag This bit is set when TCNT overflows in watchdog timer but not set in interval timer mode. Only 0 can be written [Setting condition] When TCNT overflows (H'FF → H'00) in watchdog time [Clearing condition] When this bit is read as 1 and then written by 0. (The flag must be read after writing of 0, when this bit is by the CPU using an interrupt.)
6	RSTE	0	R/W	Reset Enable Selects whether or not the LSI internal state is reset by overflow in watchdog timer mode. 0: Internal state is not reset when TCNT overflows. (Alth this LSI internal state is not reset, TCNT and TCSR o WDT are reset.) 1: Internal state is reset when TCNT overflows.
5	—	0	R/W	Reserved Although this bit is readable/writable, operation is not af by this bit.
4 to 0	—	1	R	Reserved These are read-only bits but cannot be modified.

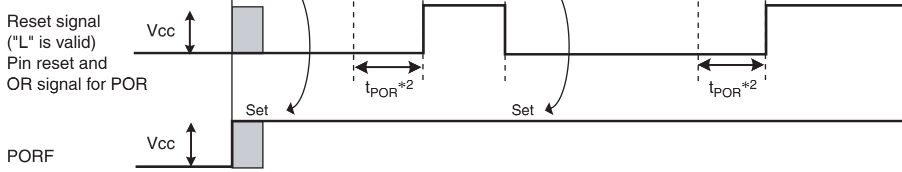
Note: * Only 0 can be written to clear the flag.

This is an internal reset generated by the power-on reset circuit.

If $\overline{RES}$ is in the high-level state when power is supplied, a power-on reset is generated. A
has exceeded Vpor and the specified period (power-on reset time) has elapsed, the chip i
from the power-on reset state. The power-on reset time is a period for stabilization of th
power supply and the LSI circuit.

If $\overline{RES}$ is at the high-level when the power-supply voltage (Vcc) falls to or below Vpor,
on reset is generated. The chip is released after Vcc has risen above Vpor and the power
time has elapsed.

After a power-on reset has been generated, the PORF bit in RSTSR is set to 1. The POR
a read-only register and is only initialized by a pin reset. Figure 4.2 shows the operation
power-on reset.



Notes: For details of the electrical characteristics, see section 29, Electrical Characteristics.

1. V_{POR} shows a level of power-on reset detection level.
2. T_{POR} shows a time for power-on reset.

Figure 4.2 Operation of a Power on Reset

4.7 Deep Software Standby Reset

This is an internal reset generated when deep software standby mode is canceled by an i

When deep software standby mode is canceled, a deep software standby reset is generat
simultaneously, clock oscillation starts. After the time specified with DPSWCR has elap
deep software standby reset is canceled.

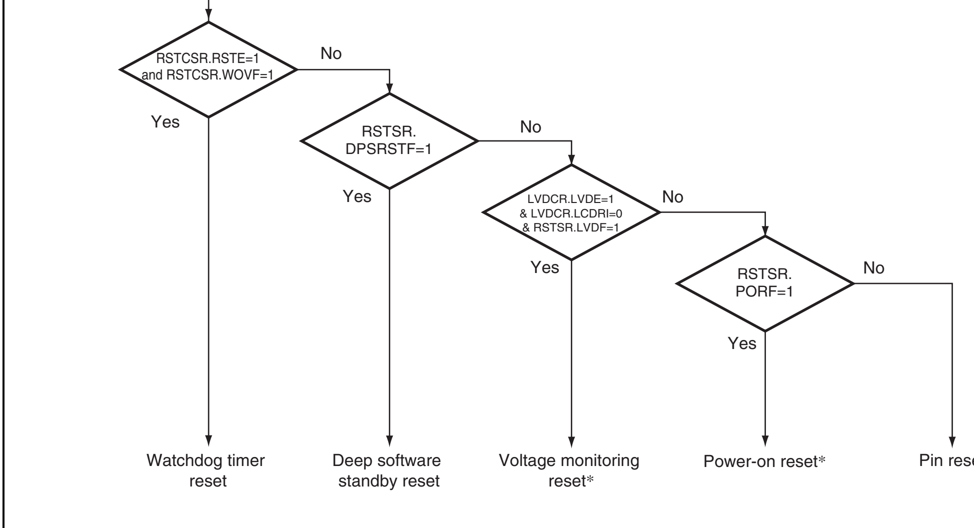
For details of the deep software standby reset, see section 27, Power-Down Modes.

4.8 Watchdog Timer Reset

This is an internal reset generated by the watchdog timer.

When the RSTE bit in RSTCSR is set to 1, a watchdog timer reset is generated by a TC
overflow. After a certain time, the watchdog timer reset is canceled.

For details of the watchdog timer reset, see section 17, Watchdog Timer (WDT).



Note: * Supported only by the H8SX/1655M Group.

Figure 4.3 Example of Reset Generation Source Determination Flow

Capable of detecting the power-supply voltage (Vcc) becoming less than or equal to

Capable of generating an internal reset or interrupt when a low voltage is detected.

A block diagram of the voltage detection circuit is shown in figure 5.1.

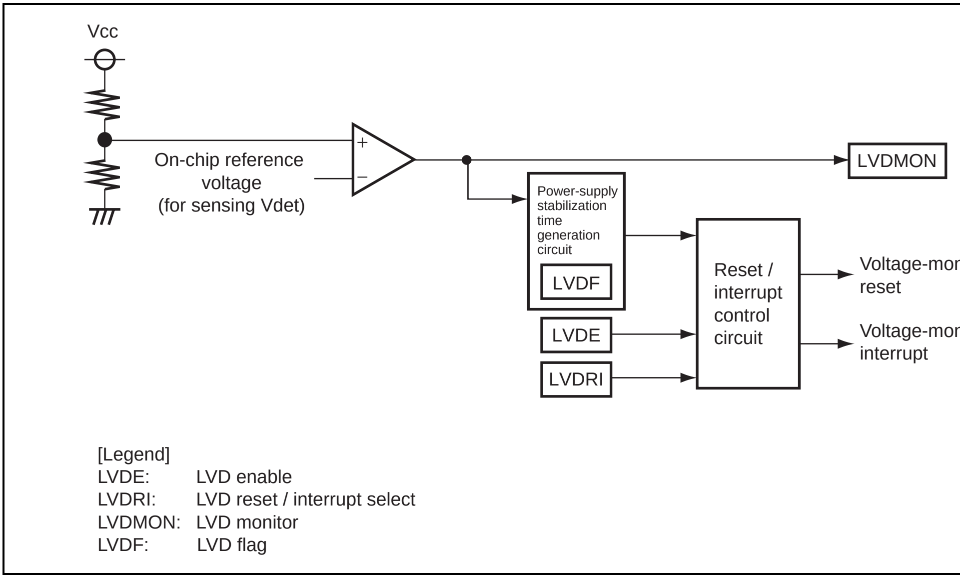


Figure 5.1 Block Diagram of Voltage-Detection Circuit

LVDE, LVDRI, and LVDMON are initialized by a pin reset or power-on reset

Bit	7	6	5	4	3	2	1
Bit name	LVDE	LVDRI	—	LVDMON	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	LVDE	0	R/W	LVD Enable This bit enables or disables issuing of a r interrupt by the voltage-detection circuit. 0: Disabled 1: Enabled
6	LVDRI	0	R/W	LVD Reset/Interrupt Select This bit selects whether an internal reset interrupt is generated when the voltage d circuit detects a low voltage. When modif LVDRI bit, ensure that low-voltage detect the disabled state (the LVDE bit is cleare 0: A reset is generated when a voltage is detected. 1: An interrupt is generated when a low v detected.
5	—	0	R/W	Reserved This bit is always read as 0 and the write should always be 0.

5.2.2 Reset Status Register (RSTSR)

RSTSR indicates the source of an internal reset or voltage monitoring interrupt.

Bit	7	6	5	4	3	2	1
Bit name	DPSRSTF	—	—	—	—	LVDF	—
Initial value:	0	0	0	0	0	Undefined	Undefined
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R/(W)*	R/W

Note: * To clear the flag, only 0 should be written to.

Bit	Bit Name	Initial Value	R/W	Description
7	DPSRSTF	0	R/W*	Deep Software Standby Reset Flag This bit indicates release from deep soft standby mode due to the interrupt sourc selected by DPSIER and DPSIEGR, an generation of an internal reset. [Setting condition] Release from deep software standby m an interrupt source. [Clearing condition] • Writing 0 to the bit after reading it a • Generation of a pin reset, power o voltage monitoring reset.

				[Clearing condition] • After Vcc has exceeded Vdet and th specified stabilization period has elap writing 0 to the bit after reading it as 1 • Generation of a pin reset or power-o
1	—	Undefined	R/W	Reserved The write value should always be 0.
0	PORF	Undefined	R	Power-on Reset Flag This bit indicates that a power-on reset h generated. [Setting condition] Generation of a power-on reset [Clearing condition] Generation of a pin reset

Note: * To clear the flag, only 0 should be written to.

Next, after Vcc has risen above Vdet, release from the voltage-monitoring reset takes pl
period for stabilization (t_{por}) has elapsed. The period for stabilization (t_{por}) is a time that is
by the voltage detection circuit in order to stabilize the Vcc and the internal circuit of the

When a voltage-monitoring reset is generated, the LVDF bit is set to 1.

For details, see section 29, Electrical Characteristics.

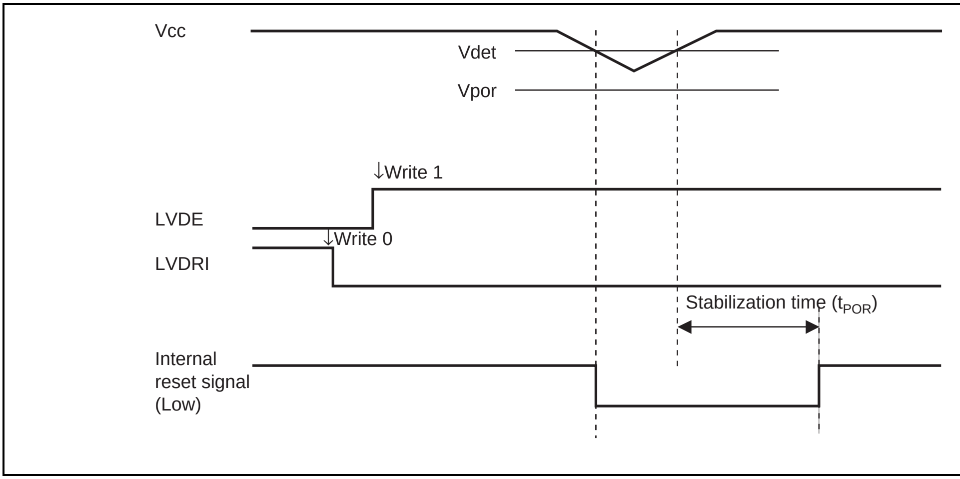


Figure 5.2 Timing of the Voltage-Monitoring Reset

the IRQ14SR and IRQ14SF bits in the ISCR to 01 (interrupt request on falling edge).

Figure 5.4 shows the procedure for setting the voltage-monitoring interrupt.

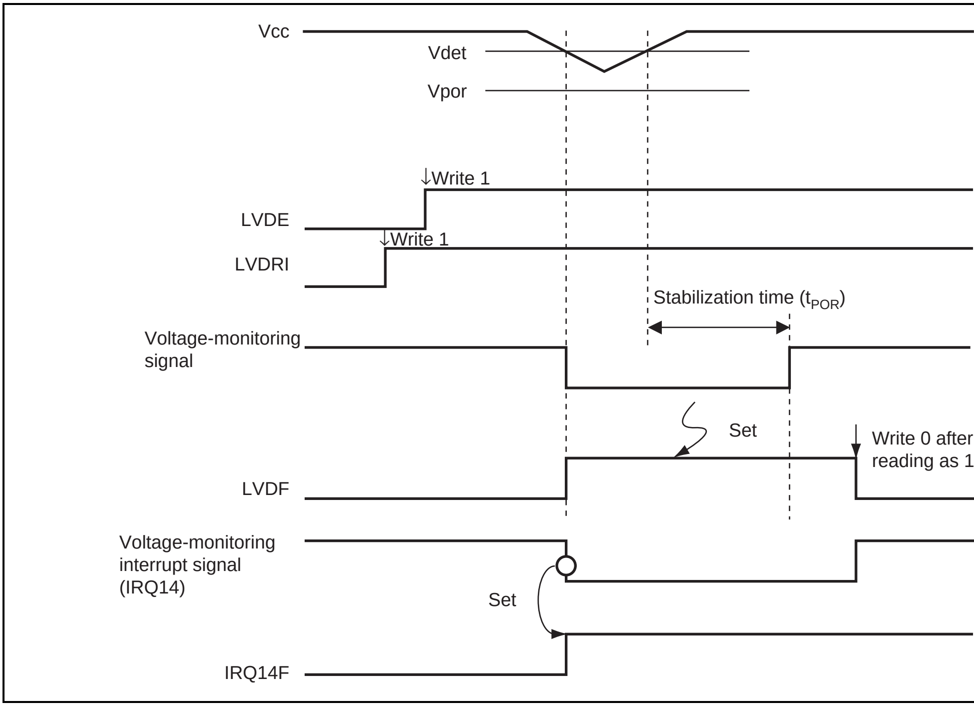
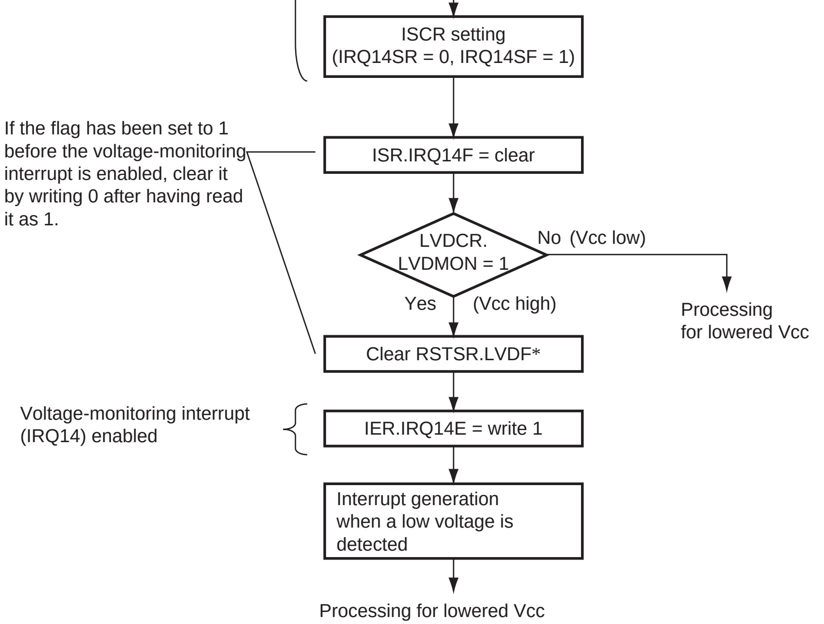


Figure 5.3 Timing of the Voltage-Monitoring Interrupt



Note: * When the LVDF cannot be cleared despite Vcc being at a higher electrical potential tha
Vdet (LVDMON = 1), the voltage-detection circuit is in the state of waiting for stabilizatio
Clear the bit again after the stabilization time (t_{POR}) has elapsed.

Figure 5.4 Example of the Procedure for Setting the Voltage-Monitoring Inte

The result of voltage detection by the voltage-detection circuit can be monitored by check
value of the LVDMON bit in LVDCR. When the LVDMON bit has been enabled by sett
LVDE bit, 0 indicates that Vcc is at or below Vdet and 1 indicates that Vcc is above Vde
bit should be read while the voltage-monitoring reset has been disabled by setting the LV
to 1.

Before clearing the LVDF bit in RSTSR to 0, confirm that the LVDMON bit is set to 1 (i
that Vcc is above Vdet). When it is impossible to clear the LVDF bit despite the LVDMO
being 1, the voltage-detection circuit is in the state of waiting for stabilization. In such ca
the bit again after the stabilization time (t_{por}) has elapsed.

Table 6.1 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 $\overline{RES}$ pin, when deep software s mode is canceled, or when the watchdog timer ov The CPU enters the reset state when the $\overline{RES}$ pir
↑	Illegal instruction	Exception handling starts when an undefined code executed.
	Trace*[1]	Exception handling starts after execution of the cu instruction or exception handling, if the trace (T) b is set to 1.
	Address error	After an address error has occurred, exception ha starts on completion of instruction execution.
	Interrupt	Exception handling starts after execution of the cu instruction or exception handling, if an interrupt re occurred.*[2]
	Sleep instruction	Exception handling starts by execution of a sleep (SLEEP), if the SSBY bit in SBYCR is set to 0 and SLPIE bit in SBYCR is set to 1.
Low	Trap instruction*[3]	Exception handling starts by execution of a trap in (TRAPA).

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

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

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

Exception Source		Vector Number	Vector Table Address Offse Normal Mode*[2]	 Advanced, M Maximum*[2]
Reset		0	H'0000 to H'0001	H'0000 to H'0
Reserved for system use		1	H'0002 to H'0003	H'0004 to H'0
		2	H'0004 to H'0005	H'0008 to H'0
		3	H'0006 to H'0007	H'000C to H'
Illegal instruction		4	H'0008 to H'0009	H'0010 to H'0
Trace		5	H'000A to H'000B	H'0014 to H'0
Reserved for system use		6	H'000C to H'000D	H'0018 to H'0
Interrupt (NMI)		7	H'000E to H'000F	H'001C to H'
Trap instruction	(#0)	8	H'0010 to H'0011	H'0020 to H'0
	(#1)	9	H'0012 to H'0013	H'0024 to H'0
	(#2)	10	H'0014 to H'0015	H'0028 to H'0
	(#3)	11	H'0016 to H'0017	H'002C to H'
CPU address error		12	H'0018 to H'0019	H'0030 to H'0
DMA address error*[3]		13	H'001A to H'001B	H'0034 to H'0
UBC break interrupt		14	H'001C to H'001D	H'0038 to H'0
Reserved for system use		15 | 17	H'001E to H'001F | H'0022 to H'0023	H'003C to H' | 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, DMAC, and EXDMAC.
4. For details of internal interrupt vectors, see section 7.5, 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 be reset by the overflow that is generated in watchdog timer mode of the wa timer. For details, see section 17, Watchdog Timer (WDT).

The chip can also be reset by the exit from deep software standby mode. For details, see s 27, Power-Down Modes.

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.

6.3.1 Reset Exception Handling

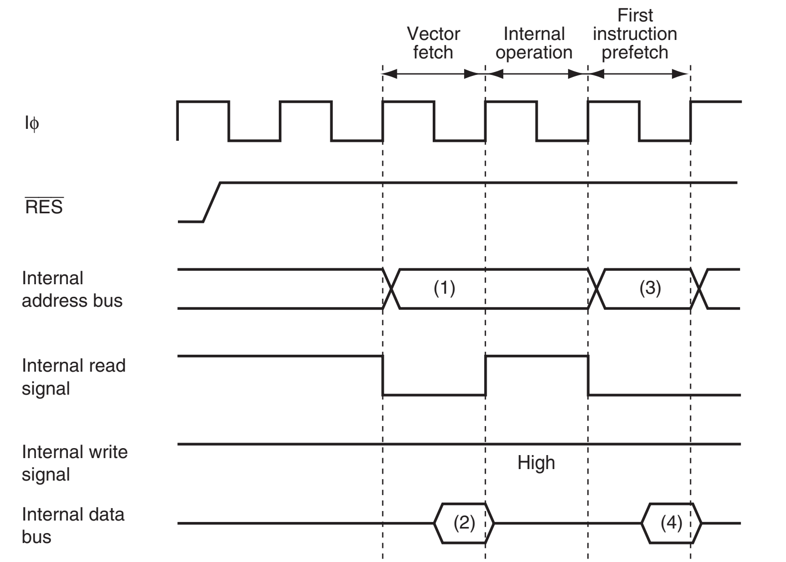
When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI starts re exception handling as follows:

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 6.1 and 6.2 show examples of the reset sequence.

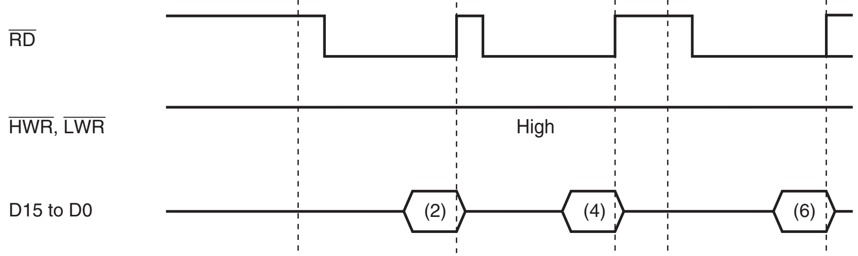
After the reset state is released, MSTPCRA and MSTPCRB are initialized to H'0FFF an
respectively, and all modules except the EXDMAC, DTC, and DMAC enter module sto

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 6.1 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 6.2 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 6.4 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	Addre
Instruction fetch	CPU	Fetches instructions from even addresses	No (no
		Fetches instructions from odd addresses	Occurs
		Fetches instructions from areas other than on-chip peripheral module space*1	No (no
		Fetches instructions from on-chip peripheral module space*1	Occurs
		Fetches instructions from external memory space in single-chip mode	Occurs
		Fetches instructions from access prohibited area.*2	Occurs
Stack operation	CPU	Accesses stack when the stack pointer value is even address	No (no
		Accesses stack when the stack pointer value is odd address.	Occurs
Data read/write	CPU	Accesses word data from even addresses	No (no
		Accesses word data from odd addresses	No (no
		Accesses external memory space in single-chip mode	Occurs
		Accesses to access prohibited area*2	Occurs

		Accesses external memory space in single-chip mode	Occur
		Accesses access prohibited area*2	Occur
		Accesses external memory space	No (n
		Accesses areas other than external memory space	Occur
Single address transfer	DMAC/ EXDMAC	Address access space is the external memory space for single address transfer	No (n
		Address access space is not the external memory space for single address transfer	Occur

Notes: 1. For on-chip peripheral module space, see section 9, Bus Controller (BSC).

2. For the access prohibited area, refer to figures 3.1 and 3.2 in section 3.4, Add

program execution starts from that address.

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

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

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

- 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 ERRF bit of EDMDR_0 in the EXDMAC is set to 1.
- The DTE bits of DMDRs for all channels in the DMAC are cleared to 0 to forcibly te transfer.
- The DTE bits of EDMDR for all channels in the EXDMAC are cleared to 0 to forcibl terminate transfer.

0: Cleared to 0

—: Retains the previous value.

NMI	NMI pin (external input)	1
UBC break interrupt	User break controller (UBC)	1
IRQ0 to IRQ11	Pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ11}}$ (external input)	12
Voltage-detection circuit	Voltage-detection circuit (LVD)*	1
On-chip peripheral module	DMA controller (DMAC)	8
	EXDMA controller (EXDMAC)	8
	Watchdog timer (WDT)	1
	A/D converter	2
	16-bit timer pulse unit (TPU)	52
	8-bit timer (TMR)	16
	Serial communications interface (SCI)	24
	I^2C bus interface 2 (IIC2)	2
	USB function module (USB)	5

Note: * Supported only by the H8SX/1655M Group.

Different vector numbers and vector table offsets are assigned to different interrupt sourc vector number and vector table offset, refer to table 7.2, Interrupt Sources, Vector Addres Offsets, and Interrupt Priority in section 7, Interrupt Controller.

3. An exception handling vector table address corresponding to the interrupt source is g
the start address of the exception service routine is loaded from the vector table to PC
program execution starts from that address.

1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the vector number speci the TRAPA instruction is generated, the start address of the exception service routine 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 3 specified in the instruction code.

Table 6.8 shows the state of CCR and EXR after execution of trap instruction exception h

Table 6.8 Status of CCR and EXR after Trap 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.

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

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

Table 6.9 shows the state of CCR and EXR after execution of sleep instruction exceptio
handling. For the detail, see section 27.10, Sleep Instruction Exception Handling.

Table 6.9 Status of CCR and EXR after Sleep Instruction 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.

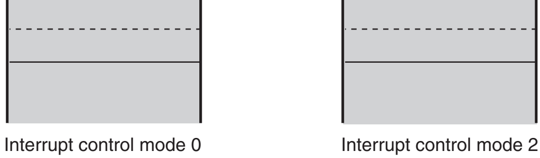
1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the occurred exception i generated, the start address of the exception service routine is loaded from the vector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

Table 6.10 shows the state of CCR and EXR after execution of illegal instruction excepti handling.

Table 6.10 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 6.3 Stack Status after Exception Handling

Use the following instructions to restore registers:

```
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.

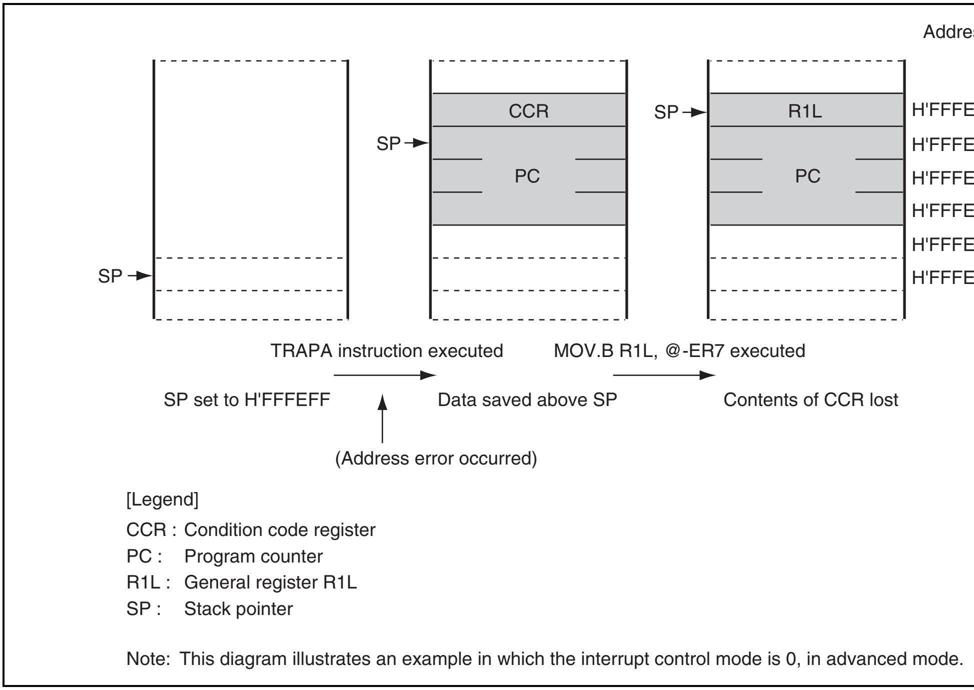


Figure 6.4 Operation when SP Value is Odd

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, DMAC, and EXDMAC)
— Sleep instruction

- Independent vector addresses

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

- Thirteen external interrupts

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falli
  detection can be selected for NMI. Falling edge, rising edge, or both edges 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, EXDMAC. The p
  level of the CPU can be automatically assigned on an exception generation. Priority
  given to the CPU interrupt exception handling over that of the DTC, DMAC, and EX
  transfer.

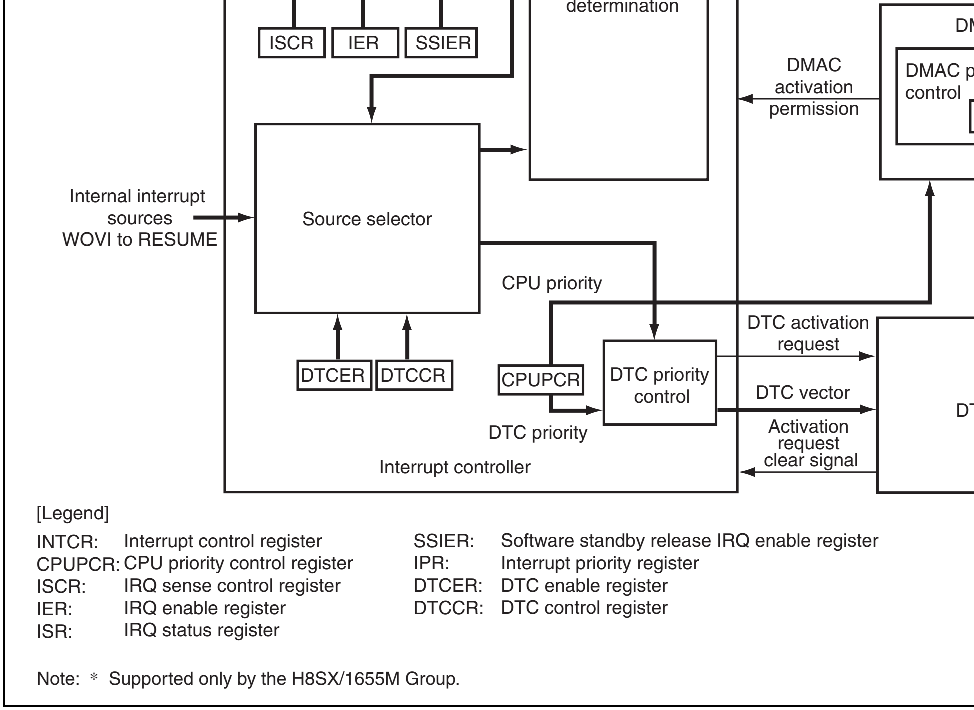


Figure 7.1 Block Diagram of Interrupt Controller

Rising, falling, or both edges, or level sensing, can be independently selected.

7.3 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 O, Q, and R (IPRA to IPRC, IPRE to IPRO, IPRQ, and IPRR)
- 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	—	0	R	Reserved
6	—	0	R	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 N 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

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	CPUPCE	0	R/W	CPU Priority Control Enable Controls the CPU priority control function. Setti to 1 enables the CPU priority control over the D DMAC, and EXDMAC. 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 C is set to 1, the CPU priority control function over DMAC, and EXDMAC becomes valid and the pri CPU processing is assigned in accordance with 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.

Bit Name	—	IPR14	IPR13	IPR12	—	IPR10	IPR9
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.

Bit	Bit Name	Initial Value	R/W	Description
				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.
2 1 0	IPR2 IPR1 IPR0	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)

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: * Supported only by the H8SX/1655M Group.

Bit	Bit Name	Initial Value	R/W	Description
15	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
14	IRQ14E*	0	R/W	IRQ14 Enable The IRQ14 interrupt request is enabled when t 1.The IRQ14 is internally connected to the volt detection interrupt.
13 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 IRQ10 interrupt request is enabled when t
10	IRQ10E	0	R/W	IRQ10 Enable The IRQ11 interrupt request is enabled when t
9	IRQ9E	0	R/W	IRQ9 Enable The IRQ9 interrupt request is enabled when th
8	IRQ8E	0	R/W	IRQ8 Enable The IRQ8 interrupt request is enabled when th

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

Note: * Supported only by the H8SX/1655M Group.

Bit	15	14	13	12	11	10	9
Bit Name	—	—	IRQ14SR*	IRQ14SF*	—	—	—
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

Note: * Supported only by the H8SX/1655M Group.

				edge. 00: Initial value 01: Interrupt request generated at falling edge 10: Setting prohibited 11: Setting prohibited
11 to 8	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
7 6	IRQ11SR IRQ11SF	0 0	R/W R/W	IRQ11 Sense Control Rise IRQ11 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 $\overline{IRQ11}$
5 4	IRQ10SR IRQ10SF	0 0	R/W R/W	IRQ10 Sense Control Rise 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 $\overline{IRQ10}$

00: Interrupt request generated by low level o

01: Interrupt request generated at falling edge

10: Interrupt request generated at rising edge

11: Interrupt request generated at both falling edges of $\overline{IRQ8}$

Note: Supported only by the H8SX/1655M Group.

13	IRQ6SR	0	R/W	IRQ6 Sense Control Rise
12	IRQ6SF	0	R/W	IRQ6 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 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 $\overline{\text{IRQ6}}$
11	IRQ5SR	0	R/W	IRQ5 Sense Control Rise
10	IRQ5SF	0	R/W	IRQ5 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 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 $\overline{\text{IRQ5}}$
9	IRQ4SR	0	R/W	IRQ4 Sense Control Rise
8	IRQ4SF	0	R/W	IRQ4 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 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 $\overline{\text{IRQ4}}$
7	IRQ3SR	0	R/W	IRQ3 Sense Control Rise
6	IRQ3SF	0	R/W	IRQ3 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 $\overline{\text{IRQ3}}$

				00: Interrupt request generated by low level of $\overline{I}$
				01: Interrupt request generated at falling edge o
				10: Interrupt request generated at rising edge o
				11: Interrupt request generated at both falling a edges of $\overline{IRQ1}$
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 $\overline{I}$ 01: Interrupt request generated at falling edge o 10: Interrupt request generated at rising edge o 11: Interrupt request generated at both falling a edges of $\overline{IRQ0}$

Bit Name	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	I
Initial Value	0	0	0	0	0	0	0	
R/W	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/

Notes: 1. Only 0 can be written, to clear the flag. The bit manipulation instructions or memory operation instructi be used to clear the flag.
2. Supported only by the H8SX/1655M Group.

Bit	Bit Name	Initial Value	R/W	Description
15	—	All 0	R/(W)*1	Reserved These bits are always read as 0. The write valu always be 0.
14	IRQ14F*2	0	R/(W)*1	[Setting condition] • When the interrupt selected by ISCR occur [Clearing conditions] • Writing 0 after reading IRQ14F = 1 • When IRQ14 interrupt exception handling i executed while falling edge sensing is sele
13, 12	—	All 0	R/(W)*1	Reserved These bits are always read as 0. The write valu always be 0.

3	IRQ3F	0	R/(W)*1	while falling-, rising-, or both-edge sensing selected.
2	IRQ2F	0	R/(W)*1	• When the DTC is activated by an IRQn int
1	IRQ1F	0	R/(W)*1	and the DISEL bit in MRB of the DTC is cl
0	IRQ0F	0	R/(W)*1	

Notes: 1. Only 0 can be written, to clear the flag.
2. Supported only by the H8SX/1655M Group.

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	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 val always be 0.
11	SSI11	0	R/W	Software Standby Release IRQ Setting
10	SSI10	0	R/W	These bits select the IRQn interrupt used to le software standby mode (n = 11 to 0).
9	SSI9	0	R/W	
8	SSI8	0	R/W	0: An IRQn request is not sampled in software mode
7	SSI7	0	R/W	
6	SSI6	0	R/W	1: When an IRQn request occurs in software s mode, this LSI leaves software standby mo the oscillation settling time has elapsed
5	SSI5	0	R/W	
4	SSI4	0	R/W	
3	SSI3	0	R/W	
2	SSI2	0	R/W	
1	SSI1	0	R/W	
0	SSI0	0	R/W	

the CPU regardless of the interrupt control mode 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 of DTCCR in the DTC to 1.
- Sets the ERRF bit of DMDR_0 in DMAC to 1.
- Sets the ERRF bit of EDMDR_0 in the EXDMAC to 1
- Clears the DTE bits of DMDRs for all channels in the DMAC to 0 to forcibly termin transfer
- Clears the DTE bits of EDMDRs for all channels in the EXDMAC to 0 to forcibly te transfer

(2) IRQn Interrupts

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

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

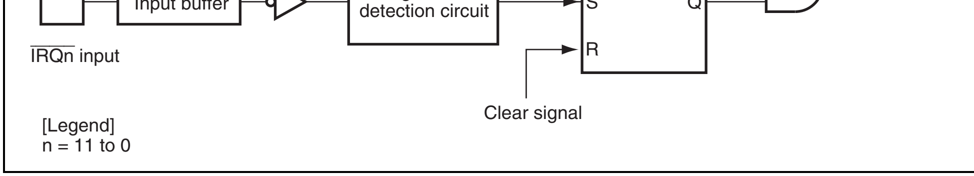


Figure 7.2 Block Diagram of Interrupts IRQn

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

7.4.2 Internal Interrupts

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

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

Classification	Interrupt Source	Vector Number	Vector Address Offset*1 Advanced Mode, Middle Mode, Maximum Mode	IPR	Priority	DTC Activation
External pin	NMI	7	H'001C	—	High	—
UBC	UBC break interrupt	14	H'0038	—	↑	—
External pin	IRQ0	64	H'0100	IPRA14 to IPRA12		O
	IRQ1	65	H'0104	IPRA10 to IPRA8		O
	IRQ2	66	H'0108	IPRA6 to IPRA4		O
	IRQ3	67	H'010C	IPRA2 to IPRA0		O
	IRQ4	68	H'0110	IPRB14 to IPRB12		O
	IRQ5	69	H'0114	IPRB10 to IPRB8		O
	IRQ6	70	H'0118	IPRB6 to IPRB4		O
	IRQ7	71	H'011C	IPRB2 to IPRB0		O
	IRQ8	72	H'0120	IPRC14 to IPRC12		O
	IRQ9	73	H'0124	IPRC10 to IPRC8		O
	IRQ10	74	H'0128	IPRC6 to IPRC4		O
	IRQ11	75	H'012C	IPRC2 to IPRC0		O
—	Reserved for system use	76	H'0130	—		—
		77	H'0134			—
LVD*2	Voltage-monitoring interrupt (IRQ14)	78	H'0138	IPRD6 to IPRD4		—
—	Reserved for system use	79	H'013C	—		—
		80	H'0140			—
WDT	WOVI	81	H'0144	IPRE10 to IPRE8	Low	—

TPU_0	TGI0A	88	H'0160	IPRF6 to IPRF4		O
	TGI0B	89	H'0164			O
	TGI0C	90	H'0168			O
	TGI0D	91	H'016C			O
	TCI0V	92	H'0170			—
TPU_1	TGI1A	93	H'0174	IPRF2 to IPRF0		O
	TGI1B	94	H'0178			O
	TCI1V	95	H'017C			—
	TCI1U	96	H'0180			—
TPU_2	TGI2A	97	H'0184	IPRG14 to IPRG12		O
	TGI2B	98	H'0188			O
	TCI2V	99	H'018C			—
	TCI2U	100	H'0190			—
TPU_3	TGI3A	101	H'0194	IPRG10 to IPRG8		O
	TGI3B	102	H'0198			O
	TGI3C	103	H'019C			O
	TGI3D	104	H'01A0			O
	TCI3V	105	H'01A4			—
TPU_4	TGI4A	106	H'01A8	IPRG6 to IPRG4		O
	TGI4B	107	H'01AC			O
	TCI4V	108	H'01B0			—
	TCI4U	109	H'01B4		Low	—

TMR_0	CMI0A	116	H'01D0	IPRH14 to IPRH12		O
	CMI0B	117	H'01D4			O
	OV0I	118	H'01D8			—
TMR_1	CMI1A	119	H'01DC	IPRH10 to IPRH8		O
	CMI1B	120	H'01E0			O
	OV1I	121	H'01E4			—
TMR_2	CMI2A	122	H'01E8	IPRH6 to IPRH4		O
	CMI2B	123	H'01EC			O
	OV2I	124	H'01F0			—
TMR_3	CMI3A	125	H'01F4	IPRH2 to IPRH0		O
	CMI3B	126	H'01F8			O
	OV3I	127	H'01FC			—
DMAC	DMTEND0	128	H'0200	IPRI14 to IPRI12		O
	DMTEND1	129	H'0204	IPRI10 to IPRI8		O
	DMTEND2	130	H'0208	IPRI6 to IPRI4		O
	DMTEND3	131	H'020C	IPRI2 to IPRI0		O
EXDMAC	EXDMTEND0	132	H'0210	IPRJ14 to IPRJ12		O
	EXDMTEND1	133	H'0214	IPRJ10 to IPRJ8		O
	EXDMTEND2	134	H'0218	IPRJ6 to IPRJ4		O
	EXDMTEND3	135	H'021C	IPRJ2 to IPRJ0		O
DMAC	DMEEND0	136	H'0220	IPRK14 to IPRK12		O
	DMEEND1	137	H'0224			O
	DMEEND2	138	H'0228			O
	DMEEND3	139	H'022C		Low	O

	RXI0	145	H'0244			O
	TXI0	146	H'0248			O
	TEI0	147	H'024C			—
SCI_1	ERI1	148	H'0250	IPRK2 to IPRK0		—
	RXI1	149	H'0254			O
	TXI1	150	H'0258			O
	TEI1	151	H'025C			—
SCI_2	ERI2	152	H'0260	IPRL14 to IPRL12		—
	RXI2	153	H'0264			O
	TXI2	154	H'0268			O
	TEI2	155	H'026C			—
—	Reserved for system use	156	H'0270	—		—
		157	H'0274			—
		158	H'0278			—
		159	H'027C			—
SCI_4	ERI4	160	H'0280	IPRL6 to IPRL4		—
	RXI4	161	H'0284			O
	TXI4	162	H'0288			O
	TEI4	163	H'028C			—
TPU_6	TGI6A	164	H'0290	IPRL2 to IPRL0		O
	TGI6B	165	H'0294			O
	TGI6C	166	H'0298			O
	TGI6D	167	H'029C			O
	TGI6V	168	H'02A0	IPRM14 to IPRM12	Low	—

	TGI8B	174	H'02B8			O
	TCI8V	175	H'02BC	IPRN14 to IPRN12		—
	TCI8U	176	H'02C0			—
TPU_9	TGI9A	177	H'02C4	IPRN10 to IPRN8		O
	TGI9B	178	H'02C8			O
	TGI9C	179	H'02CC			O
	TGI9D	180	H'02D0			O
	TCI9V	181	H'02D4	IPRN6 to IPRN4		—
TPU_10	TGI10A	182	H'02D8	IPRN2 to IPRN0		O
	TGI10B	183	H'02DC			O
	Reserved for system use	184	H'02E0	—		—
	Reserved for system use	185	H'02E4			—
	TCI10V	186	H'02E8	IPRO14 to IPRO12		O
	TCI10U	187	H'02EC			—
TPU_11	TGI11A	188	H'02F0	IPRO10 to IPRO8		O
	TGI11B	189	H'02F4			O
	TCI11V	190	H'02F8	IPRO6 to IPRO4		—
	TCI11U	191	H'02FC			—
—	Reserved for system use	192 | 215	H'0300 | H'035C	—	Low	— | —

SCI_5	RXI5	220	H'0370	IPRQ2 to IPRQ0		—	O
	TXI5	221	H'0374			—	O
	ERI5	222	H'0378			—	—
	TEI5	223	H'037C			—	—
SCI_6	RXI6	224	H'0380	IPRR14 to IPRR12		—	O
	TXI6	225	H'0384			—	O
	ERI6	226	H'0388			—	—
	TEI6	227	H'038C			—	—
TMR_4	CMIA4 or CMIB4	228	H'0390	IPRR10 to IPRR8		—	—
TMR_5	CMIA5 or CMIB5	229	H'0394			—	—
TMR_6	CMIA6 or CMIB6	230	H'0398			—	—
TMR_7	CMIA7 or CMIB7	231	H'039C			—	—
USB	USBINTN0	232	H'03A0	IPRR6 to IPRR4		—	O
	USBINTN1	233	H'03A4			—	O
	USBINTN2	234	H'03A8			—	—
	USBINTN3	235	H'03AC			—	—
—	Reserved for system use	236	H'03B0	IPRR2 to IPRR0		—	—
A/D_1	ADI1	237	H'03B4			—	O
USB	resume	238	H'03B8			—	—
—	Reserved for system use	239 | 255	H'03BC | H'03FC	—	Low	— | —	—

Notes: 1. Lower 16 bits of the start address in advanced, middle, and maximum modes.
2. Supported only by the H8SX/1655M Group.

Mode	Register	Mask Bit	Description
0	Default	I	The priority levels of the interrupt sources are default settings. The interrupts except for NMI is masked by th
2	IPR	I2 to I0	Eight priority levels can be set for interrupt so except for NMI with IPR. 8-level interrupt mask control is performed by I0.

7.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI are masked by the I bit in the CPU. Figure 7.3 shows a flowchart of the interrupt acceptance operation in this case

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

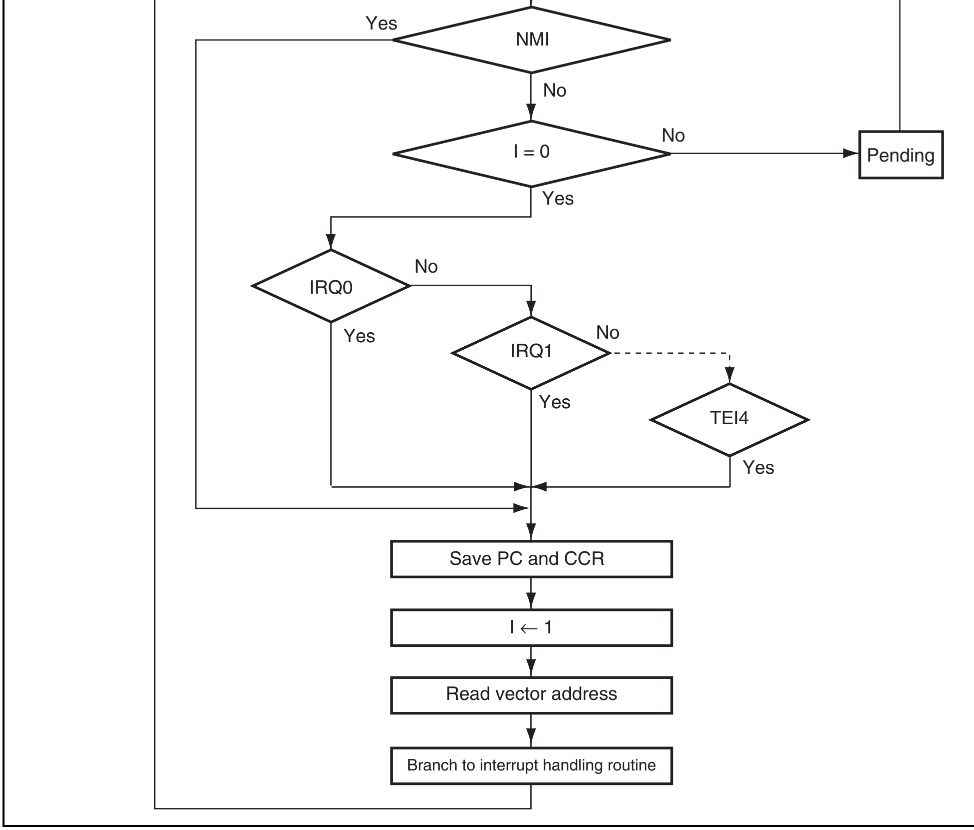


Figure 7.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

multiple interrupt requests have the same priority, an interrupt request is selected ac
the default setting shown in table 7.2.

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

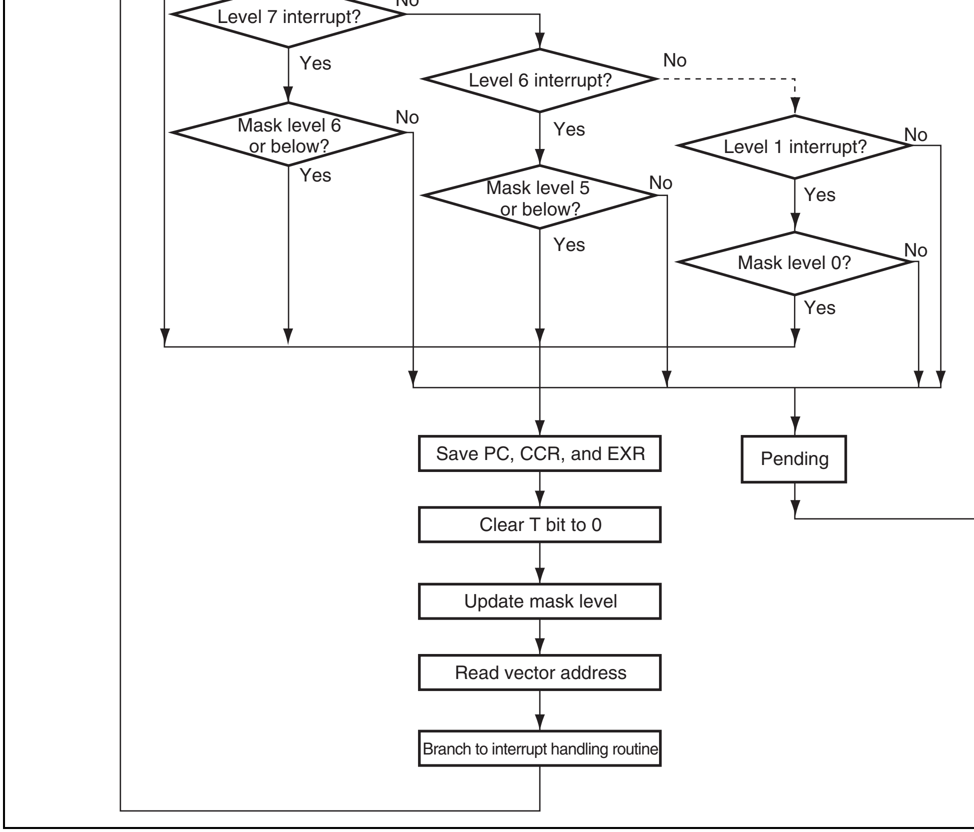


Figure 7.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

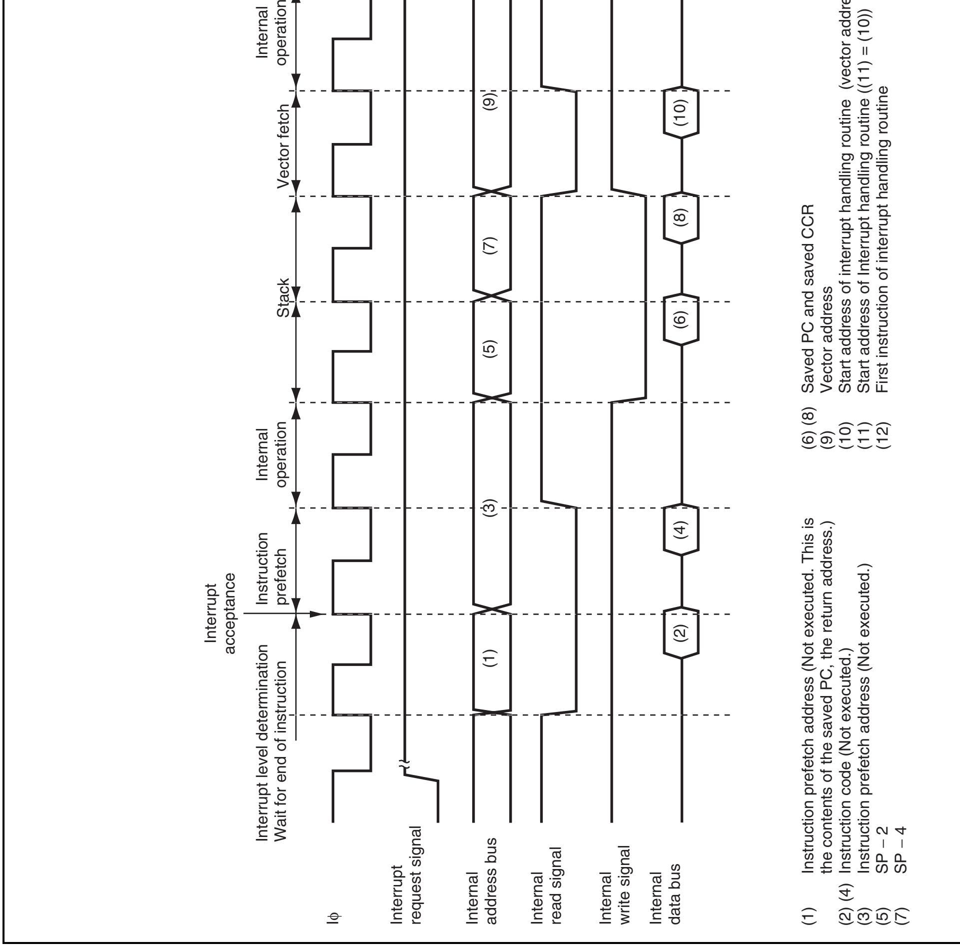


Figure 7.5 Interrupt Exception Handling

	Normal Mode*		Advanced Mode		Maximum
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 handling
4. Internal operation after interrupt acceptance or after vector fetch
5. Not available in this LSI.
6. When setting the SP value to 4n, the interrupt response time is S_K; when settin
2, the interrupt response time is $2 \cdot S_K$.

[Legend]

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

7.6.5 DTC and DMAC Activation by Interrupt

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

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

For details on interrupt requests that can be used to activate the DTC and DMAC, see ta
section 10, DMA Controller (DMAC), and section 12, Data Transfer Controller (DTC).

Figure 7.6 shows a block diagram of the DTC, DMAC, and interrupt controller.

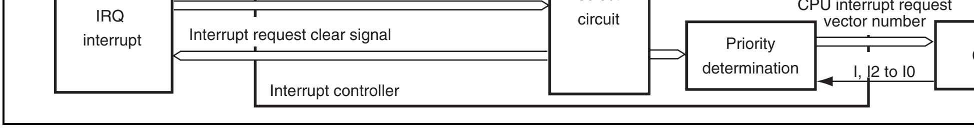


Figure 7.6 Block Diagram of DTC, DMAC, and Interrupt Controller

(1) Selection of Interrupt Sources

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

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

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

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

When the same interrupt source is set as both the DTC and DMAC activation source and
interrupt source, the DTC and DMAC must be given priority over the CPU. If the IPSETI
CPUPCR is set to 1, the priority is determined according to the IPR setting. Therefore, th
setting or the IPR setting corresponding to the interrupt source must be set to lower than o
to the DTCP and DMAP setting. If the CPU is given priority over the DTC or DMAC, th
DMAC may not be activated, and the data transfer may not be performed.

are performed independently.

Table 7.6 lists the selection of interrupt sources and interrupt source clear control by sett
DTA bit in DMDR of the DMAC, the DTCE bit in DTCERA to DTCERF of the DTC, a
DISEL bit in MRB of the DTC.

Table 7.6 Interrupt Source Selection and Clear Control

DMAC Setting	DTC Setting		Interrupt Source Selection/Clear		
DTA	DTCE	DISEL	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.

(4) Usage Note

The interrupt sources of the SCI, and A/D converter are cleared according to the setting
table 7.6, when the DTC or DMAC reads/writes the prescribed register.

To initiate multiple channels for the DTC with the same interrupt, the same priority (DT
DMAP) should be assigned.

EXDMAC is assigned by bits EDMAP2 to EDMAP0 in the EXDMA mode control regist
(EDMDR_0 to EDMDE_3) for each channel.

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

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

For the DMAC, the priority level can be specified for each channel. The DMAC activatio
is controlled according to the priority level of each DMAC channel indicated by bits DM
DMAP0 and the priority level of the CPU. If the CPU has priority, the DMAC activation
held. The DMAC is activated when the condition by which the activation source is held i
cancelled (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 higher
levels continue transfer and the activation sources for the channels of lower priority level
that of the CPU are held.

For the EXDMAC, the priority level can be specified for each channel. The EXDMAC ac
source is controlled according to the priority level of each EXDMAC channel indicated b
EDMAP2 to EDMAP0 and the priority level of the CPU. If the CPU has priority, the EX
activation source is held. The EXDMAC is activated when the condition by which the act
source is held is cancelled (CPUPCE = 1 and value of bits CPUP2 to CPUP0 is greater th
bits DMAP2 to EDMAP0). If different priority levels are specified for channels, the chan

The priority level that is automatically assigned when the IPSETE bit is 1 differs accord
interrupt control mode.

In interrupt control mode 0, the I bit in CCR of the CPU is reflected in bit CPUP2. Bits
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 7.7 shows the CPU priority control.

Table 7.7 CPU Priority Control

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

[illegible]	0	Any	Any	Any	Any	Enabled	Enabled	E
	1	B'000	B'000	B'000	B'000	Enabled	Enabled	E
		B'100	B'000	B'000	B'000	Masked	Masked	M
		B'100	B'000	B'011	B'100	Masked	Masked	E
		B'100	B'111	B'101	B'000	Enabled	Enabled	M
		B'000	B'111	B'101	B'000	Enabled	Enabled	E
2	0	Any	Any	Any	Any	Enabled	Enabled	E
	1	B'000	B'000	B'000	B'000	Enabled	Enabled	E
		B'000	B'011	B'101	B'110	Enabled	Enabled	E
		B'011	B'011	B'101	B'110	Enabled	Enabled	E
		B'100	B'011	B'101	B'110	Masked	Enabled	E
		B'101	B'011	B'101	B'110	Masked	Enabled	E
		B'110	B'011	B'101	B'110	Masked	Masked	E
		B'111	B'011	B'101	B'110	Masked	Masked	M
		B'101	B'011	B'101	B'011	Masked	Enabled	M
		B'101	B'110	B'101	B'011	Enabled	Enabled	M

be executed on completion of the instruction. However, if there is an interrupt request w
over that interrupt, interrupt exception handling will be executed for the interrupt with p
and another interrupt will be ignored. The same also applies when an interrupt source fla
cleared to 0. Figure 7.7 shows an example in which the TCIEV bit in TIER of the TPU i
to 0. The above conflict will not occur if an enable bit or interrupt source flag is cleared
the interrupt is masked.

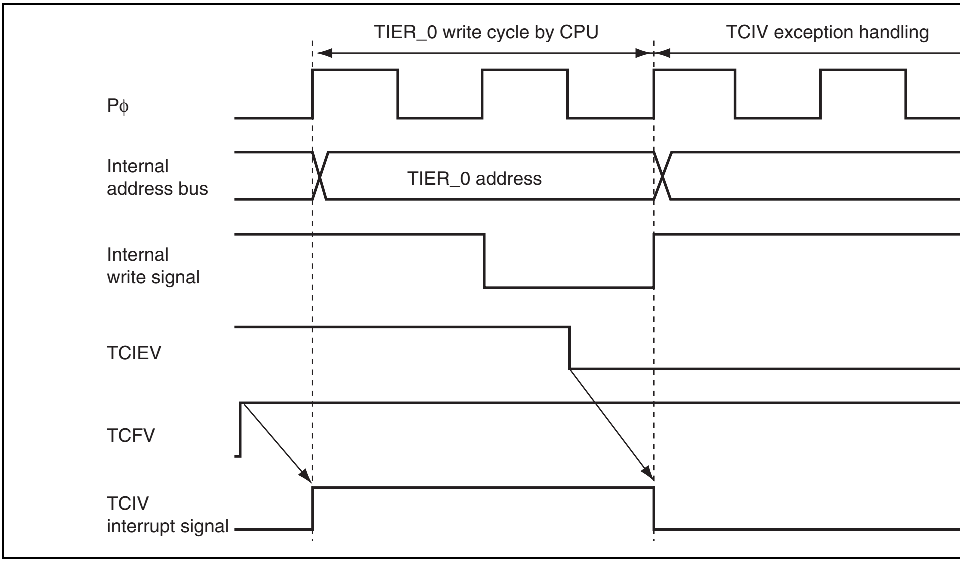


Figure 7.7 Conflict between Interrupt Generation and Disabling

Similarly, when an interrupt is requested immediately before the DTC enable bit is chan
activate the DTC, DTC activation and the interrupt exception handling by the CPU are b
executed. When changing the DTC enable bit, make sure that an interrupt is not requeste

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU ha updated the mask level with an LDC, ANDC, ORC, or XORC instruction, and for a perio writing to the registers of the interrupt controller.

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

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

7.8.5 Interrupts during Execution of MOVMD and MOVSD Instructions

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

8.1 Features

- Number of break channels: four (channels A, B, C, and D)
- Break comparison conditions (each channel)
  - — Address
  - — Bus master (CPU cycle)
  - — Bus cycle (instruction execution (PC break))
- UBC break interrupt exception handling is executed immediately before execution o
  instruction fetched from the specified address (PC break).
- Module stop state can be set

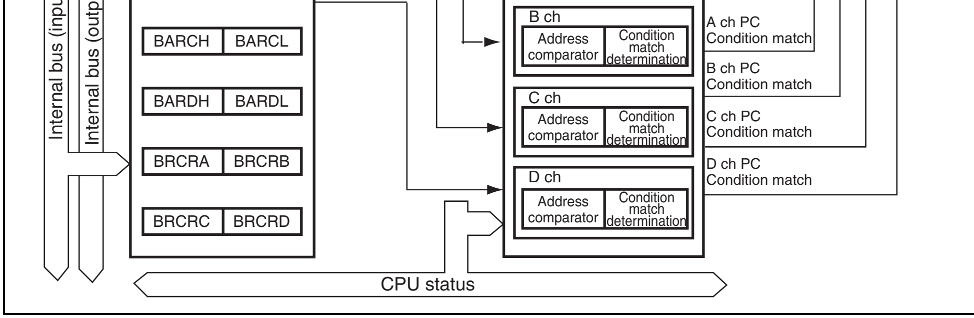


Figure 8.1 Block Diagram of UBC

	BAMRAL	R/W	H'0000	H'FFA06
Break address register B	BARBH	R/W	H'0000	H'FFA08
	BARBL	R/W	H'0000	H'FFA0A
Break address mask register B	BAMRBH	R/W	H'0000	H'FFA0C
	BAMRBL	R/W	H'0000	H'FFA0E
Break address register C	BARCH	R/W	H'0000	H'FFA10
	BARCL	R/W	H'0000	H'FFA12
Break address mask register C	BAMRCH	R/W	H'0000	H'FFA14
	BAMRCL	R/W	H'0000	H'FFA16
Break address register D	BARDH	R/W	H'0000	H'FFA18
	BARDL	R/W	H'0000	H'FFA1A
Break address mask register D	BAMRDH	R/W	H'0000	H'FFA1C
	BAMRDL	R/W	H'0000	H'FFA1E
Break control register A	BRCRA	R/W	H'0000	H'FFA28
Break control register B	BRCRB	R/W	H'0000	H'FFA2C
Break control register C	BRCRC	R/W	H'0000	H'FFA30
Break control register D	BRCRD	R/W	H'0000	H'FFA34

BARnL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	BARn15	BARn14	BARn13	BARn12	BARn11	BARn10	BARn9	BARn8	BARn7	BARn6	BARn5	BARn4	BARn3	BARn2	BARn1
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- BARnH

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BARn31 to BARn16	All 0	R/W	Break Address n31 to 16 These bits hold the upper bit values (bits 31 to the address break-condition on channel n.

[Legend]

n = Channels A to D

- BARnL

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	BARn15 to BARn0	All 0	R/W	Break Address n15 to 0 These bits hold the lower bit values (bits 15 to the address break-condition on channel n.

[Legend]

n = Channels A to D

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	BAMRn15	BAMRn14	BAMRn13	BAMRn12	BAMRn11	BAMRn10	BAMRn9	BAMRn8	BAMRn7	BAMRn6	BAMRn5	BAMRn4	BAMRn3	BAMRn2	BAMI
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/\

• BAMRnH

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BAMRn31 to BAMRn16	All 0	R/W	Break Address Mask n31 to 16 Be sure to write H'FF00 here before setting a condition in the break control register.

[Legend]

n = Channels A to D

• BAMRnL

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	BAMRn15 to BAMRn0	All 0	R/W	Break Address Mask n15 to 0 Be sure to write H'0000 here before setting a condition in the break control register.

[Legend]

n = Channels A to D

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R/W	Reserved
14	—	0	R/W	These bits are always read as 0. The write val should always be 0.
13	CMFCPn	0	R/W	Condition Match CPU Flag UBC break source flag that indicates satisfact specified CPU bus cycle condition. 0: The CPU cycle condition for channel n brea requests has not been satisfied. 1: The CPU cycle condition for channel n brea requests has been satisfied.
12	—	0	R/W	Reserved These bits are always read as 0. The write val should always be 0.
11	CPn2	0	R/W	CPU Cycle Select
10	CPn1	0	R/W	These bits select CPU cycles as the bus cycle condition for the given channel.
9	CPn0	0	R/W	000: Break requests will not be generated. 001: The bus cycle break condition is CPU cy 01x: Setting prohibited 1xx: Setting prohibited
8	—	0	R/W	Reserved
7	—	0	R/W	These bits are always read as 0. The write val should always be 0.
6	—	0	R/W	

				condition for the given channel. 00: Break requests will not be generated. 01: The bus cycle break condition is read cyc 1x: Setting prohibited
1	—	0	R/W	Reserved
0	—	0	R/W	These bits are always read as 0. The write va should always be 0.

[Legend]

n = Channels A to D

consist of CPU cycle, PC break, and reading. Condition comparison is not performed
CPU cycle setting is CPn = B'000, the PC break setting is IDn = B'00, or the read setti
RWn = B'00.

3. The condition match CPU flag (CMFCPn) is set in the event of a break condition mat
corresponding channel. These flags are set when the break condition matches but are
cleared when it no longer does. To confirm setting of the same flag again, read the fla
from the break interrupt handling routine, and then write 0 to it (the flag is cleared by
to it after reading it as 1).

[Legend]

n = Channels A to D

8.4.2 PC Break

1. When specifying a PC break, specify the address as the first address of the required in
If the address for a PC break condition is not the first address of an instruction, a brea
never be generated.
2. The break occurs after fetching and execution of the target instruction have been conf
cases of contention between a break before instruction execution and a user maskable
priority is given to the break before instruction execution.
3. A break will not be generated even if a break before instruction execution is set in a d
4. The PC break condition is generated by specifying CPU cycles as the bus condition in
control register n (BRCRn.CPn0 = 1), PC break as the break condition (IDn0 = 1), an
cycles as the bus-cycle condition (RWn0 = 1).

[Legend]

n = Channels A to D

		for channel B
BRCRC	CMFCPC (bit 13)	Indicates that the condition matches in the CPU for channel C
BRCRD	CMFCPD (bit 13)	Indicates that the condition matches in the CPU for channel D

oscillation settling time has elapsed subsequent to the transition to software standb

When an interrupt is the canceling source, interrupt exception handling is executed RTE instruction, and the instruction following the SLEEP instruction is then execu

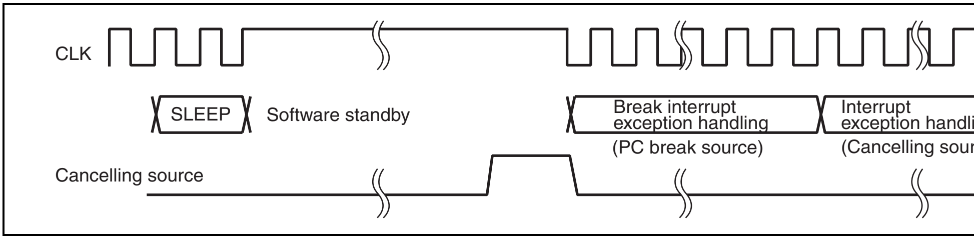


Figure 8.2 Contention between SLEEP Instruction (Software Standby) and PC

2. Prohibition on Setting of PC Break
   — Setting of a UBC break interrupt for program within the UBC break interrupt hand routine is prohibited.
3. The procedure for clearing a UBC flag bit (condition match flag) is shown below. A f cleared by writing 0 to it after reading it as 1. As the register that contains the flag bits accessible in byte units, bit manipulation instructions can be used.

Figure 8.3 Flag Bit Clearing Sequence (Condition Match Flag)

4. After setting break conditions for the UBC, an unexpected UBC break interrupt may after the execution of an illegal instruction. This depends on the value of the progran and the internal bus cycle.

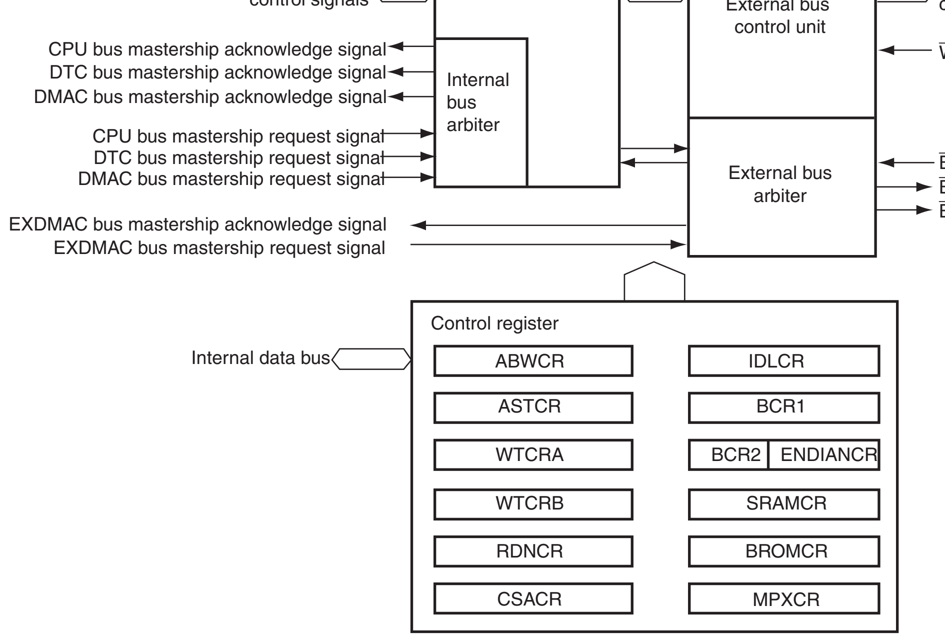
- Manages external address space in area units
  Manages the external address space divided into eight areas
  Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for each area
  Bus specifications can be set independently for each area
  8-bit access or 16-bit access can be selected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

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

- External bus release function
- Bus arbitration function

  Includes a bus arbiter that arbitrates bus mastership among the CPU, DMAC, EXDM
  DTC, and external bus master
- EXDMAC transfers to the external buses and internal accesses can be executed in par
- 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.



[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
IDLCD: 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 registe
MPXCR: Address/data multiplexed I/O contro

Figure 9.1 Block Diagram of Bus Controller

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

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

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

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

[Legend]

×: Don't care

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

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

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

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

				001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
11	—	0	R	Reserved This is a read-only bit and cannot be modified.
10 9 8	W22 W21 W20	1 1 1	R/W R/W R/W	Area 2 Wait Control 2 to 0 These bits select the number of program wait c when accessing area 2 while bit AST2 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
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 when accessing area 0 while bit AST0 in AST 000: Program wait cycle not inserted 001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted

Bit	7	6	5	4	3	2	1
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	RDN7 to RDN0 set the negation timing of the strobe in a corresponding area read access.
13	RDN5	0	R/W	As shown in figure 9.2, the read strobe for an which the RDNn bit is set to 1 is negated one cycle earlier than that for an area for which the bit is cleared to 0. The read data setup and ho are also given one half-cycle earlier.
12	RDN4	0	R/W	0: In an area n read access, the $\overline{RD}$ signal is n at the end of the read cycle
11	RDN3	0	R/W	1: In an area n read access, the $\overline{RD}$ signal is n one half-cycle before the end of the read cy
10	RDN2	0	R/W	(n = 7 to 0)
9	RDN1	0	R/W	
8	RDN0	0	R/W	
7 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modifi

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

2. In an external address space which is specified as burst ROM interface, the R setting is ignored during read accesses by the CPU and EXDMAC cluster tran the same operation when RDNn = 0 is performed.

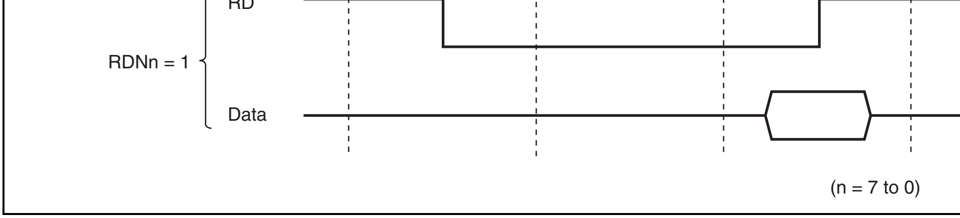


Figure 9.2 Read Strobe Negation Timing (Example of 3-State Access Spac

9.2.5 $\overline{\text{CS}}$ 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	$\overline{CS}$ and Address Signal Assertion Period Contr
6	CSXT6	0	R/W	These bits specify whether or not the Tt cycle is
5	CSXT5	0	R/W	inserted (see figure 9.3). When an area for whic
4	CSXT4	0	R/W	CSXTn is set to 1 is accessed, one Tt cycle, in
3	CSXT3	0	R/W	the $\overline{CSn}$ and address signals are retained, is in after the normal access cycle.
2	CSXT2	0	R/W	0: In access to area n, the $\overline{CSn}$ and address as
1	CSXT1	0	R/W	period (Tt) is not extended
0	CSXT0	0	R/W	1: In access to area n, the $\overline{CSn}$ and address as period (Tt) is extended (n = 7 to 0)

Note: * In burst ROM interface, the CSXTn settings are ignored during read accesses CPU and EXDMAC cluster transfer

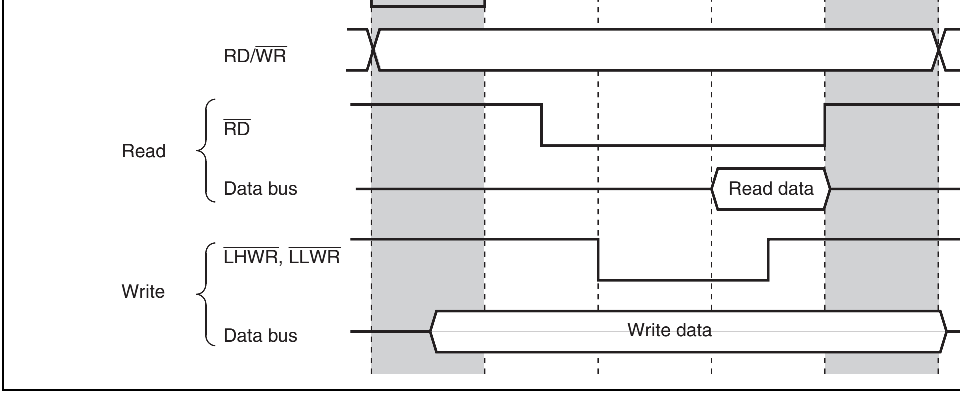


Figure 9.3 $\overline{CS}$ and Address Assertion Period Extension (Example of Basic Bus Interface, 3-State Access Space, and RDNn = 0)

Bit Name	IDLSEL7	IDLSEL6	IDLSEL5	IDLSEL4	IDLSEL3	IDLSEL2	IDLSEL1	IL
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	IDLS3	1	R/W	Idle Cycle Insertion 3 Inserts an idle cycle between the bus cycles wh DMAC/EXDMAC single address transfer (write followed by 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 wh external write cycle is followed by external read 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 wh external read cycles of different areas continue 0: No idle cycle is inserted 1: An idle cycle is inserted

				00: No idle cycle is inserted 01: 2 idle cycles are inserted 00: 3 idle cycles are inserted 01: 4 idle cycles are inserted
9 8	IDLCA1 IDLCA0	1 1	R/W R/W	Idle Cycle State Number Select A Specifies the number of idle cycles to be inser the 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 6 5 4 3 2 1 0	IDLSEL7 IDLSEL6 IDLSEL5 IDLSEL4 IDLSEL3 IDLSEL2 IDLSEL1 IDLSEL0	0 0 0 0 0 0 0 0	R/W R/W R/W R/W R/W R/W R/W R/W	Idle Cycle Number Select Specifies the number of idle cycles to be inser each area for the idle insertion condition speci IDLS1 and IDLS0. 0: Number of idle cycles to be inserted for are specified by IDLCA1 and IDLCA0. 1: Number of idle cycles to be inserted for are specified by IDLCB1 and IDLCB0. (n = 7 to 0)

Bit	7	6	5	4	3	2	1
Bit Name	DKC	EDKC	—	—	—	—	—
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 us ports 1: External bus release enabled* For details, see section 13, I/O Ports.
14	BREQOE	0	R/W	$\overline{\text{BREQO}}$ Pin Enable Controls outputting the bus request signal (BR the external bus master in the external bus rel state when an internal bus master performs a external 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

				The changed setting may not affect an externa immediately after the change. 0: Write data buffer function not used 1: Write data buffer function used
8	WAITE	0	R/W	$\overline{\text{WAIT}}$ Pin Enable Selects enabling/disabling of wait input by the pin. 0: Wait input by $\overline{\text{WAIT}}$ pin disabled $\overline{\text{WAIT}}$ pin can be used as I/O port 1: Wait input by $\overline{\text{WAIT}}$ pin enabled For details, see section 13, I/O Ports.
7	DKC	0	R/W	$\overline{\text{DACK}}$ Control Selects the timing of DMAC transfer acknowle signal assertion. 0: $\overline{\text{DACK}}$ signal is asserted at the Bϕ falling ed 1: $\overline{\text{DACK}}$ signal is asserted at the Bϕ rising ed
6	EDKC	0	R/W	$\overline{\text{EDACK}}$ Control Selects the timing of EXDMAC transfer acknov signal assertion. 0: $\overline{\text{EDACK}}$ signal is asserted at the Bϕ falling e 1: $\overline{\text{EDACK}}$ signal is asserted at the Bϕ rising e
5 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modifi

Note: When external bus release is enabled or input by the $\overline{\text{WAIT}}$ pin is enabled, make the ICR bit to 1. For details, see section 13, I/O Ports.

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

R/W: R/W R/W R/W R/W R/W R/W R/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 en
4	LE4	0	R/W	1: Data format of area n is specified as little e
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 modifi

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

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

Bit	Bit Name	Initial Value	R/W	Description
15	BCSEL7	0	R/W	Byte Control SRAM Interface Select
14	BCSEL6	0	R/W	Selects the bus interface for the corresponding
13	BCSEL5	0	R/W	When setting a bit to 1, the bus interface select BROMCR and MPXCR must 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 modifie

Bit Name	BSRM1	BSTS12	BSTS11	BSTS10	—	—	BSWD11
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSRM0	0	R/W	Area 0 Burst ROM Interface Select Specifies the area 0 bus interface. To set this clear bit BCSEL0 in SRAMCR to 0. 0: Basic bus interface or byte-control SRAM in 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 modifi

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

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	MPXE7	0	R/W	Address/Data Multiplexed I/O Interface Select
14	MPXE6	0	R/W	Specifies the bus interface for the correspondi
13	MPXE5	0	R/W	To set this bit to 1, clear the BCSELn bit in SF 0.
12	MPXE4	0	R/W	0: Area n is specified as a basic interface or a control SRAM interface.
11	MPXE3	0	R/W	1: Area n is specified as an address/data mult I/O interface
				(n = 7 to 3)
10 to 1	—	All 0	R	Reserved
				These are read-only bits and cannot be modifi
0	ADDEX	0	R/W	Address Output Cycle Extension
				Specifies whether a wait cycle is inserted for t address output cycle of address/data multiplex interface.
				0: No wait cycle is inserted for the address ou
				1: One wait cycle is inserted for the address o cycle

- Internal peripheral bus

  A bus that accesses registers in the bus controller, interrupt controller, DMAC, and EX
  and registers of peripheral modules such as SCI and timer.

- External access bus

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

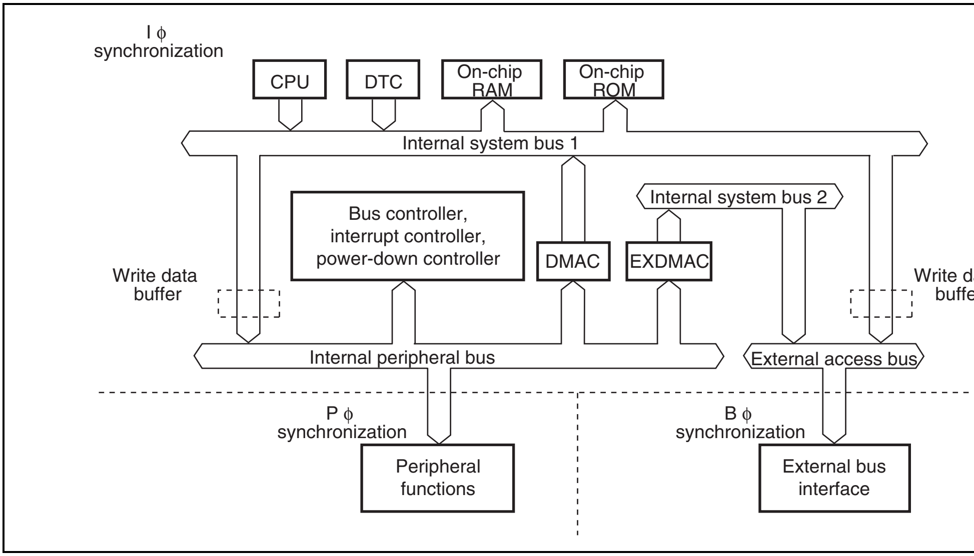


Figure 9.4 Internal Bus Configuration

	Bus controller CPU DTC DMAC EXDMAC Internal memory Clock pulse generator Power down control
Pϕ	I/O ports TPU PPG TMR WDT SCI A/D D/A IIC2 USB
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 26, 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.

the frequency rate of Iϕ and Pϕ is m : 1, 0 to m-1 cycles of Tsy may be inserted.

Figure 9.5 shows the external 2-state access timing when the frequency rate of Iϕ and Bϕ
Figure 9.6 shows the external 3-state access timing when the frequency rate of Iϕ and Bϕ

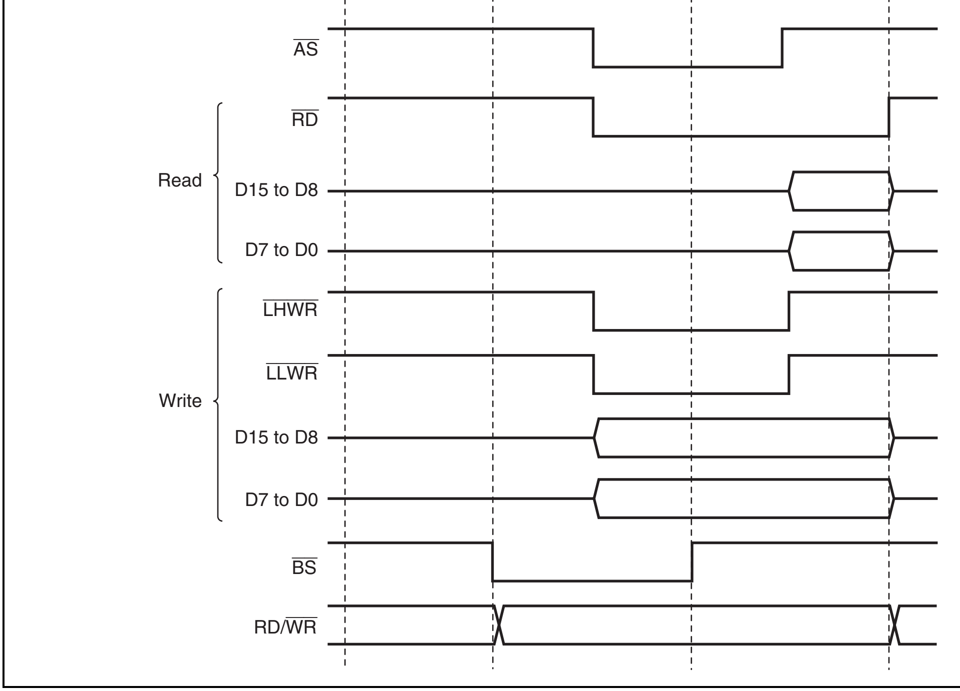


Figure 9.5 System Clock: External Bus Clock = 4:1, External 2-State Acce

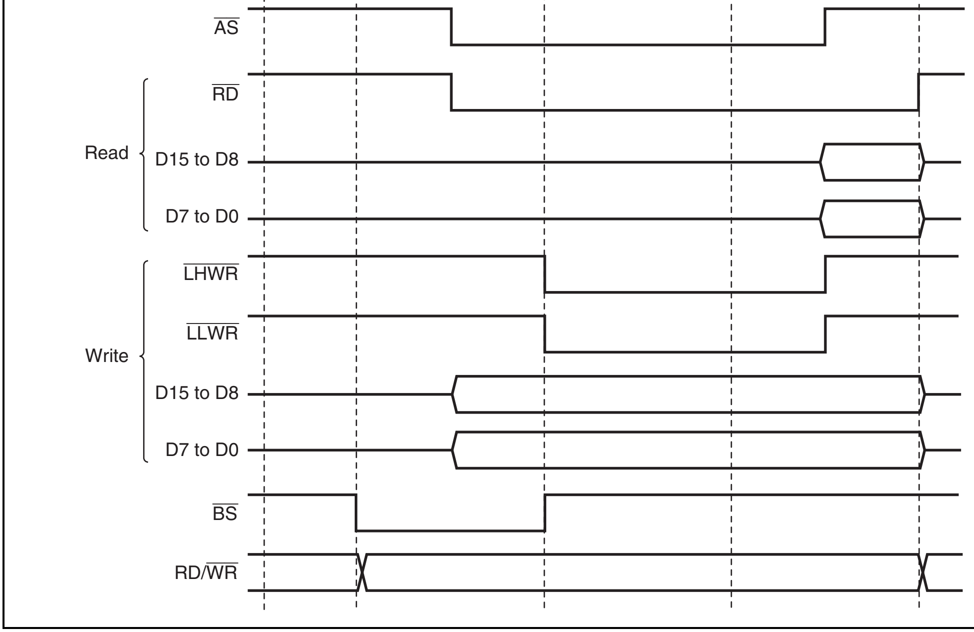


Figure 9.6 System Clock: External Bus Clock = 2:1, External 3-State Acces

Bus cycle start	$\overline{BS}$	Output	Signal indicating that the bus cycle has st
Address strobe/ address hold	$\overline{AS}$/$\overline{AH}$	Output	• Strobe signal indicating that the basic control SRAM, or burst ROM space is and address output on address bus is • Signal to hold the address during acce address/data multiplexed I/O interface
Read strobe	$\overline{RD}$	Output	Strobe signal indicating that the basic bus control SRAM, burst ROM, or address/dat multiplexed I/O space is being read
Read/write	RD/$\overline{WR}$	Output	• Signal indicating the input or output di • Write enable signal of the SRAM durin to the byte control SRAM space
Low-high write/ lower-upper byte select	$\overline{LHWR}$/$\overline{LUB}$	Output	• Strobe signal indicating that the basic ROM, or address/data multiplexed I/O written to, and the upper byte (D15 to data bus is enabled • Strobe signal indicating that the byte c SRAM space is accessed, and the upp (D15 to D8) of data bus is enabled
Low-low write/ lower-lower byte select	$\overline{LLWR}$/$\overline{LLB}$	Output	• Strobe signal indicating that the basic ROM, or address/data multiplexed I/O written to, and the lower byte (D7 to D bus is enabled • Strobe signal indicating that the byte c SRAM space is accessed, and the low (D7 to D0) of data bus is enabled

Wait	WAIT	Input	Wait request signal when accessing extern address space.
Bus request	$\overline{\text{BREQ}}$	Input	Request signal for release of bus to externa master
Bus request acknowledge	$\overline{\text{BACK}}$	Output	Acknowledge signal indicating that bus has released to external bus master
Bus request output	$\overline{\text{BREQO}}$	Output	External bus request signal used when inte master accesses external address space in external-bus released state
Data transfer acknowledge 3 (DMAC_3)	$\overline{\text{DACK3}}$	Output	Data transfer acknowledge signal for DMA single address transfer
Data transfer acknowledge 2 (DMAC_2)	$\overline{\text{DACK2}}$	Output	Data transfer acknowledge signal for DMA single address transfer
Data transfer acknowledge 1 (DMAC_1)	$\overline{\text{DACK1}}$	Output	Data transfer acknowledge signal for DMA single address transfer
Data transfer acknowledge 0 (DMAC_0)	$\overline{\text{DACK0}}$	Output	Data transfer acknowledge signal for DMA single address transfer
Data transfer acknowledge 1 (EXDMAC_1)	$\overline{\text{EDACK1}}$	Output	Data transfer acknowledge signal for EXDM single address transfer
Data transfer acknowledge 0 (EXDMAC_0)	$\overline{\text{EDACK0}}$	Output	Data transfer acknowledge signal for EXDM single address transfer
External bus clock	Bφ	Output	External bus clock

$\overline{CS3}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS4}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS5}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS6}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS7}$	—	—	—	O	O	O	—	—	O	O	
BS	—	—	—	O	O	O	O	O	O	O	
RD/$\overline{WR}$	—	—	—	O	O	O	O	O	O	O	
$\overline{AS}$	Output	Output	—	O	O	O	O	O	—	—	
$\overline{AH}$	—	—	—	—	—	—	—	—	O	O	
$\overline{RD}$	Output	Output	—	O	O	O	O	O	O	O	
$\overline{LHWR}$/$\overline{LUB}$	Output	Output	—	O	—	O	O	—	O	—	
$\overline{LLWR}$/$\overline{LLB}$	Output	Output	—	O	O	O	O	O	O	O	
$\overline{WAIT}$	—	—	—	O	O	O	O	O	O	O	Contro WAITE

[Legend]

O: Used as a bus control signal

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

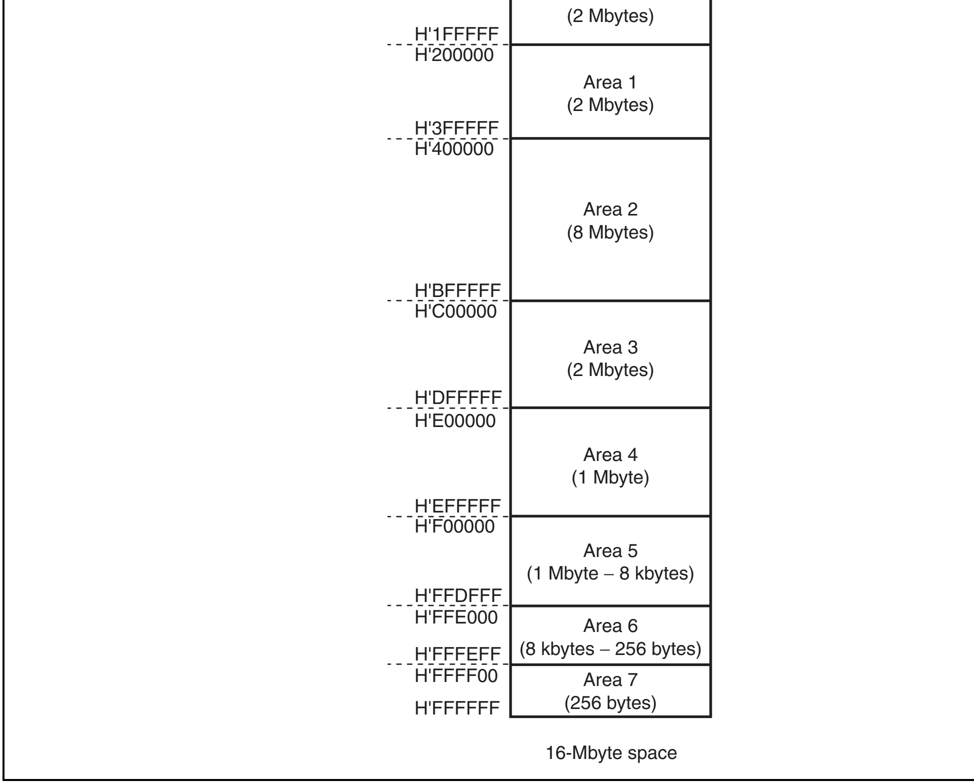


Figure 9.7 Address Space Area Division

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

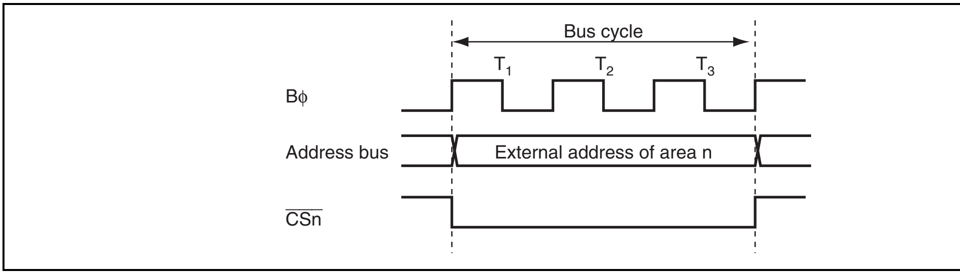


Figure 9.8 $\overline{\text{CSn}}$ Signal Output Timing (n = 0 to 7)

Figure 9.9 Timing When $\overline{CS}$ Signal is Output to the Same Pin

9.5.4 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, area name to be set for each interface. Table 9.5 areas that can be specified for each interface. The initial state of each area is a basic bus i

Table 9.4 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

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

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]

(b) Byte Control SRAM Interface

The number of access cycles in the byte control SRAM interface is the same as that in the 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]

(c) Burst ROM Interface

The number of access cycles at full access in the burst ROM interface is the same as that basic bus interface. The number of access cycles in the burst access can be specified as or 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 9.6 lists the number of access cycles for each interface.

Table 9.6 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 3)
	=	Th	+T1	+T2	+Tpw	+Ttw	+T3		+Tb	
		[0,1]	[1]	[1]	[0 to 7]	[n]	[1]		[(1 to 8) × m]	[(2 to 11 + n)
Address/data multiplexed I/O interface	= Tma	+Th	+T1	+T2				+Tt		
	[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)

9.5.5 Area and External Bus Interface

(1) Area 0

Area 0 includes on-chip ROM. All of area 0 is used as external address space in on-chip R disabled extended mode, and the space excluding on-chip ROM is external address space chip ROM enabled extended mode.

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

Either of the basic bus interface, byte control SRAM interface, or burst ROM interface ca selected for area 0 by bit BSRM0 in BROMCR and bit BCSEL0 in SRAMCR. Table 9.7 the external interface of area 0.

Table 9.7 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

Interface	Register Setting	
	BSRM1 of BROMCR	BCSEL1 of SRAMCR
Basic bus interface	0	0
Byte control SRAM interface	0	1
Burst ROM interface	1	0
Setting prohibited	1	1

(3) Area 2

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

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

Either the basic bus interface or byte control SRAM interface can be selected for area 2
BCSEL2 in SRAMCR. Table 9.9 shows the external interface of area 2.

Table 9.9 Area 2 External Interface

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

Interface	Register Setting	
	MPXE3 of MPXCR	BCSEL3 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

(5) Area 4

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

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

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexe interface can be selected for area 4 by bit MPXE4 in MPXCR and bit BCSEL4 in SRAM Table 9.11 shows the external interface of area 4.

Table 9.11 Area 4 External Interface

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

interface can be selected for area 5 by the MPXE5 bit in MPXCR and the BCSEL5 bit in SRAMCR. Table 9.12 shows the external interface of area 5.

Table 9.12 Area 5 External Interface

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

Interface	Register Setting	
	MPXE6 of MPXCR	BCSEL6 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

(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 9.14 shows the external interface of area 7.

Table 9.14 Area 7 External Interface

Interface	Register Setting	
	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 9.10 and 9.11 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 … RI… Data b… D15 D8 …
Byte	n	1	1st	Byte	[illegible]
Word	n	2	1st	Byte	[illegible]
			2nd	Byte	[illegible]
Longword	n	4	1st	Byte	[illegible]
			2nd	Byte	[illegible]
			3rd	Byte	[illegible]
			4th	Byte	[illegible]

Figure 9.10 Access Sizes and Data Alignment Control for 8-Bit Access Space (Bi

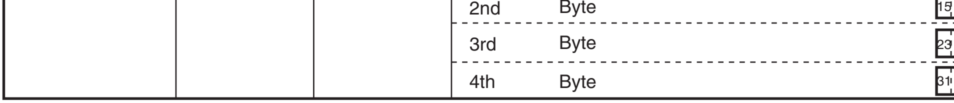


Figure 9.11 Access Sizes and Data Alignment Control for 8-Bit Access Spac
(Little Endian)

(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 9.12 and 9.13 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 9
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 third byte data bus.

Access Size	Access Address	Access Count	Bus Cycle	Data Size	Data (D15–D8)
Longword	Even (2n)	2	1st	Word	31 … 24
			2nd	Word	15 … 8
	Odd (2n+1)	3	1st	Byte	
			2nd	Word	23 … 16
			3rd	Byte	7 … 0

Figure 9.12 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
Byte	Even (2n)	1	1st	Byte	
	Odd (2n+1)	1	1st	Byte	7 … 0
Word	Even (2n)	1	1st	Word	15 … 8
	Odd (2n+1)	2	1st	Byte	7 … 0
			2nd	Byte	
Longword	Even (2n)	2	1st	Word	15 … 8
			2nd	Word	31 … 24
	Odd (2n+1)	3	1st	Byte	7 … 0
			2nd	Word	23 … 16
			3rd	Byte	

Figure 9.13 Access Sizes and Data Alignment Control for 16-Bit Access Sp (Little Endian)

or lower byte data bus (D7 to D0) is used according to the bus specifications for the area
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 9.5.6, Endian and Data Alignm

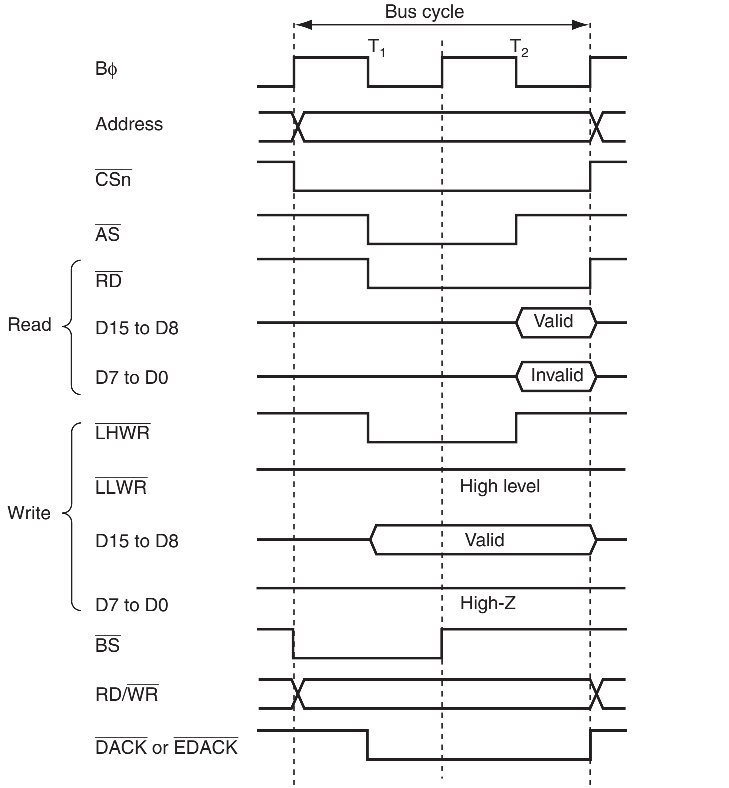
9.6.2 I/O Pins Used for Basic Bus Interface

Table 9.15 shows the pins used for basic bus interface.

Table 9.15 I/O Pins for Basic Bus Interface

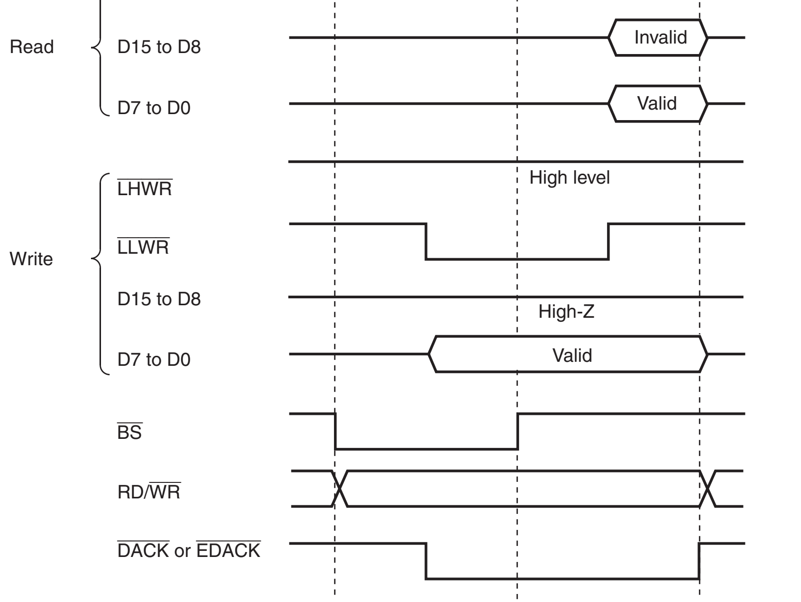
Name	Symbol	I/O	Function
Bus cycle start	$\overline{\text{BS}}$	Output	Signal indicating that the bus cycle has start
Address strobe	$\overline{\text{AS}}$*	Output	Strobe signal indicating that an address outp address bus is valid during access
Read strobe	$\overline{\text{RD}}$	Output	Strobe signal indicating the read access
Read/write	RD/$\overline{\text{WR}}$	Output	Signal indicating the data bus input or outpu direction
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 (D0) is valid during write access
Chip select 0 to 7	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	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

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



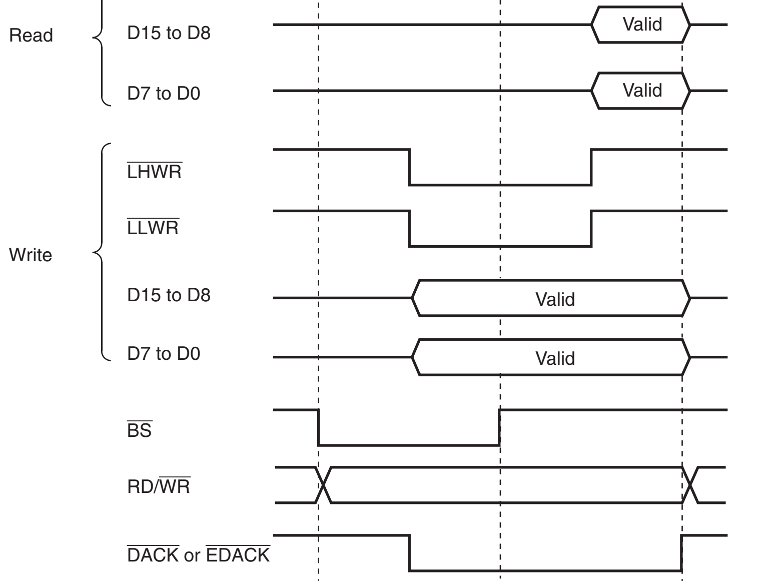
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC, EDKC = 0

Figure 9.14 16-Bit 2-State Access Space Bus Timing (Byte Access for Even Ad



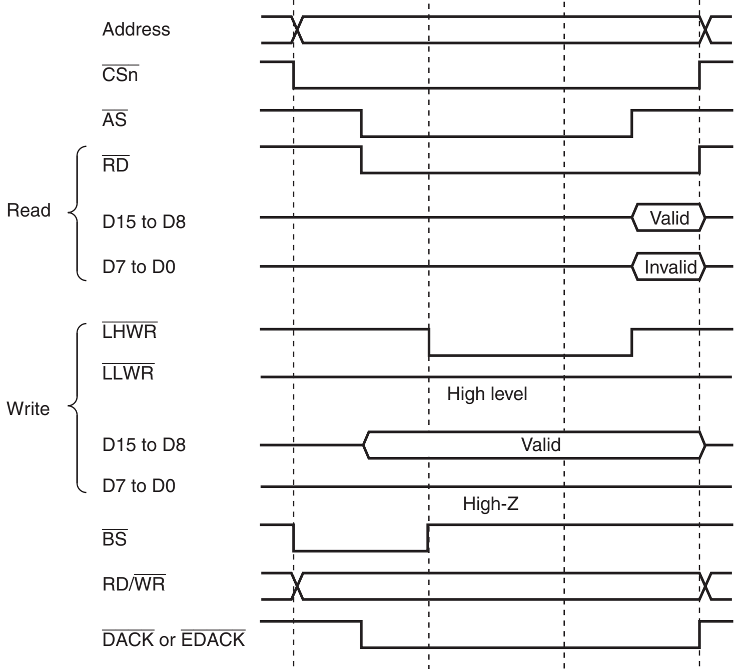
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC, EDKC = 0

Figure 9.15 16-Bit 2-State Access Space Bus Timing (Byte Access for Odd Add



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

Figure 9.16 16-Bit 2-State Access Space Bus Timing (Word Access for Even A



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

Figure 9.17 16-Bit 3-State Access Space Bus Timing (Byte Access for Even Ad

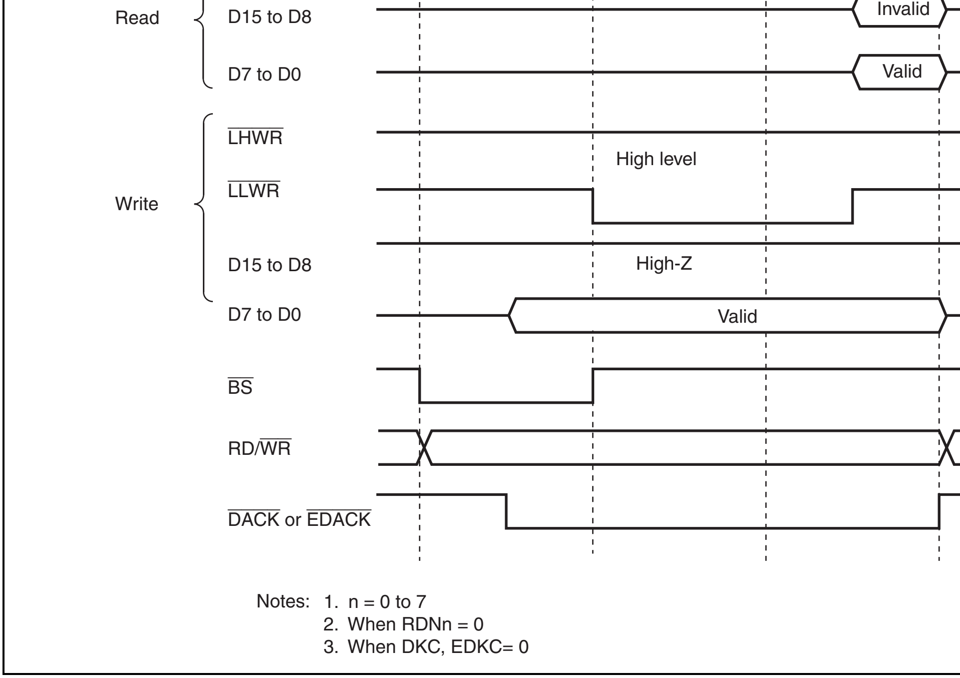
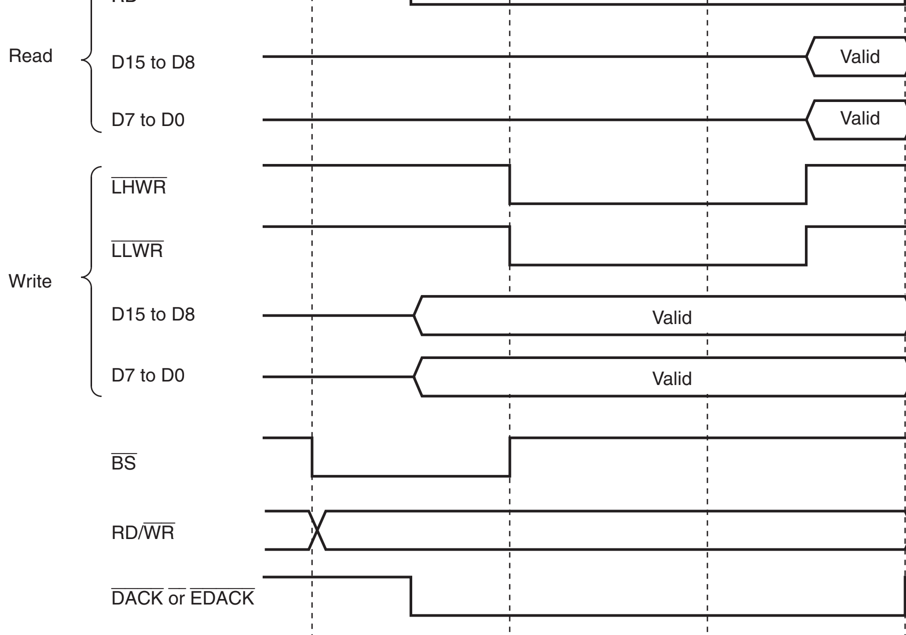


Figure 9.18 16-Bit 3-State Access Space Bus Timing (Word Access for Odd A

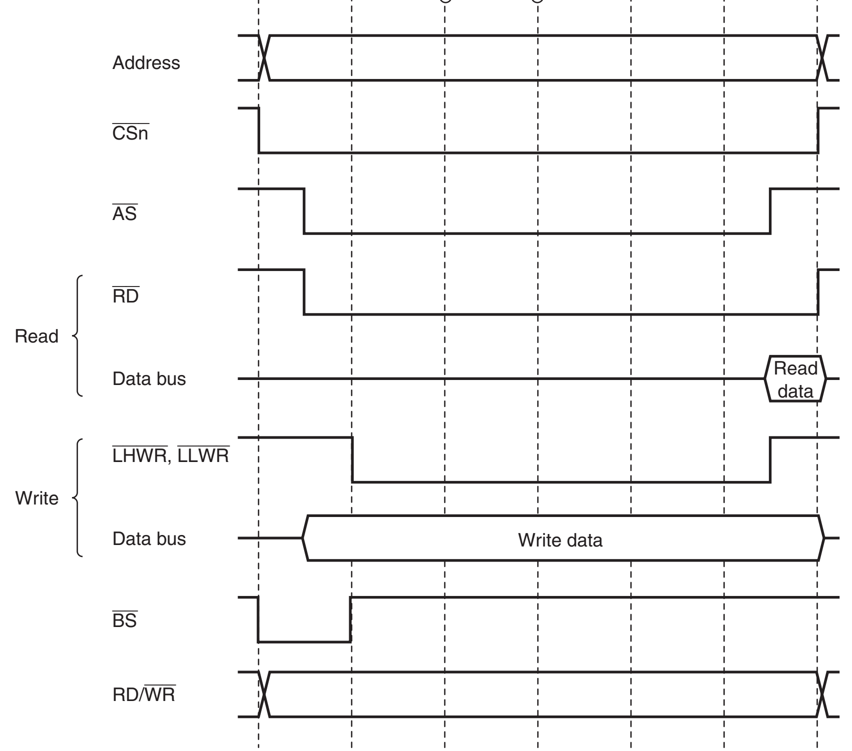


Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC, EDKC= 0

Figure 9.19 16-Bit 3-State Access Space Bus Timing (Word Access for Even Ad

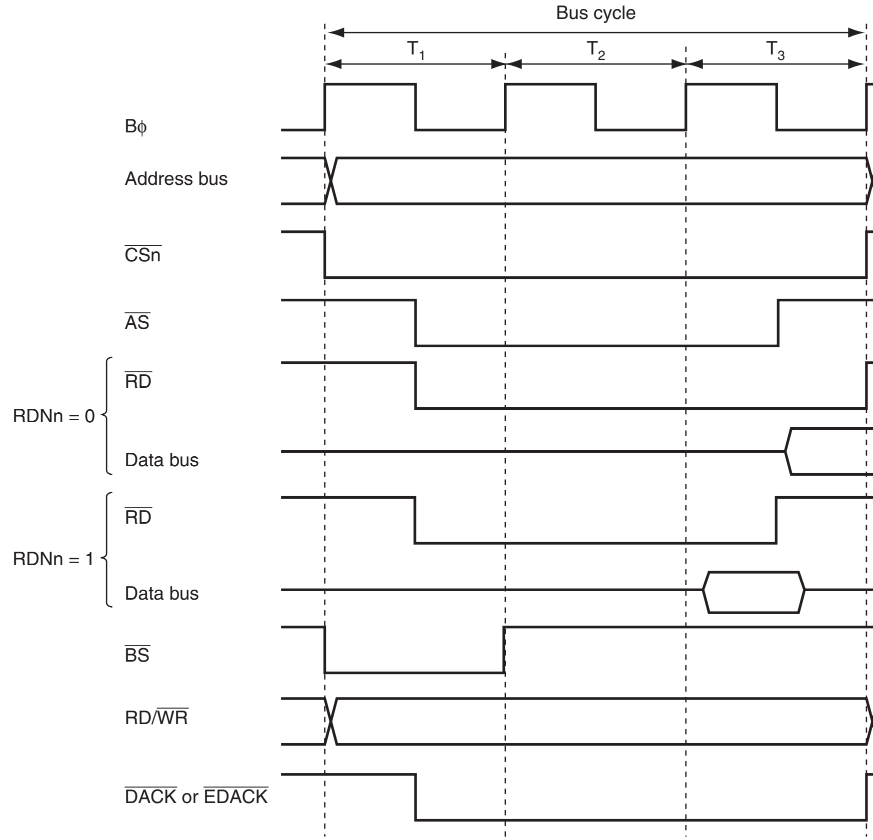
(2) Pin Wait Insertion

For 3-state access space, when the WAITE bit in BCR1 is set to 1 and the corresponding is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled. When the external address s accessed in this state, a program wait (Tpw) is first inserted according to the WTCRA ar WTCRB settings. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of Bϕ in the last T2 or Tpw another Ttw cycle is inserted until the $\overline{\text{WAIT}}$ pin is brought high. The pin wait insertion effective when the Tw cycles are inserted to seven cycles or more, or when the number c cycles to be inserted is changed according to the external devices. The WAITE bit is cor all areas. For details on ICR, see section 13, I/O Ports.



Notes: 1. Upward arrows indicate the timing of WAIT pin sampling.
2. n = 0 to 7
3. When RDNn = 0

Figure 9.20 Example of Wait Cycle Insertion Timing

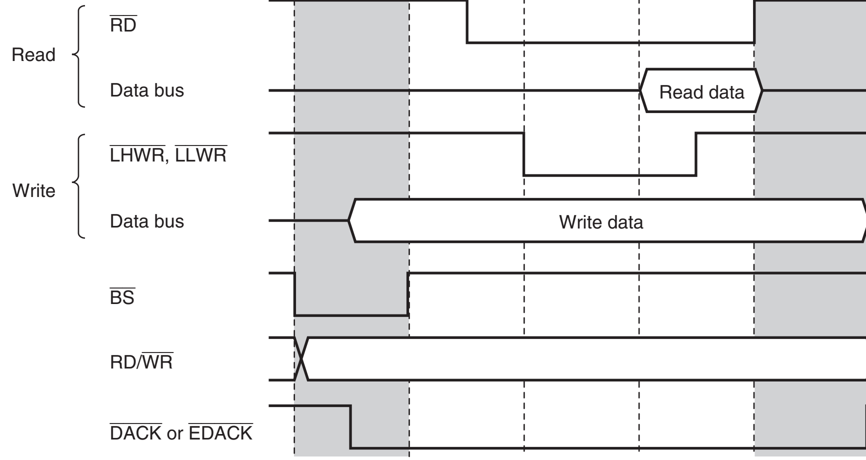


Notes: 1. n = 0 to 7
2. When DKC, EDKC = 0

Figure 9.21 Example of Read Strobe Timing

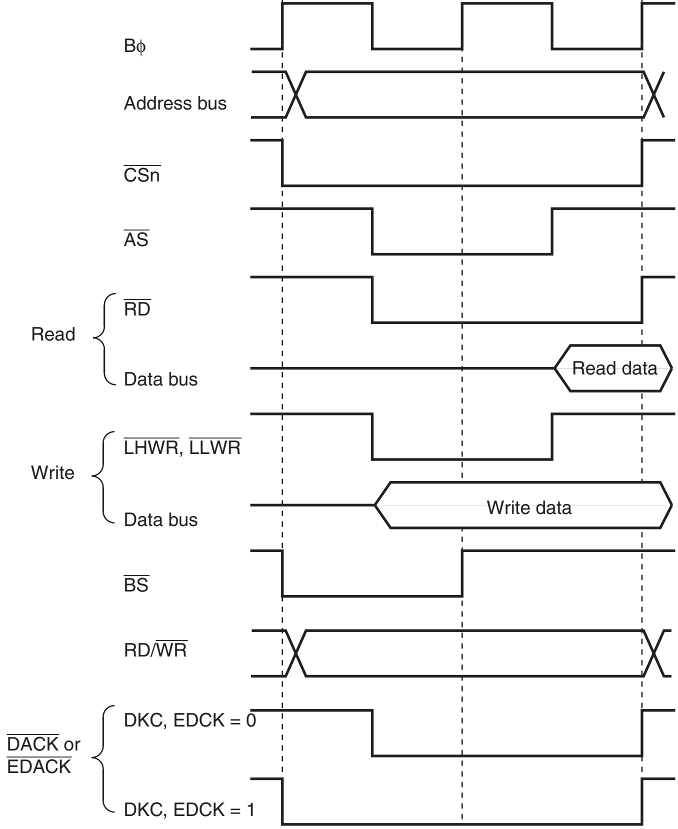
3-state access space.

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, EDKC = 0

Figure 9.22 Example of Timing when Chip Select Assertion Period is Exten



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

Figure 9.23 $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ Signal Output Timings

9.7.1 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 s
as burst ROM interface or address/data multiplexed I/O interface, the SRAMCR setting
and byte control SRAM interface cannot be used.

9.7.2 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
9.5.6, Endian and Data Alignment.

AS/AH	AS	Address strobe	Output	Strobe signal indicating that the ad output on the address bus is valid w basic bus interface space or byte c SRAM space is accessed
$\overline{\text{CSn}}$	$\overline{\text{CSn}}$	Chip select	Output	Strobe signal indicating that area n selected
$\overline{\text{RD}}$	$\overline{\text{RD}}$	Read strobe	Output	Output enable for the SRAM when control SRAM space is accessed
RD/$\overline{\text{WR}}$	RD/$\overline{\text{WR}}$	Read/write	Output	Write enable signal for the SRAM w byte control SRAM space is access
$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$	$\overline{\text{LUB}}$	Lower-upper byte select	Output	Upper byte select when the 16-bit b control SRAM space is accessed
$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$	$\overline{\text{LLB}}$	Lower-lower byte select	Output	Lower byte select when the 16-bit b control SRAM space is accessed
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal used when an address space is accessed
A20 to A0	A20 to A0	Address pin	Output	Address output pin
D15 to D0	D15 to D0	Data pin	Input/ output	Data input/output pin

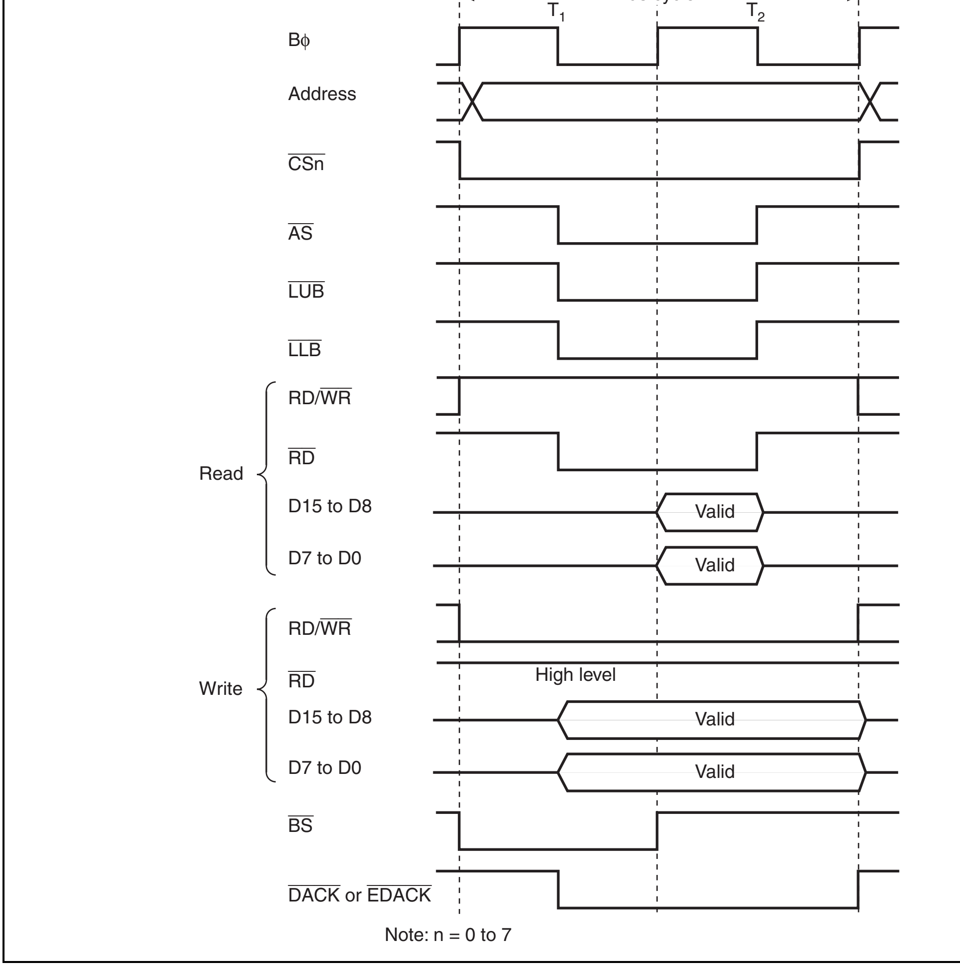


Figure 9.24 16-Bit 2-State Access Space Bus Timing

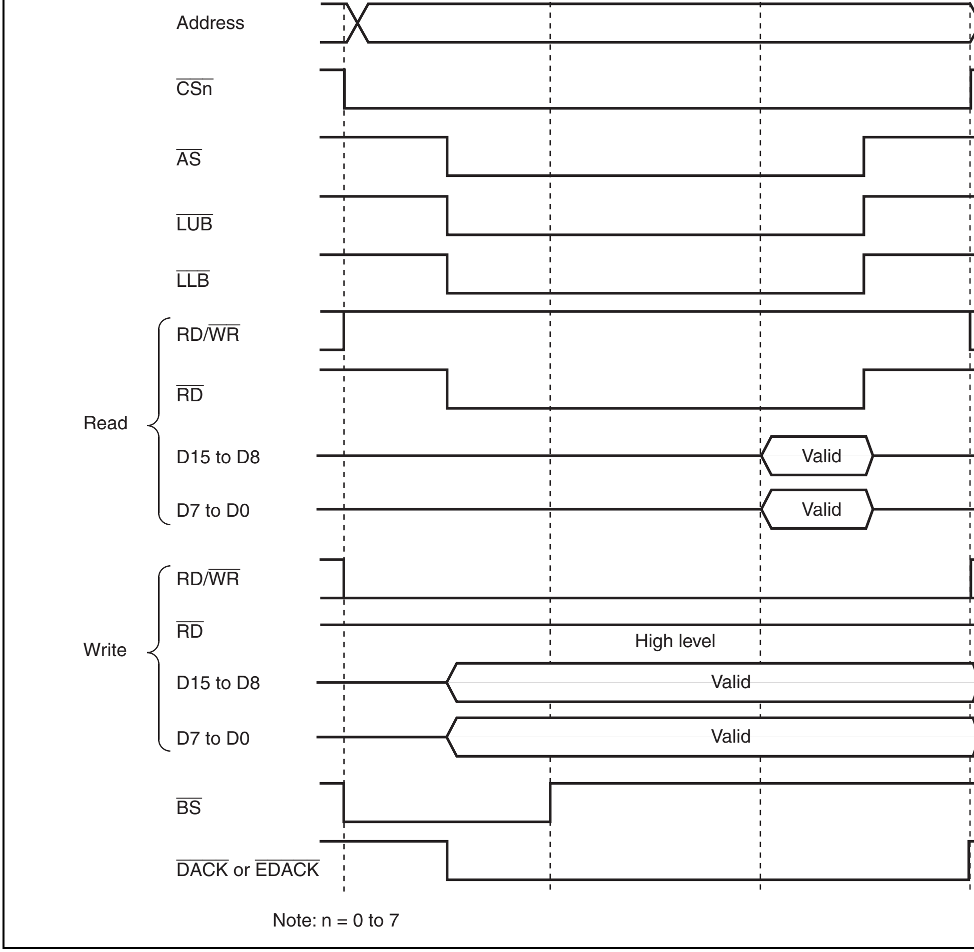
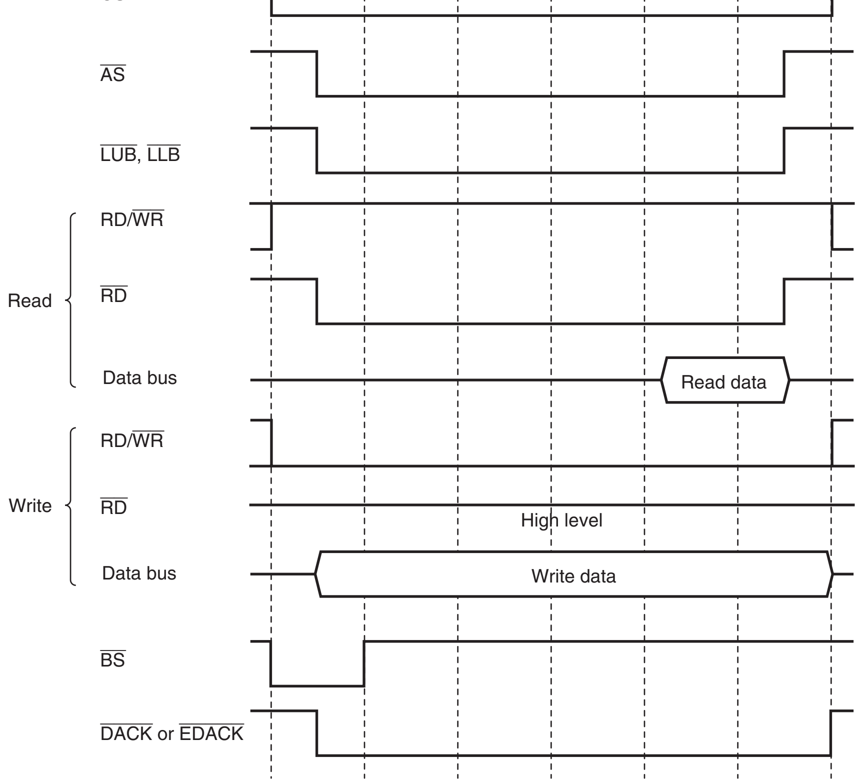


Figure 9.25 16-Bit 3-State Access Space Bus Timing

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 13, I/O Ports.

Figure 9.26 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 9.26 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 9.6.6, Extensior
Select ($\overline{\text{CS}}$) Assertion Period.

9.7.8 $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ Signal Output Timings

For DMAC or EXDMAC single address transfers, the $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$ signal assert t
be modified by using the DKC or EDKC bit in BCR1.

Figure 9.27 shows the $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ signal output timings. Setting the DKC or E
1 asserts the $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$ signal a half cycle earlier.

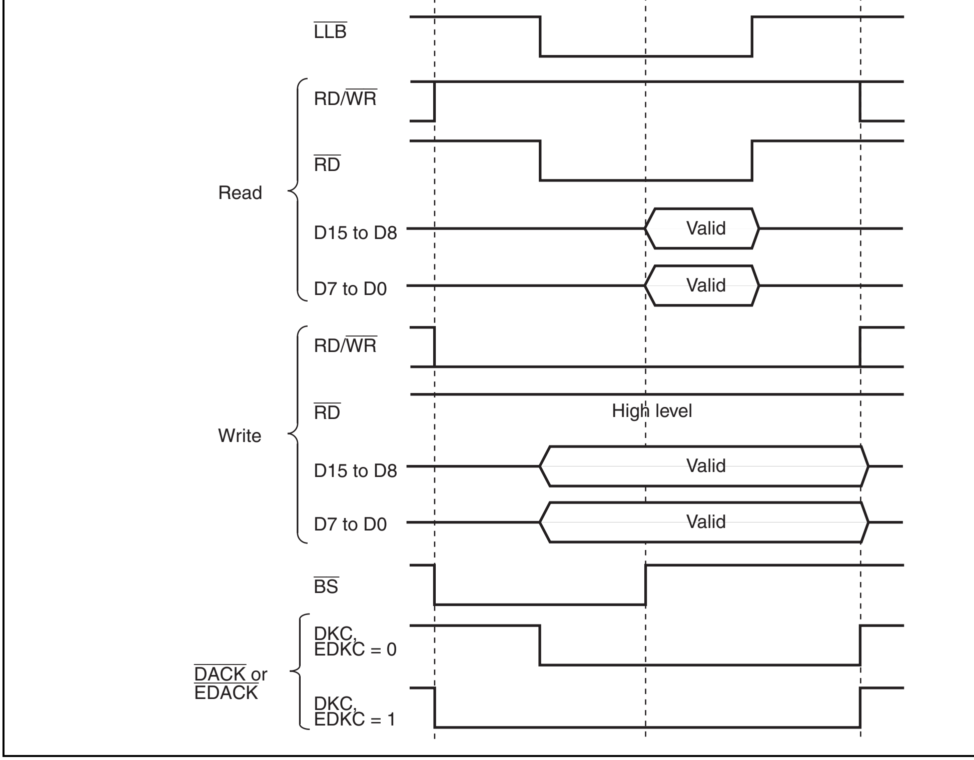


Figure 9.27 $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ Signal Output Timings

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

In the burst ROM interface, the burst access covers only CPU read accesses and cluster
read accesses of EXDMAC. Other accesses are performed with the similar method to th
interface.

9.8.1 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.

9.8.2 Data Bus

The bus width of the burst ROM space can be specified as 8-bit or 16-bit burst ROM int
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
9.5.6, Endian and Data Alignment.

Read strobe	$\overline{RD}$	Output	Strobe signal indicating the read access
Read/write	RD/$\overline{WR}$	Output	Signal indicating the data bus input or 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

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 9.28 and 9.29.

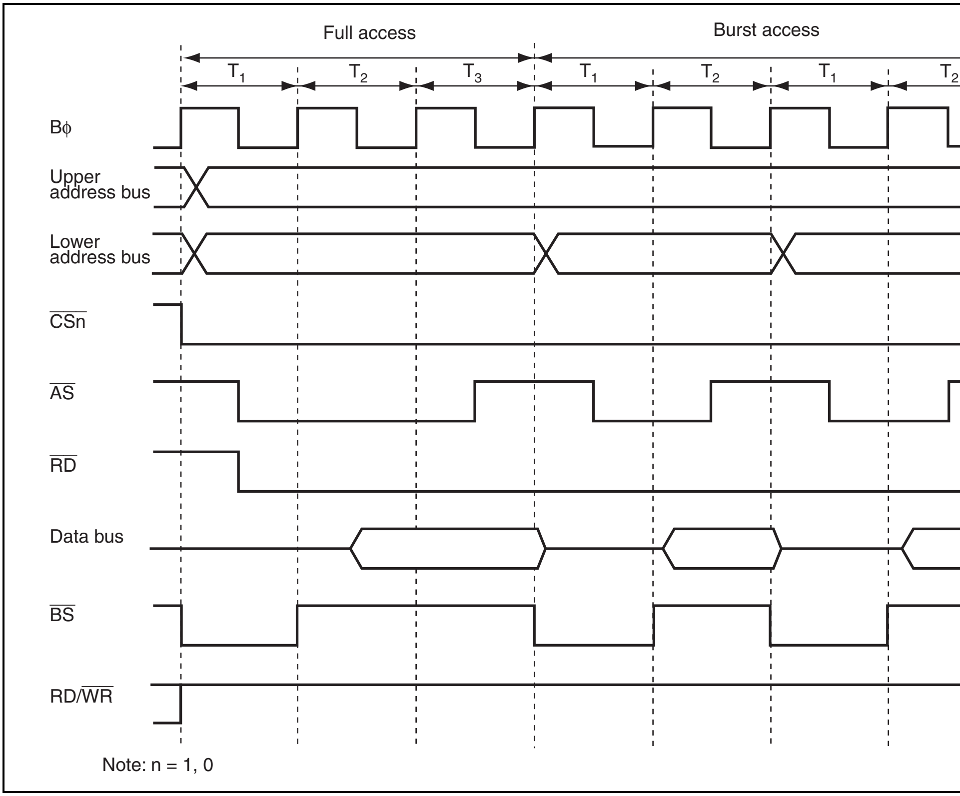


Figure 9.28 Example of Burst ROM Access Timing (ASTn = 1, Two Burst Cy

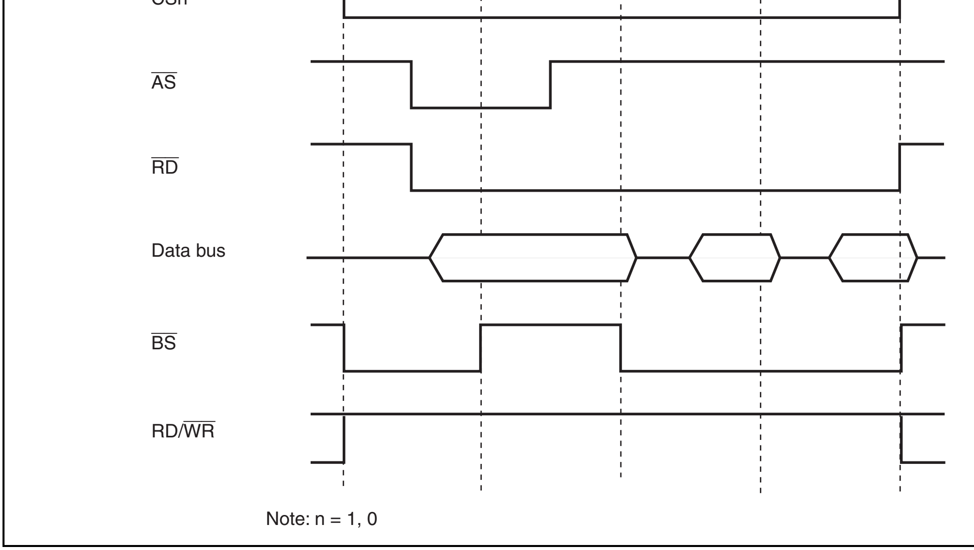


Figure 9.29   Example of Burst ROM Access Timing (ASTn = 0, One Burst Cy

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

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

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

For the burst ROM space, the burst access can be enabled only during read accesses by t EXDMAC cluster transfer. In this case, the setting of the corresponding CSXTn bit in C ignored and an extension cycle can be inserted only before the full access cycle. Note th extension cycle can be inserted before or after the burst access cycles.

For accesses except read accesses by the CPU or EXDMAC cluster transfer, the burst R is equivalent to the basic bus interface space. Accordingly, extension cycles can be inser 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.

9.9.2 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 9.18 Address/Data Multiplex

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

9.9.3 Data Bus

The bus width of the address/data multiplexed I/O space can be specified for either 8-bit a
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 9.5.6, Endian and Data Alignme

				address/data multiplexed I/O space
$\overline{\text{AS}}$/$\overline{\text{AH}}$	$\overline{\text{AH}}$*	Address hold	Output	Signal to hold an address when the addres multiplexed I/O space is specified
$\overline{\text{RD}}$	$\overline{\text{RD}}$	Read strobe	Output	Signal indicating that the address/data mult space is being read
$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$	$\overline{\text{LHWR}}$	Low-high write	Output	Strobe signal indicating that the upper byte D8) is valid when the address/data multiple space is written
$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$	$\overline{\text{LLWR}}$	Low-low write	Output	Strobe signal indicating that the lower byte is valid when the address/data multiplexed 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.
A20 to A0	A20 to A0	Address	Output	Address output pin
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal used when the external space is accessed
$\overline{\text{BS}}$	$\overline{\text{BS}}$	Bus cycle start	Output	Signal to indicate the bus cycle start
RD/$\overline{\text{WR}}$	RD/$\overline{\text{WR}}$	Read/write	Output	Signal indicating the data bus input or outp

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

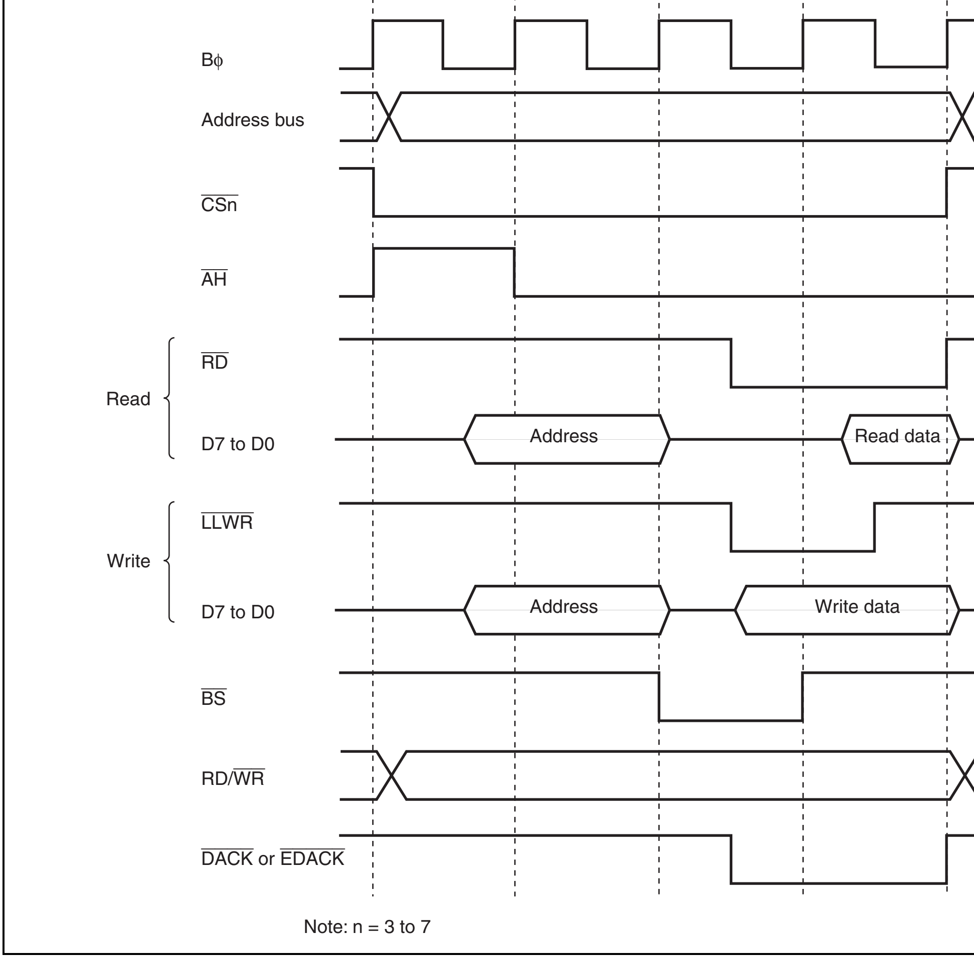


Figure 9.30 8-Bit Access Space Access Timing (ABWHn = 1, ABWLn = 1)

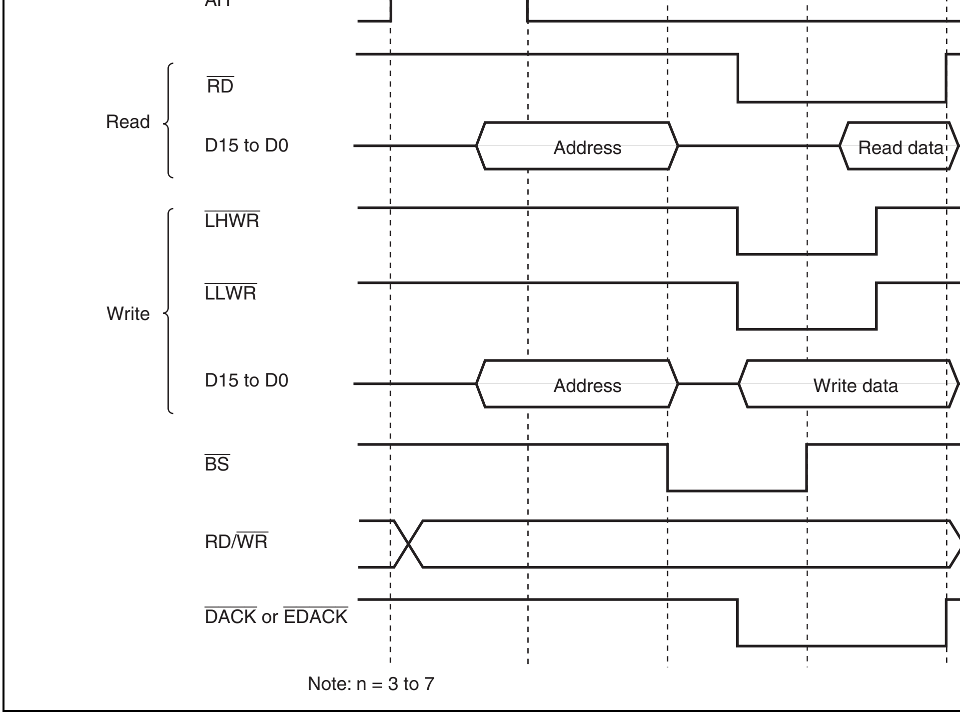


Figure 9.31 16-Bit Access Space Access Timing (ABWHn = 0, ABWLn =

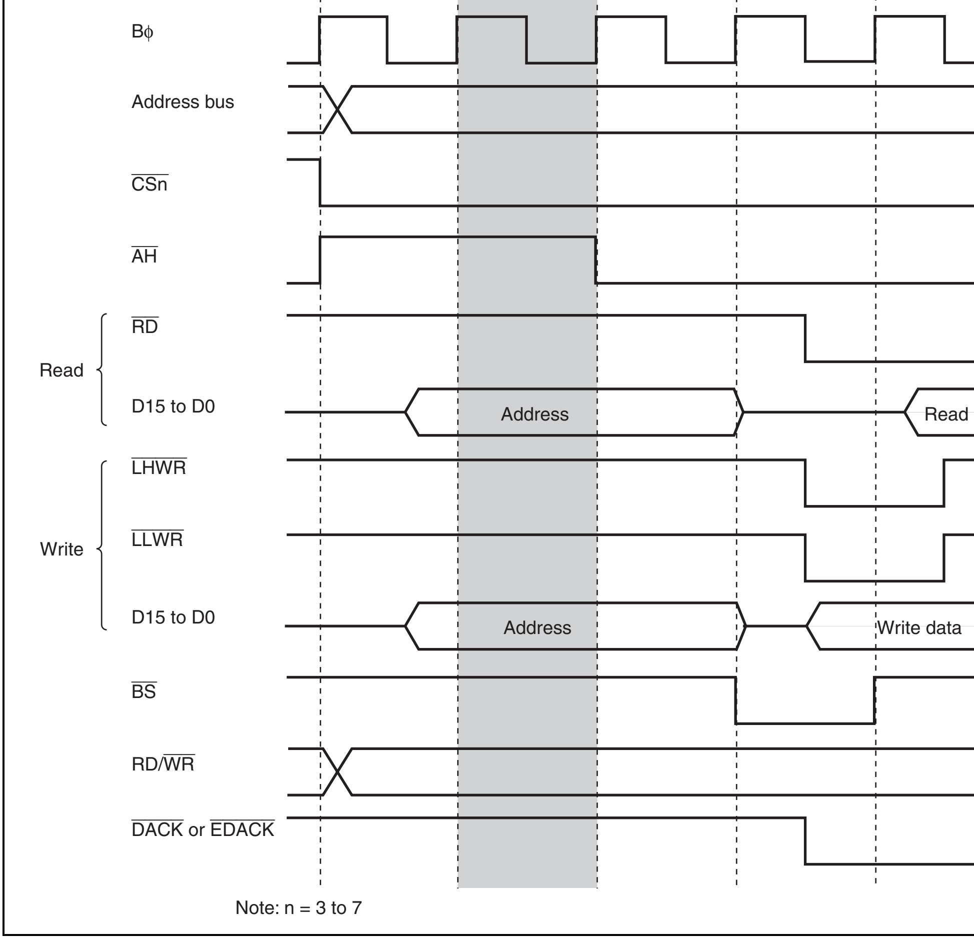


Figure 9.32 Access Timing of 3 Address Cycles (ADDEX = 1)

In the address/data multiplexed I/O interface, the read strobe timing of data cycles can b
in the same way as in basic bus interface. For details, see section 9.6.5, Read Strobe ($\overline{\mathrm{RD}}$

Figure 9.33 shows an example when the read strobe timing is modified.

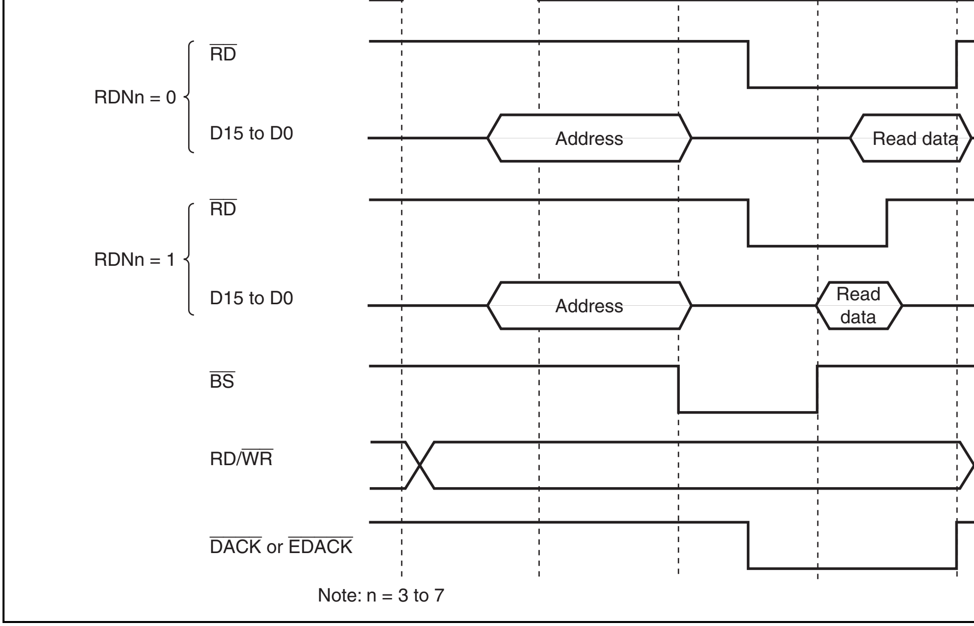


Figure 9.33 Read Strobe Timing

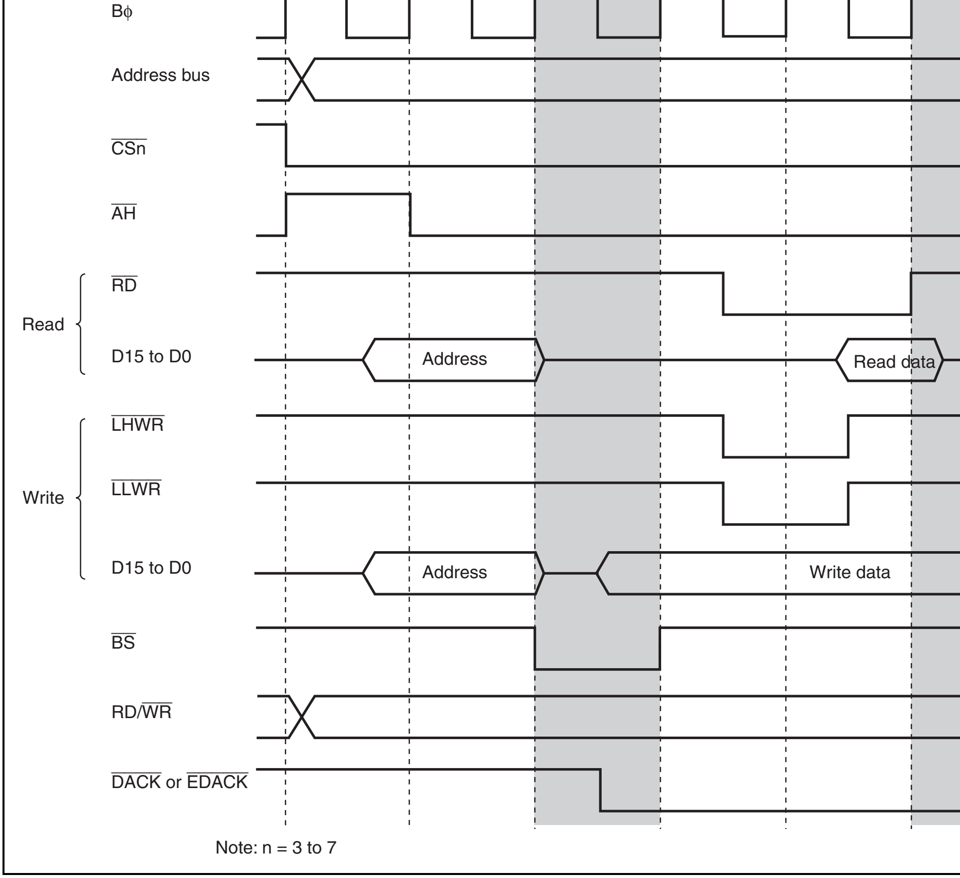


Figure 9.34 Chip Select ($\overline{CS}$) Assertion Period Extension Timing in Data Cy

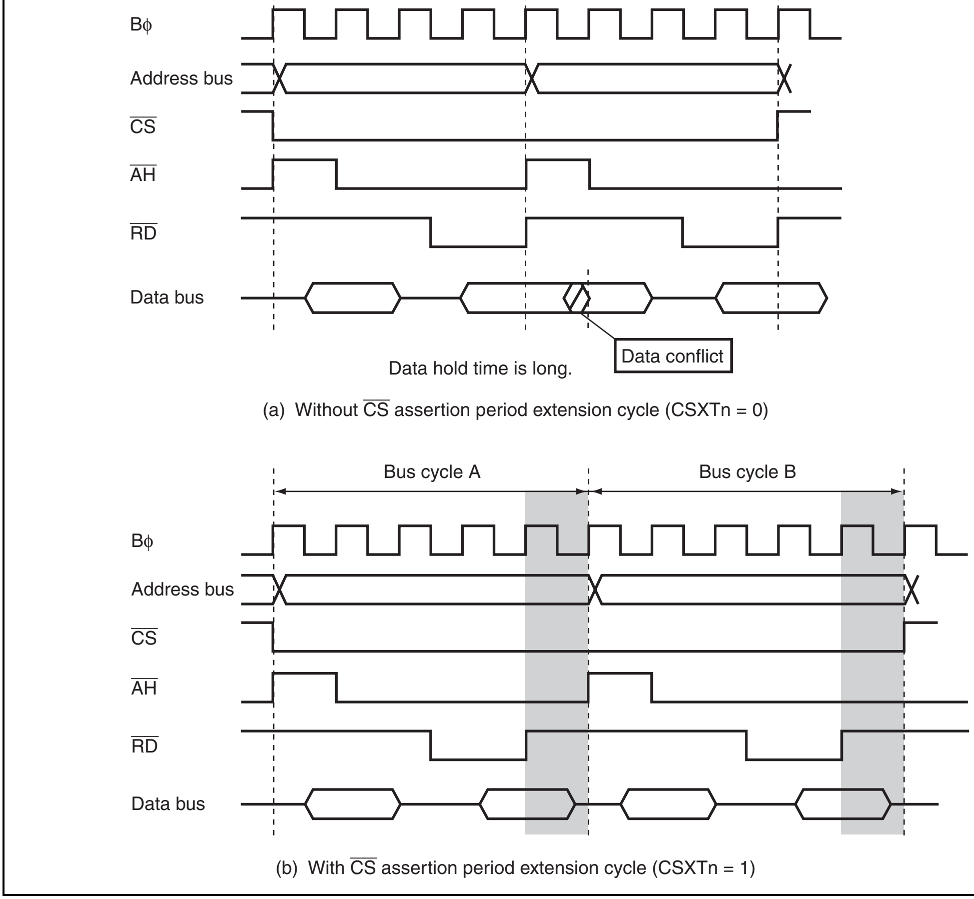


Figure 9.35 Consecutive Read Accesses to Same Area (Address/Data Multiplexed I/O Space)

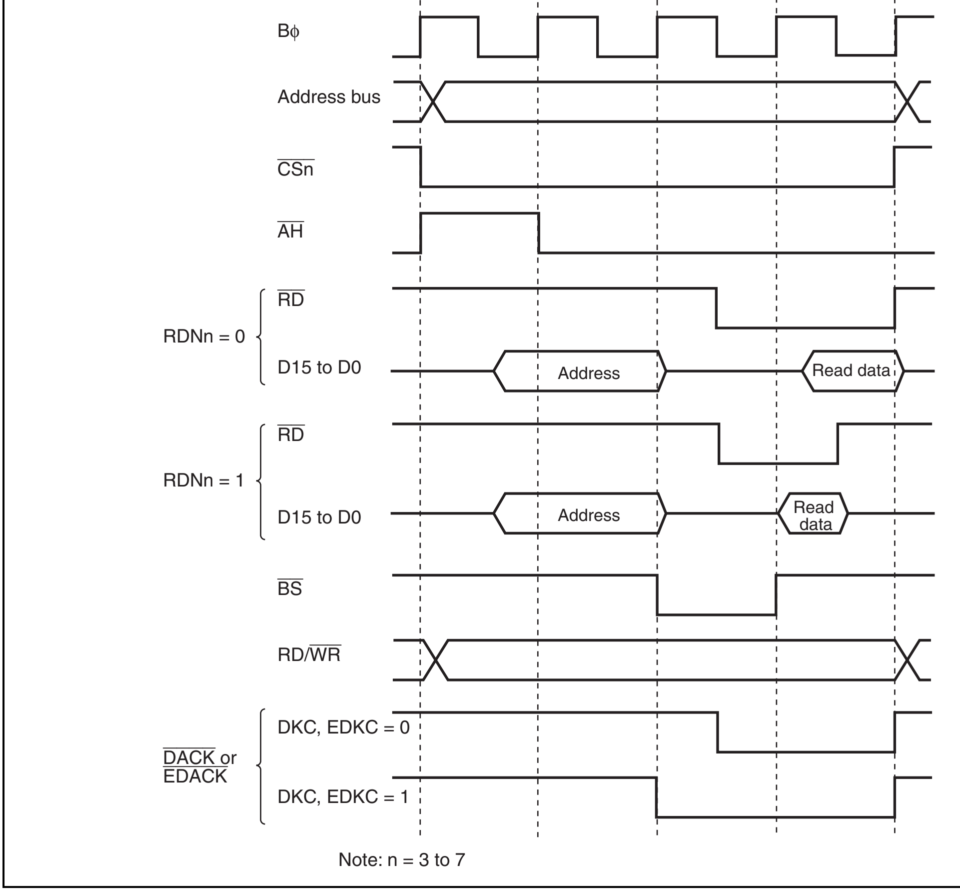


Figure 9.36 DACK and EDACK Signal Output Timings

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 or EXDMAC single addr
   transfer (write cycle)

Up to four idle cycles can be inserted under the conditions shown above. The number of i
cycles to be inserted should be specified to prevent data conflicts between the output data
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 nur
idle cycles can be selected from setting A specified by bits IDLCA1 and IDLCA0 in IDL
setting B specified by bits IDLCB1 and IDLCB0 in IDLCR: Setting A can be selected fro
four cycles, and setting B can be selected from one or two to four cycles. Setting A or B c
specified for each area by setting bits IDLSEL7 to IDLSEL0 in IDLCR. Note that bits ID
to IDLSEL0 correspond to the previously accessed area of the consecutive accesses.

The number of idle cycles to be inserted under conditions 3 and 4, which are conditions to
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 conditior
shown above.

Table 9.20 shows the correspondence between conditions 1 to 4 and number of idle cycle
inserted for each area. Table 9.21 shows the correspondence between the number of idle c
be inserted specified by settings A and B, and number of cycles to be inserted.

			1	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 9.21 Number of Idle Cycle Insertions

Bit Settings				
A		B		
IDLCA1	IDLCA0	IDLCB1	IDLCB0	Number of Cy
—	—	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.

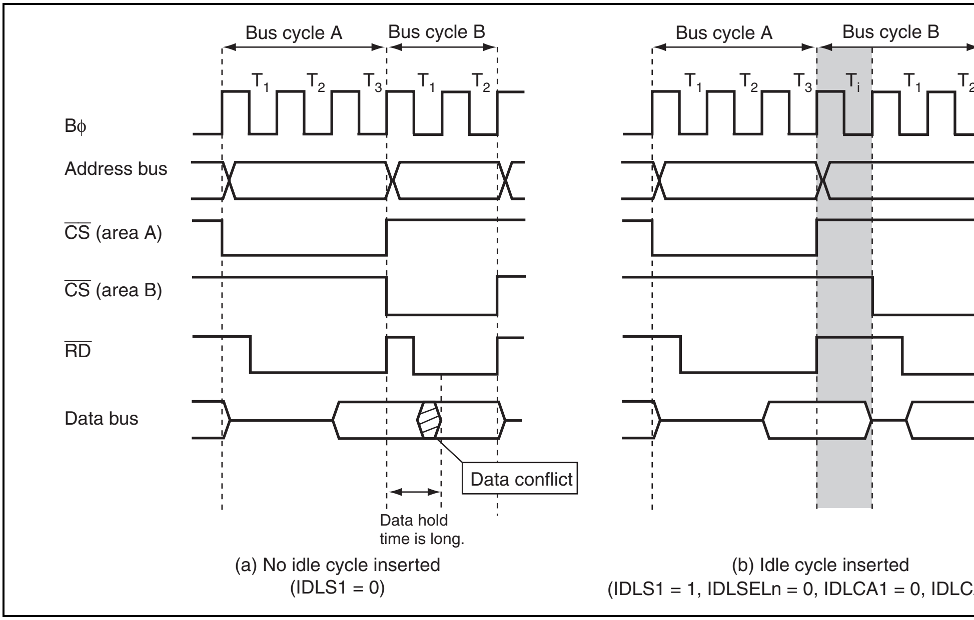


Figure 9.37 Example of Idle Cycle Operation (Consecutive Reads in Different A

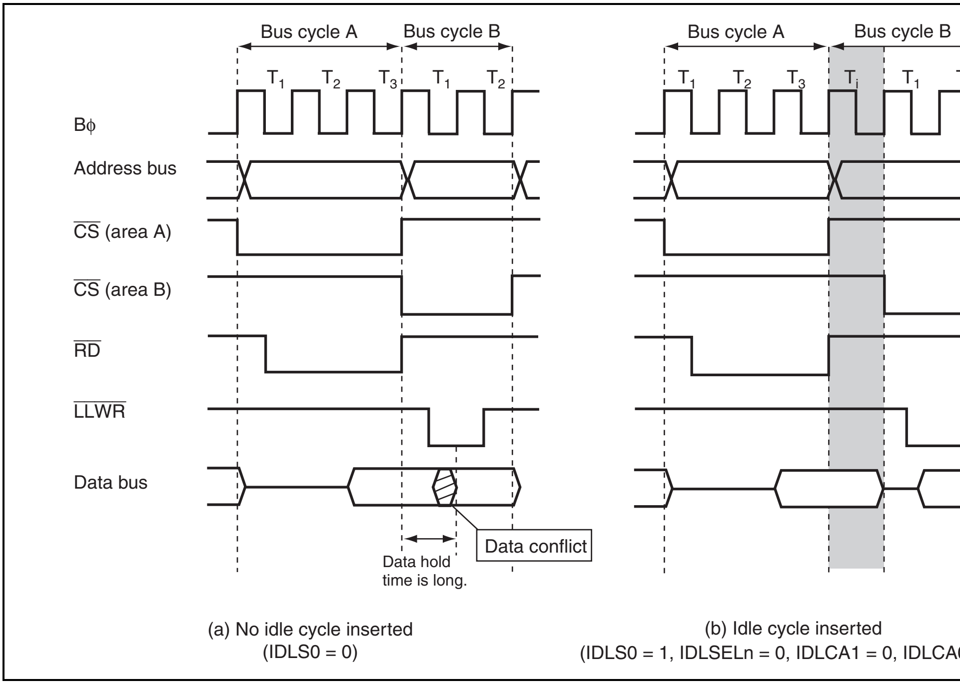


Figure 9.38 Example of Idle Cycle Operation (Write after Read)

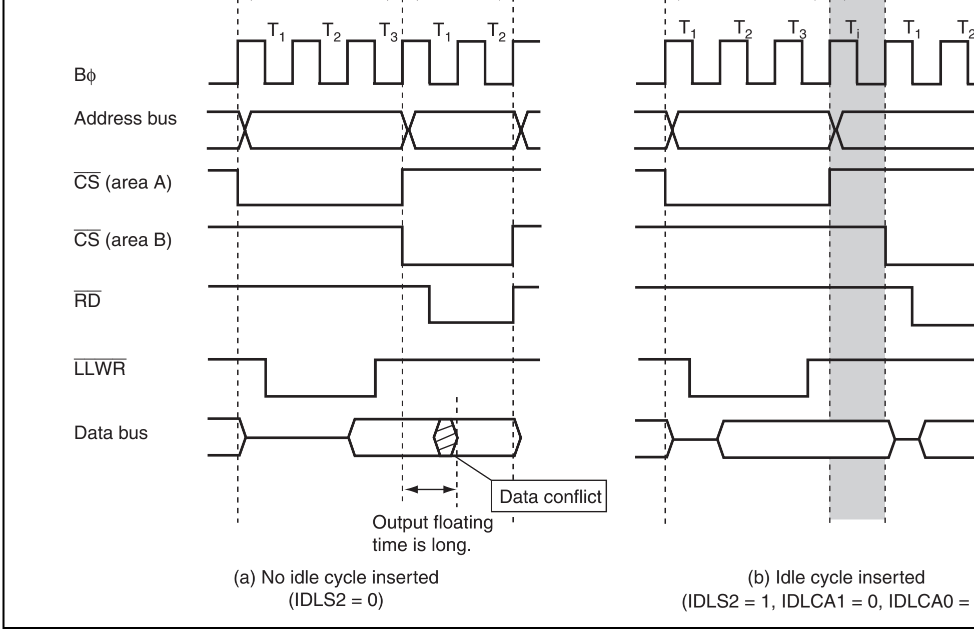


Figure 9.39 Example of Idle Cycle Operation (Read after Write)

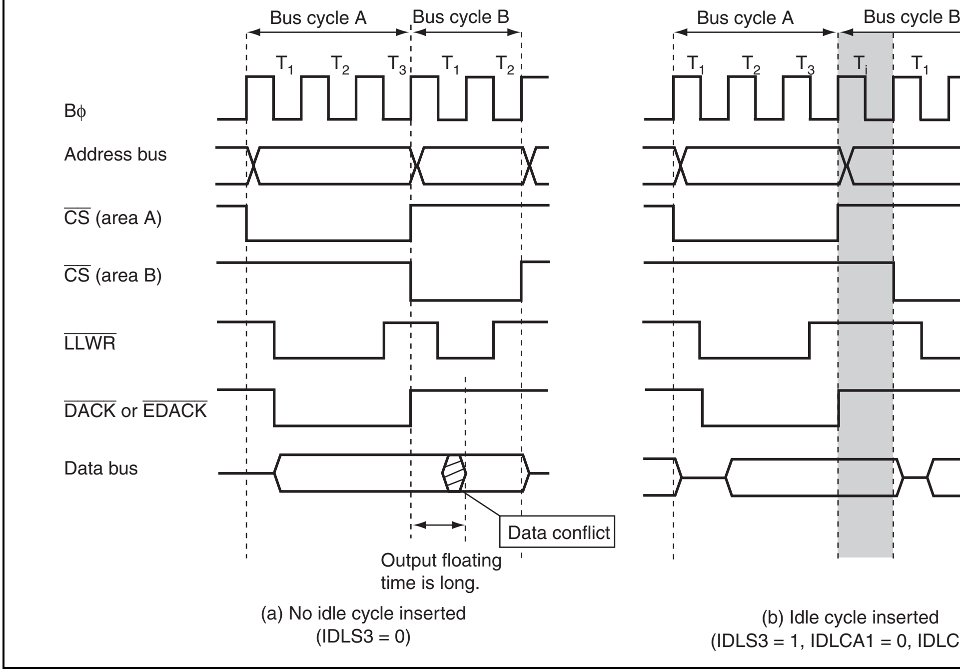


Figure 9.40 Example of Idle Cycle Operation (Write after Single Address Transf

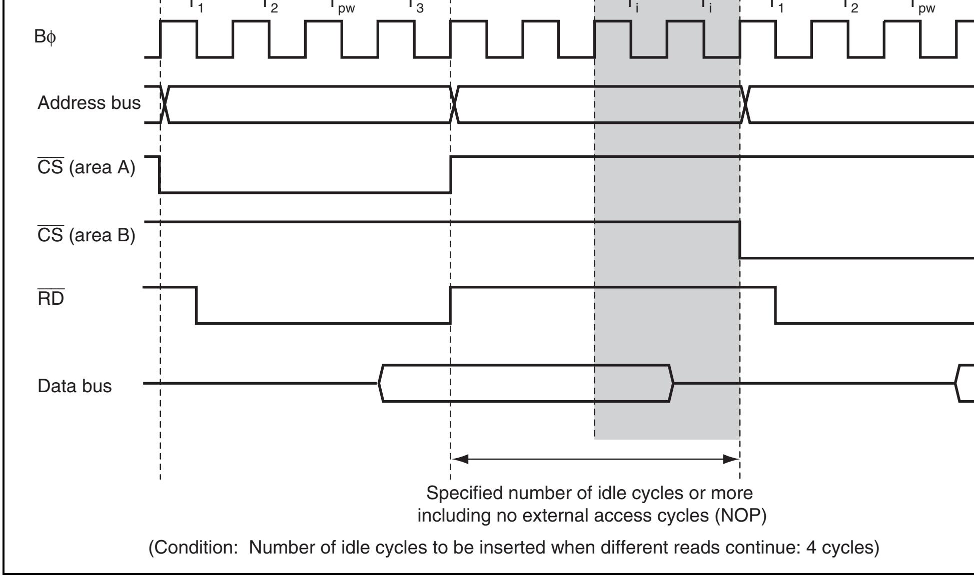


Figure 9.41 Idle Cycle Insertion Example

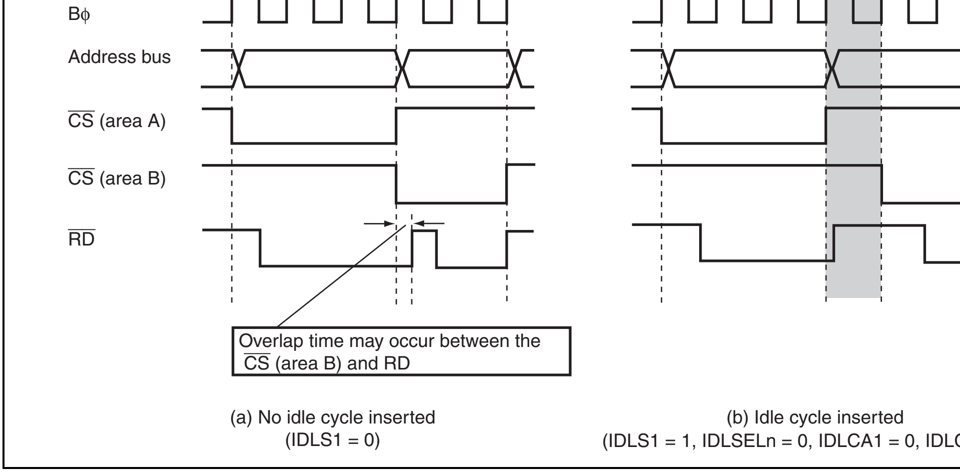


Figure 9.42 Relationship between Chip Select ($\overline{\text{CS}}$) and Read ($\overline{\text{RD}}$)

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

$\overline{AS}$	High
$\overline{RD}$	High
$\overline{BS}$	High
RD/$\overline{WR}$	High
$\overline{AH}$	low
$\overline{LHWR}$, $\overline{LLWR}$	High
$\overline{DACKn}$ (n = 3 to 0)	High
$\overline{EDACKn}$ (n = 1 to 0)	High

In external extended mode, when the BRLE bit in BCR1 is set to 1 and the ICR bits for th corresponding pin are set to 1, the bus can be released to the external. Driving the $\overline{\text{BREQ}}$ issues an external bus request to this LSI. When the $\overline{\text{BREQ}}$ pin is sampled, at the prescrib timing, the $\overline{\text{BACK}}$ pin is driven low, and the address bus, data bus, and bus control signal placed in the high-impedance state, establishing the external bus released state. For detail DDR and ICR, see section 13, I/O Ports.

In the external bus released state, the CPU, DTC, and DMAC can access the internal spac the internal bus. When the CPU, DTC, DMAC, or EXDMAC attempts to access the exter address space, it temporarily defers initiation of the bus cycle, and waits for the bus reque the external bus master to be canceled.

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

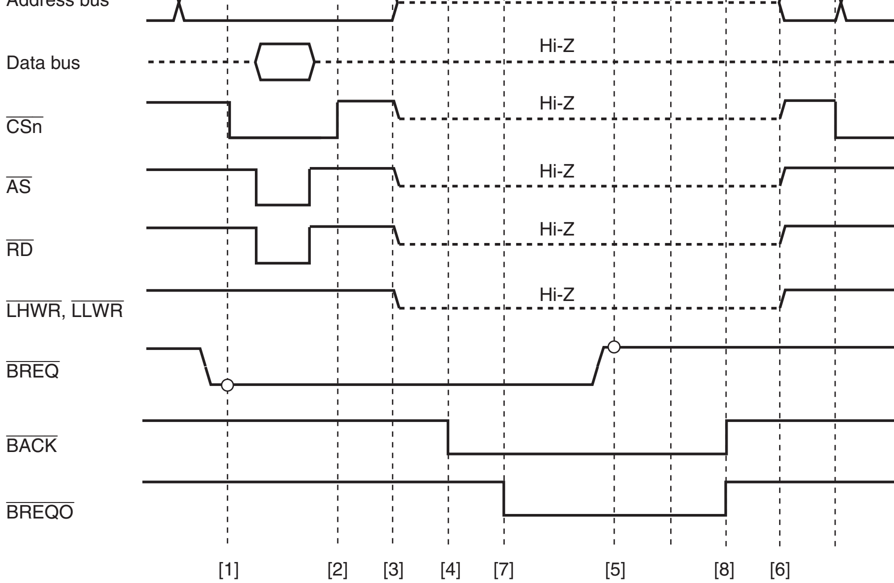
- When the CPU, DTC, DMAC, or EXDMAC attempts to access the external address s
- When a SLEEP instruction is executed to place the chip in software standby mode or module-clock-stop mode
- When SCKCR is written to for setting the clock frequency

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

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

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

$\overline{\mathrm{CSn}}$ (n = 7 to 0)	High impedance
$\overline{\mathrm{AS}}$	High impedance
$\overline{\mathrm{AH}}$	High impedance
RD/$\overline{\mathrm{WR}}$	High impedance
$\overline{\mathrm{RD}}$	High impedance
$\overline{\mathrm{LUB}}$, $\overline{\mathrm{LLB}}$	High impedance
$\overline{\mathrm{LHWR}}$, $\overline{\mathrm{LLWR}}$	High impedance
$\overline{\mathrm{DACKn}}$ (n = 3 to 0)	High level
$\overline{\mathrm{EDACKn}}$ (n = 1 to 0)	High level



[1] A low level of the 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 or more after the low level of the BREQ signal is sampled.
[3] The BACK signal is driven low, releasing bus to the external bus master.
[4] The BREQ signal state sampling is continued in the external bus released state.
[5] A high level of the BREQ signal is sampled.
[6] The external bus released cycles are ended one cycle after the BREQ signal is driven high.
[7] When the external space is accessed by an internal bus master during external bus released while the BR bit is set to 1, the BREQO signal goes low.
[8] Normally the BREQO signal goes high at the rising edge of the BACK signal.

Figure 9.43 Bus Released State Transition Timing

Table 9.25 Number of Access Cycles for On-Chip Memory Spaces

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

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

Table 9.26 lists the number of access cycles for registers of on-chip peripheral modules.

Table 9.26 Number of Access Cycles for Registers of On-Chip Peripheral Module

	Number of Cycles		
Module to be Accessed	**Read**	**Write**	**Write Data Buffer**
DMAC and EXDMAC registers	Two Iϕ	Two Iϕ	Disabled
MCU operating mode, clock pulse generator, power-down control registers, interrupt controller, bus controller, and DTC registers	Two Iϕ	Three Iϕ	Disabled
I/O port registers of PFCR and WDT	Two Pϕ	Three Pϕ	Disabled
I/O port registers other than PFCR and PORTM, PPG0, TPU, TMR0, TMR1, SCI0 to SCI2, SCI4, IIC2, D/A, and A/D_0 registers	Two Pϕ	Two Pϕ	Enabled
I/O port registers of PORTM, TMR2, TMR3, USB, SCI5, SCI6, A/D_1, and PPG1 registers	Three Pϕ	Three Pϕ	Enabled

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

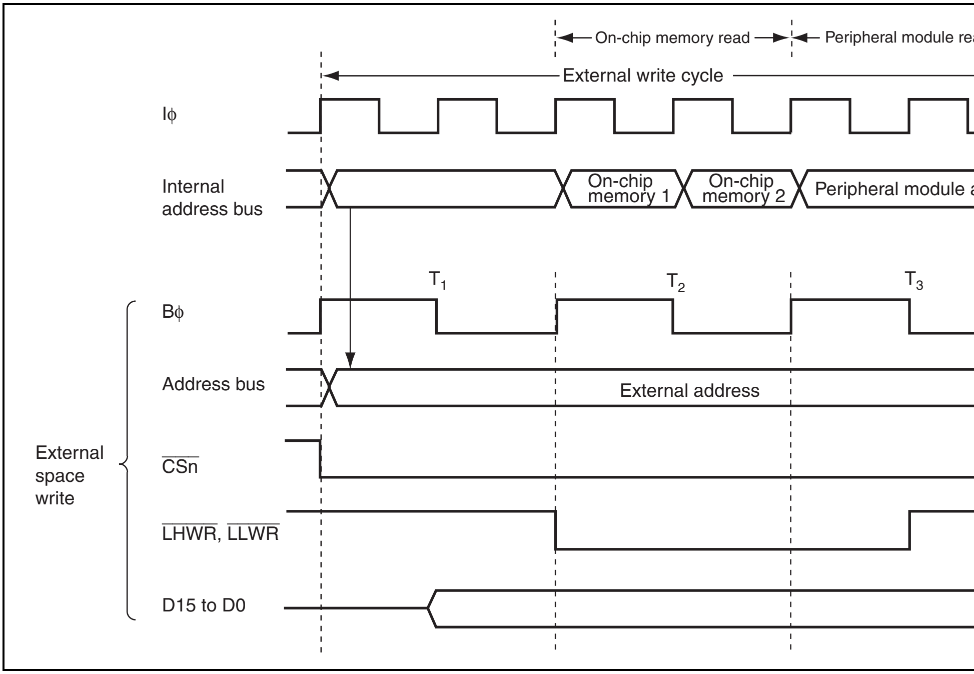


Figure 9.44 Example of Timing when Write Data Buffer Function is Used

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

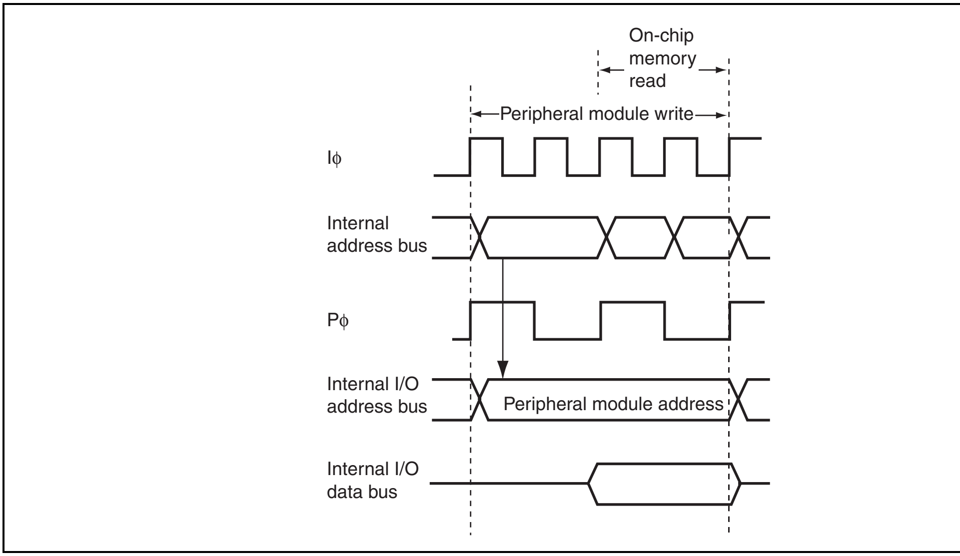


Figure 9.45 Example of Timing when Peripheral Module Write Data Buffer Function is Used

9.14.1 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, se request acknowledge signal to the bus master. If there are bus requests from more than or 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 th is canceled.

The priority of the internal bus arbitration:

(High) DMAC > DTC > CPU (Low)

The priority of the external bus arbitration:

(High) EXDMAC > External bus release request > External access by the CPU, DTC DMAC (Low)

If the DMAC or DTC accesses continue, the CPU can be given priority over the DMAC o to execute the bus cycles alternatively between them by setting the IBCCS bit in BCR2. I case, the priority between the DMAC and DTC does not change. If the external bus releas or EXDMAC accesses continue, the external access by the CPU, DTC, and DMAC can b priority over the EXDMAC or external bus release request to execute the bus cycles alter between them by setting the EBCCS bit in BCR2. In this case, the priority between the E and external bus release request does not change.

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

EXDMAC that issued the request.

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 instructio instruction.

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

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

(2) DTC

The DTC sends the internal bus arbiter a request for the bus when an activation request i generated. When the DTC accesses an external bus space, the DTC first takes control of 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 cycle master whose priority is higher than the DTC requests the bus, the DTC transfers the bu higher priority bus master. If the IBCCS bit in BCR2 is set to 1, the DTC transfers the b CPU.

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

After the DMAC takes control of the bus, it may continue the transfer processing cycles o
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 to cycle

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

- 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) EXDMAC

The EXDMAC sends the external bus arbiter a request for the bus when an activation req
generated. During external access by the internal bus master, the bus is transferred to the EXDMAC at the timing the bus can be transferred.

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

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

- Between the read cycle in the dual-address mode and the write cycle corresponding to cycle
- During transfers in the cluster transfer mode

(3) External Bus Release

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

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

other than an instruction fetch access.

(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 and MSTPCR H'FFFFFFFF) or for operation of the 8-bit timer module alone (MSTPCRA and MSTPCF H'F[F to C]FFFFFF), and a transition is made to the sleep state, the all-module-clock-stop entered in which the clock is also stopped for the bus controller and I/O ports. For details section 27, Power-Down Modes.

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

(3) External Bus Release Function and Software Standby

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

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

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

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

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

  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

- 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 $\overline{\text{DACK}}$ signal a 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 ad 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

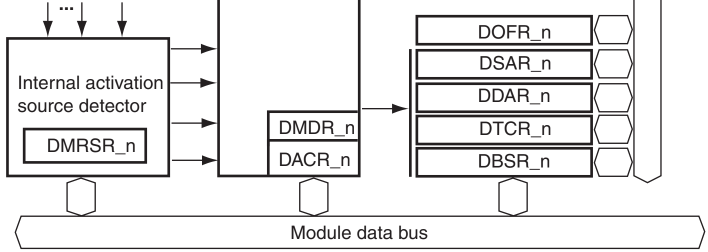
  Up to 65,536 bytes/words/longwords can be set as bloc respective boundary

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.

- Module stop state can be set.



[Legend]

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

Figure 10.1 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)

10.3.1 DMA Source Address Register (DSAR)

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

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

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

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 longword
must not be written to while data for the channel is being transferred.

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

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

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

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

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
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	All 0	R/W	Specify the repeat size or block size. When H'0001 is set, the repeat or block size is one word, or one longword. When H'0000 is se means the maximum value (refer to table 10.1) the DMA is in operation, the setting is fixed.
15 to 0	BKSZ15 to BKSZ0	All 0	R/W	Indicate the remaining repeat or block size whil DMA is in operation. The value is decremented every time data is transferred. When the remair becomes 0, the value of the BKSZH bits is load the same value as the BKSZH bits.

DMDR controls the DMAC operation.

- DMDR_0

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

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

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

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

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

Bit Name	DTSZ1	DTSZ0	MDS1	MDS0	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.

RENESAS

transfer.

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

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

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

0: Disables a data transfer

1: Enables a data transfer (DMA is in operatio

[Clearing conditions]

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

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

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

				1: Enables 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 o $\overline{\text{DREQ}}$ signal used in external request mode is 0: Low level detection 1: Falling edge detection (the first transfer after transfer enabled is detected on a low level)
26	NRD	0	R/W	Next Request Delay Selects the accepting timing of the next transfe 0: Starts accepting the next transfer request aft completion of the current transfer 1: Starts accepting the next transfer request on of Bϕ after completion of the current transfer
25, 24	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
23	ACT	0	R	Active State Indicates the operating state for the channel. 0: Waiting for a transfer request or a transfer di state by clearing the DTE bit to 0 1: Active state
22 to 20	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

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

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

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

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

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.

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

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

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

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

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

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

				When an overflow in the extended repeat area with the SARIE bit set to 1, an interrupt can be requested. Table 10.3 shows the settings and 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
6, 5	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

becomes the start address and the end addres
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 10.3 shows the settings and a
the extended repeat area.

00101	32 bytes specified as extended repeat area by the lower 5 bits of the addres
00110	64 bytes specified as extended repeat area by the lower 6 bits of the addres
00111	128 bytes specified as extended repeat area by the lower 7 bits of the addre
01000	256 bytes specified as extended repeat area by the lower 8 bits of the addre
01001	512 bytes specified as extended repeat area by the lower 9 bits of the addre
01010	1 Kbyte specified as extended repeat area by the lower 10 bits of the addres
01011	2 Kbytes specified as extended repeat area by the lower 11 bits of the addre
01100	4 Kbytes specified as extended repeat area by the lower 12 bits of the addre
01101	8 Kbytes specified as extended repeat area by the lower 13 bits of the addre
01110	16 Kbytes specified as extended repeat area by the lower 14 bits of the add
01111	32 Kbytes specified as extended repeat area by the lower 15 bits of the add
10000	64 Kbytes specified as extended repeat area by the lower 16 bits of the add
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 adc
11001	32 Mbytes specified as extended repeat area by the lower 25 bits of the adc
11010	64 Mbytes specified as extended repeat area by the lower 26 bits of the adc
11011	128 Mbytes specified as extended repeat area by the lower 27 bits of the ac
111××	Setting prohibited

[Legend]

×: Don't care

address	• Repeat transfer • Block transfer Repeat or block size = 1 to 65,536 bytes, 1 to 65,536 words, or 1 to 65,536 longwords	(activated by CPU) • On-chip module interrupt • External request	size: 1 to 4 Gbytes or not specified • Offset addition • Extended repeat area function	
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 b can be selected. When the total transfer size is not specified (DTCR = H'00000000), the counter is stopped and the transfer is continued without the limitation of the transfer cou divided into multiple bus cycles).

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

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

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

Figure 10.2 shows an example of the signal timing in dual address mode and figure 10.3 s
operation in dual address mode.

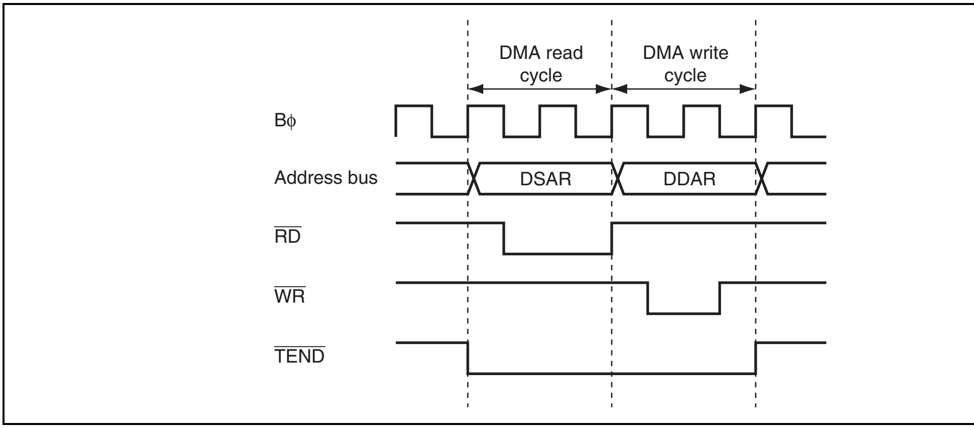


Figure 10.2 Example of Signal Timing in Dual Address Mode

Figure 10.3 Operations in Dual Address Mode

(2) Single Address Mode

In single address mode, data between an external device and an external memory is dire
transferred using the $\overline{\text{DACK}}$ pin instead of DSAR or DDAR. A transfer at a time is perf
one bus cycle. In this mode, the data bus width must be the same as the data access size.
details on the data bus width, see section 9, Bus Controller (BSC).

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

The transfer direction is decided by the DIRS bit in DACR which specifies an external d
the $\overline{\text{DACK}}$ pin as the transfer source or destination. When DIRS = 0, data is transferred
external memory (DSAR) to an external device with the $\overline{\text{DACK}}$ pin. When DIRS = 1, da
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 DMD
$\overline{\text{DACK}}$ signal is low active.

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

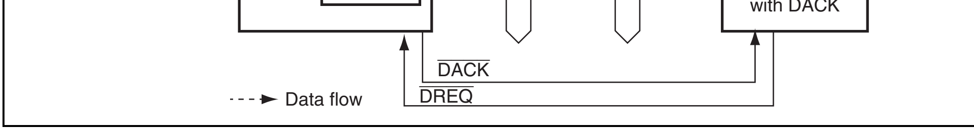


Figure 10.4 Data Flow in Single Address Mode

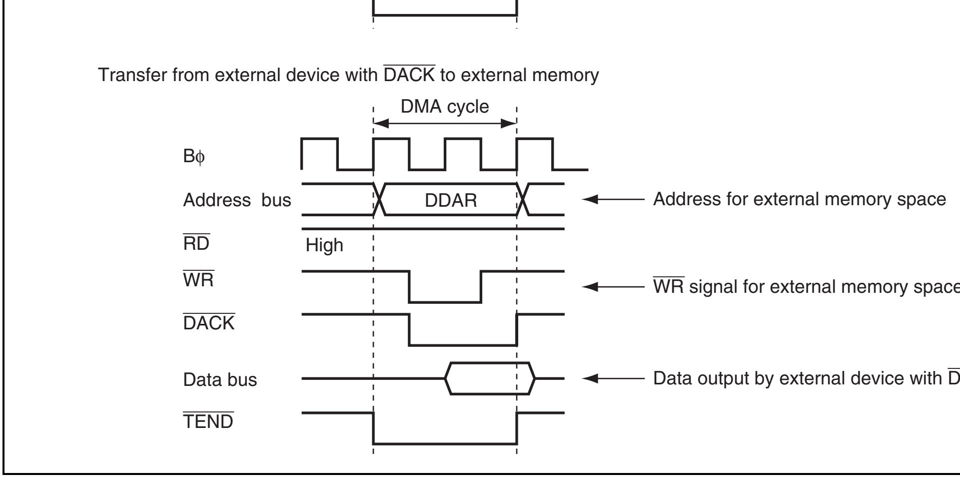


Figure 10.5 Example of Signal Timing in Single Address Mode

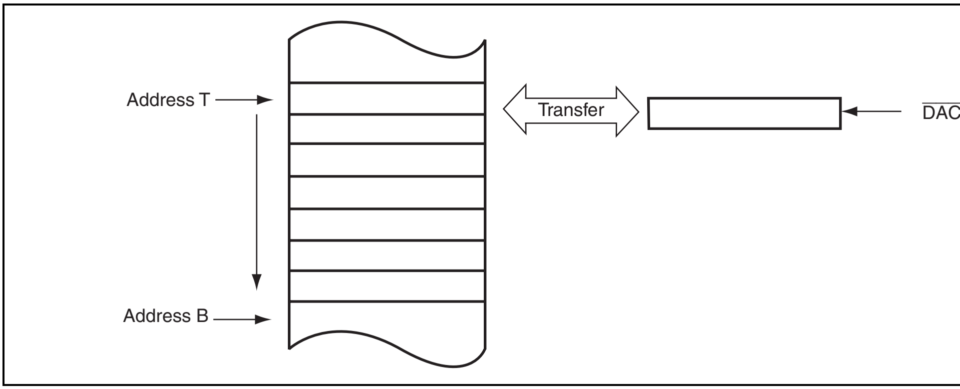


Figure 10.6 Operations in Single Address Mode

the operation in normal transfer mode.

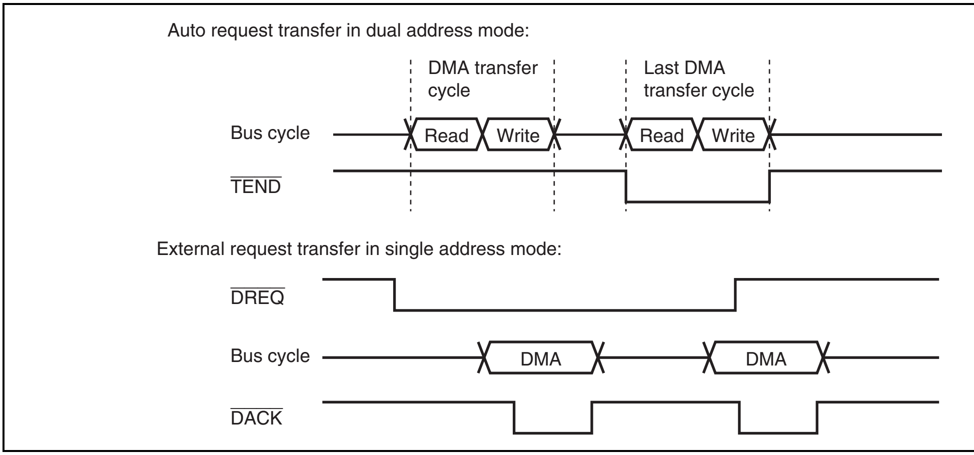


Figure 10.7 Example of Signal Timing in Normal Transfer Mode

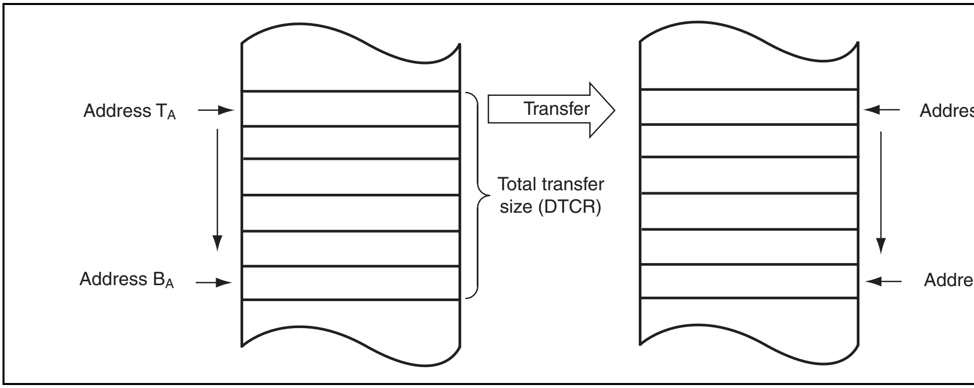


Figure 10.8 Operations in Normal Transfer Mode

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 timing of the $\overline{\text{TEND}}$ signal is the same as in normal transfer mode.

Figure 10.9 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 10.8. In this case, a repe
end interrupt can also be requested to the CPU when the repeat size of transfers is compl

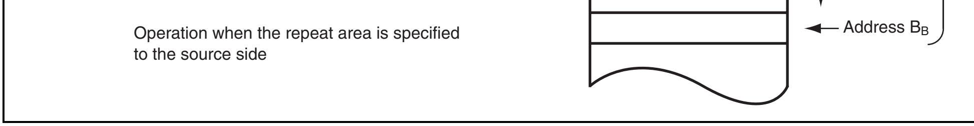


Figure 10.9 Operations in Repeat Transfer Mode

(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 64 k data access size.

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

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

The $\overline{\text{TEND}}$ signal is output every time 1-block data is transferred in the last DMA transfe

When an interrupt request by an extended repeat area overflow is used in block transfer m
settings should be selected carefully. For details, see section 10.5.5, Extended Repeat Are
Function.

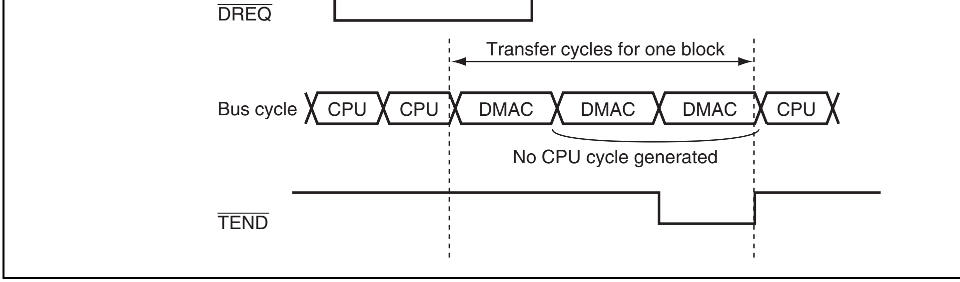


Figure 10.10 Operations in Block Transfer Mode

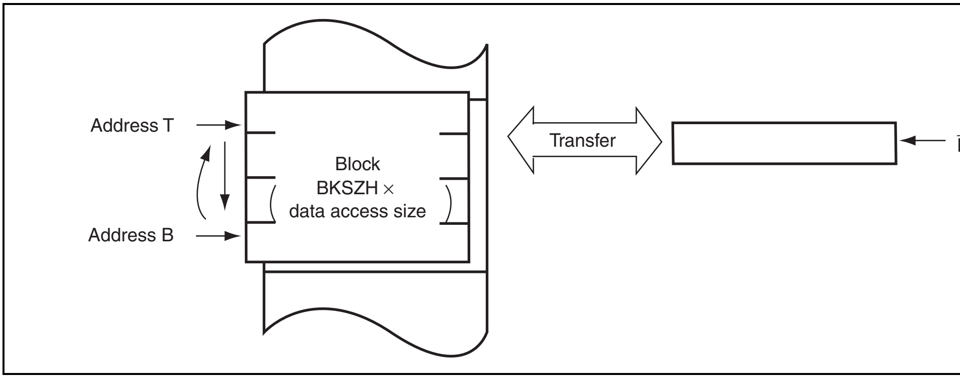


Figure 10.11 Operation in Single Address Mode in Block Transfer Mode (Block Area Specified)

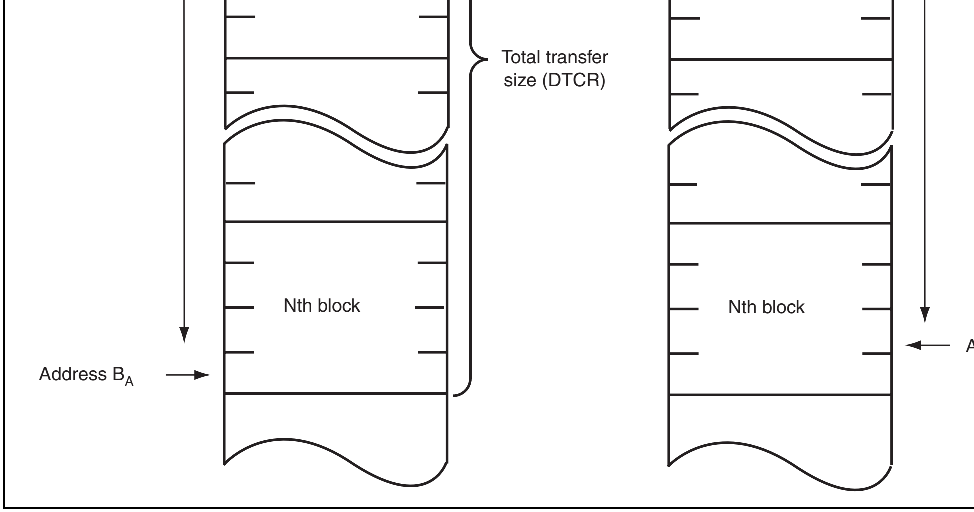


Figure 10.12 Operation in Dual Address Mode in Block Transfer Mode (Block Area Not Specified)

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 10.5 is a list of on-chip module interrupts for the DMAC. The interrupt request selected activation source can generate an interrupt request simultaneously to the CPU or DTC. refer to section 7, 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.

[illegible]	[illegible]	
TGI5A (TGI5A input capture/compare match)	TPU_5	11
RXI0 (receive data full interrupt for SCI channel 0)	SCI_0	14
TXI0 (transmit data empty interrupt for SCI channel 0)	SCI_0	14
RXI1 (receive data full interrupt for SCI channel 1)	SCI_1	14
TXI1 (transmit data empty interrupt for SCI channel 1)	SCI_1	15
RXI2 (receive data full interrupt for SCI channel 2)	SCI_2	15
TXI2 (transmit data empty interrupt for SCI channel 2)	SCI_2	15
RXI4 (receive data full interrupt for SCI channel 4)	SCI_4	16
TXI4 (transmit data empty interrupt for SCI channel 4)	SCI_4	16
TGI6A (TGI6A input capture/compare match)	TPU_6	16
TGI7A (TGI7A input capture/compare match)	TPU_7	16
TGI8A (TGI8A input capture/compare match)	TPU_8	17
TGI9A (TGI9A input capture/compare match)	TPU_9	17
TGI10A (TGI10A input capture/compare match)	TPU_10	18
TGI11A (TGI11A input capture/compare match)	TPU_11	18
RXI5 (receive data full interrupt for SCI channel 5)	SCI_5	22
TXI5 (transmit data empty interrupt for SCI channel 5)	SCI_5	22
RXI6 (receive data full interrupt for SCI channel 6)	SCI_6	22
TXI6 (transmit data empty interrupt for SCI channel 6)	SCI_6	22
USBINTN0 (EP1FIFO full interrupt)	USB	23
USBINTN1 (EP2FIFO empty interrupt)	USB	23
ADI1 (conversion end interrupt for A/D converter unit 1)	A/D_1	23

When an external request is selected as an activation source, clear the DDR bit to 0 and
ICR bit to 1 for the corresponding pin. For details, see section 13, I/O Ports.

10.5.4 Bus Access Modes

There are two types of bus access modes: cycle stealing and burst.

When an activation source is the auto request, the cycle stealing or burst mode is selecte
DTF0 in DMDR. When an activation source is the on-chip module interrupt or external
the cycle stealing mode is selected.

(1) Cycle Stealing Mode

In cycle stealing mode, the DMAC releases the bus every time one unit of transfers (byt
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 10.5.8, Priority of Channels.

Bus released temporarily for the CPU

Figure 10.13   Example of Timing in Cycle Stealing Mode

(2) Burst Access Mode

In burst mode, once it takes the bus, the DMAC continues a transfer without releasing the the transfer end condition is satisfied. Even if a transfer is requested from another channe priority, the transfer is not stopped once it is started. The DMAC releases the bus in the n after the transfer for the channel in burst mode is completed. This is similarly to operatio stealing mode. However, setting the IBCCS bit in BCR2 of the bus controller makes the release the bus to pass the bus to another bus master.

In block transfer mode, the burst mode setting is ignored (operation is the same as that in mode during one block of transfers). The DMAC is always operated in cycle stealing mo

Clearing the DTE bit in DMDR stops a DMA transfer. A transfer requested before the D 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 10.14 shows an example of timing in burst mode.

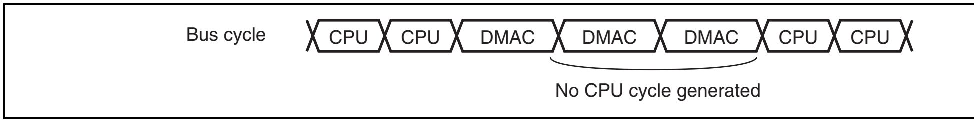


Figure 10.14   Example of Timing in Burst Mode

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.

A DMA transfer is stopped and an interrupt by an extended repeat area overflow can be
to the CPU when the contents of the address register reach the end address of the extend
area. When an overflow on the extended repeat area set in DSAR occurs while the SARI
DACR is set to 1, the ESIF bit in DMDR is set to 1 and the DTE bit in DMDR is cleared
stop the transfer. At this time, if the ESIE bit in DMDR is set to 1, an interrupt by an ext
repeat area overflow is requested to the CPU. When the DARIE bit in DACR is set to 1,
overflow on the extended repeat area set in DDAR occurs, meaning that the destination
target. During the interrupt handling, setting the DTE bit in DMDR resumes the transfer

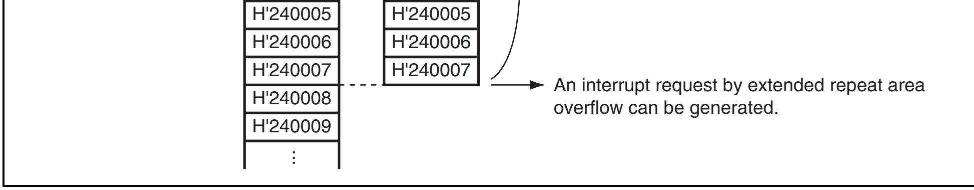


Figure 10.15 Example of Extended Repeat Area Operation

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

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

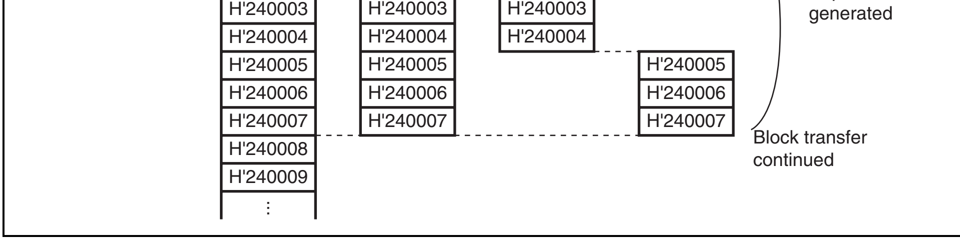


Figure 10.16 Example of Extended Repeat Area Function in Block Transfer I

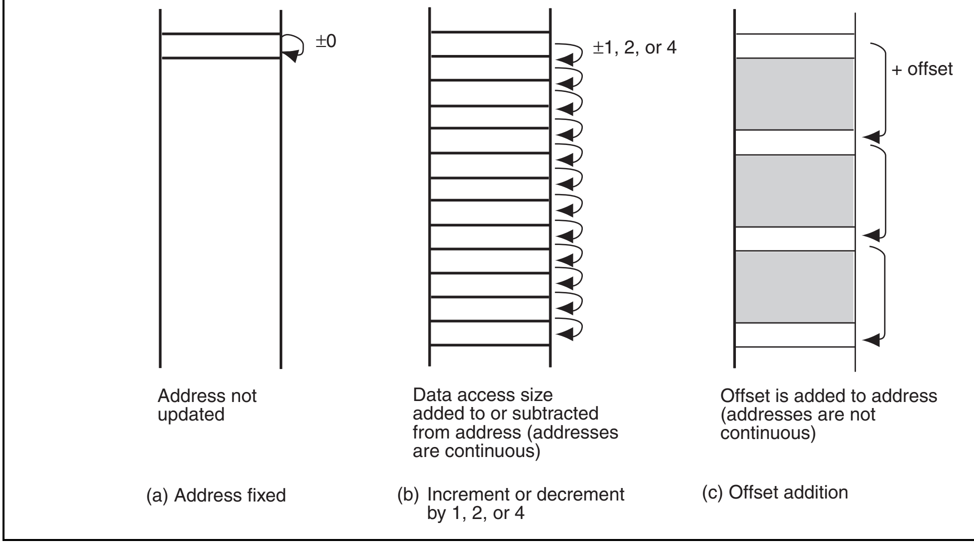


Figure 10.17 Address Update Method

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

In item (b), Increment or decrement by 1, 2, or 4, the transfer source or destination addres
incremented or decremented by the value according to the data access size at each transfe
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 p
consecutive areas.

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

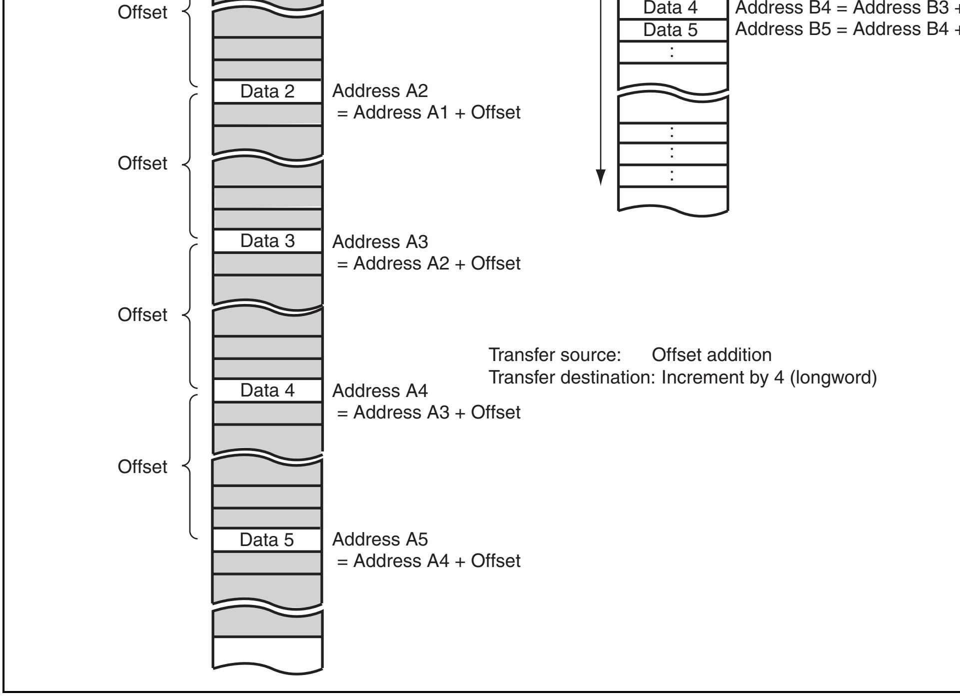


Figure 10.18 Operation of Offset Addition

In figure 10.18, the offset addition is selected as the transfer source address update and i
or decrement by 1, 2, or 4 is selected as the transfer destination address. The address upd
that data at the address which is away from the previous transfer source address by the o
read from. The data read from the address away from the previous address is written to t
consecutive area in the destination side.

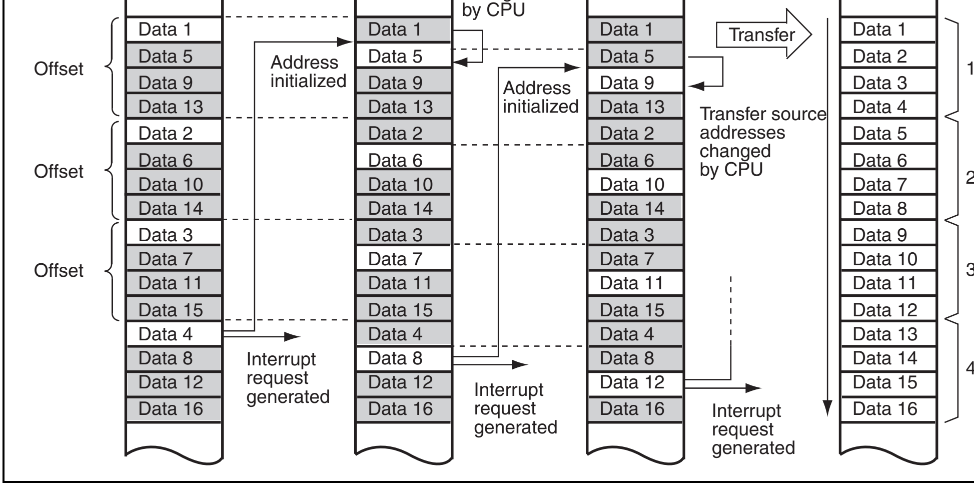


Figure 10.19 XY Conversion Operation Using Offset Addition in Repeat Transfe

In figure 10.19, 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 s longword, H'00000010 is set in DOFR, as an example). The repeat size is set to 4 × data a size (when the data access size is longword, the repeat size is set to 4 × 4 = 16 bytes, as a 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 RPTIE bit in DACR is set to 1 and the r size of transfers is completed.

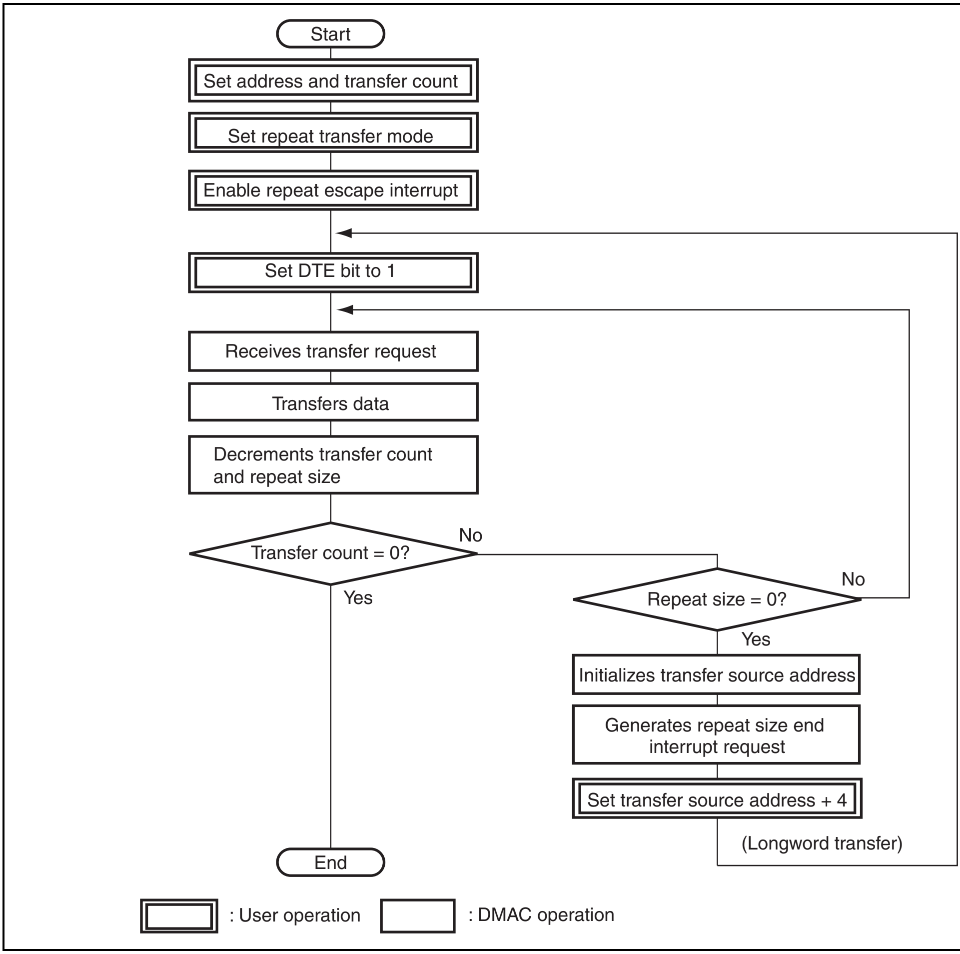


Figure 10.20   XY Conversion Flowchart Using Offset Addition in Repeat Transf

10.5.7 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. 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.

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

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 10.21 show the procedure for changing the register settings for the channel being transferred.

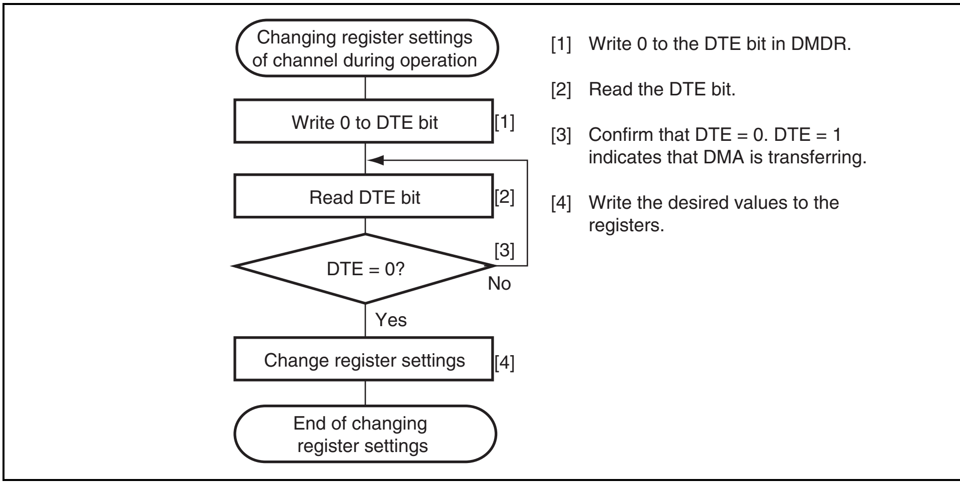


Figure 10.21 Procedure for Changing Register Setting For Channel being Tran

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.

However, when the DMAC is in the module stop state, the ERRF bit is not set to 1 for ad
errors or the NMI.

(8) ESIF Bit in DMDR

When an interrupt by an transfer size error, a repeat size end, or an extended repeat area c
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 10.8, Interrupt Sources.

For details on interrupts, see section 10.8, Interrupt Sources.

10.5.8 Priority of Channels

The channels of the DMAC are given following priority levels: channel 0 > channel 1 > channel 2 > channel3. Table 10.6 shows the priority levels among the DMAC channels.

Table 10.6 Priority among DMAC Channels

Channel	Pr
Channel 0	Hig
Channel 1	
Channel 2	
Channel 3	Lo

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.

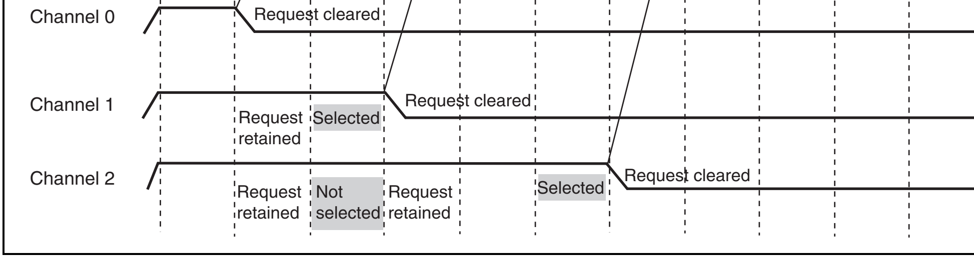


Figure 10.22 Example of Timing for Channel Priority

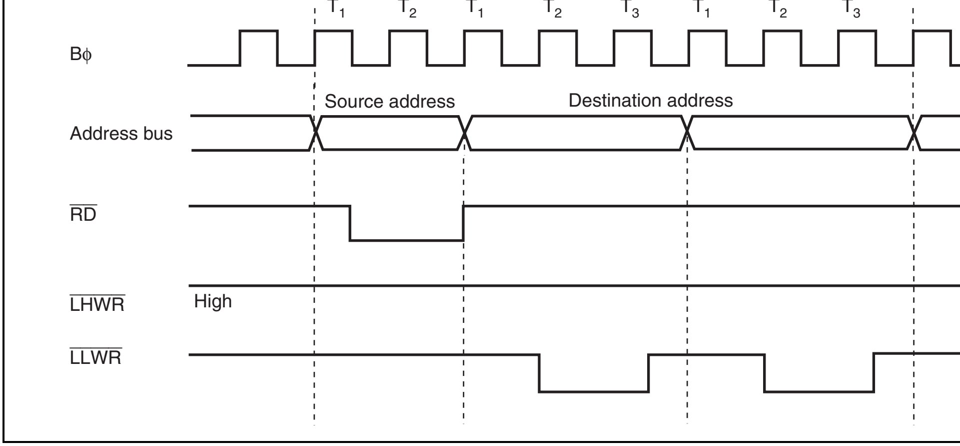


Figure 10.23 Example of Bus Timing of DMA Transfer

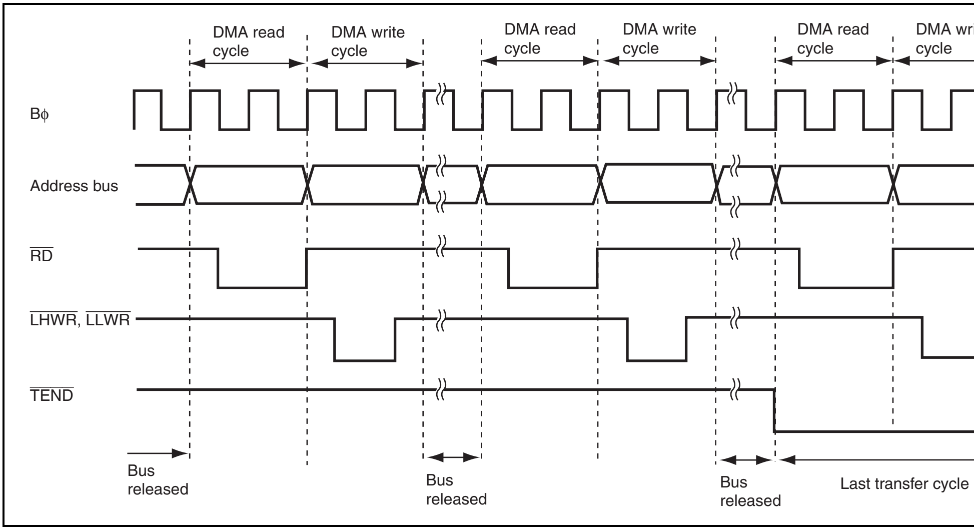


Figure 10.24 Example of Transfer in Normal Transfer Mode by Cycle Steali

In figures 10.25 and 10.26, the $\overline{\text{TEND}}$ signal output is enabled and data is transferred in l
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 10.25, the transfer source (DSAR) is not aligned with a longword boundary and
transfer destination (DDAR) is aligned with a longword boundary.

In figure 10.26, the transfer source (DSAR) is aligned with a longword boundary and the
destination (DDAR) is not aligned with a longword boundary.

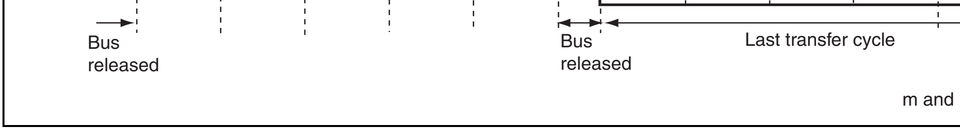


Figure 10.25 Example of Transfer in Normal Transfer Mode by Cycle Steal
(Transfer Source DSAR = Odd Address and Source Address Increment)

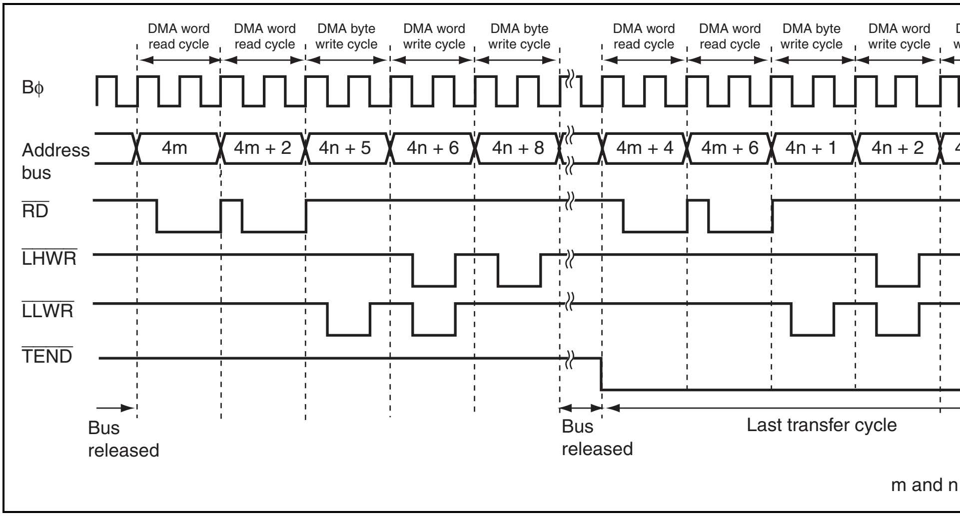


Figure 10.26 Example of Transfer in Normal Transfer Mode by Cycle Steal
(Transfer Destination DDAR = Odd Address and Destination Address Decrem

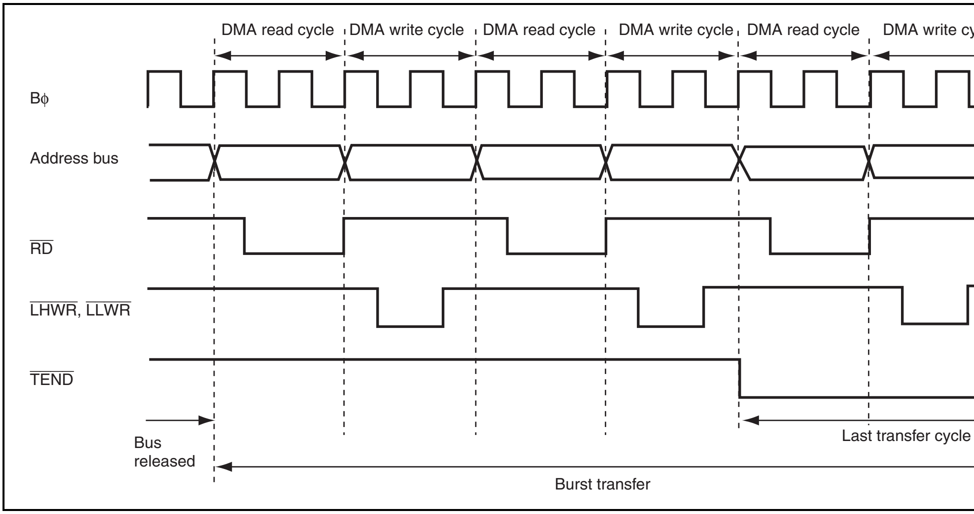


Figure 10.27 Example of Transfer in Normal Transfer Mode by Burst Acce

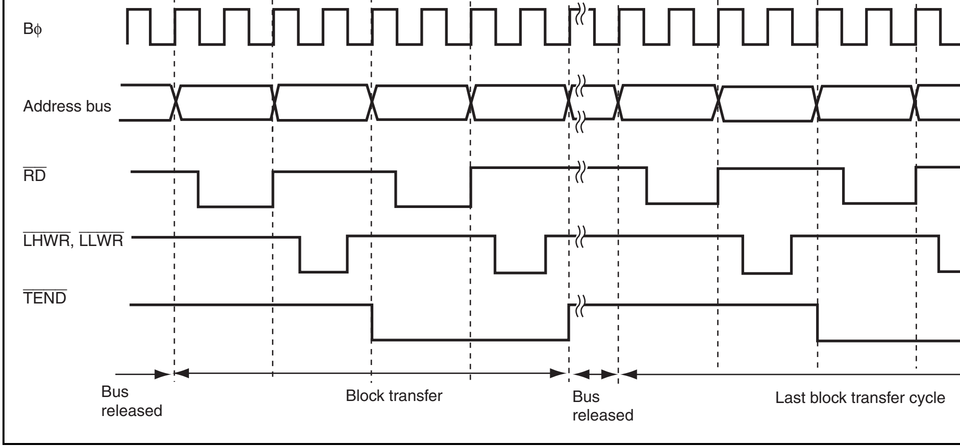
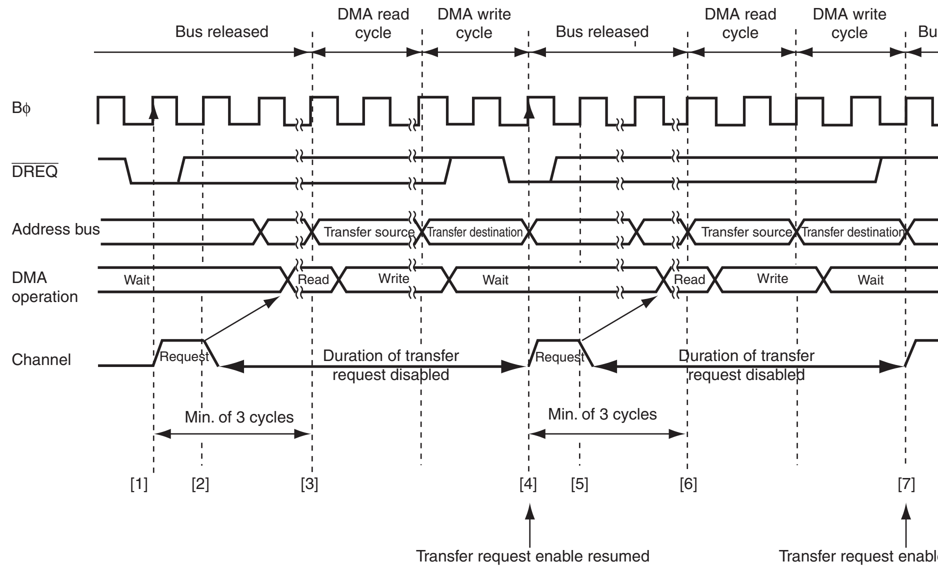


Figure 10.28 Example of Transfer in Block Transfer Mode

a high level of the DREQ signal has been detected until completion of the DMA write cy
receiving the next transfer request resumes and then a low level of the DREQ signal is de
This operation is repeated until the transfer is completed.



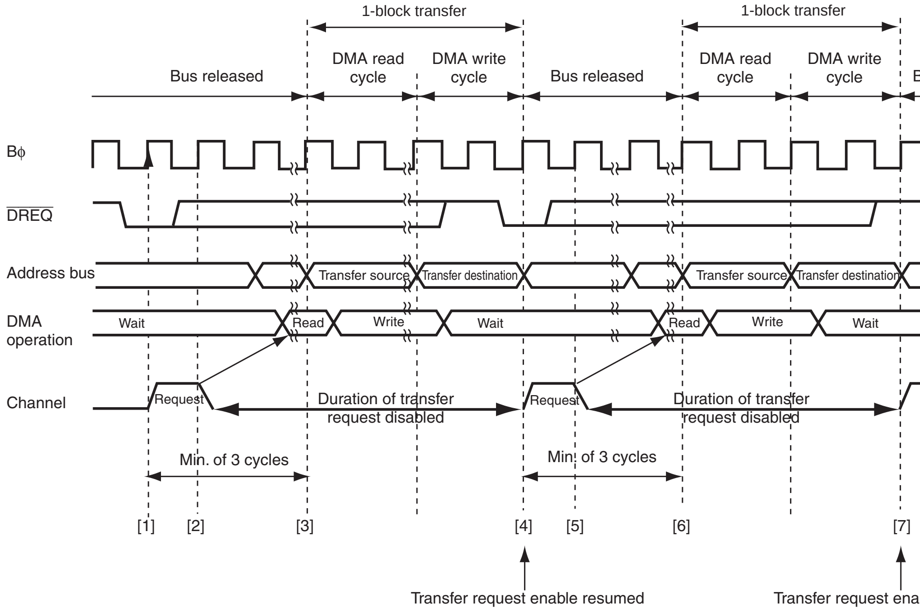
[1] After DMA transfer request is enabled, a low level of the DREQ signal is detected at the rising edge of the Bφ signal and a
request is held.

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

[3][6] A DMA cycle is started and sampling the DREQ signal at the rising edge of the Bφ signal is started to detect a high level c
DREQ signal.

[4][7] When a high level of the DREQ signal has been detected, transfer request enable is resumed after completion of the write
(A low level of the DREQ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the sa

Figure 10.29 Example of Transfer in Normal Transfer Mode Activated by DREQ Falling Edge



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

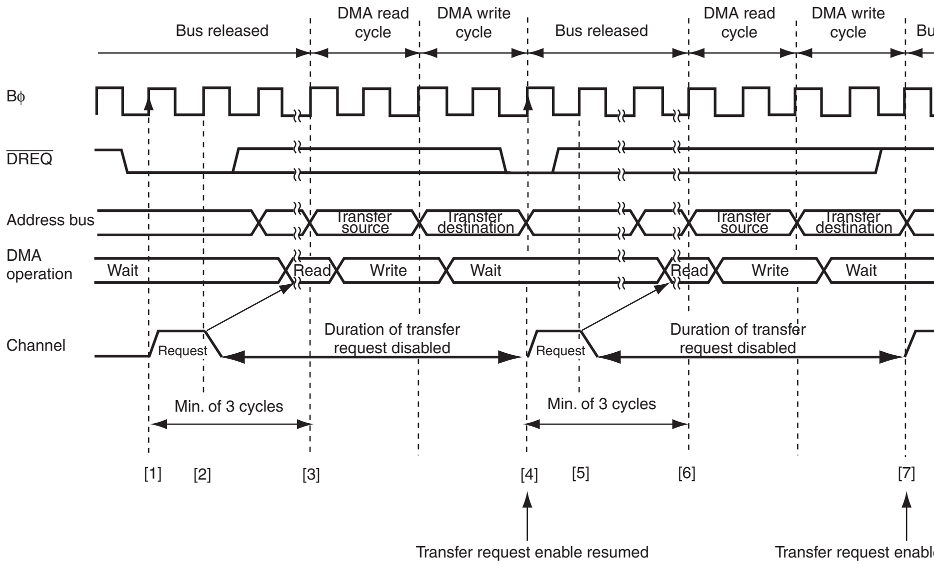
[2][5] The DMAC is activated and the transfer request is cleared.

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

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

Figure 10.30 Example of Transfer in Block Transfer Mode Activated by $\overline{DREQ}$ Falling Edge

completed.



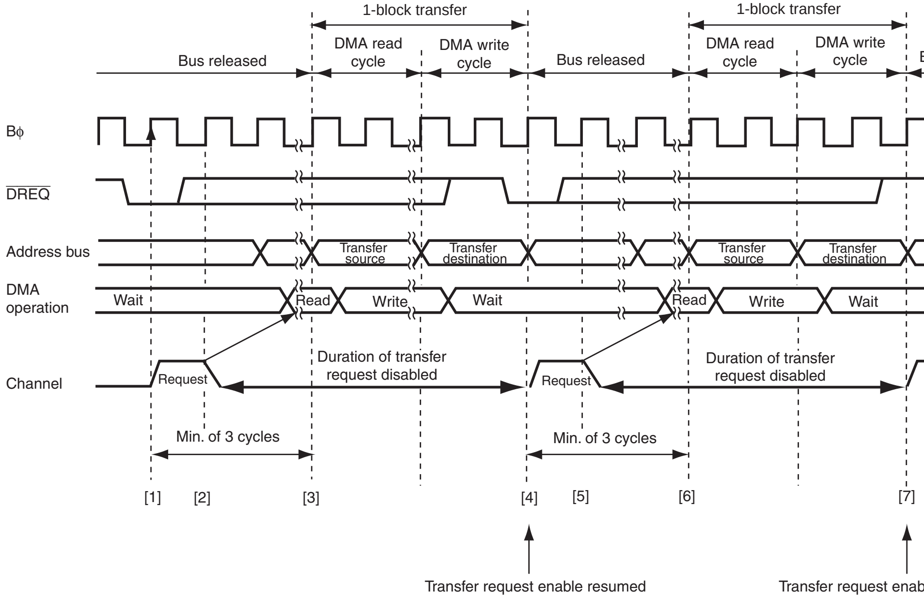
[1] After DMA transfer request is enabled, a low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bϕ signal and a
request is held.

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

[3][6] A DMA cycle is started.

[4][7] Transfer request enable is resumed after completion of the write cycle.
(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 10.31 Example of Transfer in Normal Transfer Mode Activated by $\overline{DREQ}$ Low Level



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

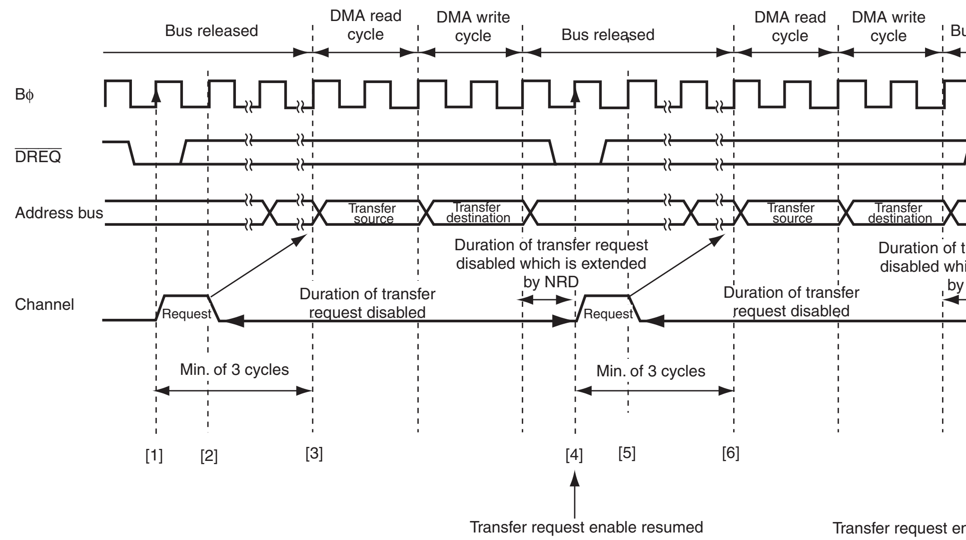
[2][5] The DMAC is activated and the transfer request is cleared.

[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 DREQ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the same a

Figure 10.32 Example of Transfer in Block Transfer Mode Activated by 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 $\overline{DREQ}$ signal is detected. This operation is repeated until the t
completed.



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

Figure 10.33 Example of Transfer in Normal Transfer Mode Activated by $\overline{DREQ}$ Low Level with NRD = 1

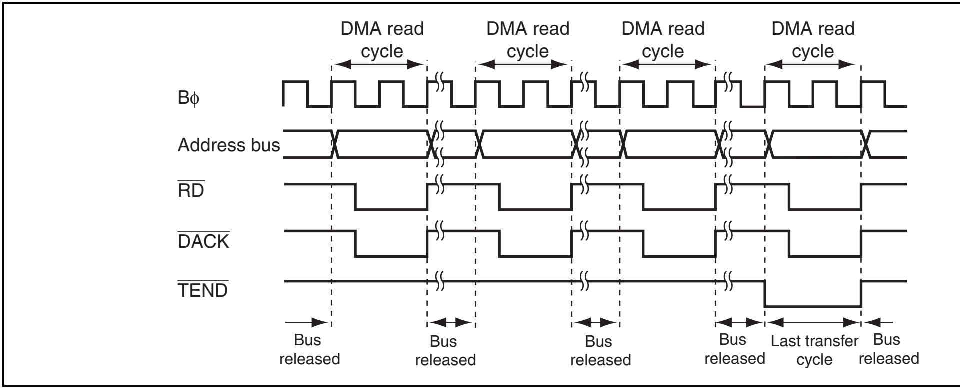


Figure 10.34 Example of Transfer in Single Address Mode (Byte Read)

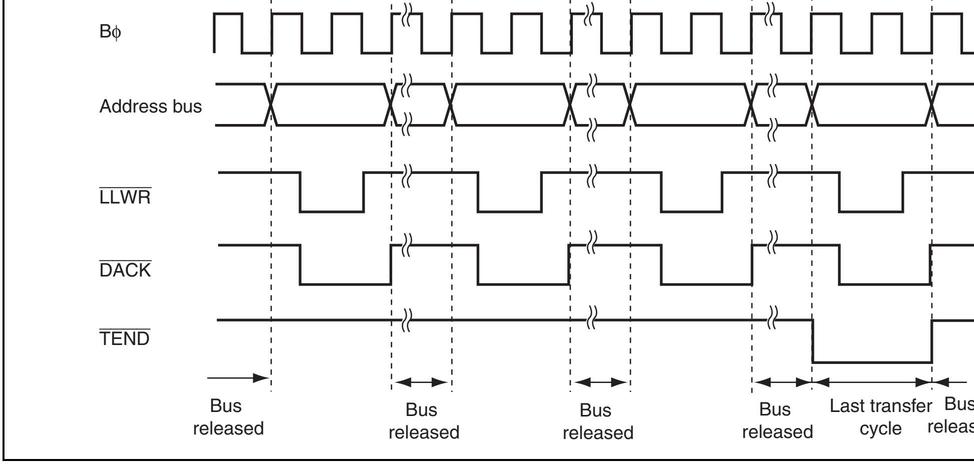
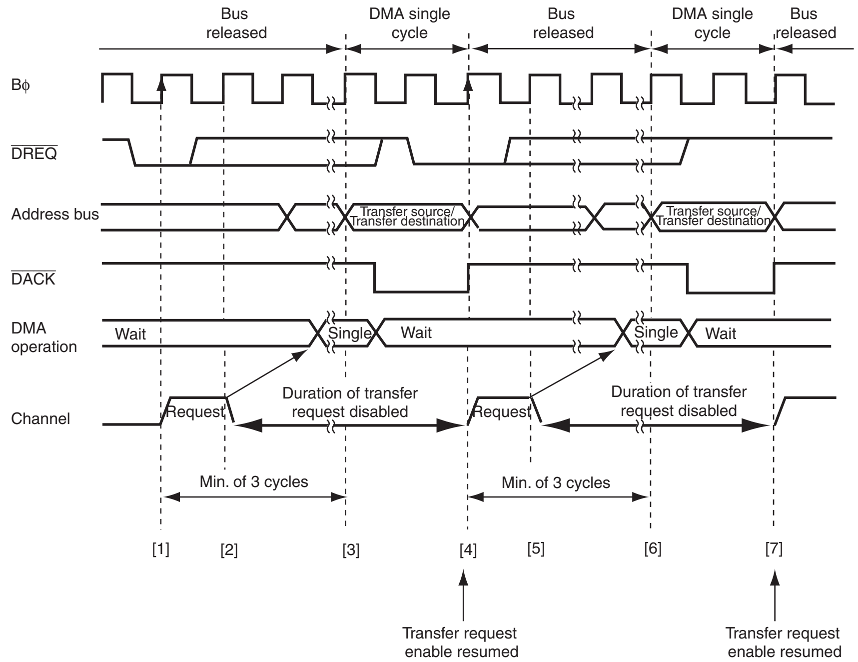


Figure 10.35 Example of Transfer in Single Address Mode (Byte Write)

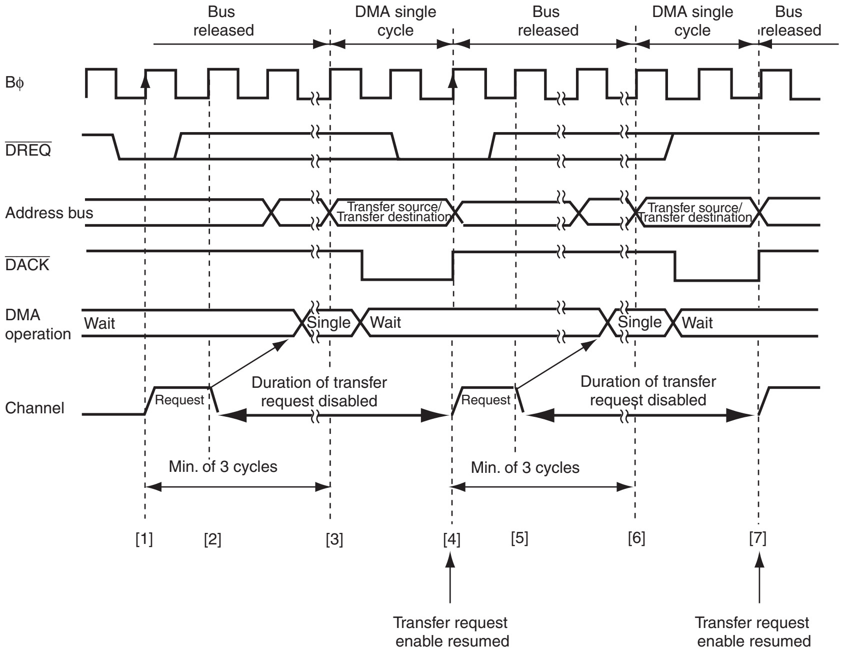
the next transfer request resumes and then a low level of the $\overline{DREQ}$ signal is detected. T
operation is repeated until the transfer is completed.



[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 enable is resumed after completion of the write cycle.
(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 10.36 Example of Transfer in Single Address Mode Activated by $\overline{DREQ}$ Falling Edge

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 tr request is held.

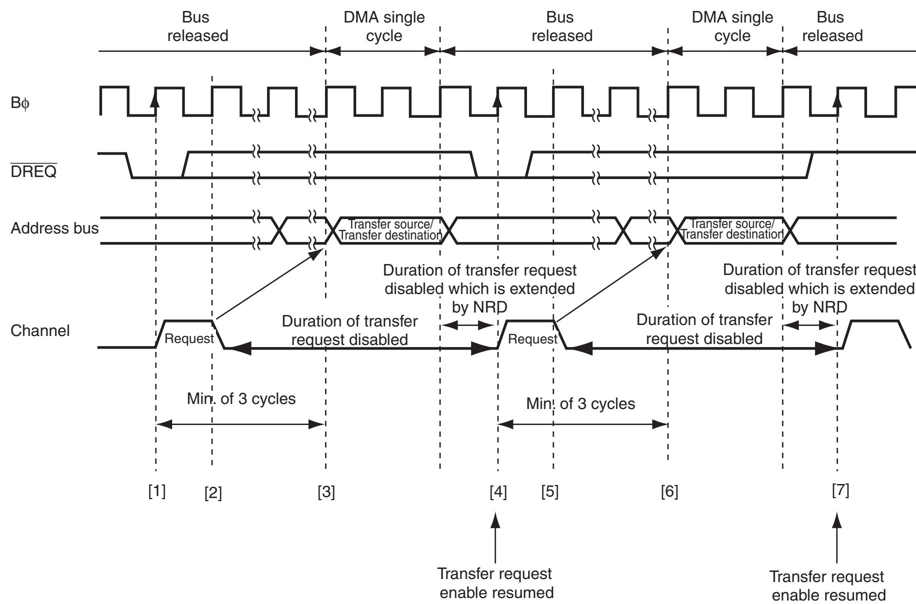
[2][5] The DMAC is activated and the transfer request is cleared.

[3][6] A DMA cycle is started.

[4][7] Transfer request enable is resumed after completion of the single cycle.
(A low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the sam

Figure 10.37 Example of Transfer in Single Address Mode Activated by $\overline{DREQ}$ Low Level

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



[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bφ signal and 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 10.38 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 DMDR is
to 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 disabled
  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 t
  access size, data is transferred in bytes
- In block transfer mode, when the DTCR value is less than the block size, the specified
  data in DTCR is transferred instead of transferring the block size of data. The transfer
  performed in bytes.

When an overflow on the extended repeat area occurs while the extended repeat area is s
and the SARIE or DARIE bit in DACR is set to 1, an interrupt by an extended repeat are
overflow is requested. When the interrupt is requested, the DMA transfer is terminated,
bit in DMDR is cleared to 0, and the ESIF bit in DMDR is set to 1.

In dual address mode, even if an interrupt by an extended repeat area overflow occurs du
read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow occurs d
block transfer, the remaining data is transferred. The transfer is not terminated by an ext
repeat area overflow interrupt unless the current transfer is complete.

(5) Transfer End by Clearing DTE Bit in DMDR

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

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

transfer unit.

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

(b) Block Transfer Mode

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

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

(7) Transfer End by Address Error

When an address error occurs, the DTE bits for all the channels are cleared to 0 and the E
in DMDR_0 is set to 1. When an address error occurs during a DMA transfer, the transfe
forced to stop. To perform a DMA transfer after an address error occurs, clear the ERRF
and then set the DTE bits for the channels.

The transfer end timing after an address error is the same as that after an NMI interrupt.

(8) Transfer End by Hardware Standby Mode or Reset

The DMAC is initialized by a reset and a transition to the hardware standby mode. A DM
transfer is not guaranteed.

The priority level of the CPU is specified by bits CPUP2 to CPUP0. The value of bits C
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 h
over the DMAC, a transfer request for the corresponding channel is masked and the tran
activated. When another channel has priority over or the same as the CPU, a transfer req
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 th
channel is given priority over the CPU by changing priority levels of the CPU or channe
transfer request is received and the transfer is resumed. Writing 0 to the DTE bit clears t
suspended transfer request.

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

of 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 ove CPU and DTC, accesses to the external space is suspended (the IBCCS bit in the bus con register 2 (BCR2) is cleared to 0).

When the bus is passed to another channel or an auto request transfer by cycle stealing, b 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 s IBCCS bit in BCR2, the bus is used alternatively except the bus cycles which are not sep For details, see section 9, Bus Controller (BSC).

A conflict may occur between external space access of the DMAC, and the EXDMAC cy external bus release cycle. Even if a burst or block transfer is performed by the DMAC, t transfer is stopped temporarily and the EXDMAC cycle or external bus release cycle is in the BSC according to the external bus priority (when the CPU external access and the DT external access do not have priority over a DMAC transfer, the transfers are not operated 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 t EXDMAC cycle or 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

Each interrupt is enabled or disabled by the DTIE and ESIE bits in DMDR for the corres channel. A DMTEND interrupt is generated by the combination of the DTIF and DTIE DMDR. A DMEEND interrupt is generated by the combination of the ESIF and ESIE b DMDR. The DMEEND interrupt sources are not distinguished. The priority among cha decided by the interrupt controller and it is shown in table 10.7. For details, see section Controller.

An interrupt other than the transfer end interrupt by the transfer counter is generated whe
ESIF bit in DMDR is set to 1. The ESIF bit is set to 1 when the conditions are satisfied by
transfer while the enable bit is set to 1.

A transfer size error interrupt is generated when the next transfer cannot be performed be
DTCR value is less than the data access size, meaning that the data access size of transfer
be performed. In block transfer mode, the block size is compared with the DTCR value fo
transfer error decision.

A repeat size end interrupt is generated when the next transfer is requested after completi
repeat size of transfers in repeat transfer mode. Even when the repeat area is not specified
address register, the transfer can be stopped periodically according to the repeat size. At t
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 addresses
generated when the address exceeds the extended repeat area (overflow). At this time, wh
transfer end interrupt by the transfer counter, the ESIF bit is set to 1.

Figure 10.39 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 transfer
setting the DTE bit in DMDR after setting the register. Figure 10.40 shows procedure to r
the transfer by clearing an interrupt.

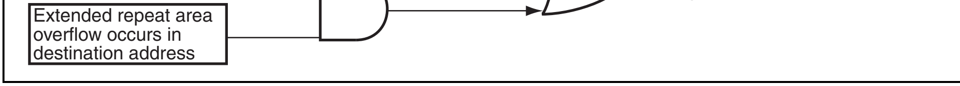
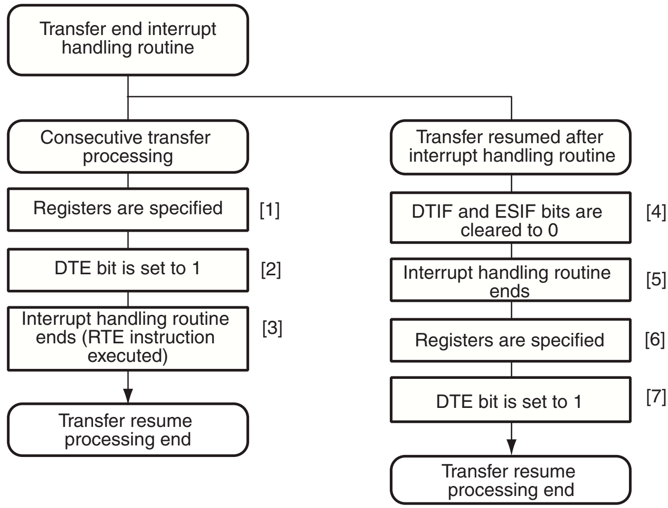


Figure 10.39 Interrupt and Interrupt Sources



[1] Specify the values in the registers such as transfer counter and address register.
[2] Set the DTE bit in DMDR to 1 to resume DMA operation. Setting the DTE bit to 1 automatically clears the 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 10.40 Procedure Example of Resuming Transfer by Clearing Interrupt

enters the module stop state. However, when a transfer for a channel is enabled or wh
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 fol
register settings are valid in the module stop state. Disable them before entering the m
stop state, if necessary.

— TENDE bit in DMDR is 1 (the $\overline{\text{TEND}}$ signal output enabled)
— DACKE bit in DMDR is 1 (the $\overline{\text{DACK}}$ 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 tra
      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 transition
      made.

   After a DMAC transfer enabled, a transition to 1. is made. Therefore, the $\overline{\text{DREQ}}$ sign
   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 o
   setting of $\overline{\text{DREQ}}$ falling edge or low level detection. Therefore, if the $\overline{\text{DREQ}}$ signal is
   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.

- Up to 4-Gbyte address space accessible
- Selection of byte, word, or longword transfer data length
- Total transfer size of up to 4 Gbytes (4,294,967,295 bytes)
  Selection of free-running mode (with no total transfer size specified)
- Selection of auto-requests or external requests for activating the EXDMAC
  Auto-request: Activation from the CPU (Cycle steal mode or burst mode can be sele
  External request: Low level sensing or falling edge sensing for the $\overline{\text{EDREQ}}$ signal ca
  selected.
  Only channel 0 or 1 can accept external requests.
- Selection of dual address mode or single address mode
  Dual address mode: Both the transfer source and destination addresses are specified
  data.
  Single address mode: The $\overline{\text{EDACK}}$ signal is used to access the transfer source or des
  peripheral device and the address of the other device is specified to transfer data.
  Only channel 0 or 1 can be selected for single address mode.
- Normal, repeat, block, or cluster transfer (only for the EXDMAC) can be selected as
  mode

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

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

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

  Up to 64-Kbyte transfers can be set as repeat size (65,536
  bytes/words/longwords)

- Selection of address update methods: Increment/decrement by 1, 2 or 4, fixed, or offs
  addition

  When offset addition is used to update addresses, the mid-addresses can be skipped du
  transfer.

- Transfer of word or longword data to addresses beyond each data boundary

  Data can be divided into an optimal data size (byte or word) according to addresses w
  transferring data.

- Two kinds of interrupts requested to the CPU

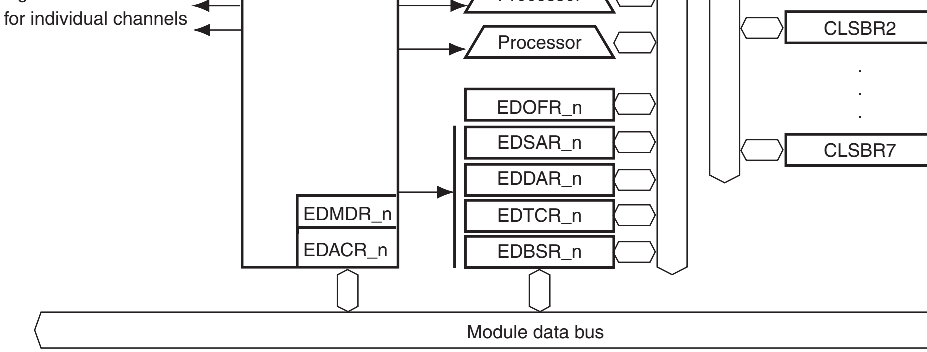
  Transfer end interrupt: Requested after the number of data set by the transfer counter
  completely transferred

  Transfer escape end interrupt: Requested when the remaining transfer size is smaller t
  size set for a single transfer request, after a repeat-size transfer is completed, or when
  extended repeat area overflow occurs.

- Acceptance of a transfer request can be reported to an external device via the $\overline{\text{EDRAK}}$
  (only for the EXDMAC).

- Operation of EXDMAC, connected to a dedicated bus, in parallel with a bus master su
  CPU, DTC, or DMAC (only for the EXDMAC).

- Module stop state can be set.



[Legend]

EDSAR_n:	EXDMA source address register
EDDAR_n:	EXDMA destination address register
EDOFR_n:	EXDMA offset register
EDTCR_n:	EXDMA transfer count register
EDBSR_n:	EXDMA block size register
EDMDR_n:	EXDMA mode control register
EDACR_n:	EXDMA address control register
CLSBR0 to CLSBR7:	Cluster buffer registers 0 to 7

$\overline{\text{EDREQm}}$: EXDMA transfer reques
$\overline{\text{EDACKm}}$: EXDMA transfer acknow
$\overline{\text{ETENDm}}$: EXDMA transfer end
$\overline{\text{EDRAKm}}$: $\overline{\text{EDREQ}}$ acceptance ack
(n: 0 to 3)
(m:0 or 1)

Figure 11.1 Block Diagram of EXDMAC

	$\overline{\text{EDREQ0}}$ acceptance acknowledge	$\overline{\text{EDRAK0}}$	Output	Notification to external de channel 0 external reque acceptance and start of e
1	EXDMA transfer request 1	$\overline{\text{EDREQ1}}$	Input	Channel 1 external reque
	EXDMA transfer acknowledge 1	$\overline{\text{EDACK1}}$	Output	Channel 1 single address acknowledge
	EXDMA transfer end 1	$\overline{\text{ETEND1}}$	Output	Channel 1 transfer end
	$\overline{\text{EDREQ1}}$ acceptance acknowledge	$\overline{\text{EDRAK1}}$	Output	Notification to external de channel 1 external reque acceptance and start of e

- EXDMA block size register_0 (EDBSR_0)
- EXDMA mode control register_0 (EDMDR_0)
- EXDMA address control register_0 (EDACR_0)

Channel 1

- EXDMA source address register_1 (EDSAR_1)
- EXDMA destination address register_1 (EDDAR_1)
- EXDMA offset register_1 (EDOFR_1)
- EXDMA transfer count register_1 (EDTCR_1)
- EXDMA block size register_1 (EDBSR_1)
- EXDMA mode control register_1 (EDMDR_1)
- EXDMA address control register_1 (EDACR_1)

Channel 2

- EXDMA source address register_2 (EDSAR_2)
- EXDMA destination address register_2 (EDDAR_2)
- EXDMA offset register_2 (EDOFR_2)
- EXDMA transfer count register_2 (EDTCR_2)
- EXDMA block size register_2 (EDBSR_2)
- EXDMA mode control register_2 (EDMDR_2)
- EXDMA address control register_2 (EDACR_2)

Common register

- Cluster buffer registers 0 to 7 (CLSBR0 to CLSBR7)

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	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							
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

EDTCR can be read at all times by the CPU. When reading EDTCR for a channel on whi
EXDMA transfer processing is in progress, a longword-size read must be executed. Do n
to EDTCR for a channel on which EXDMA transfer is in progress.

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

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	BI
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	E
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	E
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	All 0	R/W	Sets the repeat size, block size, or cluster siz When these bits are set to H'0001, one byte-, word-, or one longword-size is set. When thes set to H'0000, the maximum values are set (s 11.2). These bits are always fixed during an E operation.
15 to 0	BKSZ15 to BKSZ0	All 0	R/W	In an EXDMA operation, the remaining repeat block size, or cluster size is indicated. The va decremented by one each time of a data trans the remaining size becomes zero, the BKSZH loaded. Set the same initial value as for the B when writing.

11.3.6 EXDMA Mode Control Register (EDMDR)

EDMDR controls EXDMAC operations.

- EDMDR_0

Bit	31	30	29	28	27	26	25	
Bit Name	DTE	EDACKE	ETENDE	EDRAKE	EDREQS	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	D
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R/(W)*	R	R/(W)*	R/
Bit	15	14	13	12	11	10	9	
Bit Name	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	D
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	F
Bit	7	6	5	4	3	2	1	
Bit Name	DTF1	DTF0	—	—	—	EDMAP2	EDMAP1	ED
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R	R	R/W	R/W	F

Note: * Only 0 can be written to this bit after having been read as 1, 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	R/W

Bit	7	6	5	4	3	2	1
Bit Name	DTF1	DTF0	—	—	—	EDMAP2	EDMAP1
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.

operation, transfer is halted.

If this bit is cleared to 0 during an EXDMA ope
block transfer mode, this bit is cleared to 0 on
completion of the currently executing one-block
transfer. When this bit is cleared to 0 during an
operation in cluster transfer mode, this bit is cle
0 on completion of the currently executing one-
transfer.

If an external source that ends (aborts) transfe
this bit is automatically cleared to 0 and transfe
terminated.

Do not change the operating mode, transfer mo
other parameters while this bit is set to 1.

0: Data transfer disabled

1: Data transfer enabled (during an EXDMA op

[Clearing conditions]

- When transfer of the total transfer size spec ends
- When operation is halted by a repeat size e interrupt
- When operation is halted by an extended re area overflow interrupt
- When operation is halted by a transfer size interrupt
- When 0 is written to terminate transfer

  In block transfer mode, the value written is after one-block transfer ends.

  In cluster transfer mode, the value written is effective after one-cluster transfer ends.
- When an address error or NMI interrupt occ
- Reset, hardware standby mode

				Enables or disables output from the $\overline{\text{ETEND}}$ p This bit should be set to 0 for EDMDR_2 or E 0: $\overline{\text{ETEND}}$ pin output disabled 1: $\overline{\text{ETEND}}$ pin output enabled
28	EDRAKE	0	R/W	$\overline{\text{EDRAK}}$ Pin Output Enable Enables or disables output from the $\overline{\text{EDRAK}}$ p This bit should be set to 0 for EDMDR_2 or E 0: $\overline{\text{EDRAK}}$ pin output disabled 1: $\overline{\text{EDRAK}}$ pin output enabled
27	EDREQS	0	R/W	$\overline{\text{EDREQ}}$ Select Selects whether a low level or the falling edge $\overline{\text{EDREQ}}$ signal used in external request mode detected. This bit should be set to 0 for EDMDR_2 or E 0: Low-level detection 1: Falling edge detection (the first transfer is c on a low level after a transfer is enabled.)
26	NRD	0	R/W	Next Request Delay Selects the timing of the next transfer request accepted. 0: Next transfer request starts to be accepted transfer of the bus cycle in progress ends. 1: Next transfer request starts to be accepted cycle of Bϕ from the completion of the bus progress.
25, 24	—	All 0	R	Reserved They are always read as 0 and cannot be mo

19	ERRF	0	R/(W)*	System Error Flag Flag that indicates the occurrence of an addres or NMI interrupt. This bit is only enabled in EDI When this bit is set to 1, write to the DTE bit fo channels is disabled. This bit is reserved in ED to EDMDR_3. They are always read as 0 and c be modified. 0: Address error or NMI interrupt is not genera 1: Address error or NMI interrupt is generated [Clearing condition] • Writing 0 to ERRF after reading ERRF = 1 [Setting condition] • When an address error or NMI interrupt occ However, when an address error or an NMI int has been generated in EXDMAC module stop this bit is not set to 1.
18	—	0	R	Reserved They are always read as 0 and cannot be mod

				• Writing 1 to the DTE bit • Writing 0 to ESIF while reading ESIF = 1 [Setting conditions] • Transfer size error interrupt request is ger • Repeat size end interrupt request is gener • Extended repeat area overflow end interru is generated
16	DTIF	0	R/(W)*	Data Transfer Interrupt Flag Flag indicating that a transfer end interrupt re occurred by the transfer counter. 0: Transfer end interrupt request is not genera the transfer counter 1: Transfer end interrupt request is generated transfer counter [Clearing conditions] • Writing 1 to the DTE bit • Writing 0 to DTIF while reading DTIF = 1 [Setting condition] • When EDTCR becomes 0 and transfer ha
15	DTSZ1	0	R/W	Data Access Size 1 and 0
14	DTSZ0	0	R/W	Selects the data access size. 00: Byte-size (8 bits) 01: Word-size (16 bits) 10: Longword-size (32 bits) 11: Setting prohibited

				request. When this bit is set to 1 and the transfer counte becomes smaller than the data access size for transfer request by EXDMAC transfer, the DTE cleared to 0 by the next transfer request. At the time, the ESIF bit is set to 1 to indicate that a tr size error interrupt request is generated. When cluster transfer read/write address mode specified, this bit should be set to 1. Transfer size error interrupt request occurs in t following conditions: • In normal transfer and repeat transfer mode total transfer size set in EDTCR is smaller t data access size • In block transfer mode, the total transfer siz EDTCR is smaller than the block size • In cluster transfer mode, the total transfer s EDTCR is smaller than the cluster size 0: Transfer size error interrupt request disabled 1: Transfer size error interrupt request enabled
10	—	0	R	Reserved They are always read as 0 and cannot be mod

8	DTIE	0	R/W	Data Transfer Interrupt Enable Enables or disables a transfer end interrupt re the transfer counter. When this bit is set to 1 a DTIF bit is set to 1, a transfer end interrupt is to the CPU or DTC. The transfer end interrupt is canceled by clearing this bit or the DTIF bit 0: Transfer end interrupt request disabled 1: Transfer end interrupt request enabled
7 6	DTF1 DTF0	0 0	R/W R/W	Data Transfer Factor 1 and 0 Selects a source to activate EXDMAC. For ex requests, a sampling method is selected by th EDREQS bit. External requests should not be selected for E or EDMDR_3. 00: Auto-request (cycle steal mode) 01: Auto-request (burst mode) 10: Setting prohibited 11: External request
5	—	0	R/W	Reserved The initial value should not be changed.

independently for each channel. This bit is ena
when the CPUPCE bit in CPUPCR is 1.

000: Priority level 0 (lowest)

001: Priority level 1

010: Priority level 2

011: Priority level 3

100: Priority level 4

101: Priority level 5

110: Priority level 6

111: Priority level 7 (highest)

Note: * Only 0 can be written to these bits after 1 is read to clear the flag.

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 single address mode or dual address When single address mode is selected, $\overline{\text{EDAC}}$ valid due to the EDACKE bit setting in EDMD 0: Dual address mode 1: Single address mode
30	DIRS	0	R/W	Single Address Direction Select Specifies the data transfer direction in single a mode. In dual address mode, the specificatior bit is ignored. In cluster transfer mode, the internal cluster b be the source or destination in place of the ex device with $\overline{\text{DACK}}$. 0: EDSAR transferred as a source address 1: EDDAR transferred as a destination addres

				Even if the repeat area is not specified (ARS1, B'10), the repeat size end interrupt can be requ the end of a repeat-size transfer. When this bit is set to 1 and the next transfer s generated at the end of a block- or cluster-size in block transfer or cluster transfer mode, the D EDMDR is cleared to 0. At the same time, the in EDMDR is set to 1 to indicate that the repea end interrupt is requested. 0: Repeat size end interrupt request disabled 1: Repeat size end interrupt request enabled
25	ARS1	0	R/W	Area Select 1 and 0
24	ARS0	0	R/W	Select the block area or repeat area in block tr repeat transfer or cluster transfer mode. 00: Block area/repeat area on the source addr 01: Block area/repeat area on the destination a side 10: Block area/repeat area not specified 11: Setting prohibited
23, 22	—	All 0	R	Reserved They are always read as 0 and cannot be mod

				11: Decremented (–1, –2, or –4 according to t access size)
19, 18	—	All 0	R	Reserved They are always read as 0 and cannot be mo
17	DAT1	0	R/W	Destination Address Update Mode 1 and 0
16	DAT0	0	R/W	These bits specify incrementing/decrementing transfer destination address (EDDAR). When transfer source is not specified in EDDAR in s address mode, the specification by these bits ignored. 00: Fixed 01: Offset added 10: Incremented (+1, +2, or +4 according to th access size) 11: Decremented (–1, –2, or –4 according to t access size)

				When used together with block transfer mode, interrupt is requested at the end of a block-size If the DTE bit is set to 1 in EDMDR for the char which transfer is terminated by an interrupt, tra be resumed from the state in which it ended. If a source address extended repeat area is no designated, the specification by this bit is ignor 0: Source address extended repeat area overfl interrupt request disabled 1: Source address extended repeat area overfl interrupt request enabled
14, 13	—	All 0	R	Reserved They are always read as 0 and cannot be mod
12 11 10 9 8	SARA4 SARA3 SARA2 SARA1 SARA0	0 0 0 0 0	R/W R/W R/W R/W R/W	Source Address Extended Repeat Area These bits specify the source address (EDSAR extended repeat area. The extended repeat are function updates the specified lower address b leaving the remaining upper address bits alway same. An extended repeat area size of 4 bytes Mbytes can be specified. The setting interval is power-of-two number of bytes. When extended repeat area overflow results fr incrementing or decrementing an address, the address is the start address of the extended re area in the case of address incrementing, or th address of the extended repeat area in the cas address decrementing. If SARIE bit is set to 1, an interrupt can be requ when an extended repeat area overflow occurs Table 11.3 shows the settings and ranges of th extended repeat area.

				When used together with block transfer mode, an interrupt is requested at the end of a block-siz… If DTE bit is set to 1 in EDMDR for the channe… which transfer is terminated by an interrupt, tr… be resumed from the state in which it ended. … destination address extended repeat area is n… designated, the specification by this bit is igno… 0: Destination address extended repeat area … interrupt request disabled 1: Destination address extended repeat area … interrupt request enabled
6, 5	—	All 0	R	Reserved They are always read as 0 and cannot be mo…
4	DARA4	0	R/W	Destination Address Extended Repeat Area
3	DARA3	0	R/W	These bits specify the destination address (E… extended repeat area.
2	DARA2	0	R/W	
1	DARA1	0	R/W	The extended repeat area function updates th… specified lower address bits, leaving the rema… upper address bits always the same. An exter… repeat area size of 4 bytes to 128 Mbytes can… specified. The setting interval is a power-of-tw… of bytes.
0	DARA0	0	R/W	When extended repeat area overflow results f… incrementing or decrementing an address, the… address is the start address of the extended r… area in the case of address incrementing, or t… address of the extended repeat area in the ca… address decrementing. If the DARIE bit is set to 1, an interrupt can be… requested when an extended repeat area ove… occurs. Table 11.3 shows the settings and ranges of t… extended repeat area.

00110	Lower 6 bit (64-byte area) designated as extended repeat a
00111	Lower 7 bit (128-byte area) designated as extended repeat
01000	Lower 8 bit (256-byte area) designated as extended repeat
01001	Lower 9 bit (512-byte area) designated as extended repeat
01010	Lower 10 bit (1-Kbyte area) designated as extended repeat
01011	Lower 11 bit (2-Kbyte area) designated as extended repeat
01100	Lower 12 bit (4-Kbyte area) designated as extended repeat
01101	Lower 13 bit (8-Kbyte area) designated as extended repeat
01110	Lower 14 bit (16-Kbyte area) designated as extended repe
01111	Lower 15 bit (32-Kbyte area) designated as extended repe
10000	Lower 16 bit (64-Kbyte area) designated as extended repe
10001	Lower 17 bit (128-Kbyte area) designated as extended rep
10010	Lower 18 bit (256-Kbyte area) designated as extended rep
10011	Lower 19 bit (512-Kbyte area) designated as extended rep
10100	Lower 20 bit (1-Mbyte area) designated as extended repea
10101	Lower 21 bit (2-Mbyte area) designated as extended repea
10110	Lower 22 bit (4-Mbyte area) designated as extended repea
10111	Lower 23 bit (8-Mbyte area) designated as extended repea
11000	Lower 24 bit (16-Mbyte area) designated as extended repe
11001	Lower 25 bit (32-Mbyte area) designated as extended repe
11010	Lower 26 bit (64-Mbyte area) designated as extended repe
11011	Lower 27 bit (128-Mbyte area) designated as extended rep
111XX	Setting prohibited

[Legend] X: Don't care

In cluster transfer mode, the same CLSBR is used for all channels. When the CPU write
to CLSBR conflicts with cluster transfer, the contents of transferred data are not guarant
cluster transfer read/write address mode is specified and if another channel is set for clus
transfer, the transferred data may be overwritten.

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

Mode	Transfer Mode	Activation Source	Common Function	Source	D
Dual address mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode (Repeat size/ block size = 1 to 65,536 bytes/ word/longword)	• Auto-request (activated by the CPU) • External request*	• Total transfer size: 1 to 4 Gbytes, or no specification • Offset addition • Extended repeat area function	EDSAR	ED
Single address mode*	• Direct data transfer to/from external devices using $\overline{\text{EDACK}}$ pin instead of source or destination address register • Above transfer mode can be specified in addition to address register setting • One transfer possible in one bus cycle (Transfer mode variations are the same as in dual address mode.)			EDSAR/ $\overline{\text{EDACK}}$	$\overline{\text{ED}}$ ED

Note * Only channel 0 or 1 can be selected.

When the activation source is an auto-request, cycle steal mode or burst mode can be sele

When the total transfer size is not specified (EDTCR = H'00000000), the transfer counter and the transfer count is not restricted, allowing continuous transfer.

Dual address mode	(activated by the CPU) • External request*	One access size (byte/word/longword) to 32 bytes • Total transfer size 1 to 4 Gbytes, or no specification • Offset addition • Extended repeat area function		the transfer source and written to the transfer destination	
Cluster transfer Read address mode (DIRS = 0)			EDSAR	Read from the transfer source	–
Cluster transfer Write address mode (DIRS = 1)			—	Written to the transfer destination	E

Note * Only channel 0 or 1 can be selected.

In cluster transfer mode, the specified cluster size is transferred in response to a single tr
request. The cluster size can be from one access size (byte, word, or longword) to 32 by
a cluster, a cluster-size transfer is performed in burst transfer mode. With a cluster-size a
cluster transfer mode (dual address mode), block transfer mode (dual address mode) is u

With auto-requests, cycle steal mode is set.

In a transfer operation, the data on the transfer source address is read in the first bus cycle written to the transfer destination address in the next bus cycle.

These consecutive read and write cycles are indivisible: another bus cycle (external acces another bus master, refresh cycle, or external bus release cycle) does not occur between th cycles.

ETEND pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. is output for two consecutive bus cycles. When an idle cycle is inserted before the bus cy ETEND signal is also output in the idle cycle. The EDACK signal is not output.

Figure 11.2 shows an example of the timing in dual address mode and figure 11.3 shows address mode operation.

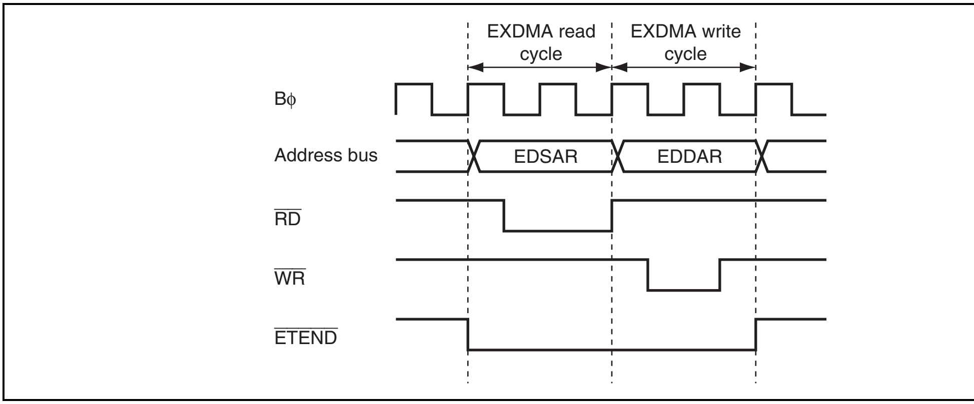


Figure 11.2 Example of Timing in Dual Address Mode

(2) Single Address Mode

In single address mode, the $\overline{\text{EDACK}}$ pin is used instead of EDSAR or EDDAR to transf
directly between an external device and external memory. One transfer operation is exec
one bus cycle.

Only channel 0 or 1 can be selected for single address mode. In this mode, the data bus w
be the same as the data access size. For details on the data bus width, see section 9, Bus
(BSC).

In this mode, the EXDMAC accesses the transfer source or transfer destination external
outputting the strobe signal ($\overline{\text{EDACK}}$) for the external device with $\overline{\text{DACK}}$, and at the sar
accesses the other external device in the transfer by outputting an address. In this way, E
transfer can be executed in one bus cycle. In the example of transfer between external m
an external device with $\overline{\text{DACK}}$ shown in figure 11.4, data is output to the data bus by the
device and written to external memory in the same bus cycle.

The transfer direction, that is whether the external device with $\overline{\text{DACK}}$ is the transfer sou
transfer destination, can be specified with the DIRS bit in EDACR. Transfer is performe
external memory (EDSAR) to the external device with $\overline{\text{DACK}}$ when DIRS = 0, and fron
external device with $\overline{\text{DACK}}$ to the external memory (EDDAR) when DIRS = 1. The sett
source or destination address register not used in the transfer is ignored.

The $\overline{\text{EDACK}}$ pin output is valid by the setting of EDACKE bit in EDMDR when single
mode is selected. The $\overline{\text{EDACK}}$ pin output is active-low.

$\overline{\text{ETEND}}$ pin output can be enabled or disabled by means of the ETENDE bit in EDMDR
is output for one bus cycle. When an idle cycle is inserted before the bus cycle, the $\overline{\text{ETE}}$
is also output in the idle cycle.

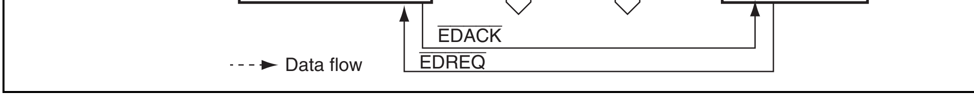


Figure 11.4 Data Flow in Single Address Mode

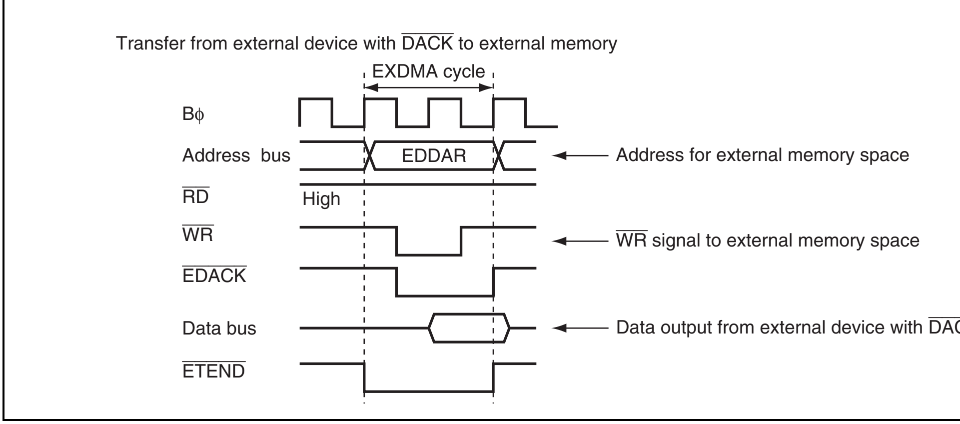


Figure 11.5 Example of Timing in Single Address Mode

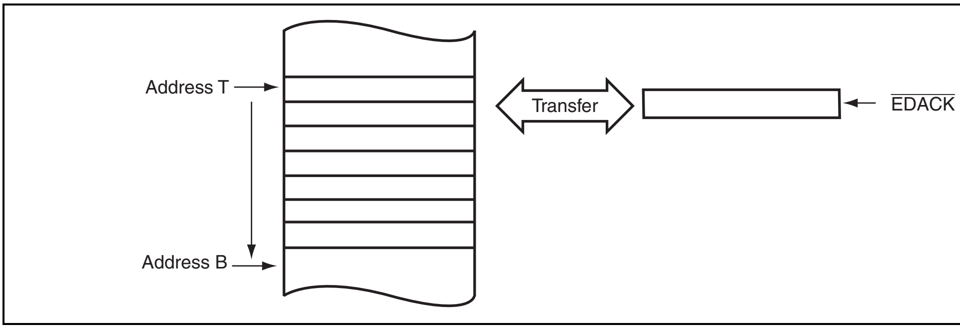


Figure 11.6 Single Address Mode Operation

Figure 11.7 shows examples of transfer timing in normal transfer mode and figure 11.8 sh
normal transfer mode operation in dual address mode.

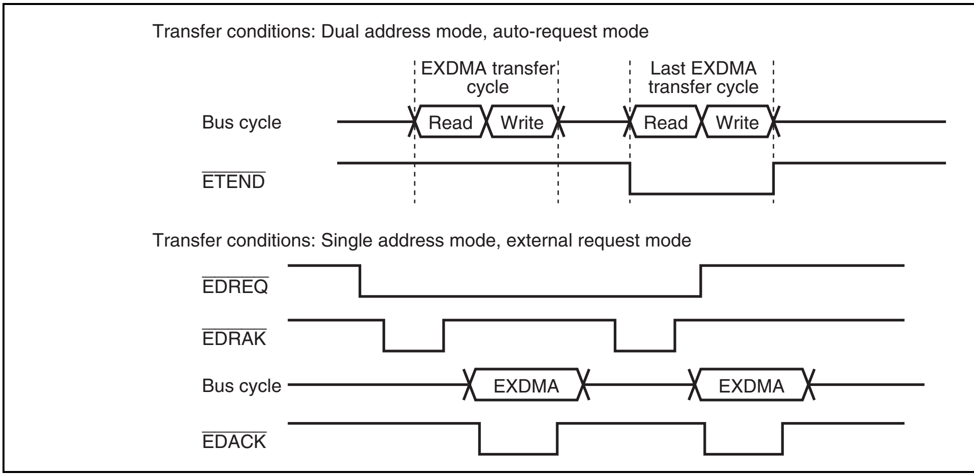


Figure 11.7 Examples of Timing in Normal Transfer Mode

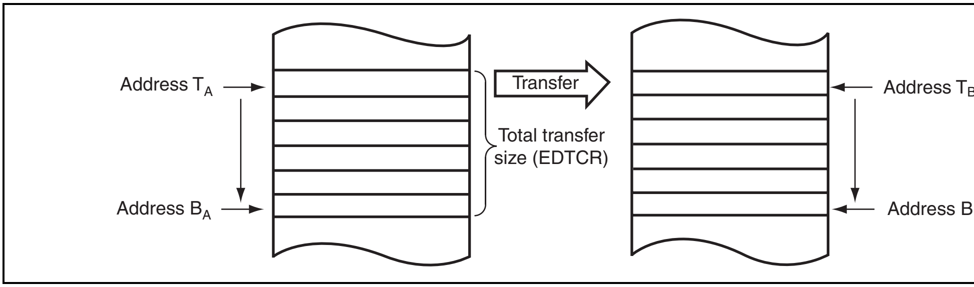


Figure 11.8 Normal Transfer Mode Operation

At the end of a repeat-size transfer, the EXDMA transfer is halted temporarily and a rep
end interrupt is requested to the CPU or DTC. When the RPTIE bit in EDACR is set to
next transfer request is generated at the end of a repeat-size transfer, the ESIF bit in EDM
to 1 and the DTE bit in EDMDR is cleared to 0 to terminate the transfer. At this time, ar
is requested to the CPU or DTC when the ESIE bit in EDMDR is set to 1.

The timing of EXDMA transfer including the $\overline{\text{ETEND}}$ or $\overline{\text{EDRAK}}$ output is the same as
normal transfer mode.

Figure 11.9 shows the repeat transfer mode operation in dual address mode.

The operation without specifying a repeat area on the source or destination address side
same as for the normal transfer mode operation shown in figure 11.8. In this case, a repe
interrupt can also be generated at the end of a repeat-size transfer.

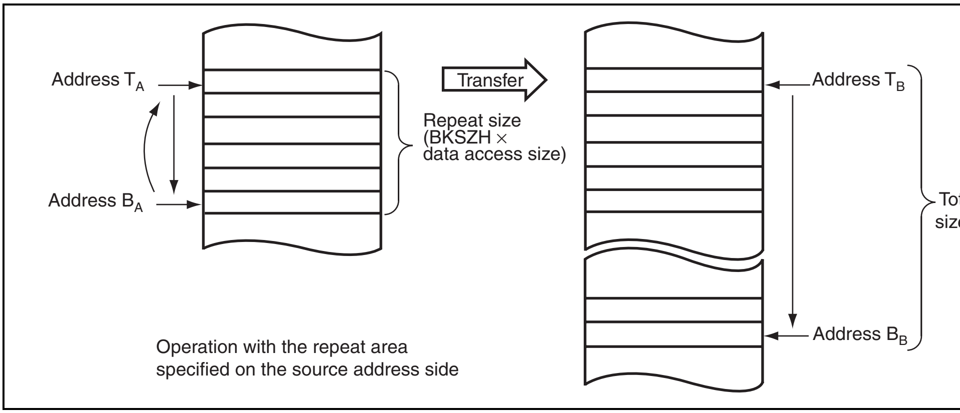


Figure 11.9 Repeat Transfer Mode Operation

time one-block transfer completes. When no repeat area is specified on the source and de
address sides, the address is not restored to the transfer start address and the operation pro
the next sequence. A repeat size end interrupt can be generated.

The $\overline{\text{ETEND}}$ signal is output for each block transfer in the EXDMA transfer cycle in whic
block ends. The $\overline{\text{EDRAK}}$ signal is output once for one transfer request (for transfer of one

Caution is required when setting the extended repeat area overflow interrupt in block tran
mode. For details, see section 11.5.5, Extended Repeat Area Function.

Figure 11.10 shows an example of EXDMA transfer timing in block transfer mode. The t
conditions are as follows:

Address mode: Single address mode

Data access size: In bytes

One block size: 3 bytes

Figure 11.10 Example of Block Transfer Mode

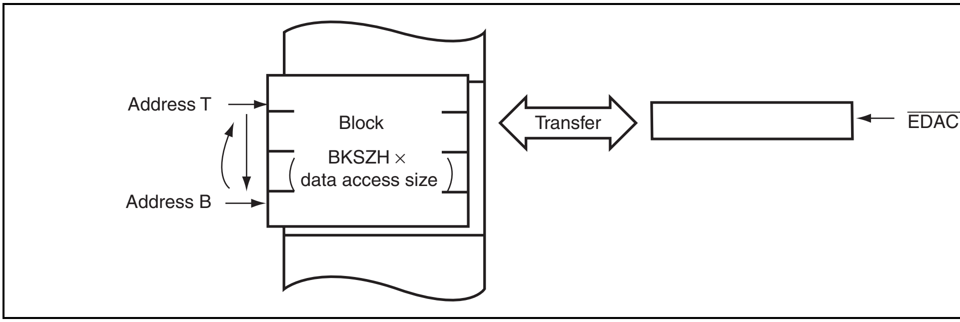


Figure 11.11 Block Transfer Mode Operation in Single Address Mode (with Block Area Specified)

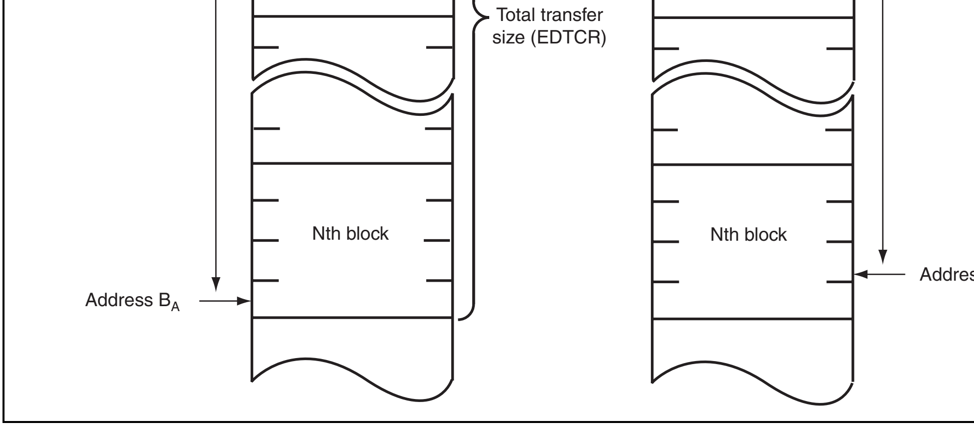


Figure 11.12 Block Transfer Mode Operation in Dual Address Mode (without Block Area Specified)

request activation. The bus mode can be selected from cycle steal mode and burst mode
request activation.

(2) Activation by External Request

Transfer is started by the transfer request signal ($\overline{\text{EDREQ}}$) from the external device for a
by an external request. When the EXDMA transfer is enabled (DTE = 1), the EXDMA t
starts by $\overline{\text{EDREQ}}$ input. Only channel 0 or 1 can be selected for activation by an externa

The transfer request signal is accepted by the $\overline{\text{EDREQ}}$ pin. The EDREQS bit in EDMDR
whether the $\overline{\text{EDREQ}}$ is detected by falling edge sensing or low level sensing.

When the EDRAKE bit in EDMDR is set to 1, the signal notifying transfer request acce
output from the $\overline{\text{EDRAK}}$ pin. The $\overline{\text{EDRAK}}$ signal is accepted for one external request an
output when transfer processing starts.

When specifying an external request as an activation source, set the DDR bit to 0 and th
to 1 on the corresponding pin in advance. For details, see section 13, I/O Ports.

transfer request, the EXDMAC takes back the bus mastership, performs another transfer-transfer, and then releases the bus mastership again at the end of the transfer. This proced repeated until the transfer end condition is satisfied.

If a transfer request occurs in another channel during EXDMA transfer, the bus mastershi temporarily released for another bus master, then transfer is performed on the channel for the transfer request was issued. For details on the operation when there are transfer reque number of channels, see section 11.5.8, Channel Priority Order.

Figure 11.13 shows an example of the timing in cycle steal mode. The transfer conditions follows:

- Address mode: Single address mode
- Sampling method on the $\overline{\text{EDREQ}}$ pin: Low level sensing
- CPU internal bus master is operating in external space

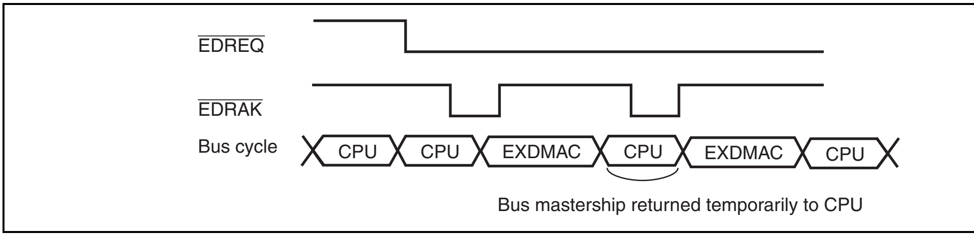


Figure 11.13 Example of Timing in Cycle Steal Mode

In block transfer mode and cluster transfer mode, the setting of burst mode is invalid (o
one-cluster transfer is processed in the same way as in burst mode). The EXDMAC alwa
operates in cycle steal mode.

When the DTE bit is cleared to 0 in EDMDR, EXDMA transfer is halted. However, EX
transfer is executed for all transfer requests generated within the EXDMAC until the DT
cleared to 0. If a transfer size error interrupt, a repeat size end interrupt, or extended rep
overflow interrupt is generated, the DTE bit is cleared to 0 and transfer is terminated.

Figure 11.14 shows an example of the timing in burst mode.

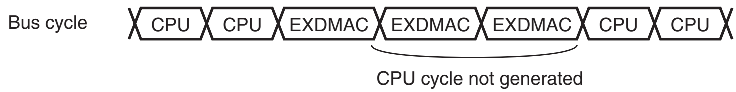


Figure 11.14 Example of Timing in Burst Mode

11.5.5 Extended Repeat Area Function

The EXDMAC has a function for designating an extended repeat area for source addres
destination addresses. When an extended repeat area is designated, the address register
repeat within the range specified as the extended repeat area. Normally, when a ring buf
involved in a transfer, an operation is required to restore the address register value to the
start address each time the address register value becomes the last address in the buffer (
ring buffer address overflow occurs). However, if the extended repeat area function is us
operation that restores the address register value to the buffer start address is processed
automatically within the EXDMAC.

The extended repeat area function can be set independently for the source address regist
(EDSAR) and the destination address register (EDDAR).

Figure 11.15 illustrates the operation of the extended repeat area function.

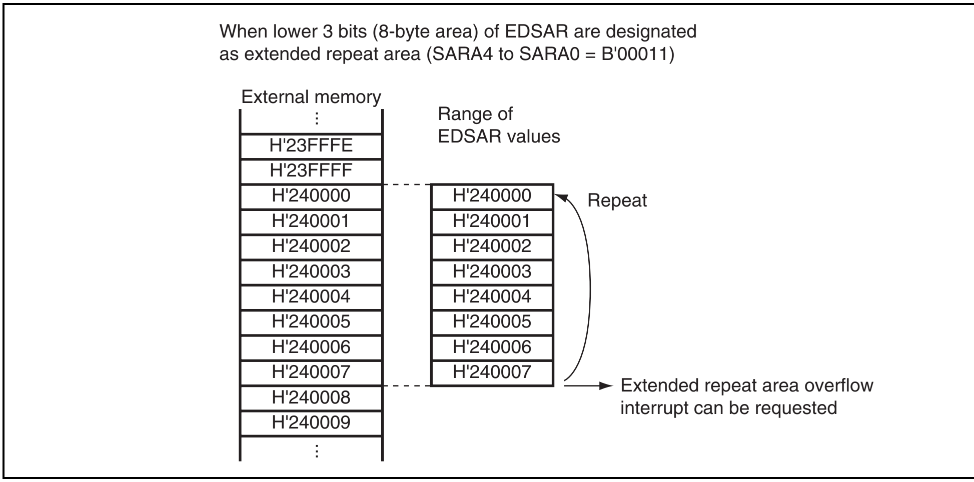


Figure 11.15 Example of Extended Repeat Area Function Operation

Figure 11.16 shows an example in which block transfer mode is used together with the e
repeat area function.

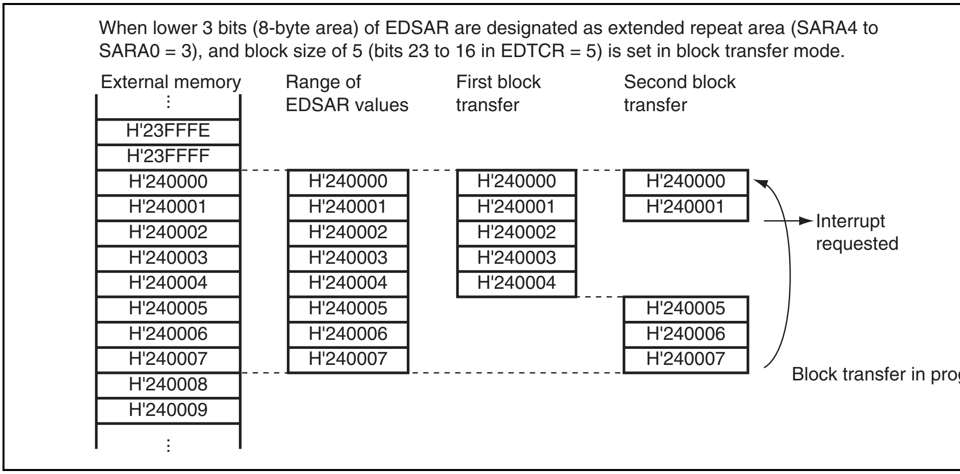


Figure 11.16 Example of Extended Repeat Area Function Operation in Block T Mode

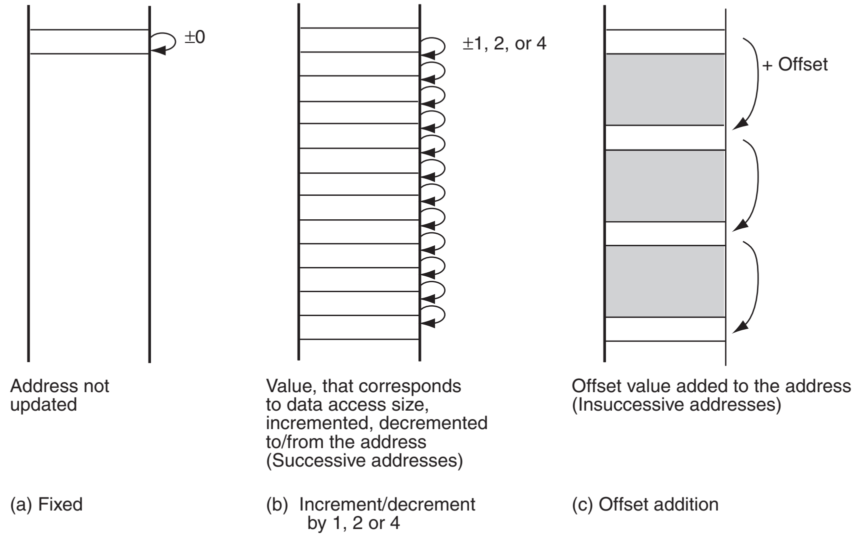


Figure 11.17 Address Update Method

For the fixed method (a), the same address is always indicated without the transfer destin source address being updated.

For the method of increment/decrement by 1, 2 or 4 (b), the value corresponding to the da size is incremented or decremented to or from the transfer destination or source address e the data is transferred. A byte, word, or longword can be specified for the data access size value used for increment or decrement of an address is 1 for a byte-size , 2 for a word-siz for a longword-size transfer. This function allows continuous address transfer of EXDMA

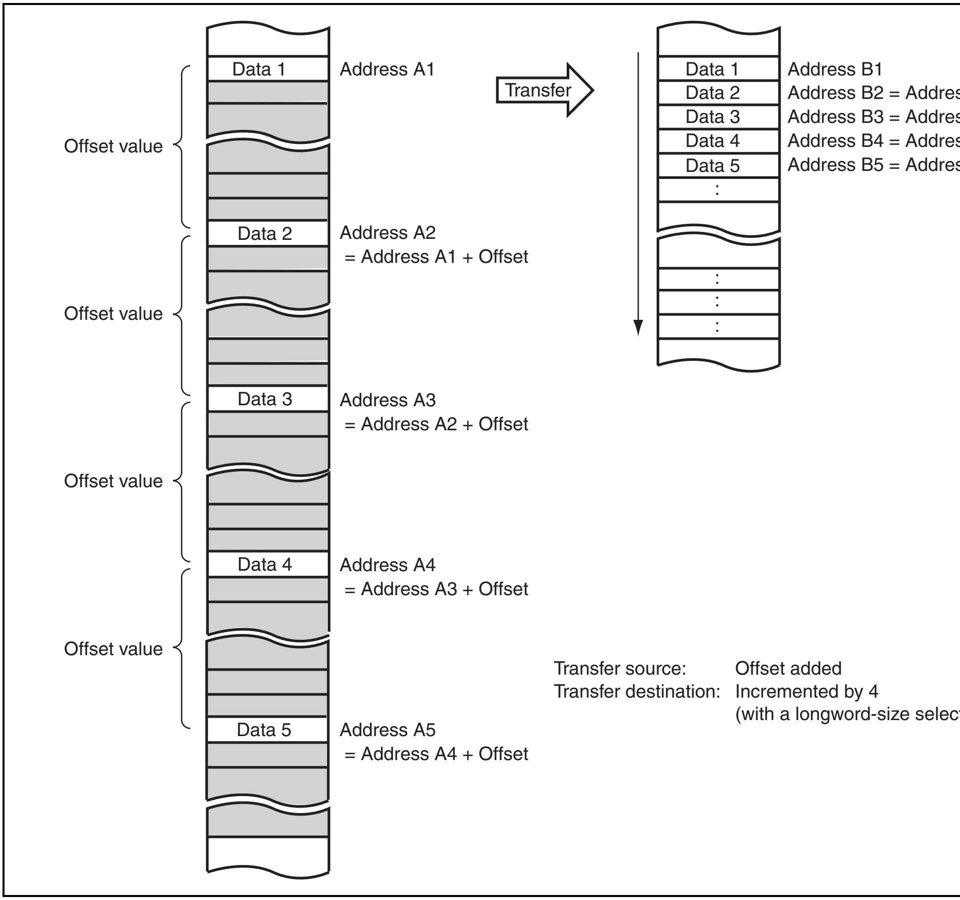


Figure 11.18 Address Update Function Using Offset

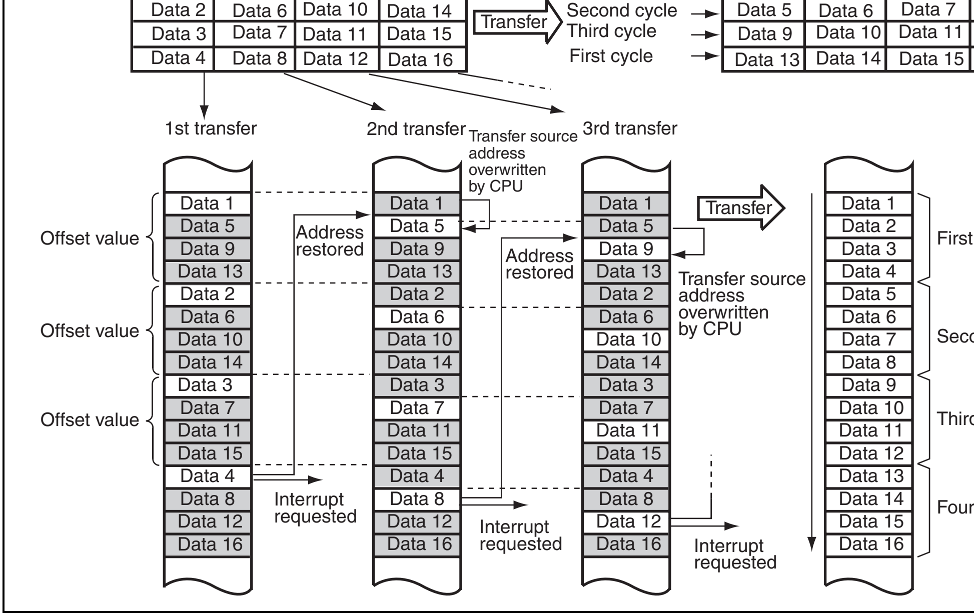


Figure 11.19 XY Conversion by Combining Repeat Transfer Mode and Offset Ac

In figure 11.19, the source address side is set as a repeat area in EDACR and the offset ac
set in EDACR. The offset value is the address that corresponds to 4 × data access size (ex
for a longword-size transfer, H'00000010 is specified in EDOFR). The repeat size is 4 × d
access size (example: for a longword-size transfer, 4 × 4 = 16 bytes are specified as a repe
The increment by 1, 2 or 4 is set for the transfer destination. The RPTIE bit in EDACR is
to generate a repeat size end interrupt request at the end of a repeat-size transfer.

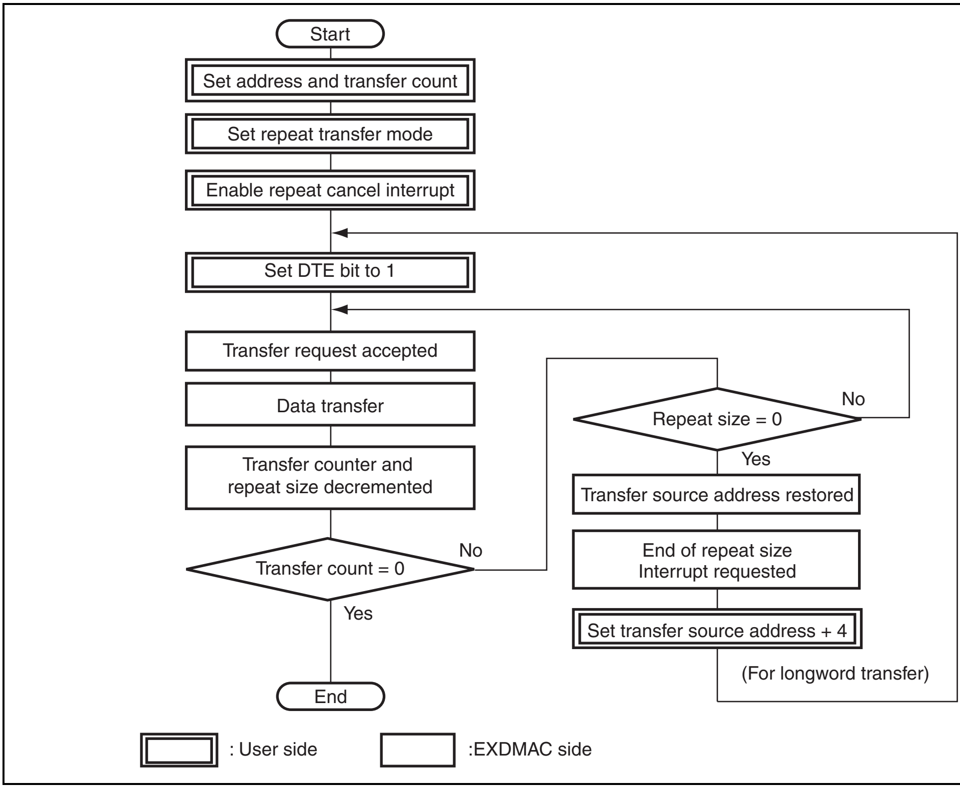


Figure 11.20 Flow of XY Conversion Combining Repeat Transfer Mode and (
Addition

A twos complement can be derived by the NEG.L instruction of the CPU.

11.5.7 Registers during EXDMA Transfer Operation

EXDMAC register values are updated as EXDMA transfer processing is performed. The values depend on various settings and the transfer status. The following registers and bits updated: EDSAR, EDDAR, EDTCR, bits BKSZH and BKSZ in EDBSR, and bits DTE, A ERRF, ESIF and DTIF in EDMDR.

(1) EXDMA Source Address Register (EDSAR)

When the EDSAR address is accessed as the transfer source, the EDSAR value is output, EDSAR is updated with the address to be accessed next.

Bits SAT1 and SAT0 in EDACR specify incrementing or decrementing. The address is fi when SAT1 and SAT0 = B′00, incremented by offset register value when SAT1 and SAT B′01, incremented when SAT1 and SAT0 = B′10, and decremented when SAT1 and SAT B′11. (The increment or decrement value is determined by the data access size.)

The DTSZ1 and DTSZ0 bits in EDMDR set the data access size. When DTSZ1 and DTS B′00, the data is byte-size and the address is incremented or decremented by 1. When DT DTSZ0 = B′01, the data is word-size and the address is incremented or decremented by 2 DTSZ1and DTSZ0 = B′10, the data is longword-size and the address is incremented or decremented by 4. When a word-size or longword-size is specified but the source address the word or longword boundary, the data is divided into bytes or words for reading. When or longword is divided for reading, the address is incremented or decremented by 1 or 2 a to an actual byte-or word-size read. After a word-size or longword-size read, the address incremented or decremented to or from the read start address according to the setting of S SAT0.

EDSAR for a channel on which a transfer operation is in progress.

(2) EXDMA Destination Address Register (EDDAR)

When the EDDAR address is accessed as the transfer destination, the EDDAR value is o
then EDDAR is updated with the address to be accessed next.

Bits DAT1 and DAT0 in EDACR specify incrementing or decrementing. The address is
when DAT1 and DAT0 = B′00, incremented by offset register value when DAT1 and D
B′01, incremented when DAT1 and DAT0 = B′10, and decremented when DAT1 and D
B′11. (The increment or decrement value is determined by the data access size.)

The DTSZ1 and DTSZ0 bits in EDMDR set the data access size. When DTSZ1 and DT
B′00, the data is byte-size and the address is incremented or decremented by 1. When D
DTSZ0 = B′01, the data is word-size and the address is incremented or decremented by
DTSZ1 and DTSZ0 = B′10, the data is longword-size and the address is incremented or
decremented by 4. When a word-size or longword-size is specified but the destination a
not at the word or longword boundary, the data is divided into bytes or words for writing
word or a longword is divided for writing, the address is incremented or decremented by
according to an actual byte- or word-size written. After a word-size or longword-size wr
address is incremented or decremented to or from the write start address according to th
SAT1 and SAT0.

When a block area (repeat area) is set for the destination address in block transfer mode
transfer mode), the destination address is restored to the transfer start address at the end
size (repeat-size) transfer and is not affected by address updating.

When an extended repeat area is set for the destination address, the operation conforms
setting. The upper addresses set for the extended repeat area is fixed, and is not affected
address updating.

EDTCR value does not change.

All of the bits of EDTCR may change, so when EDTCR is read by the CPU during EXDM
transfer, a longword access must be used. During a transfer operation, EDTCR may be up
without regard to accesses from the CPU, and the correct values may not be read if the up
lower words are read separately. Do not write to EDTCR for a channel on which a transfe
operation is in progress.

If there is conflict between an address update associated with EXDMA transfer and a wri
CPU, the CPU write has priority.

In the event of conflict between an EDTCR update from 1, 2, or 4 to 0 and a write (of a n
value) by the CPU, the CPU write value has priority as the EDTCR value, but transfer is
terminated.

(4) EXDMA Block Size Register (EDBSR)

EDBSR is valid in block transfer or repeat transfer mode. EDBSR31 and EDBSR16 are u
BKSZH and EDBSR15 and EDBSR0 for BKSZ. The 16 bits of BKSZH holds a block si
repeat size and their values do not change. The 16 bits of BKSZ functions as a block size
size counter, the value of which is decremented by 1 when one data transfer is performed
the BKSZ value is determined as 0 during EXDMA transfer, the EXDMAC does not stor
BKSZ and stores the BKSZH value.

The upper 16 bits of EDBSR is never updated, allowing a word-size access.

Do not write to EDBSR for a channel on which a transfer operation is in progress.

- When an extended repeat area overflow interrupt is requested, and transfer ends
- When an NMI interrupt is generated, and transfer halts
- When an address error is generated, and transfer halts
- A reset
- Hardware standby mode
- When 0 is written to the DTE bit, and transfer halts

Writes (except to the DTE bit) are prohibited to registers of a channel for which the DT
to 1. When changing register settings after a 0-write to the DTE bit, it is necessary to co
the DTE bit has been cleared to 0.

Figure 11.21 shows the procedure for changing register settings in an operating channel.

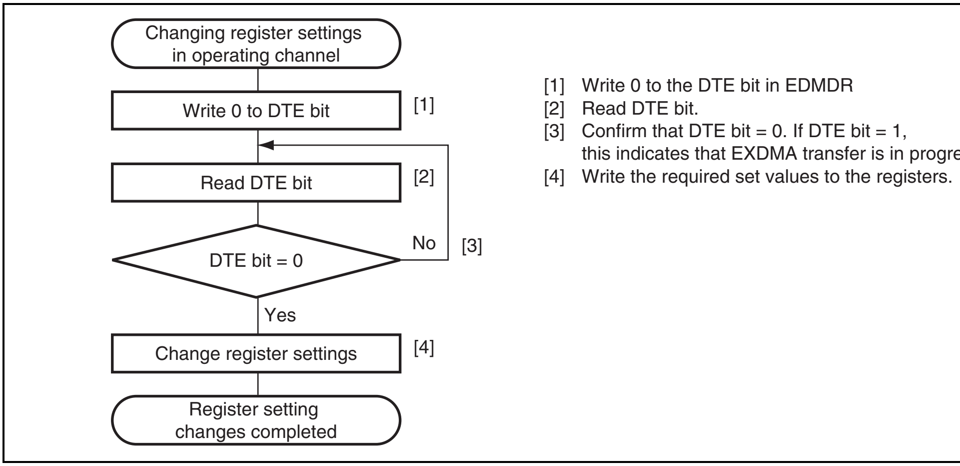


Figure 11.21 Procedure for Changing Register Settings in Operating Chan

the bus cycle in which 0 is written to the DTE bit has been processed. The ACT bit is held between termination of the last EXDMA cycle and 0-write in the DTE bit.

(7) ERRF bit in EDMDR

This bit specifies termination of transfer by EXDMAC clearing the DTE bit to 0 for all ch an address error or NMI interrupt is generated. The EXDMAC also sets 1 to the ERRF bi EDMDR_0 regardless of the EXDMAC operation to indicate that an address error or NM interrupt is generated. However, when an address error or an NMI interrupt has been gene EXDMAC module stop mode, the ERRF bit is not set to 1.

(8) ESIF bit in EDMDR

The ESIF bit in EDMDR is set to 1 when a transfer size interrupt, repeat size end interrup extended repeat area overflow interrupt is requested. When the ESIF bit is set to 1 and the bit in EDMDR is set to 1, a transfer escape interrupt is requested to the CPU or DTC.

The timing that the ESIF bit is set to 1 is when the EXDMA transfer bus cycle (the source interrupt request) terminates, the ACT bit in EDMDR is set to 0, and transfer is terminate

When the DTE bit is set to 1 to resume transfer during interrupt processing, the ESIF bit i automatically cleared to 0 to cancel the interrupt request.

For details on interrupts, see section 11.9, Interrupt Sources.

For details on interrupts, see section 11.9, Interrupt Sources.

11.5.8 Channel Priority Order

The priority order of the EXDMAC channels is: channel 0 > channel 1 > channel 2 > ch

Table 11.6 shows the EXDMAC channel priority order.

Table 11.6 EXDMAC Channel Priority Order

Channel	Channel
Channel 0	High
Channel 1	↑
Channel 2	
Channel 3	Low

If transfer requests occur simultaneously for a number of channels, the highest-priority c
according to the priority order is selected for transfer. Transfer starts after the channel in
releases the bus. If a bus request is issued from another bus master other than EXDMAC
transfer operation, another bus master cycle is initiated.

Channels are not switched during burst transfer, a block-size transfer in block transfer m
cluster-size transfer in cluster transfer mode.

Figure 11.22 shows an example of the transfer timing when transfer requests occur simu
for channels 0, 1, and 2.

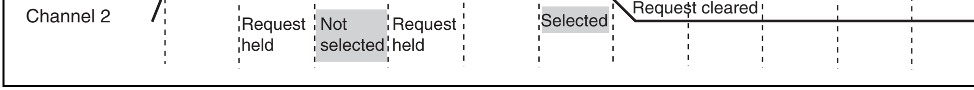


Figure 11.22 Example of Channel Priority Timing

11.5.9 Basic Bus Cycles

An example of the basic bus cycle timing is shown in figure 11.23. In this example, word
transfer is performed from 16-bit, 2-state access space to 8-bit, 3-state access space. Whe
mastership is transferred from the CPU to the EXDMAC, a source address read and desti
address write are performed. The bus is not released in response to another bus request, e
between these read and write operations. As like CPU cycles, EXDMAC cycles conform
bus controller settings.

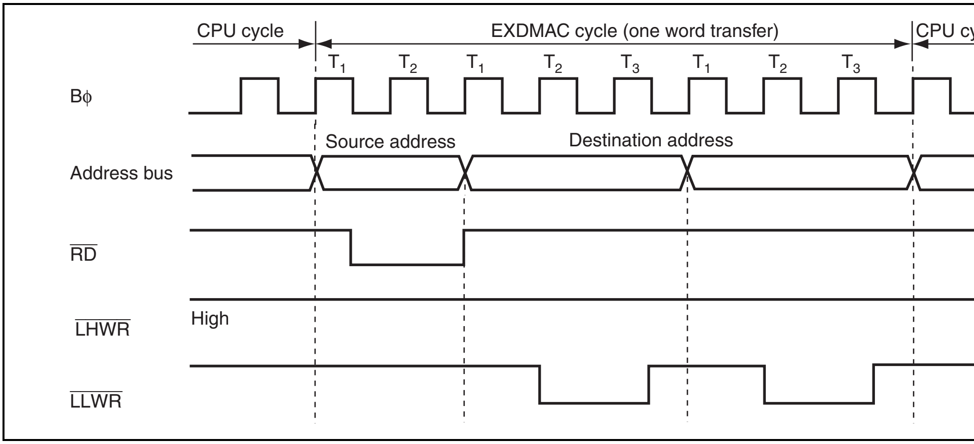


Figure 11.23 Example of EXDMA Transfer Bus Timing

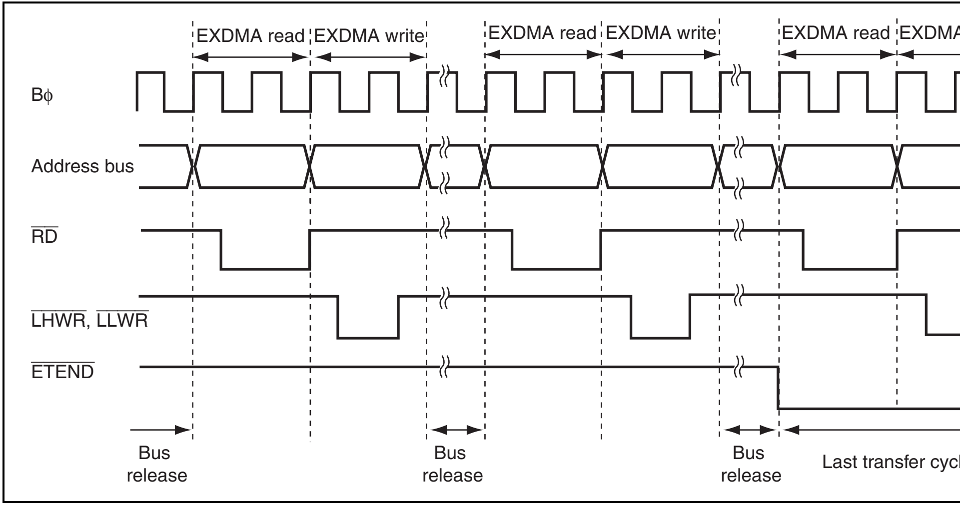


Figure 11.24 Example of Normal Transfer Mode (Cycle Steal Mode) Trans

Figures 11.25 and 11.26 show examples of transfer when $\overline{\text{ETEND}}$ output is enabled, anc longword-size, normal transfer mode (cycle steal mode) is performed from external 16-b access space to external 16-bit, 2-state access space.

In figure 11.25, the transfer source (SAR) address is not at a longword boundary and the destination (DAR) address is at the longword boundary.

In figure 11.26, the transfer source (SAR) address is at the longword boundary and the t destination (DAR) address is not at the longword boundary.

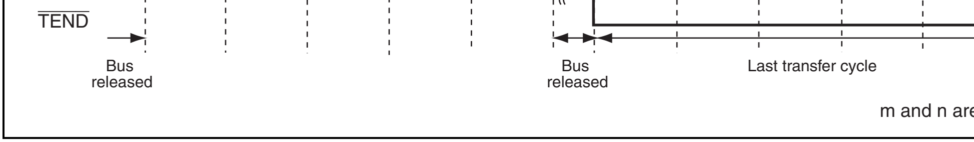


Figure 11.25 Example of Normal Transfer Mode (Cycle Steal Mode) Transf
(Transfer Source EDSAR = Odd Address, Source Address Incremented)

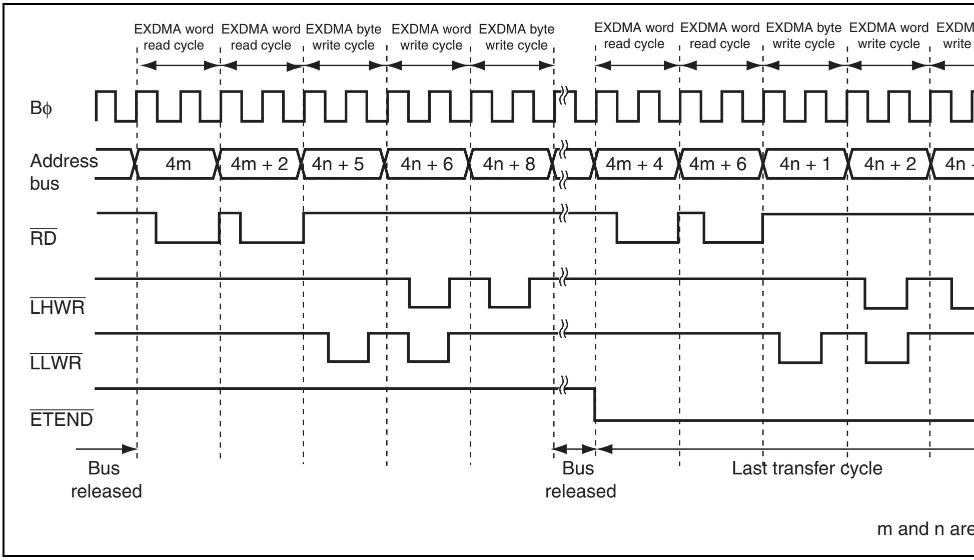


Figure 11.26 Example of Normal Transfer Mode (Cycle Steal Mode) Transf
(Transfer Destination EDDAR = Odd Address, Destination Address Decremen

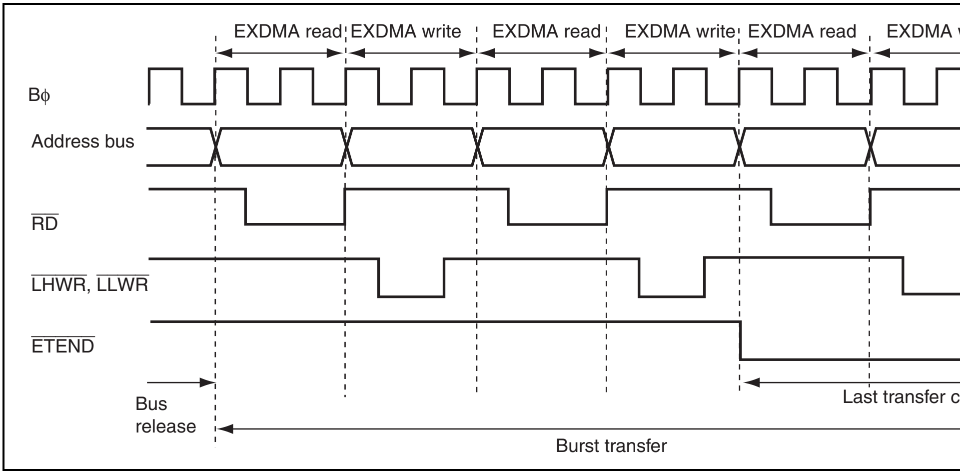


Figure 11.27 Example of Normal Transfer Mode (Burst Mode) Transfe

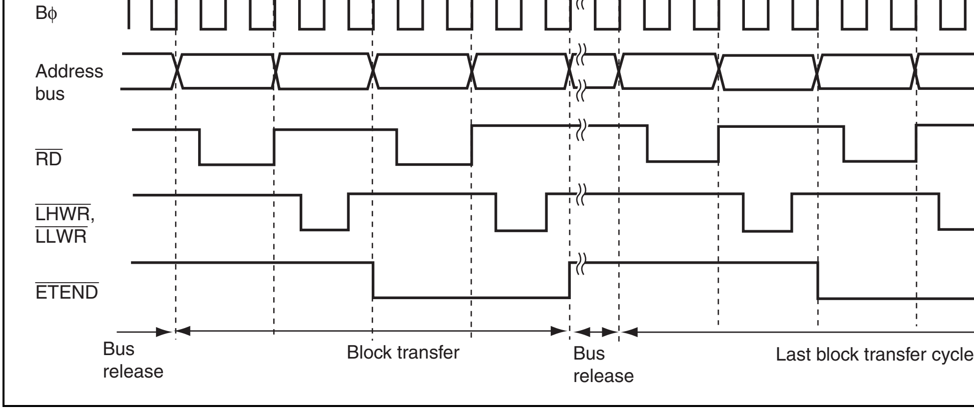
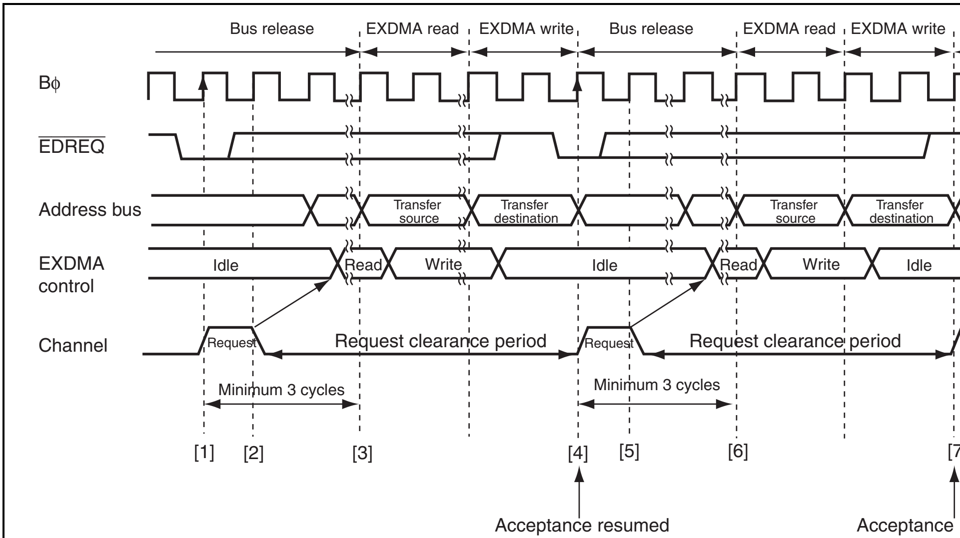


Figure 11.28 Example of Block Transfer Mode Transfer

is started. If $\overline{\text{EDREQ}}$ pin high level sampling is completed by the end of the EXDMA w
acceptance resumes after the end of the write cycle, and $\overline{\text{EDREQ}}$ pin low level sampling
performed again. This sequence of operations is repeated until the end of the transfer.



[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of Bφ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] EXDMA cycle starts; $\overline{\text{EDREQ}}$ pin high level sampling is started at rise of Bφ.
[4], [7] When $\overline{\text{EDREQ}}$ pin high level has been sampled, acceptance is resumed after completion of write cycle.
(As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of Bφ, and request is held.)

Figure 11.29 Example of Normal Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Falling Edge

performed again. This sequence of operations is repeated until the end of the transfer.

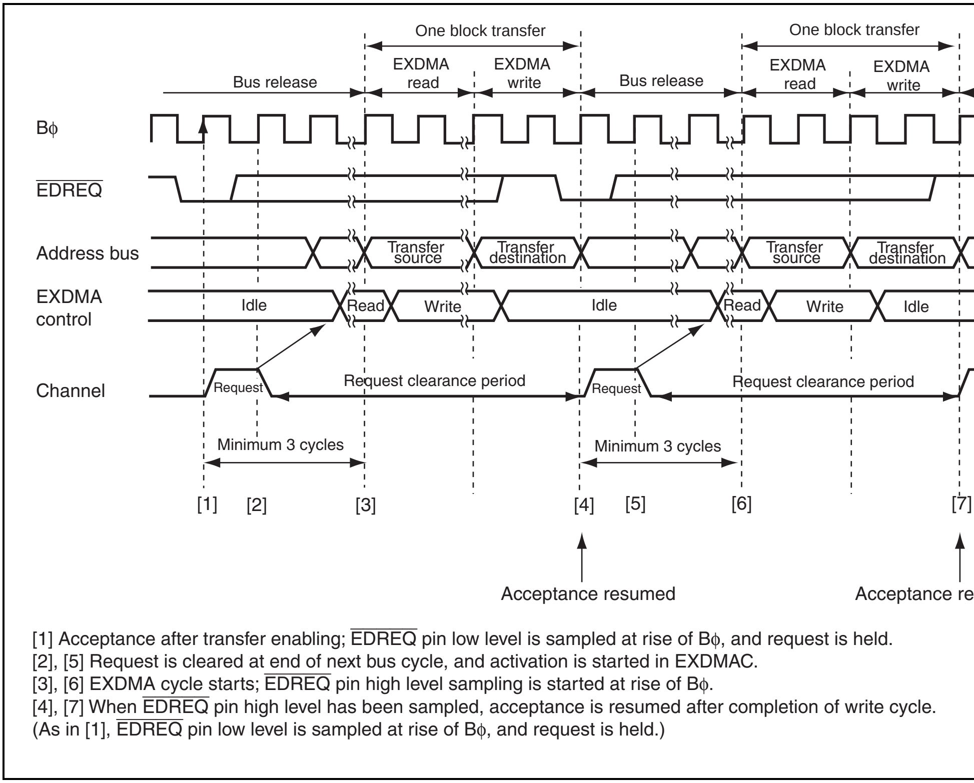
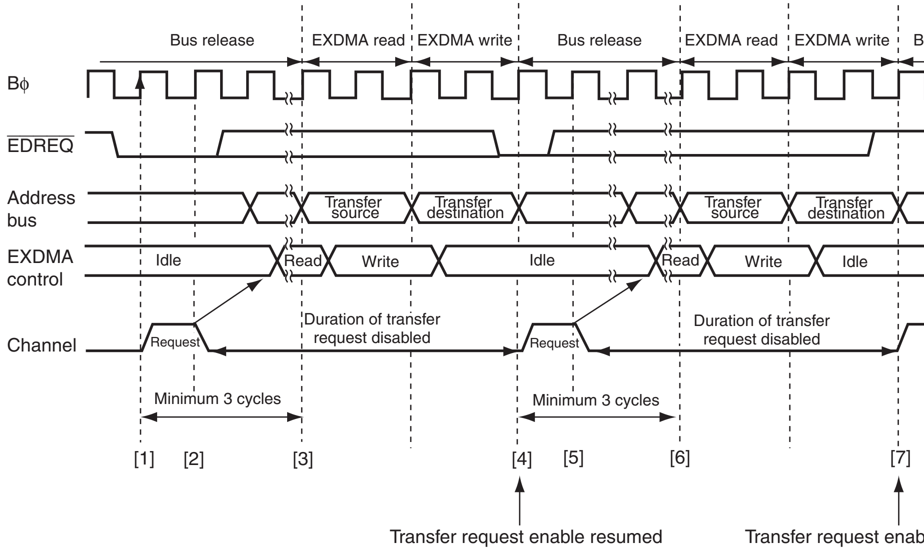


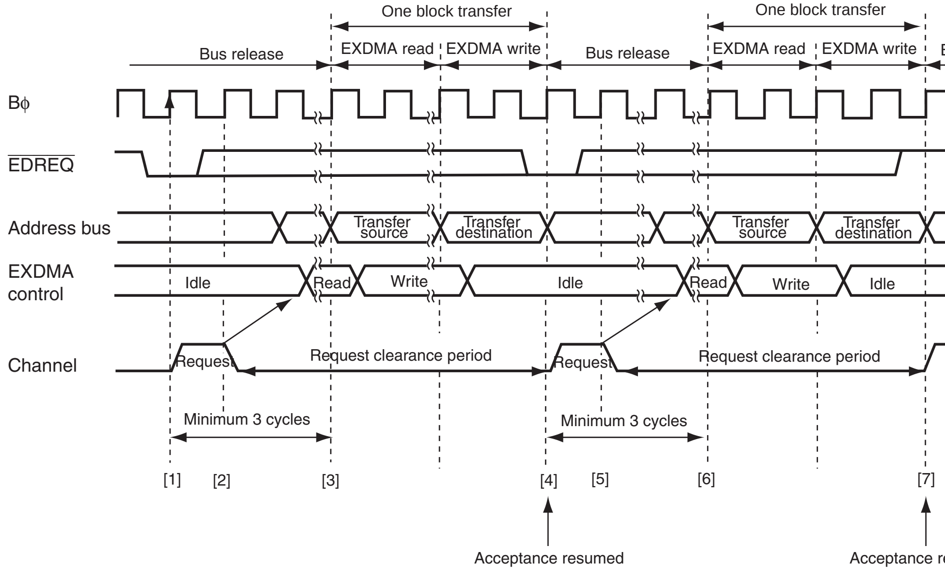
Figure 11.30 Example of Block Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Falling Edge

and EDREQ pin low-level sampling is performed again. This sequence of operations is r
until the end of the transfer.



[1] Acceptance after transfer enabling; $\overline{EDREQ}$ pin low level is sampled at rise of Bφ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] EXDMA cycle starts.
[4], [7] Acceptance is resumed after completion of write cycle.
(As in [1], $\overline{EDREQ}$ pin low level is sampled at rise of Bφ, and request is held.)

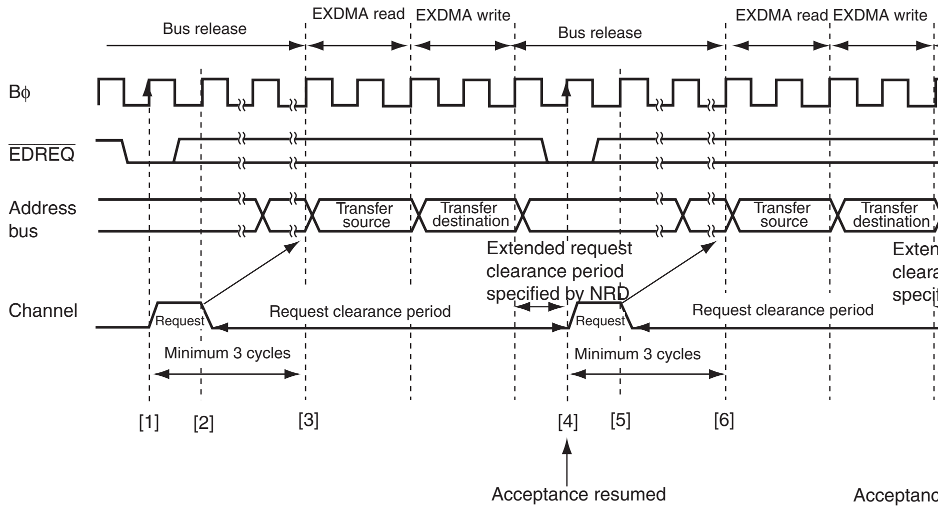
Figure 11.31 Example of Normal Transfer Mode Transfer Activated by $\overline{EDREQ}$ Pin Low Level



[1] Acceptance after transfer enabling; $\overline{EDREQ}$ pin low level is sampled at rise of Bφ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] EXDMA cycle starts.
[4], [7] Acceptance is resumed after completion of write cycle.
(As in [1], $\overline{EDREQ}$ pin low level is sampled at rise of Bφ, and request is held.)

Figure 11.32 Example of Block Transfer Mode Transfer Activated by $\overline{EDREQ}$ Pin Low Level

EDREQ pin is possible, the request is held within the EXDMAC. Then when activation within the EXDMAC, the request is cleared. After the end of the write cycle, acceptance when one cycle of the request clearance period specified by NRD = 1 expires and EDRE level sampling is performed again. This sequence of operations is repeated until the end transfer.



[1] Acceptance after transfer enabling; EDREQ pin low level is sampled at rise of Bϕ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] EXDMA cycle starts.
[4], [7] Acceptance is resumed after completion of write cycle plus one cycle.
(As in [1], EDREQ pin low level is sampled at rise of Bϕ, and request is held.)

Figure 11.33 Example of Normal Transfer Mode Transfer Activated by EDREQ Pin Low Level with NRD = 1 Specified

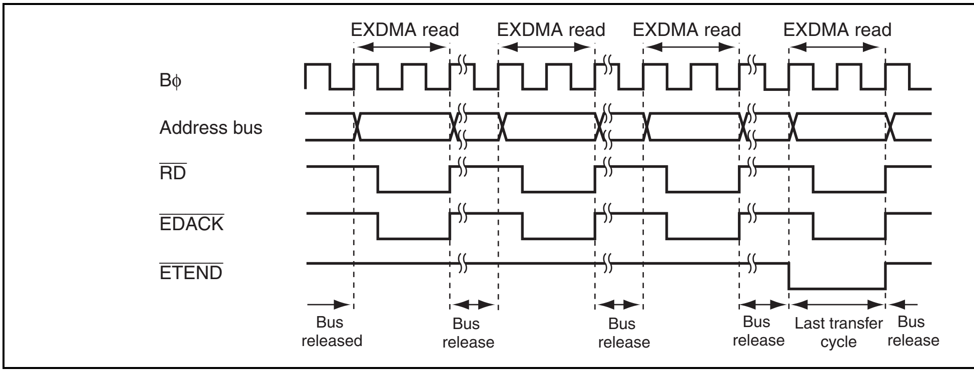


Figure 11.34 Example of Single Address Mode (Byte Read) Transfer

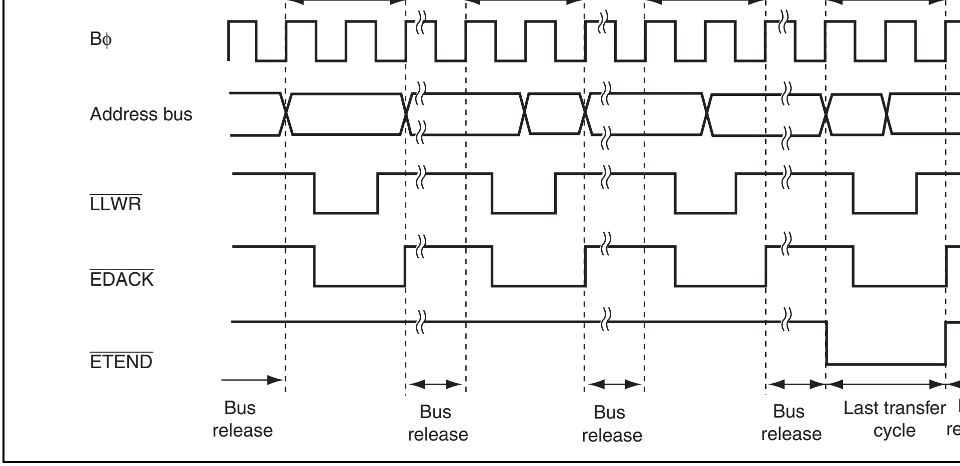
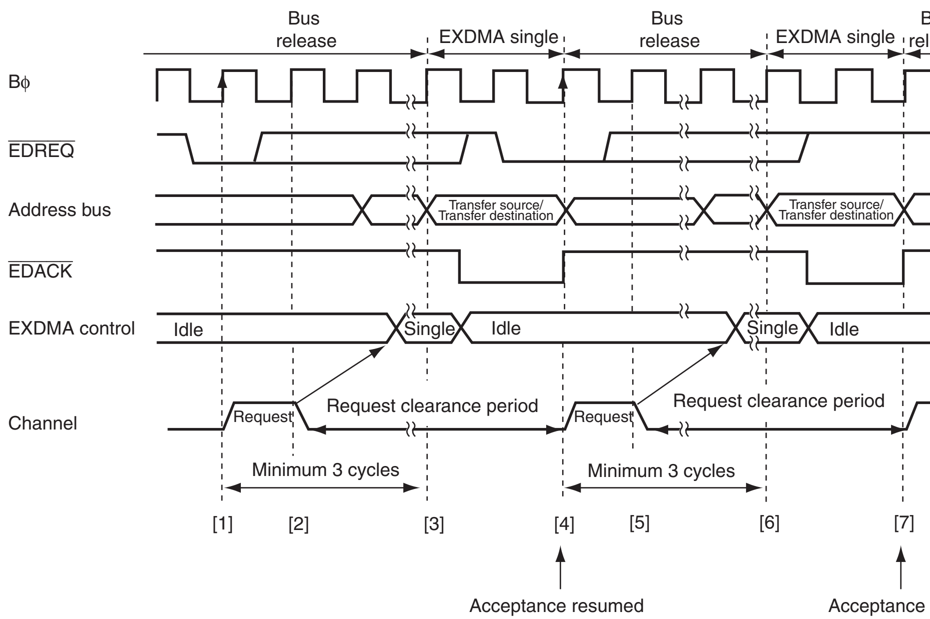


Figure 11.35 Example of Single Address Mode (Byte Write) Transfer

is started. If EDREQ pin high level sampling is completed by the end of the EXDMA single cycle, acceptance resumes after the end of the single cycle, and $\overline{EDREQ}$ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.



[1] Acceptance after transfer enabling; $\overline{EDREQ}$ pin low level is sampled at rise of Bφ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] EXDMA cycle starts; $\overline{EDREQ}$ pin high level sampling is started at rise of Bφ.
[4], [7] When $\overline{EDREQ}$ pin high level has been sampled, acceptance is resumed after completion of write cycle.
(As in [1], $\overline{EDREQ}$ pin low level is sampled at rise of Bφ, and request is held.)

Figure 11.36 Example of Single Address Mode Transfer Activated by $\overline{EDREQ}$ Pin Falling Edge

and $\overline{EDREQ}$ pin low-level sampling is performed again. This sequence of operations is r
until the end of the transfer.

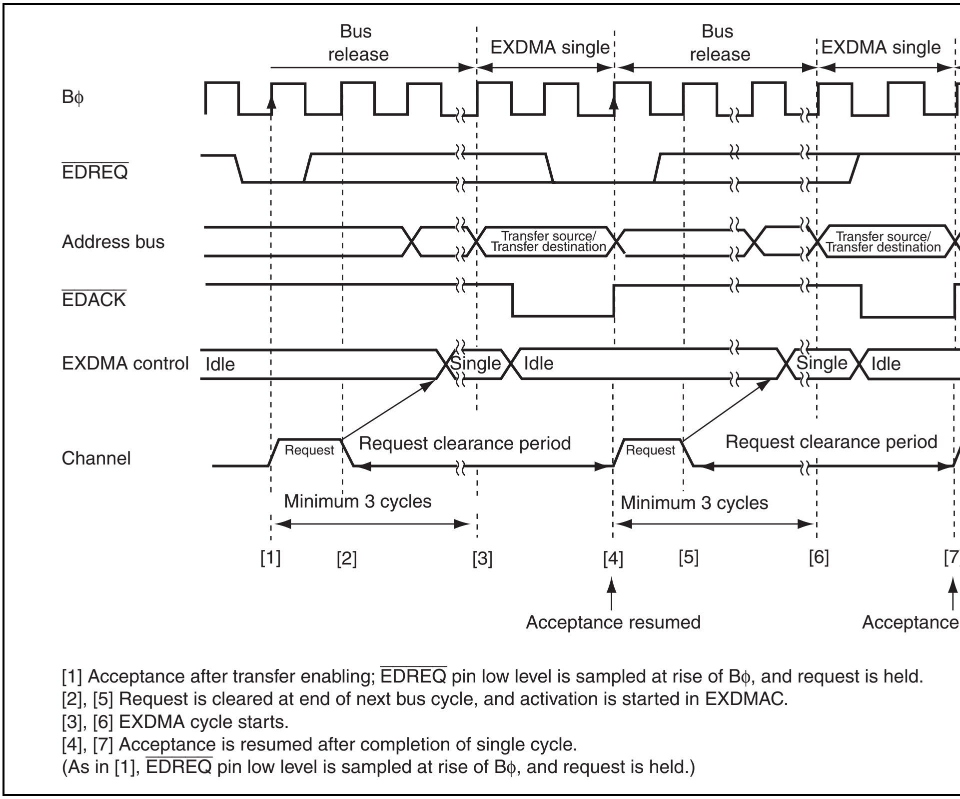
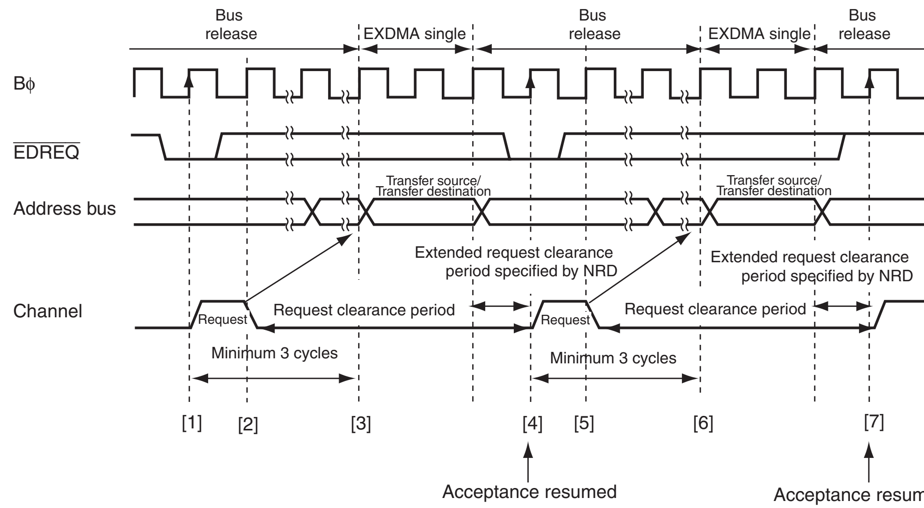


Figure 11.37 Example of Single Address Mode Transfer Activated by $\overline{EDREQ}$ Pin Low Level

$\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is within the EXDMAC, the request is cleared. After the end of the single cycle, acceptance when one cycle of the request clearance period specified by NRD = 1 expires and $\overline{\text{EDREQ}}$ level sampling is performed again. This sequence of operations is repeated until the end of transfer.



[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of Bφ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] EXDMA cycle starts.
[4], [7] Acceptance is resumed after completion of write cycle plus one cycle.
(As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of Bφ, and request is held.)

Figure 11.38 Example of Single Address Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level with NRD = 1 Specified

is resumed on completion of the higher-priority channel transfer.

Figures 11.39 and 11.40 show operation timing examples for various conditions.

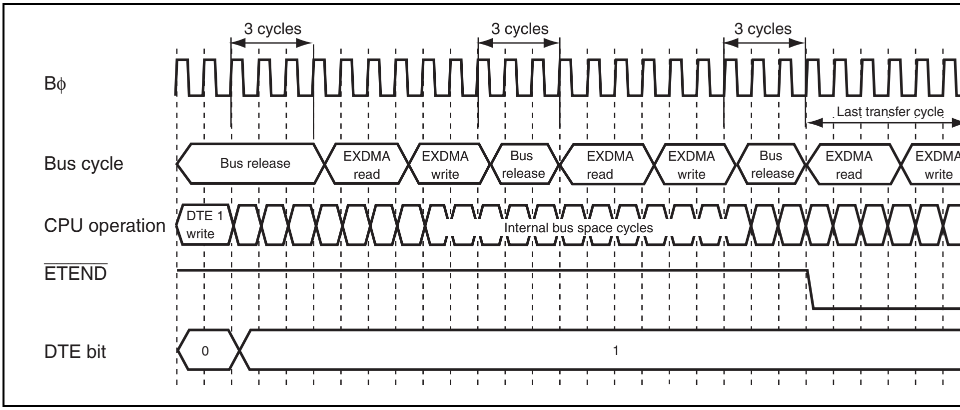


Figure 11.39 Auto-Request/Normal Transfer Mode/Cycle Steal Mode (No Conflict/Dual Address Mode)

Figure 11.40 Auto-Request/Normal Transfer Mode/Cycle Steal Mode (CPU Cycles/Single Address Mode)

(2) Auto-Request/Normal Transfer Mode/Burst Mode

With auto-request (in burst mode), when the DTE bit is set to 1 in EDMDR, an EXDMA cycle is started a minimum of three cycles later. Once transfer is started, it continues (as a until the transfer end condition is satisfied. Transfer requests for other channels are held p until the end of transfer on the current channel.

Figures 11.41 to 11.43 show operation timing examples for various conditions.

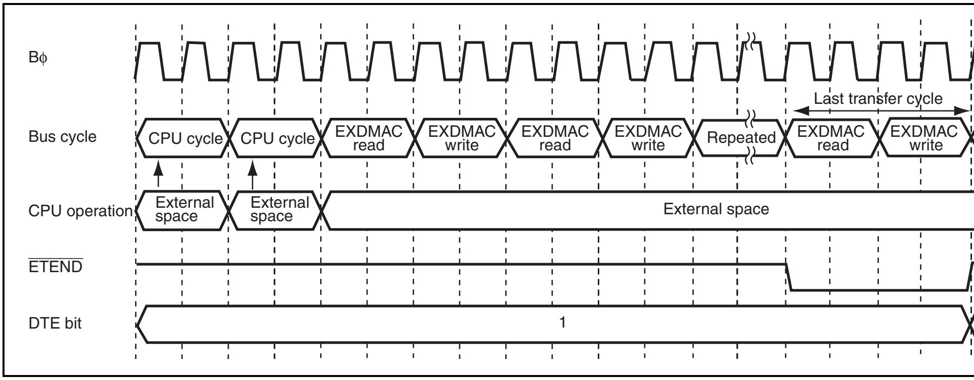


Figure 11.41 Auto-Request/Normal Transfer Mode/Burst Mode (CPU Cycles/Dual Address Mode)

Figure 11.42 Auto-Request/Normal Transfer Mode/Burst Mode (Conflict with Another Channel/Single Address Mode)

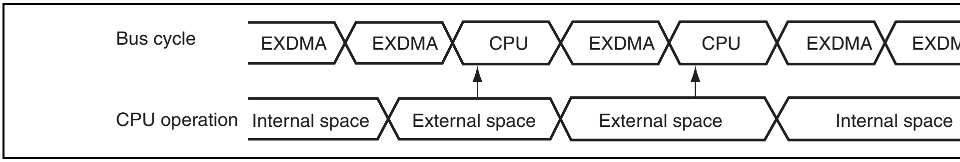


Figure 11.43 External Bus Master Cycle Steal Function (Auto-Request/Normal Mode/Burst Mode with CPU Cycles/Single Address Mode/EBCCS = 1)

(3) External Request/Normal Transfer Mode/Cycle Steal Mode

In external request mode, an EXDMA transfer cycle is started a minimum of three cycle transfer request is accepted. The next transfer request is accepted after the end of a one-t unit EXDMA cycle. For external bus space CPU cycles, at least one bus cycle is generat the next EXDMA cycle.

If a transfer request is generated for another channel, an EXDMA cycle for the other cha generated before the next EXDMA cycle.

The $\overline{\text{EDREQ}}$ pin sensing timing is different for low level sensing and falling edge sensin same applies to transfer request acceptance and transfer start timing.

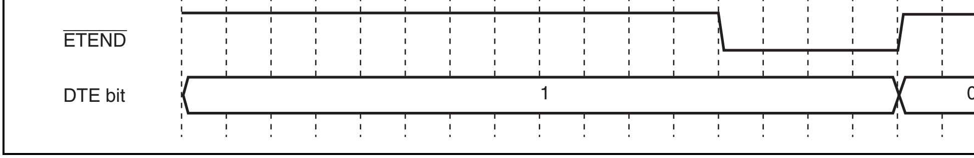


Figure 11.44 External Request/Normal Transfer Mode/Cycle Steal Mode (No Conflict/Dual Address Mode/Low Level Sensing)

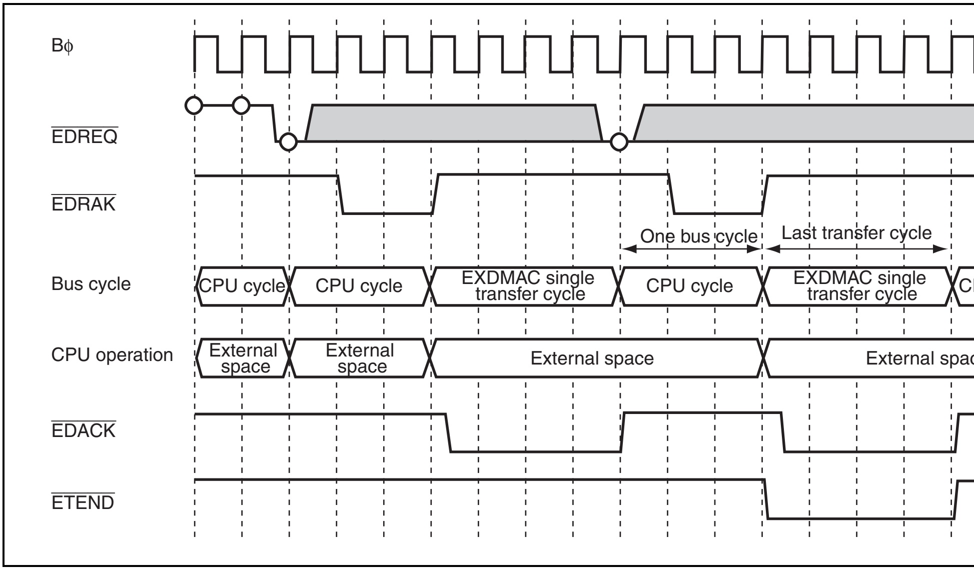


Figure 11.45 External Request/Normal Transfer Mode/Cycle Steal Mode (CPU Cycles/Single Address Mode/Low Level Sensing)

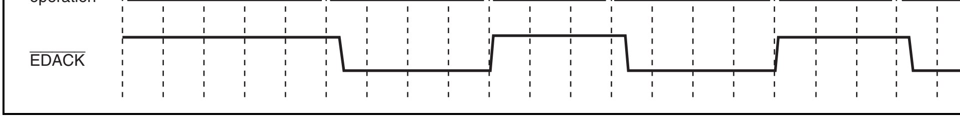


Figure 11.46 External Request/Normal Transfer Mode/Cycle Steal Mode (
Conflict/Single Address Mode/Falling Edge Sensing)

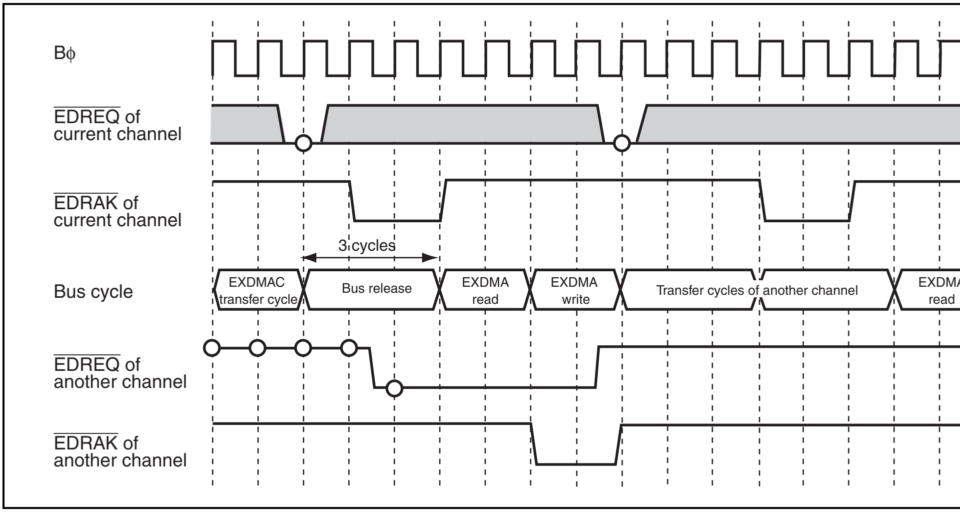


Figure 11.47 External Request/Normal Transfer Mode/Cycle Steal Mod
(Conflict with Another Channel/Dual Address Mode/Low Level Sensing)

Figures 11.48 to 11.52 show operation timing examples for various conditions.

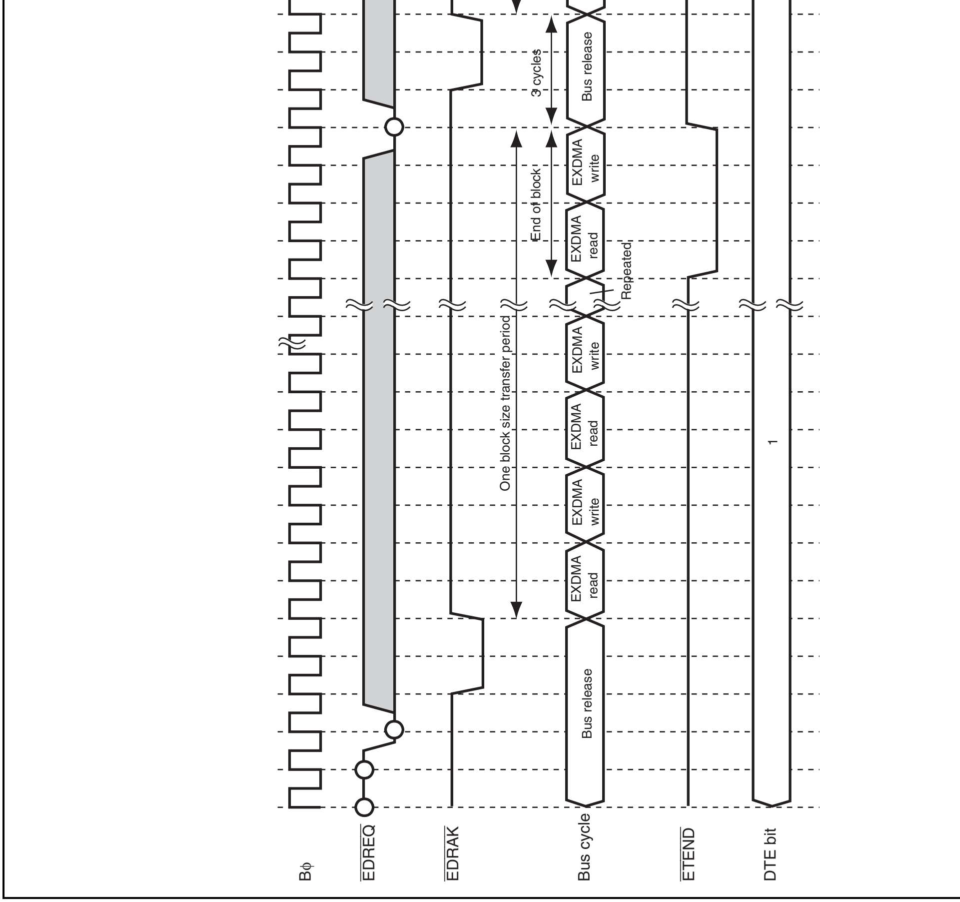


Figure 11.48 External Request/Block Transfer Mode/Cycle Steal Mode (No Conflict/Dual Address Mode/Low Level Sensing)

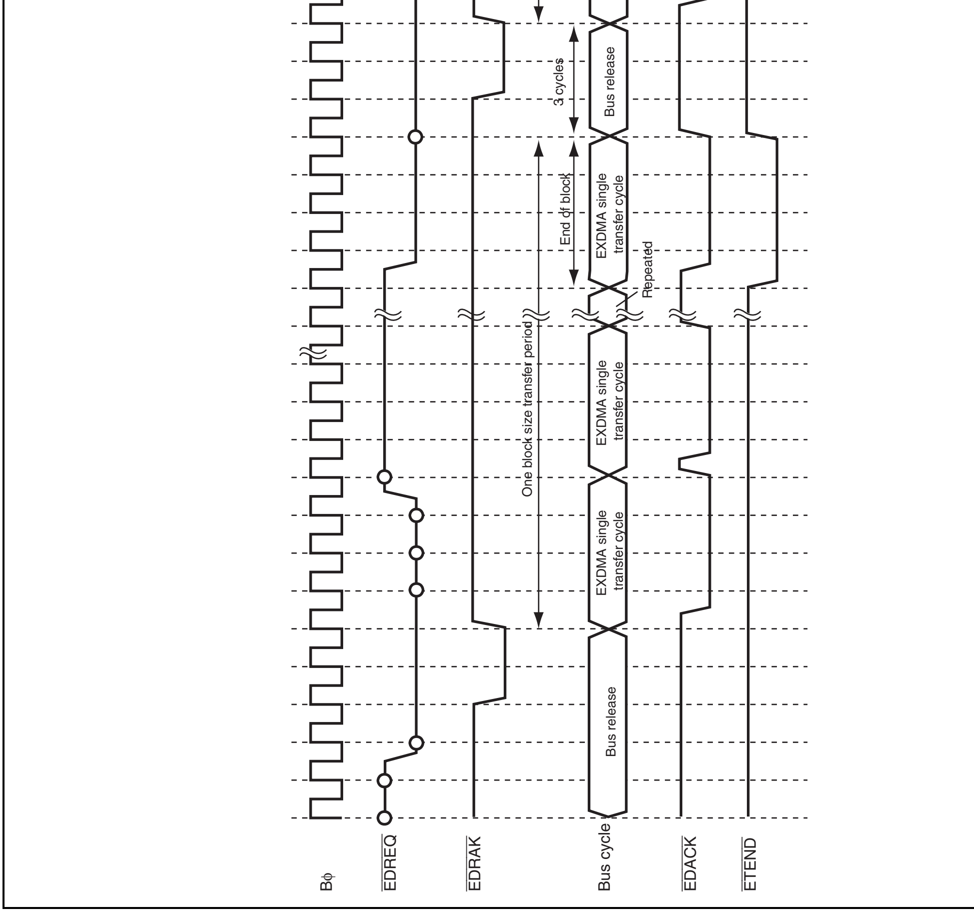


Figure 11.49 External Request/Block Transfer Mode/Cycle Steal Mode (No Conflict/Single Address Mode/Falling Edge Sensing)

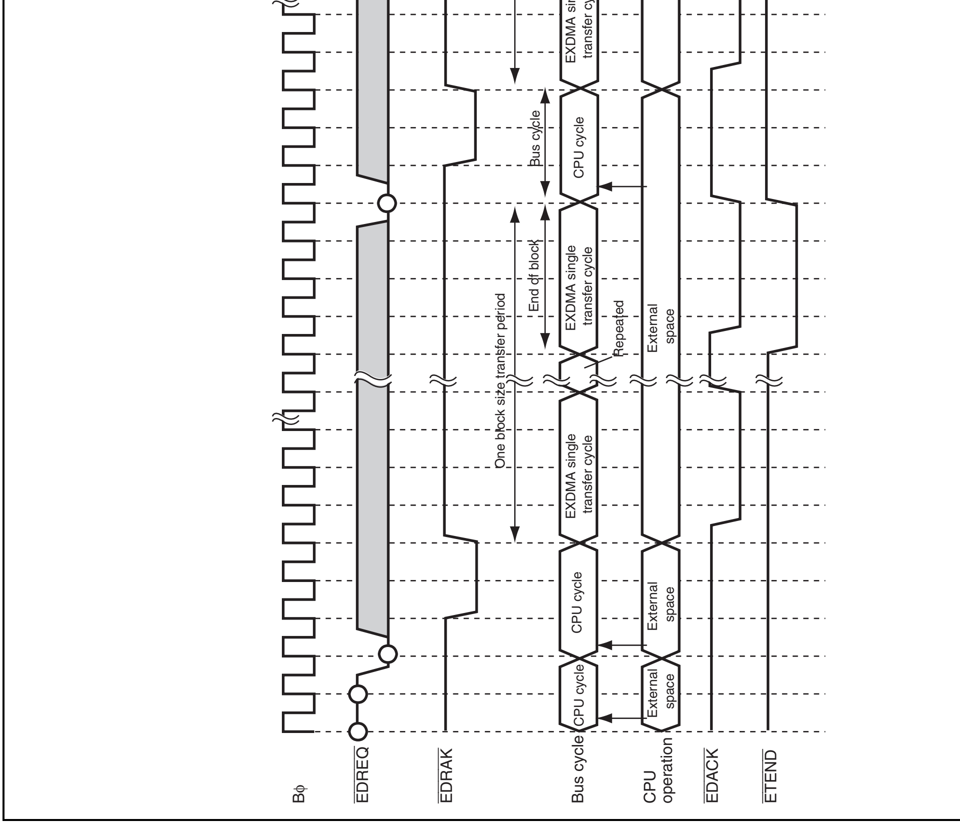


Figure 11.50 External Request/Block Transfer Mode/Cycle Steal Mode
(CPU Cycles/Single Address Mode/Low Level Sensing)

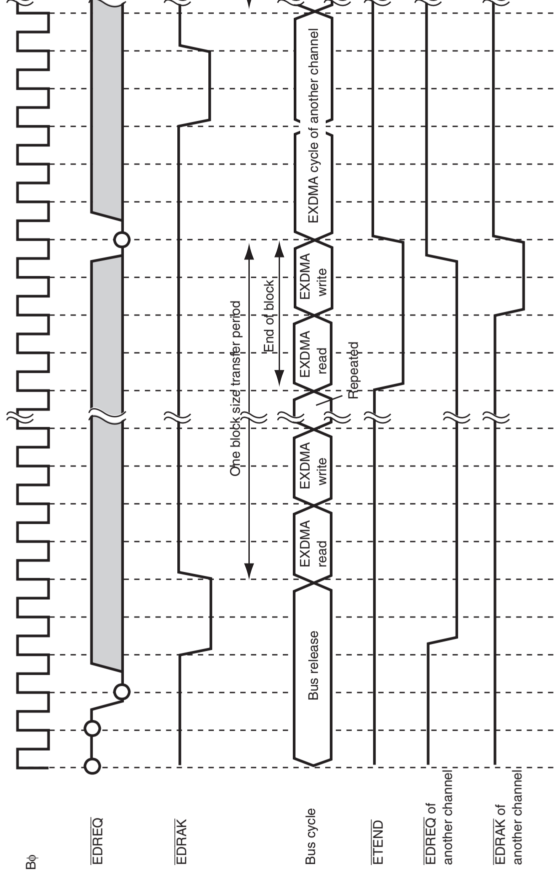


Figure 11.51 External Request/Block Transfer Mode/Cycle Steal Mode (Conflict with Another Channel/Dual Address Mode/Low Level Sensing)

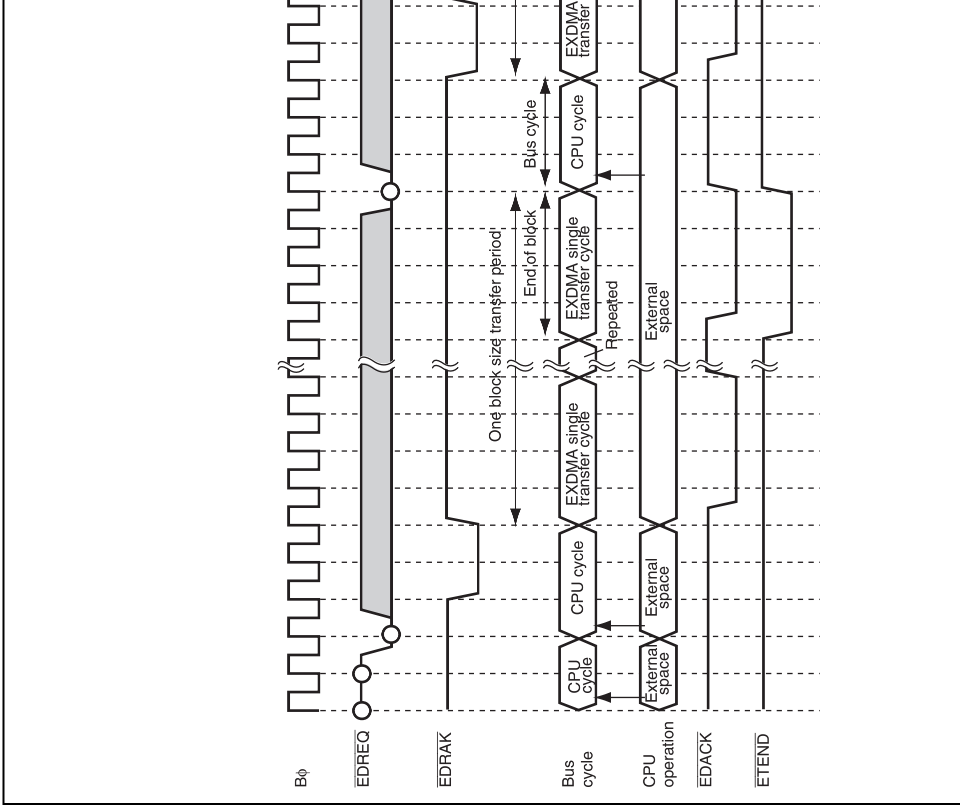


Figure 11.52 External Request/Block Transfer Mode/Cycle Steal Mode
(CPU Cycles/EBCCS = 1/Single Address Mode/Low Level Sensing)

In this mode, both the transfer source and destination addresses are specified for transfer
EXDMAC internal registers. The transfer source address is set in the source address regis
(EDSAR), and the transfer destination address is set in the destination address register (E

The transfer is processed by performing the consecutive read of a cluster-size from the tra
source address and then the consecutive write of that data to the transfer destination addre
data access size to 32 bytes can be specified as a cluster size. When one data access size i
specified as a cluster size, block transfer mode (dual address mode) is used.

The cycles in a cluster-size transfer are indivisible: another bus cycle (external access by
bus master, refresh cycle, or external bus release cycle) does not occur in a cluster-size tr

$\overline{\text{ETEND}}$ pin output can be enabled or disabled by means of the ETENDE bit in EDMDR.
is output for the last write cycle. The $\overline{\text{EDACK}}$ signal is not output.

Figure 11.53 shows the data flow in the cluster transfer mode (dual address mode), figure
shows an example of the timing in cluster transfer dual address mode, and figure 11.55 sh
cluster transfer dual address mode operation.

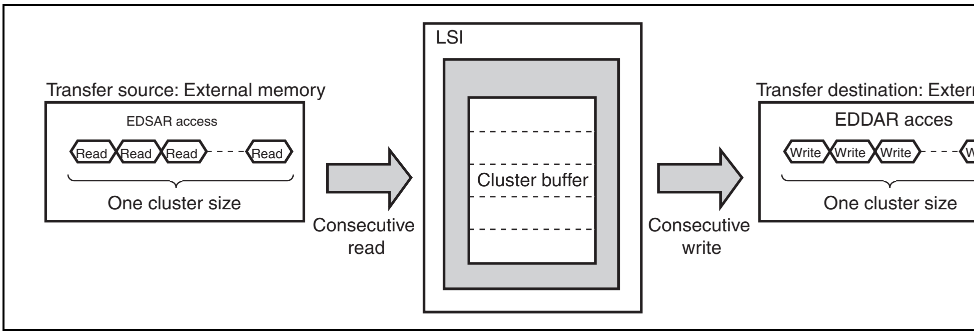


Figure 11.53 Data Flow in Cluster Transfer Dual Address Mode

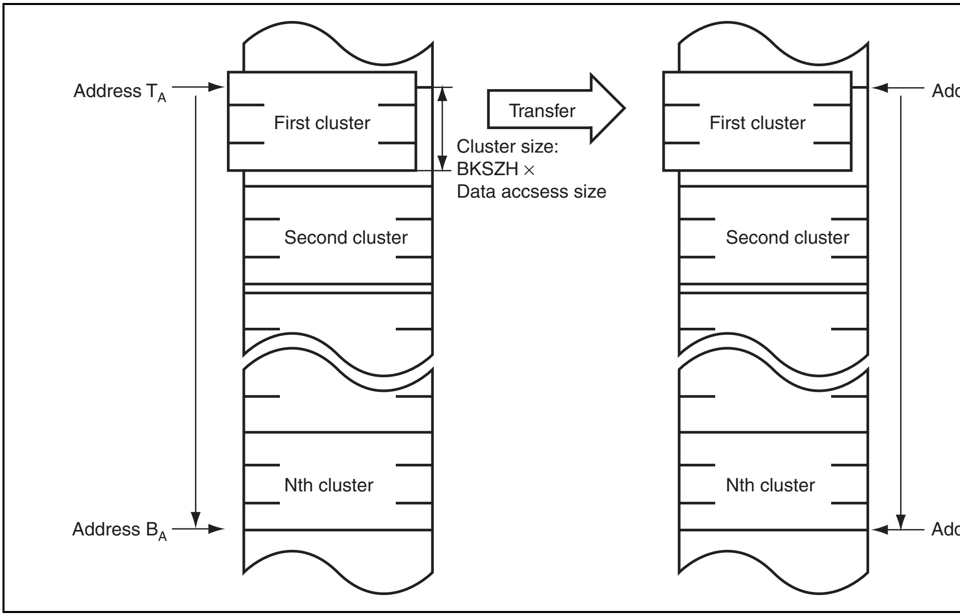


Figure 11.55 Cluster Transfer Dual Address Mode Operation

When a word or longword is specified as a data access size but the source or destination not at the word or longword boundary, use the appropriate data access size for efficient transfer.

In an example shown in figure 11.56, a longword-size transfer is performed with 4-long specified as a cluster size in the cluster transfer dual address mode from the lower two b to B'10.

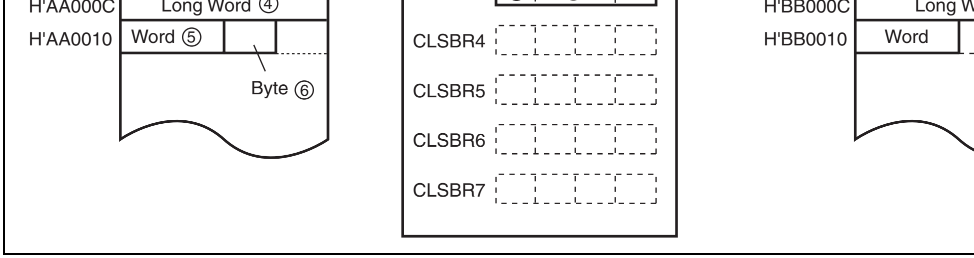


Figure 11.56 Odd Address Transfer

(2) Cluster Transfer Read Address Mode (AMS = 1, DIRS = 0)

In this mode, the transfer source address is specified in the source address register (EDSA
data is read from the transfer source and transferred to the cluster buffer. In this mode, the
bit in the mode control register (EDMDR) must be set to 1.

Two data access size to 32 bytes can be specified as a cluster size for the consecutive read
operation.

The cycles in a cluster-size transfer are indivisible: another bus cycle (external access by
bus master, refresh cycle, or external bus release cycle) does not occur in a cluster-size tr

$\overline{\text{ETEND}}$ pin output can be enabled or disabled by means of the ETENDE bit in EDMDR.
is output for the last read cycle. When an idle cycle is inserted before the last read cycle,
$\overline{\text{ETEND}}$ signal is also output in the idle cycle.

In this mode, the EDACKE bit in EDMDR must be set to 0 to disable the $\overline{\text{EDACK}}$ pin ou

Figure 11.57 shows the data flow in the cluster transfer read address mode (from the exte
memory to the cluster buffer), and figure 11.58 shows an example of the timing in cluster
read address mode.

Figure 11.57 Data Flow in Cluster Transfer Read Address Mode (from External Memory to Cluster Buffer)

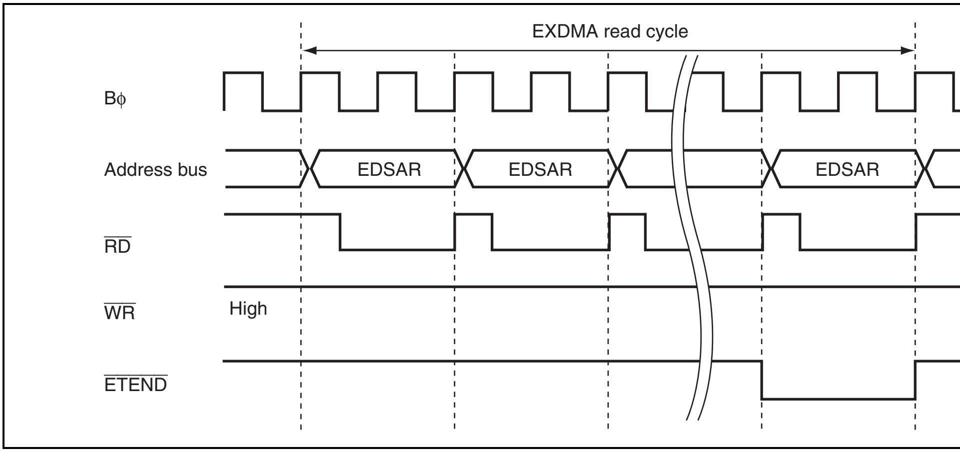


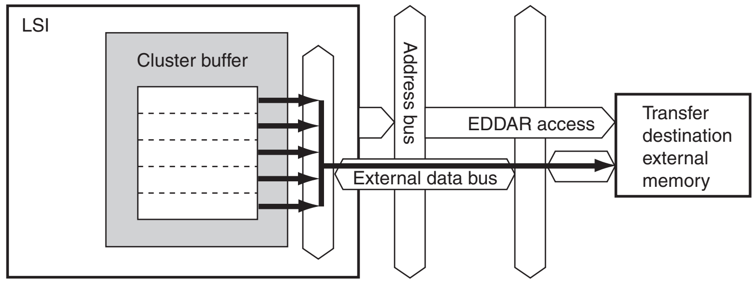
Figure 11.58 Timing in Cluster Transfer Read Address Mode (from External Memory to Cluster Buffer)

bus master, refresh cycle, or external bus release cycle) does not occur in a cluster size tr

$\overline{\text{ETEND}}$ pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. is output for the last write cycle. When an idle cycle is inserted before the last write cycle $\overline{\text{ETEND}}$ signal is also output in the idle cycle.

In this mode, the EDACKE bit in EDMCR must be set to 0 to disable the $\overline{\text{EDACK}}$ pin ou

Figure 11.59 shows the data flow in the cluster transfer write address mode (from the clus buffer to the external memory), and figure 11.60 shows an example of the timing in clust transfer write address mode.



When initializing an area by the specified data, write the specified data from cluster buffer 0 into a register sequentially. Then, specify the buffer size written in the register as a cluster size and the area to be initializ as DAR, and then execute transfer in this mode.

Figure 11.59 Data Flow in Cluster Transfer Write Address Mode (from Cluster Buffer to External Memory)

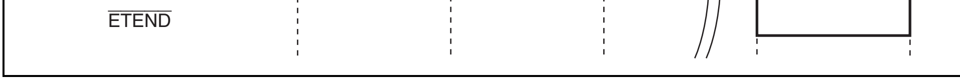


Figure 11.60 Timing in Cluster Transfer Write Address Mode (from Cluster Buffer to External Memory)

11.6.2 Setting of Address Update Mode

The cluster transfer mode transfer is restricted by the address update mode function. The following four address update methods: increment, decrement, fixed, and offset addition

When the address increment method is specified and if the specified address is not at the boundary for the data access size (odd address for a word-size transfer, address beyond t boundary for a longword-size transfer), the bus cycle is divided for transfer until the add becomes at the address boundary. When the address matches the boundary, transfer is pr units of data access sizes. At the end of transfer, the bus cycle is divided again to transfe remaining data in cluster transfer mode.

With address decrement, fixed, or offset addition method, specify the address, that match address boundary for the data access size, in EDSAR and EDDAR. When specifying the that is not at the address boundary for the data access size, in EDSAR and EDDAR, fix t bit to 0 (lower one bit for a word-size transfer, and lower two bits for a longword-size tr the address register so that the transfer is processed in units of data access sizes. The blo mode must be used for transfer of data by dividing the bus cycle according to the addres boundary.

When the EDTCR value is smaller than the cluster size, a transfer size error occurs. In th when the TSEIE bit in EDMDR is cleared to 0, the cluster transfer mode is switched to t transfer mode to process the remaining data. With the decrement, fixed, or offset additio transfer is performed without fixing the lower bit to 0.

(1) Cluster transfer mode

In cluster transfer mode, a cluster-size transfer is processed in response to one transfer rec

In an example shown in figure 11.61, the $\overline{\text{ETEND}}$ pin output is enabled, and word-size tra
performed with 4-byte cluster size in cluster transfer mode from the external 16-bit, 2-sta
space to the external 16-bit, 2-state access space.

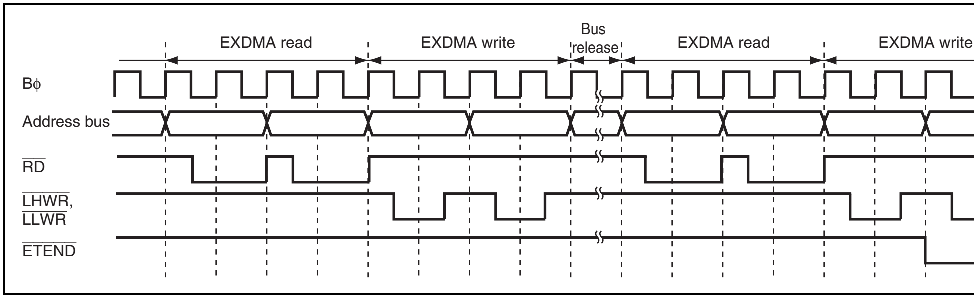
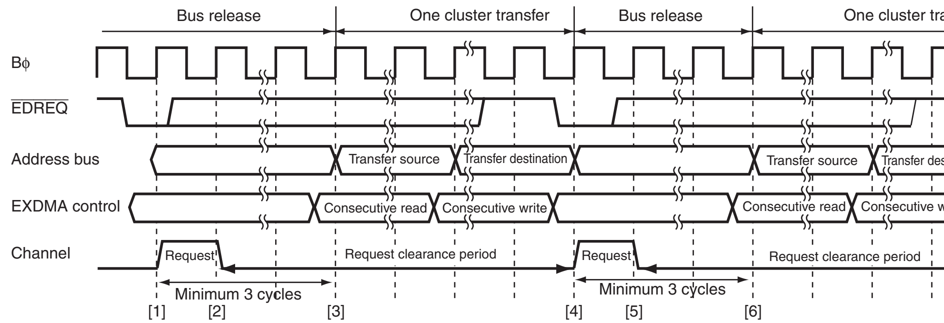


Figure 11.61 Example of Cluster Transfer Mode Transfer

is started. If $\overline{EDREQ}$ pin high level sampling is completed by the end of the last cluster
cycle, acceptance resumes after the end of the write cycle, and $\overline{EDREQ}$ pin low level sa
performed again. This sequence of operations is repeated until the end of the transfer.



[1] Acceptance after transfer enabling; $\overline{EDREQ}$ pin low level is sampled at rise of Bφ, and request is held.
[2] [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3] [6] EXDMA cycle starts; $\overline{EDREQ}$ pin high level sampling is started at rise of Bφ.
[4] [7] When $\overline{EDREQ}$ pin high level has been sampled, acceptance is resumed after completion of write cycle.
(As in [1], $\overline{EDREQ}$ pin low level is sampled at rise of Bφ, and request is held.)

Figure 11.62 Example of Cluster Transfer Mode Transfer Activated by $\overline{EDREQ}$ Pin Falling Edge

resumes and EDREQ pin low level sampling is performed again. This sequence of operat
repeated until the end of the transfer.

When NRD bit = 0 in EDMDR, acceptance resumes at the end of the last cluster write cy
EDREQ pin low level sampling is performed again. This sequence of operations is repeat
the end of the transfer.

When NRD bit = 1 in EDMDR, acceptance resumes after one cycle from the end of the la
write cycle, and EDREQ pin low level sampling is performed again. This sequence of ope
is repeated until the end of the transfer.

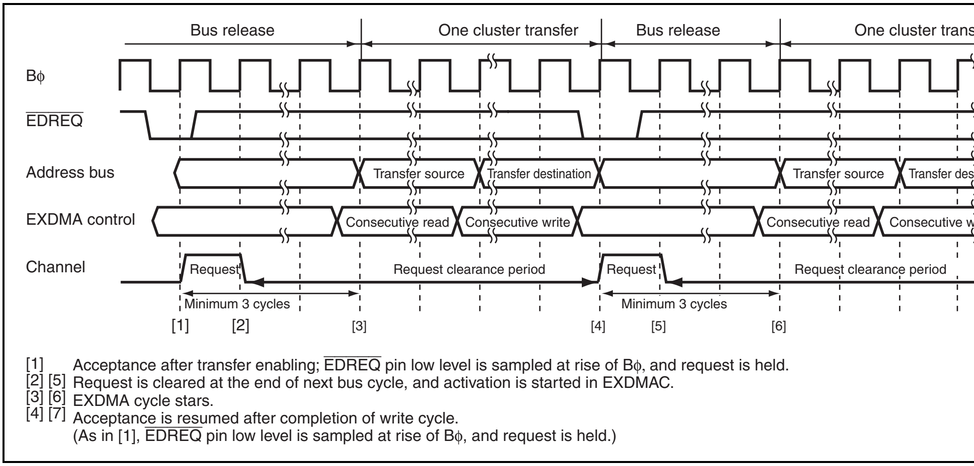


Figure 11.63 Example of Cluster Transfer Mode Transfer Activated by EDREQ Pin Low Level

channel is resumed on completion of the higher-priority channel transfer.

The cluster transfer mode (read address mode and write address mode) can not be used v
cluster transfer mode (dual address mode) among more than one channel at the same tim
using the cluster transfer mode (read address mode and write address mode), do not set t
transfer mode for another channel.

Figures 11.64 to 11.66 show operation timing examples for various conditions.

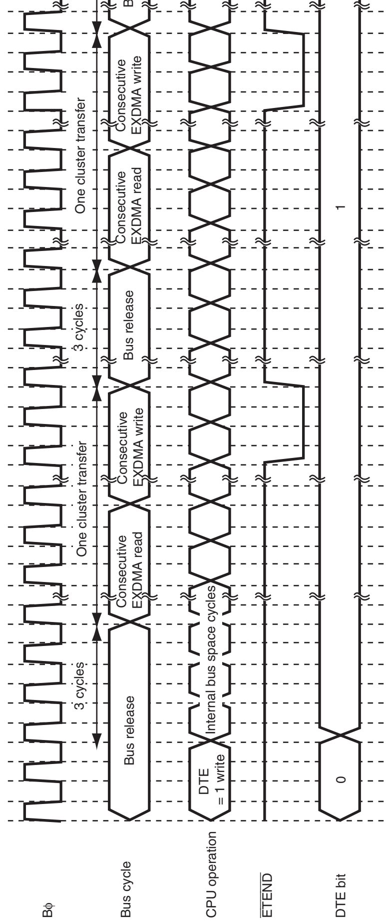


Figure 11.64 Auto-Request/Cluster Transfer Mode/Cycle Steal Mode (No Confict/Dual Address Mode)

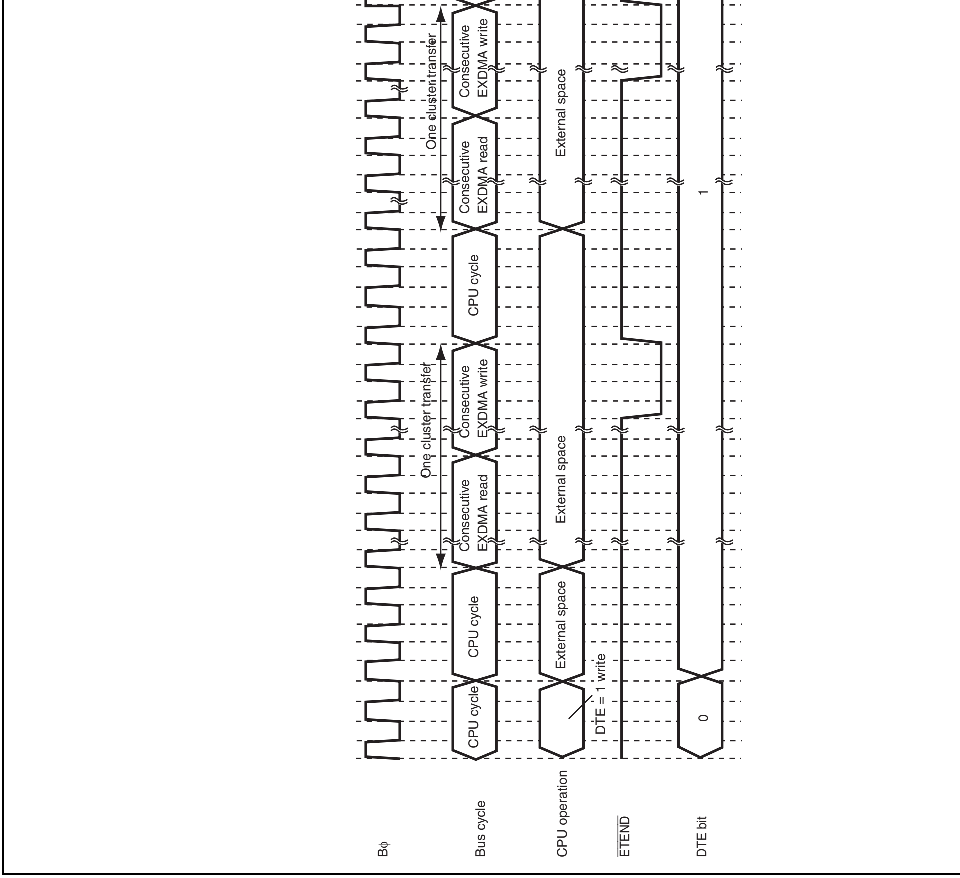


Figure 11.65 Auto-Request/Cluster Transfer Mode/Cycle Steal Mode (CPU Cycles/Dual Address Mode)

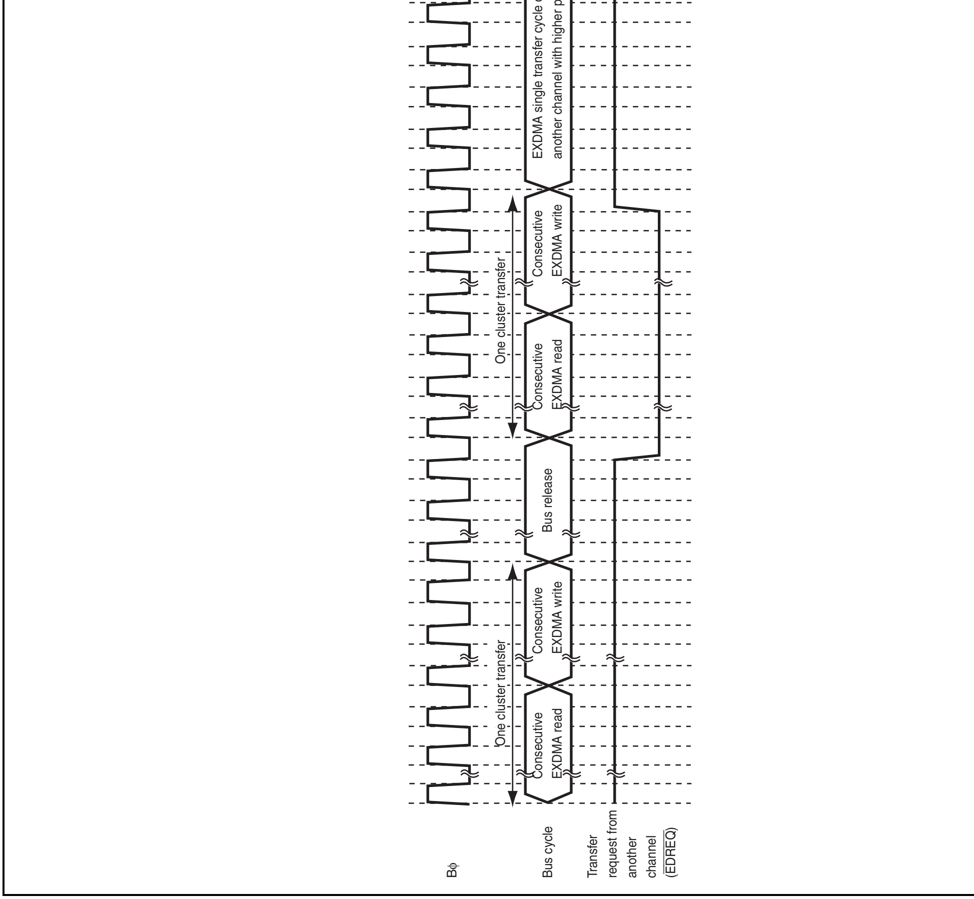


Figure 11.66 Auto-Request/Cluster Transfer Mode/Cycle Steal Mode (Conflict with Another Channel/Dual Address Mode)

transfer mode for another channel.

The $\overline{\text{EDREQ}}$ pin sensing timing is different for low level sensing and falling edge sensin
same applies to transfer request acceptance and transfer start timing.

Figures 11.67 to 11.69 show operation timing examples for various conditions.

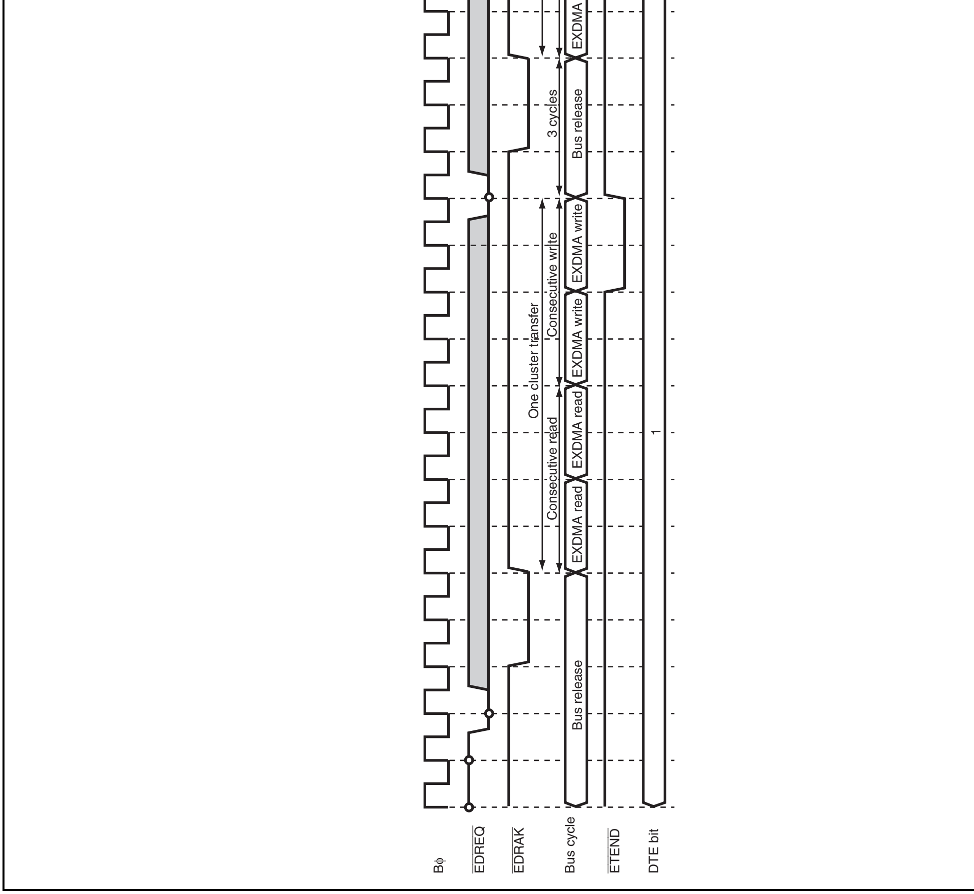


Figure 11.67 External Request/Cluster Transfer Mode/Cycle Steal Mode (No Conflict/Dual Address Mode/Low Level Sensing)

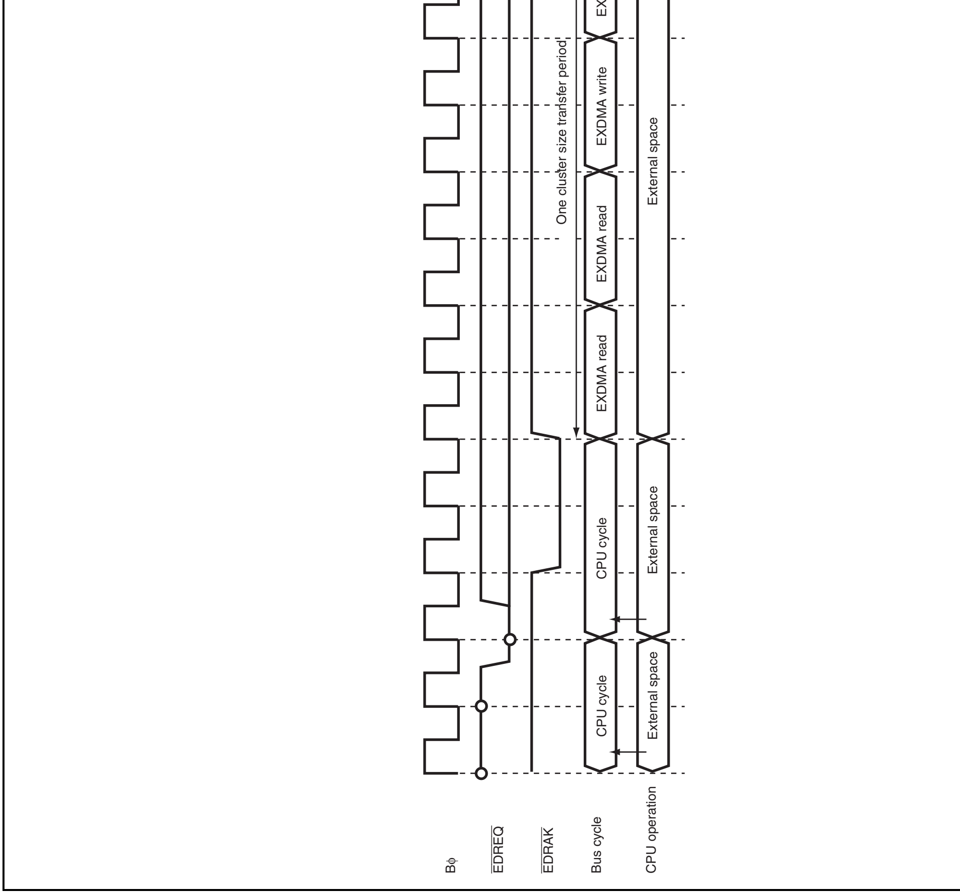


Figure 11.68 External Request/Cluster Transfer Mode/Cycle Steal Mode (CPU Cycles/Dual Address Mode/Low Level Sensing)

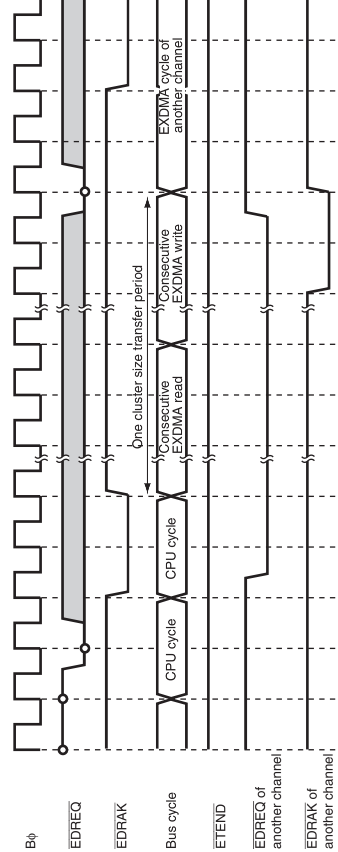


Figure 11.69 External Request/Cluster Transfer Mode/Cycle Steal Mode (Conflict with Another Channel/Dual Address Mode/Low Level Sensing)

generated by the transfer counter. EXDMA transfer does not end if the EDTCR value ha
since before the start of transfer.

(2) Transfer End by Transfer Size Error Interrupt

When the following conditions are satisfied while the TSEIE bit in EDMDR is set to 1, size error occurs and an EXDMA transfer is terminated. At this time, the DTE bit in ED cleared to 0 and the ESIF bit in EDMDR is set to 1.

- In normal transfer mode and repeat transfer mode, when the next transfer is requeste transfer is disabled due to the EDTCR value less than the data access size.
- In block transfer mode, when the next transfer is requested while a transfer is disable the EDTCR value less than the block size.
- In cluster transfer mode, when the next transfer is requested while a transfer is disabl the EDTCR value less than the cluster size.

When the TSEIE bit in EDMDR is cleared to 0, data is transferred until the EDTCR valu 0. A transfer size error is not generated. Operation in each transfer mode is described be

- In normal transfer mode and repeat mode, when the EDTCR value is less than the da size, data is transferred in bytes.
- In block transfer mode, when the EDTCR value is less than the block size, the specif data in EDTCR is transferred instead of transferring the block size of data. When the value is less than the data access size, data is transferred in bytes.
- In cluster transfer mode, when the EDTCR value is less than the cluster size, the spe of data in EDTCR is transferred instead of transferring the cluster size of data. When EDTCR value is less than the data access size, data is transferred in bytes.

is generated at the end of a cluster-size transfer, a repeat size end interrupt request is gene

(4) Transfer End by Extended Repeat Area Overflow Interrupt

If an address overflows the extended repeat area when an extended repeat area specificati been made and the SARIE or DARIE bit in EDACR is set to 1, an extended repeat area o interrupt is requested. The interrupt request terminates EXDMA transfer, the DTE bit in I is cleared to 0, and the ESIF bit in EDMDR is set to 1 at the same time.

In dual address mode, if an extended repeat area overflow interrupt is requested during a cycle, the following write cycle processing is still executed.

In block transfer mode, if an extended repeat area overflow interrupt is requested during t of a block, transfer continues to the end of the block. Transfer end by means of an extend area overflow interrupt occurs between block-size transfers.

In cluster transfer mode, if an extended repeat area overflow interrupt is requested during of a cluster, transfer continues to the end of the cluster. Transfer end by means of an exte repeat area overflow interrupt occurs between cluster-size transfers.

(5) Transfer End by 0-Write to DTE Bit in EDMDR

When 0 is written to the DTE bit in EDMDR by the CPU, etc., transfer ends after comple the EXDMA cycle in which transfer is in progress or a transfer request was accepted.

In block transfer mode, EXDMA transfer ends after completion of one-block-size transfe progress.

In cluster transfer mode, EXDMA transfer ends after completion of one-cluster-size trans progress.

units of transfers.

In single address mode, EXDMA transfer ends at the end of the EXDMA transfer bus cy
units of transfers.

(b) Block transfer mode

A block size EXDMA transfer is aborted. A block size transfer is not correctly executed
matching between the actual transfer and the transfer request is not guaranteed.

In dual address mode, a write cycle corresponding to a read cycle is executed as well as
normal transfer mode.

(c) Cluster transfer mode

A cluster size EXDMA transfer is aborted. If transfer is aborted in a read cycle, the read
guaranteed. If transfer is aborted in a write cycle, the data not transferred is not guarante
Matching between the transfer counter and the address register is not guaranteed since th
processing cannot be controlled.

(7) Transfer End by Address Error

If an address error occurs, the EXDMAC clears the DTE bit to 0 in all channels, and set
bit in EDMDR_0 to 1. An address error during EXDMA transfer forcibly terminates the
To perform EXDMA transfer after an address error occurs, clear the ERRF bit to 0 and t
the DTE bit to 1 in each channel.

The transfer end timing after address error detection is the same as for the one when an I
interrupt occurs.

control register (CPUPCR). For details, see section 7.7, CPU Priority Control Function O
DMAC, and EXDMAC.

The EXDMAC priority level can be set independently for each channel by the EDMAP2
EDMAP0 bits in EDMDR.

The CPU priority level, which corresponds to the priority level of exception handling, ca
by updating the values of the CPUP2 to CPUP0 bits in CPUPCR with the interrupt mask
values.

When the CPUPCE bit in CPUPCR is set to 1 to enable the CPU priority level control an
EXDMAC priority level is lower than the CPU priority level, the transfer request of the
corresponding channel is masked and the channel activation is disabled. When the priorit
another channel is the same or higher than the CPU priority level, the transfer request for
channel is accepted and transfer is enabled regardless of the priority levels of channels.

The CPU priority level control function holds pending the transfer source, which masked
transfer request. When the CPU priority level becomes lower than the channel priority lev
updating one of them, the transfer request is accepted and transfer starts. The transfer req
pending is cleared by writing 0 to the DTE bit.

When the CPUPCE bit is cleared to 0, the lowest CPU priority level is assumed.

These consecutive EXDMA read and write cycles are indivisible: external bus release cy
external space access cycle by internal bus master (CPU, DTC, DMAC) does not occur
read cycle and a write cycle.

In cluster transfer mode, the transfer cycle in one cluster is indivisible.

In block transfer mode and auto-request burst mode, the EXDMA transfer bus cycles co
this period, the bus priority level of the internal bus master is lower than the EXDMAC
external space access is held pending (when EBCCS = 0 in the bus control register 2 (B

When switching to another channel, or in the auto-request cycle steal mode, the EXDM
cycles and internal bus master cycles are alternatively executed. When the internal bus
not issuing an external space access cycle, the EXDMA transfer bus cycles are continuo
executed in the allowable range.

When the EBCCS bit in BCR2 is set to 1 to enable the arbitration function between the
and the internal bus master, the bus mastership is released, except for indivisible bus cyc
transferred between the EXDMAC and the internal bus master alternatively. For details,
section 9, Bus Controller (BSC).

EXDMTEND1	Transfer end indicated by channel 1 transfer counter	
EXDMTEND2	Transfer end indicated by channel 2 transfer counter	
EXDMTEND3	Transfer end indicated by channel 3 transfer counter	
EXDMEEND0	Channel 0 transfer size error	
	Channel 0 repeat size end	
	Channel 0 source address extended repeat area overflow	
	Channel 0 destination address extended repeat area overflow	
EXDMEEND1	Channel 1 transfer size error	
	Channel 1 repeat size end	
	Channel 1 source address extended repeat area overflow	
	Channel 1 destination address extended repeat area overflow	
EXDMEEND2	Channel 2 transfer size error	
	Channel 2 repeat size end	
	Channel 2 source address extended repeat area overflow	
	Channel 2 destination address extended repeat area overflow	
EXDMEEND3	Channel 3 transfer size error	
	Channel 3 repeat size end	
	Channel 3 source address extended repeat area overflow	
	Channel 3 destination address extended repeat area overflow	L

Interrupt source can be enabled or disabled by setting the DTIE and ESIE bits in EDMDR
relevant channels. The DTIE bit can be combined with the DTIF bit in EDMDR to genera
EXDMTEND interrupt. The ESIE bit can be combined with the ESIF bit in EDMDR to g
an EXDMEEND interrupt. Interrupt sources in EXDMEEND are not identified as comm
interrupts. The interrupt priority order among channels is determined by the interrupt con
shown in table 11.7. For detains see section 7, Interrupt Controller.

corresponding interrupt enable bit is set to 1, the condition for that interrupt is satisfied, the ESIF bit in EDMDR is set to 1.

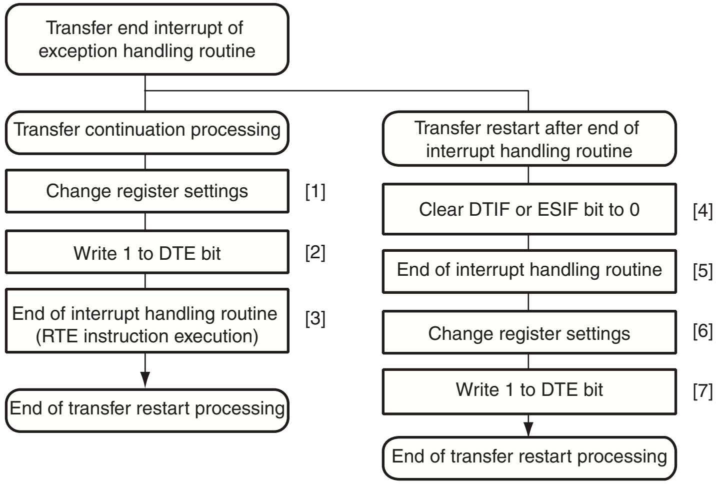
The transfer size error interrupt occurs when the EDTCR value is smaller than the data a and a data-access-size transfer for one request cannot be performed for a transfer reques transfer mode, the block size is compared to the EDTCR value to determine a transfer si In cluster transfer mode, the cluster size is compared to the EDTCR value to determine a size error.

The repeat size end interrupt occurs when the next transfer request is generated after the repeat size transfer in repeat transfer mode. When the repeat area is not set in the addres transfer can be aborted periodically based on the set repeat size value. If the transfer end by the transfer counter occurs at the same time, the ESIF bit is set to 1.

The source/destination address extended repeat area overflow interrupt occurs when the overflow the specified extended repeat area. If the transfer end interrupt by the transfer c occurs at the same time, the ESIF bit is set to 1.

Figure 11.70 shows the block diagram of various interrupts and their interrupt flags. The end interrupt can be cleared either by clearing the DTIF or ESIF bit to 0 in EDMDR wit interrupt handling routine, or by re-setting the address registers and then setting the DTE in EDMDR to perform transfer continuation processing. An example of the procedure fo the transfer end interrupt and restarting transfer is shown in figure 11.71.

Figure 11.70 Interrupts and Interrupt Sources



[1] Write set values to the registers (transfer counter, address registers, etc.)

[2] Write 1 to the DTE bit in EDMDR to restart EXDMA operation. When 1 is written to the DTE bit, the DTIF or ESIF bit in EDMDR is automatically cleared to 0 and the interrupt source is cleared.

[3] The interrupt handling routine is ended with an RTE instruction, etc.

[4] Write 0 to the DTIF or ESIF bit in EDMDR by first reading 1 from it.

[5] After the interrupt handling routine is ended with an RTE instruction, etc., interrupt masking is cleared.

[6] Write set values to the registers (transfer counter, address registers, etc.).

[7] Write 1 to the DTE bit in EDMDR to restart EXDMA operation.

Figure 11.71 Procedure for Clearing Transfer End Interrupt and Restarting Tr

initial value is "enabled".

When the MSTPA14 bit is set to 1 in MSTPCRA, the EXDMAC clock stops and the EX
enters the module stop state. However, 1 cannot be written to the MSTPA14 bit when an
EXDMAC's channels is enabled for transfer, or when an interrupt is being requested. Be
setting the MSTPA14 bit, first clear the DTE bit in EDMDR to 0, then clear the DTIF or
in EDMDR to 0.

When the EXDMAC clock stops, EXDMAC registers can no longer be accessed. The fo
EXDMAC register settings remain valid in the module stop state, and so should be disab
necessary, before making the module stop transition.

- ETENDE = 1 in EDMDR ($\overline{\text{ETEND}}$ pin enable)
- EDRAKE = 1 in EDMDR ($\overline{\text{EDRAK}}$ pin enable)
- EDACKE = 1 in EDMDR ($\overline{\text{EDACK}}$ pin enable)

(3) $\overline{\text{EDREQ}}$ Pin Falling Edge Activation

Falling edge sensing on the $\overline{\text{EDREQ}}$ pin is performed in synchronization with EXDMAC
operations, as indicated below.

1. Activation request standby state: Waits for low level sensing on $\overline{\text{EDREQ}}$ pin, then go
2. Transfer standby state: Waits for EXDMAC data transfer to become possible, then g
3. Activation request disabled state: Waits for high level sensing on $\overline{\text{EDREQ}}$ pin, then g

After EXDMAC transfer is enabled, the EXDMAC goes to state [1], so low level sensin
for the initial activation after transfer is enabled.

In cluster transfer mode, the same cluster buffer is used for all channels. When more than cluster transfer conflicts, the cluster buffer register holds the value of the last cluster trans When the transfer between the transfer source/destination and the cluster buffer conflicts another cluster transfer, the transferred data in the cluster buffer may be overwritten by ar channel cluster transfer. Therefore, in the cluster transfer mode (single address mode), do the cluster transfer mode for any other channels.

(6) Cluster Transfer Mode and Endian

In cluster transfer mode, only a transfer to the areas in the big endian format is supported. cluster transfer mode is specified, do not specify the areas in the little endian format for E and EDDAR. For details on the endian, see section 9, Bus Controller (BSC).

- 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

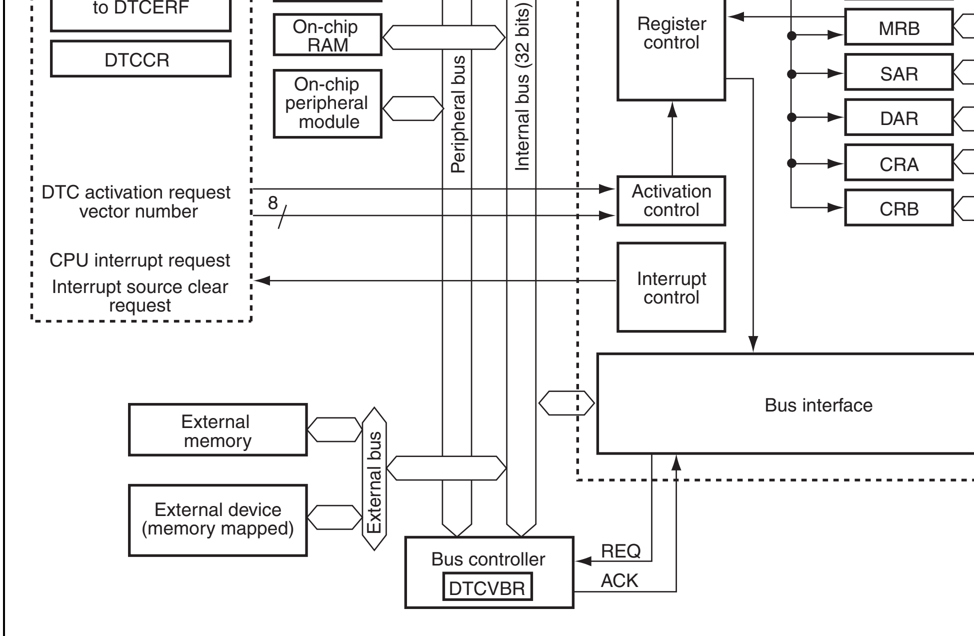
    The transfer source and destination addresses can be specified by 32 bits to selec
    Gbyte address space directly
- Size of data for data transfer can be specified as byte, word, or longword

  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 state 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 DTCERF:	DTC enable registers A to F
DTCCR:	DTC control register
DTCVBR:	DTC vector base register

Figure 12.1 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 F (DTCERA to DTCERF)
- 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 wh and Sz0 = B'01; by +4 when Sz1 and Sz0 = 11: SAR is decremented after a transfer (by –1 when Sz1 and Sz0 = B'00; by –2 wh and Sz0 = B'01; by –4 when Sz1 and Sz0 =
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]

x: Don't care

Bit	Bit Name	Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable Specifies the chain transfer. For details, see s 12.5.7, Chain Transfer. The chain transfer co selected by the CHNS bit. 0: Disables the chain transfer 1: Enables the chain transfer
6	CHNS	Undefined	—	DTC Chain Transfer Select Specifies the chain transfer condition. If the fo transfer is a chain transfer, the completion ch specified transfer count is not performed and source flag or DTCER is not cleared. 0: Chain transfer every time 1: Chain transfer only when transfer counter =
5	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt requ generated every time after a data transfer enc this bit is set to 0, a CPU interrupt request is c generated when the specified number of data ends.
4	DTS	Undefined	—	DTC Transfer Mode Select Specifies either the source or destination as r block area during repeat or block transfer mo 0: Specifies the destination as repeat or block 1: Specifies the source as repeat or block are

				(by –1 when Sz1 and Sz0 = B'00; by –2 wh and Sz0 = B'01; by –4 when Sz1 and Sz0 =
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]
x: Don't care

12.2.3 DTC Source Address Register (SAR)

SAR is a 32-bit register that designates the source address of data to be transferred by the

In full address mode, 32 bits of SAR are valid. In short address mode, the lower 24 bits o valid and bits 31 to 24 are ignored. At this time, the upper eight bits are filled with the va bit 23.

If a word or longword access is performed while an odd address is specified in SAR or if longword access is performed while address 4n + 2 is specified in SAR, the bus cycle is d into multiple cycles to transfer data. For details, see section 12.5.1, Bus Cycle Division.

SAR cannot be accessed directly from the CPU.

into multiple cycles to transfer data. For details, see section 12.5.1, Bus Cycle Division.

DAR cannot be accessed directly from the CPU.

12.2.5 DTC Transfer Count Register A (CRA)

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

In normal transfer mode, CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and bit DTCEn (n = 15 to 0) correspond activation source is cleared and then an interrupt is requested to the CPU when the coun H'0000. The transfer count is 1 when CRA = H'0001, 65,535 when CRA = H'FFFF, and when CRA = H'0000.

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

In block transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and eight bits (CRAL). CRAH holds the block size while CRAL functions as an 8-bit block- counter (1 to 256 for byte, word, or longword). CRAL is decremented by 1 every time a (word or longword) data is transferred, and the contents of CRAH are sent to CRAL wh 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.

12.2.7 DTC enable registers A to F (DTCERA to DTCERF)

DTCER, which is comprised of eight registers, DTCERA to DTCERF, is a register that s
DTC activation interrupt sources. The correspondence between interrupt sources and DT
shown in table 12.1. Use bit manipulation instructions such as BSET and BCLR to read o
DTCE bit. If all interrupts are masked, multiple activation sources can be set at one time (
the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	15	14	13	12	11	10	9	
Bit Name	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	D
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	D
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	

7	DTCE7	0	R/W	These bits are not cleared when the DISEL bit … the specified number of transfers have not ende…
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	

12.2.8 DTC Control Register (DTCCR)

DTCCR specifies transfer information read skip.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	RRS	RCHNE	—	—
Initial Value	0	0	0	0	0	0	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.

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 va… always be 0.

				0: Transfer read skip is not performed. 1: Transfer read skip is performed when the ve numbers match.
3	RCHNE	0	R/W	Chain Transfer Enable After DTC Repeat Tran Enables/disables the chain transfer while trans counter (CRAL) is 0 in repeat transfer mode. In repeat transfer mode, the CRAH value is wri CRAL when CRAL is 0. Accordingly, chain trar not occur when CRAL is 0. If this bit is set to 1, chain transfer is enabled when CRAH is writter CRAL. 0: Disables the chain transfer after repeat trans 1: Enables the chain transfer after repeat trans
2, 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	ERR	0	R/(W)*	Transfer Stop Flag Indicates that an address error or an NMI interr occurs. If an address error or an NMI interrupt the DTC stops. 0: No interrupt occurs 1: An interrupt occurs [Clearing condition] • When writing 0 after reading 1

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

Bit Name															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

12.3 Activation Sources

The DTC is activated by an interrupt request. The interrupt source is selected by DTCER activation source can be selected by setting the corresponding bit in DTCER; the CPU in source can be selected by clearing the corresponding bit in DTCER. At the end of a data (or the last consecutive transfer in the case of chain transfer), the activation source interr corresponding DTCER bit is cleared.

activation source, and then reads the transfer information from the start address. Figure 12
correspondences between the DTC vector address and transfer information.

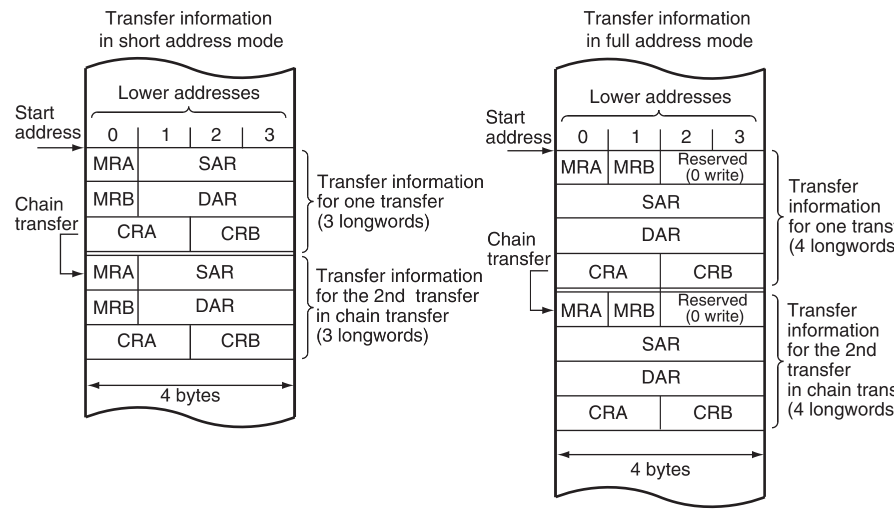


Figure 12.2 Transfer Information on Data Area

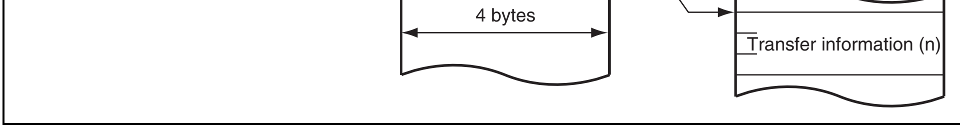


Figure 12.3 Correspondence between DTC Vector Address and Transfer Infor

	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_0	ADI0 (A/D_0 conversion end)	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	Lo

	TGI5B	111	H'5BC	DTCEC14
TMR_0	CMI0A	116	H'5D0	DTCEC13
	CMI0B	117	H'5D4	DTCEC12
TMR_1	CMI1A	119	H'5DC	DTCEC11
	CMI1B	120	H'5E0	DTCEC10
TMR_2	CMI2A	122	H'5E8	DTCEC9
	CMI2B	123	H'5EC	DTCEC8
TMR_3	CMI3A	125	H'5F4	DTCEC7
	CMI3B	126	H'5F8	DTCEC6
DMAC	DMTEND0	128	H'600	DTCEC5
	DMTEND1	129	H'604	DTCEC4
	DMTEND2	130	H'608	DTCEC3
	DMTEND3	131	H'60C	DTCEC2
EXDMAC	EXDMTEND0	132	H'610	DTCEC1
	EXDMTEND1	133	H'614	DTCEC0
	EXDMTEND2	134	H'618	DTCEC15
	EXDMTEND3	135	H'61C	DTCEC14
DMAC	DMEEND0	136	H'620	DTCED13
	DMEEND1	137	H'624	DTCED12
	DMEEND2	138	H'628	DTCED11
	DMEEND3	139	H'62C	DTCED10
EXDMAC	EXDMEEND0	140	H'630	DTCECD9
	EXDMEEND1	141	H'634	DTCECD8
	EXDMEEND2	142	H'638	DTCED7
	EXDMEEND3	143	H'63C	DTCED6 L

	TXI4	162	H'688	DTCEE12	
TPU_6	TGI6A	164	H'690	DTCEE11	
	TGI6B	165	H'694	DTCEE10	
	TGI6C	166	H'698	DTCEE9	
	TGI6D	167	H'69C	DTCEE8	
TPU_7	TGI7A	169	H'6A4	DTCEE7	
	TGI7B	170	H'6A8	DTCEE6	
TPU_8	TGI8A	173	H'6B4	DTCEE5	
	TGI8B	174	H'6B8	DTCEE4	
TPU_9	TGI9A	177	H'6C4	DTCEE3	
	TGI9B	178	H'6C8	DTCEE2	
	TGI9C	179	H'6CC	DTCEE1	
	TGI9D	180	H'6D0	DTCEE0	
TPU_10	TGI10A	182	H'6D8	DTCEF15	
	TGI10B	183	H'6DC	DTCEF14	
	TGI10V	186	H'6E8	DTCEF11	
TPU_11	TGI11A	188	H'6F0	DTCEF10	
	TGI11B	189	H'6F4	DTCEF9	Lo

Note: * The DTCE bits with no corresponding interrupt are reserved, and the write valu always be 0. To leave software standby mode or all-module-clock-stop mode w interrupt, write 0 to the corresponding DTCE bit.

Table 12.2 shows the DTC transfer modes.

Table 12.2 DTC Transfer Modes

Transfer Mode	Size of Data Transferred at One Transfer Request	Memory Address Increment or Decrement	T C
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 contin operation.

Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers w 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 12.4 shows a flowchart of DTC operation, and table 12.3 summarizes the chain tr conditions (combinations for performing the second and third transfers are omitted).

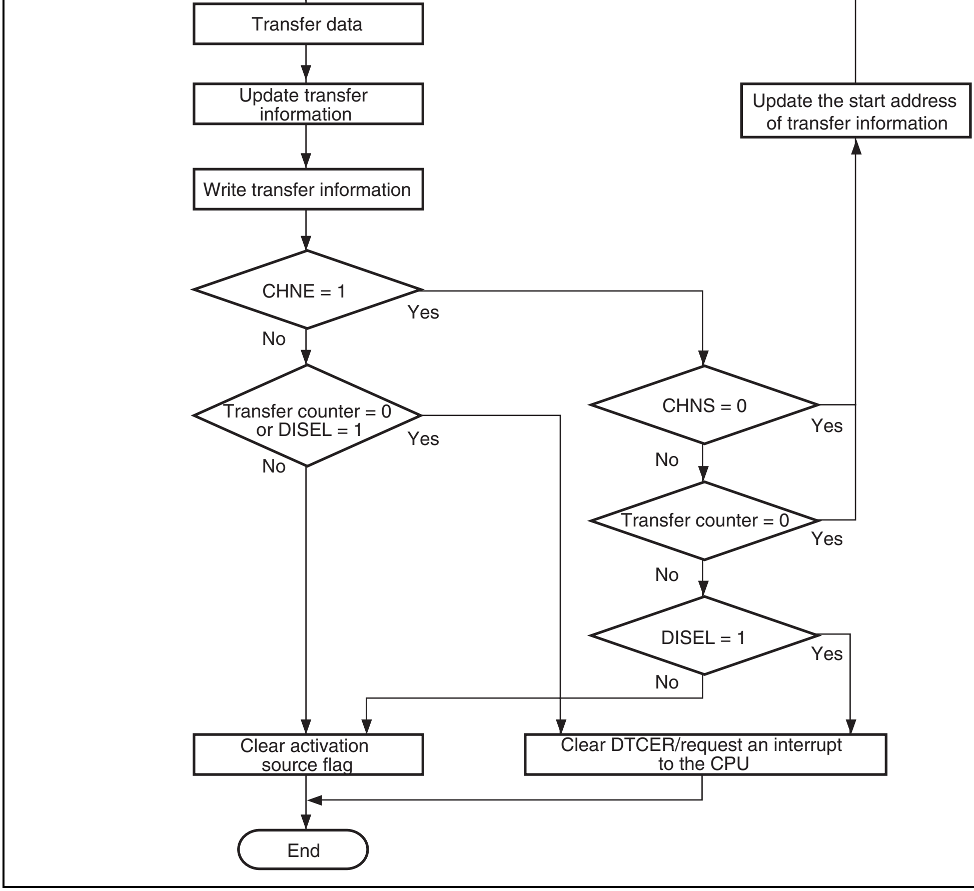


Figure 12.4 Flowchart of DTC Operation

1	1	0	Not 0	—	—		—	Ends at 1st tra
1	1	—	0*[2]	0	—	0	Not 0	Ends at 2nd tr
				0	—	0	0*[2]	Ends at 2nd tr
				0	—	1		Interrupt reque
1	1	1	Not 0	—	—	—	—	Ends at 1st tra Interrupt reque

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 moc

12.5.1 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.

Table 12.4 shows the relationship among, SAR, DAR, transfer data size, bus cycle divis access data size. Figure 12.5 shows the bus cycle division example.

Table 12.4 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)

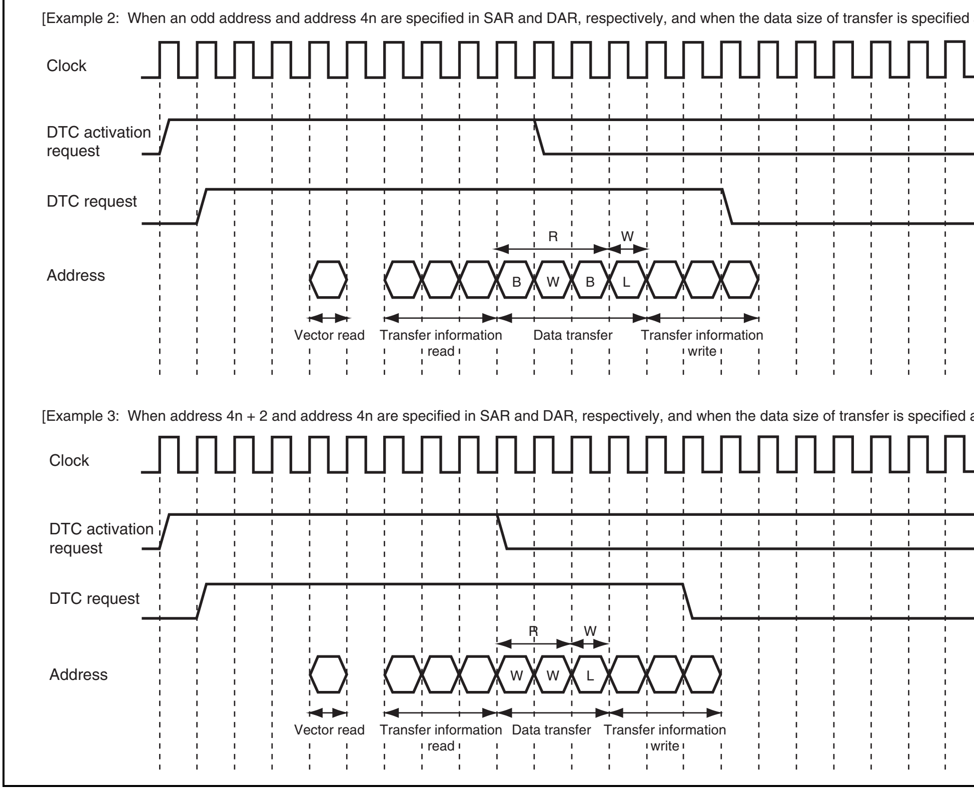


Figure 12.5 Bus Cycle Division Example

cleared to 0, the stored vector number is deleted, and the updated vector table and transf
information are read at the next activation.

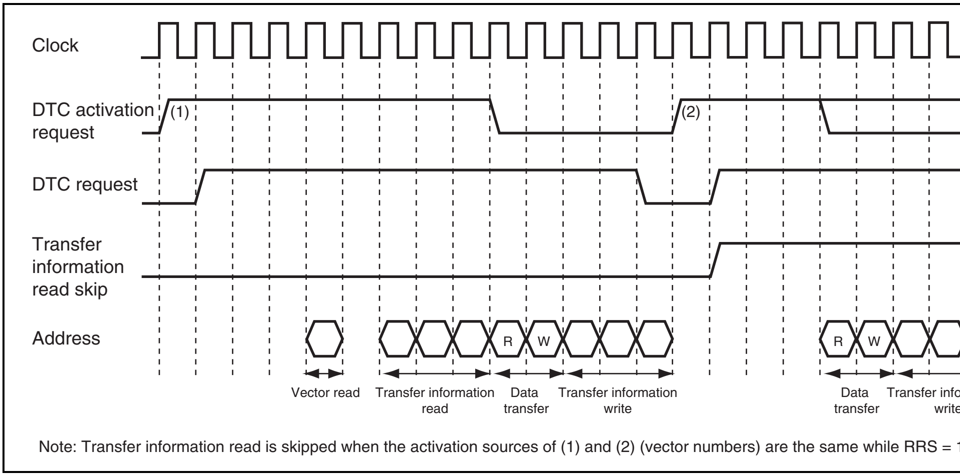


Figure 12.6 Transfer Information Read Skip Timing

SM1	DM1	SAR	DAR
0	0	Skipped	Skipped
0	1	Skipped	Written back
1	0	Written back	Skipped
1	1	Written back	Written back

12.5.4 Normal Transfer Mode

In normal transfer mode, one operation transfers one byte, one word, or one longword of From 1 to 65,536 transfers can be specified. The transfer source and destination addresse specified as incremented, decremented, or fixed. When the specified number of transfers interrupt can be requested to the CPU.

Table 12.6 lists the register function in normal transfer mode. Figure 12.7 shows the mem in normal transfer mode.

Table 12.6 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 12.7 Memory Map in Normal Transfer Mode

12.5.5 Repeat Transfer Mode

In repeat transfer mode, one operation transfers one byte, one word, or one longword of
the DTS bit in MRB, either the source or destination can be specified as a repeat area. F
256 transfers can be specified. When the specified number of transfers ends, the transfer
and address register specified as the repeat area is restored to the initial state, and transf
repeated. The other address register is then incremented, decremented, or left fixed. In r
transfer mode, the transfer counter (CRAL) is updated to the value specified in CRAH w
CRAL becomes H'00. Thus the transfer counter value does not reach H'00, and therefor
interrupt cannot be requested when DISEL = 0.

Table 12.7 lists the register function in repeat transfer mode. Figure 12.8 shows the mem
in repeat transfer mode.

CRAH	Transfer count storage	CRAH	CRAH
CRAL	Transfer count A	CRAL – 1	CRAH
CRB	Transfer count B	Not updated	Not updated

Note: * Transfer information writeback is skipped.

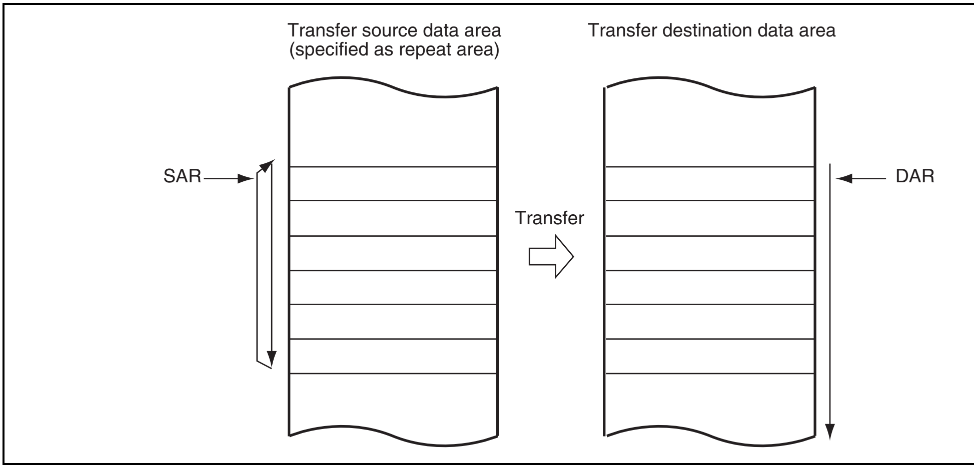


Figure 12.8 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

Table 12.8 lists the register function in block transfer mode. Figure 12.9 shows the mem
in block transfer mode.

Table 12.8 Register Function in Block Transfer Mode

Register	Function	Written Back Value
SAR	Source address	DTS =0: Incremented/decremented/fixed*
		DTS = 1: SAR initial value
DAR	Destination address	DTS = 0: DAR initial value
		DTS =1: Incremented/decremented/fixed*
CRAH	Block size storage	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB – 1

Note: * Transfer information writeback is skipped.

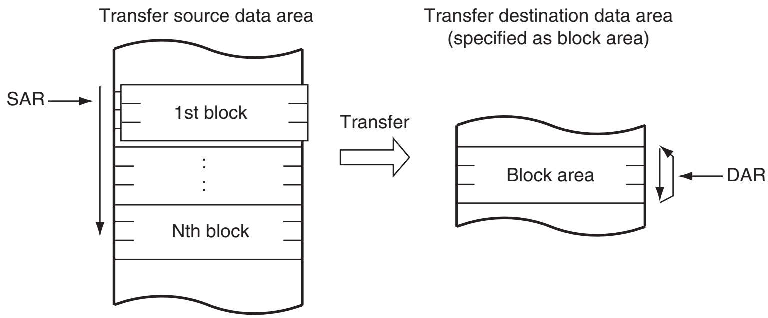


Figure 12.9 Memory Map in Block Transfer Mode
(When Transfer Destination is Specified as Block Area)

In repeat transfer mode, setting the RCHNE bit in DTCCR and the CHNE and CHNS bits to 1 enables a chain transfer after transfer with transfer counter = 1 has been completed.

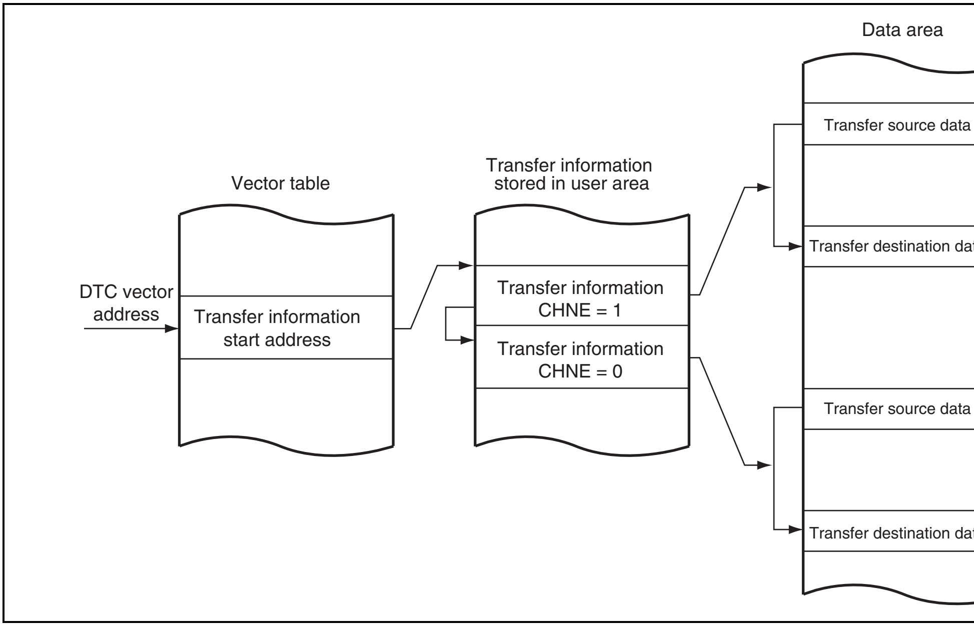


Figure 12.10 Operation of Chain Transfer

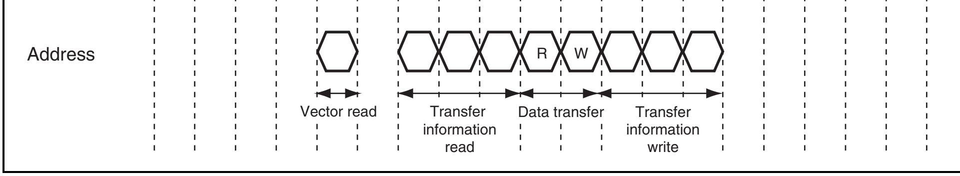


Figure 12.11 DTC Operation Timing
(Example of Short Address Mode in Normal Transfer Mode or Repeat Transfer

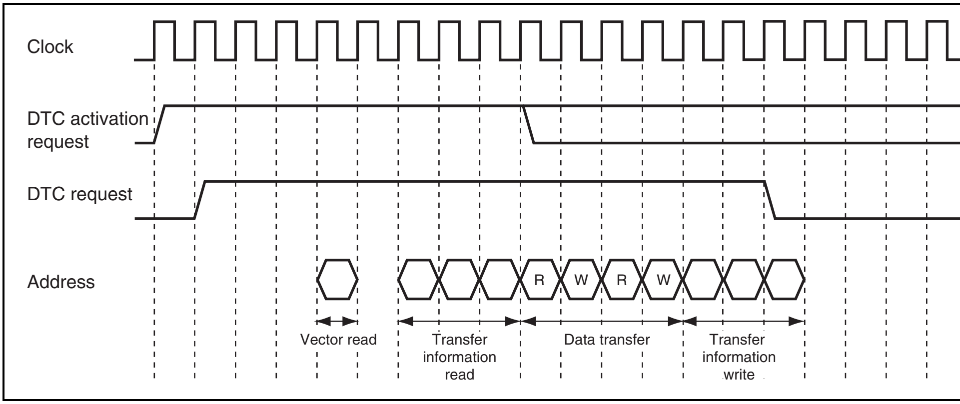


Figure 12.12 DTC Operation Timing
(Example of Short Address Mode in Block Transfer Mode with Block Size o

Figure 12.13 DTC Operation Timing (Example of Short Address Mode in Chain T

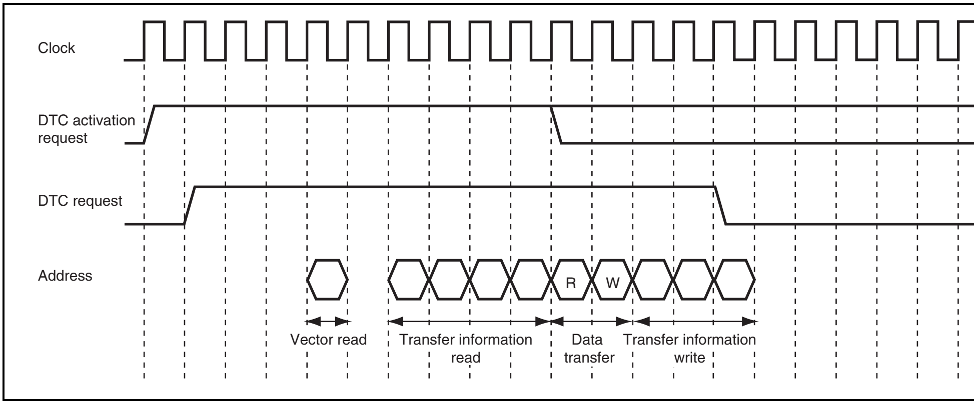


Figure 12.14 DTC Operation Timing
(Example of Full Address Mode in Normal Transfer Mode or Repeat Transfer M

Normal	1	0*	4*	3*	0*	3*	2*	1*	3*	2*	1	3*	2*	1
Repeat	1	0*[1]	4*[2]	3*[3]	0*[1]	3*[2.3]	2*[4]	1*[5]	3*[6]	2*[7]	1	3*[6]	2*[7]	1
Block transfer	1	0*[1]	4*[2]	3*[3]	0*[1]	3*[2.3]	2*[4]	1*[5]	3•P*[6]	2•P*[7]	1•P	3•P*[6]	2•P*[7]	1•P

[Legend]

P: Block size (CRAH and CRAL value)

Note: 1. When transfer information read is skipped
2. In full address mode operation
3. In short address mode operation
4. When the SAR or DAR is in fixed mode
5. When the SAR and DAR are in fixed mode
6. When a longword is transferred while an odd address is specified in the addre register
7. When a word is transferred while an odd address is specified in the address r when a longword is transferred while address 4n + 2 is specified

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 9, Bus Controller (BSC).)

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

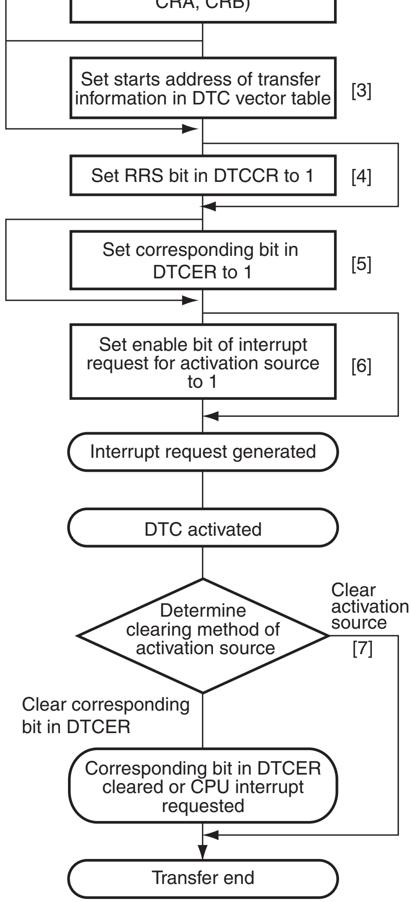
$$\text{Number of execution cycles} = I \bullet S_I + \Sigma\,(J \bullet S_J + K \bullet S_K + L \bullet S_L + M \bullet S_M) + N \bullet S$$

12.5.10 DTC Bus Release Timing

The DTC requests the bus mastership to the bus arbiter when an activation request occurs
DTC releases the bus after a vector read, transfer information read, a single data transfer,
transfer information writeback. The DTC does not release the bus during transfer informa
read, single data transfer, or transfer information writeback.

12.5.11 DTC Priority Level Control to the CPU

The priority of the DTC activation sources over the CPU can be controlled by the CPU pr
level specified by bits CPUP2 to CPUP0 in CPUPCR and the DTC priority level specifie
DTCP2 to DTCP0. For details, see section 7, Interrupt Controller.



information in the data area. For details on setting tra
information, see section 12.2, Register Descriptions. F
on location of transfer information, see section 12.4, L
Transfer Information and DTC Vector Table.

[3] Set the start address of the transfer information in the
vector table. For details on setting DTC vector table, s
12.4, Location of Transfer Information and DTC Vecto

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

[5] Set the bit in DTCER corresponding to the DTC activa
interrupt source to 1. For the correspondence of interr
DTCER, refer to table 12.1. The bit in DTCER may be
the second or later transfer. In this case, setting the bi
needed.

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

[7] After the end of one data transfer, the DTC clears the
source flag or clears the corresponding bit in DTCER
requests an interrupt to the CPU. The operation after t
depends on the transfer information. For details, see s
12.2, Register Descriptions and figure 12.4.

Figure 12.15 DTC with Interrupt Activation

the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any v

2. Set the start address of the transfer information for an RXI interrupt at the DTC vecto
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the end (RXI) interrupt. Since the generation of a receive error during the SCI reception c will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set RXI interrupt is generated, and the DTC is activated. The receive data is transferred fr to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Terminatio processing should be performed in the interrupt handling routine.

12.7.2 Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the Chain transfer can be used to perform pulse output data transfer and PPG output trigger c updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the cha transfer, and normal mode transfer to the TPU's TGR in the second half. This is because c of the activation source and interrupt generation at the end of the specified number of tra restricted to the second half of the chain transfer (transfer when CHNE = 0).

3. Locate the TPU transfer information consecutively after the NDR transfer informatio

4. Set the start address of the NDR transfer information to the DTC vector address.
5. Set the bit corresponding to the TGIA interrupt in DTCER to 1.
6. Set TGRA as an output compare register (output disabled) with TIOR, and enable th interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in D NDER for which output is to be performed to 1. Using PCR, select the TPU compar be used as the output trigger.
8. Set the CST bit in TSTR to 1, and start the TCNT count operation.
9. Each time a TGRA compare match occurs, the next output value is transferred to ND set value of the next output trigger period is transferred to TGRA. The activation sou flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is CPU. Termination processing should be performed in the interrupt handling routine.

12.7.3 Chain Transfer when Counter = 0

By executing a second data transfer and performing re-setting of the first data transfer o the counter value is 0, it is possible to perform 256 or more repeat transfers.

An example is shown in which a 128-Kbyte input buffer is configured. The input buffer assumed to have been set to start at lower address H'0000. Figure 12.16 shows the chain when the counter value is 0.

for the first data transfer reaches 0, the second data transfer is started. Set the upper ei
of the transfer source address for the first data transfer to H'21. The lower 16 bits of th
transfer destination address of the first data transfer and the transfer counter are H'000

5. Next, execute the first data transfer the 65536 times specified for the first data transfe means of interrupts. When the transfer counter for the first data transfer reaches 0, the data transfer is started. Set the upper eight bits of the transfer source address for the fi transfer to H'20. The lower 16 bits of the transfer destination address of the first data t and the transfer counter are H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data t no interrupt request is sent to the CPU.

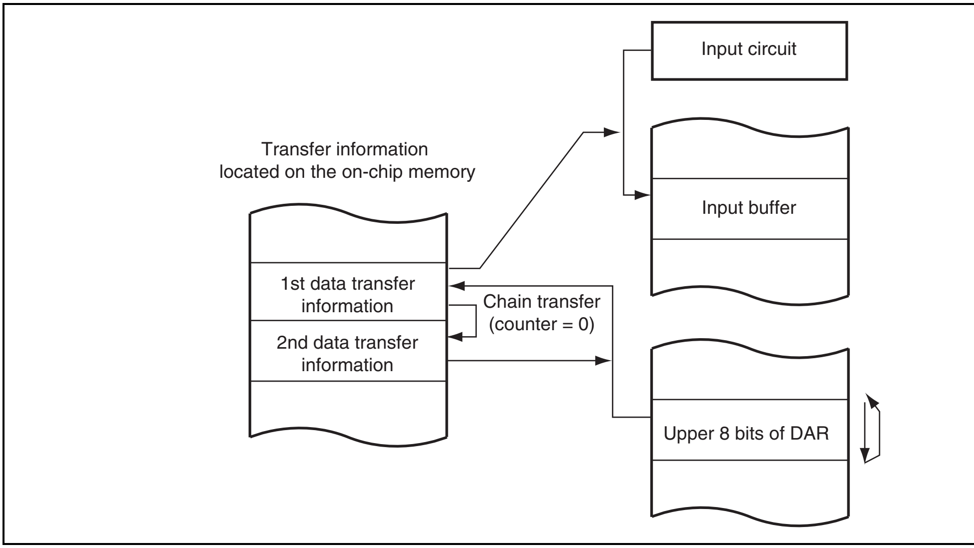


Figure 12.16 Chain Transfer when Counter = 0

Operation of the DTC can be disabled or enabled using the module stop control register. initial setting is for operation of the DTC to be enabled. Register access is disabled by se module stop state. The module stop state cannot be set while the DTC is activated. For d refer to section 27, Power-Down Modes.

12.9.2 On-Chip RAM

Transfer information can be located in on-chip RAM. In this case, the RAME bit in SYS not be cleared to 0.

12.9.3 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.

When the DTC is activated by a DMAC transfer end interrupt, even if DISEL=0, an auto clearing of the relevant activation source flag is not automatically cleared by the DTC. T write 1 to the DTE bit by the DTC transfer and clear the activation source flag to 0.

12.9.4 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.

12.9.6 Transfer Information Start Address, Source Address, and Destination Add

The transfer information start address to be specified in the vector table should be address address other than address 4n is specified, the lower 2 bits of the address are regarded as

The source and destination addresses specified in SAR and DAR, respectively, will be tra in the divided bus cycles depending on the address and data size.

12.9.7 Transfer Information Modification

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

12.9.8 Endian Format

The DTC supports big and little endian formats. The endian formats used when transfer information is written to and when transfer information is read from by the DTC must be same.

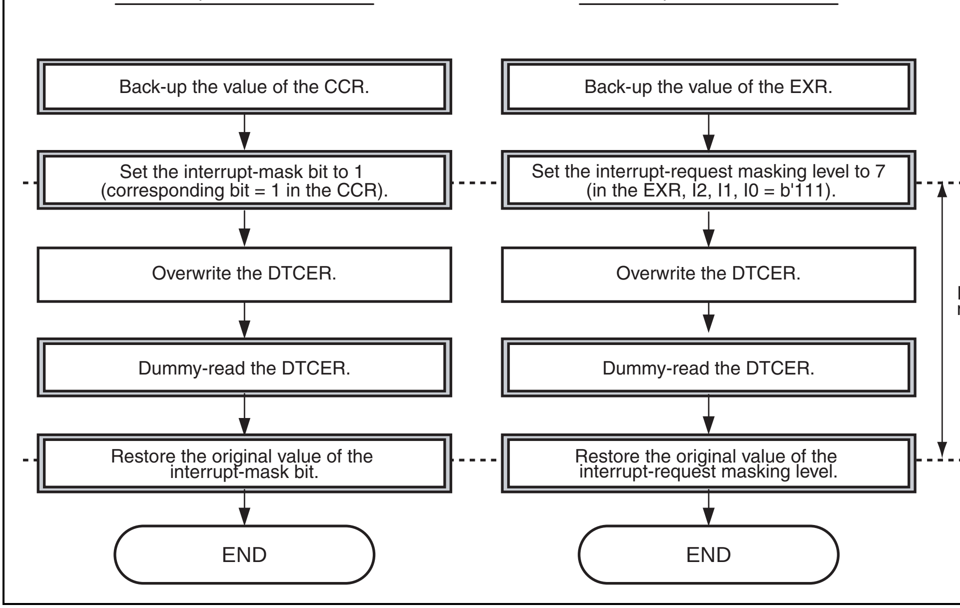


Figure 12.17 Example of Procedures for Overwriting the DTCER

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. Also, a
I/O ports can drive Darlington transistors when functioning as output ports.

Port 2, J, and K are Schmitt-trigger input. Schmitt-trigger inputs for other ports are enab
used as the IRQ, TPU, TMR, or IIC2 input.

Table 13.1 Port Functions

Port	Description	Bit	Function			Schmitt-Trigger Input*[1]	Input Pull-up MOS Function
			I/O	Input	Output		
Port 1	General I/O port also functioning as interrupt inputs, SCI I/Os, DMAC I/Os, EXDMAC I/Os, A/D converter inputs, TPU inputs, and IIC2 I/Os	7	P17/SCL0	IRQ7-A/ TCLKD/ ADTRG1	EDRAK1	IRQ7-A, TCLKD, SCL0	—
		6	P16/SDA0	IRQ6-A/ TCLKC	DACK1/ EDACK1-A	IRQ6-A, TCLKC, SDA0	
		5	P15/SCL1	IRQ5-A/ TCLKB/ RxD5/ IrRxD	TEND1/ ETEND1-A	IRQ5-A, TCLKB, SCL1	
		4	P14/SDA1	DREQ1/ IRQ4-A/ TCLKA/ EDREQ1-A	TxD5/ IrTxD	IRQ4-A, TCLKA, SDA1	
		3	P13	ADTRG0/ IRQ3-A	EDRAK0	IRQ3-A	

	IIC2 I/Os						
Port 2	General I/O port also functioning as interrupt inputs, PPG outputs, TPU I/Os, TMR I/Os, and SCI I/Os	7	P27/ TIOCB5	TIOCA5	PO7	P27, TIOCB5, TIOCA5	—
		6	P26/ TIOCA5	—	PO6/TMO1/ TxD1	All input functions	
		5	P25/ TIOCA4	TMCI1/ RxD1	PO5	P25, TIOCA4, TMCI1	
		4	P24/ TIOCB4/ SCK1	TIOCA4/ TMRI1	PO4	P24, TIOCB4, TIOCA4, TMRI1	
		3	P23/ TIOCD3	$\overline{\text{IRQ11}}$-A/ TIOCC3	PO3	P23, TIOCD3, $\overline{\text{IRQ11}}$-A	
		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	

				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	

		3	P63	TMRI3/ $\overline{\text{DREQ3}}$/ $\overline{\text{IRQ11}}$-B/ TMS/ $\overline{\text{EDREQ1}}$-B	—	TMRI3, $\overline{\text{IRQ11}}$-B, TMS	
		2	P62/SCK4	$\overline{\text{IRQ10}}$-B/ $\overline{\text{TRST}}$	TMO2/ $\overline{\text{DACK2}}$/ $\overline{\text{EDACK0}}$-B	$\overline{\text{IRQ10}}$-B, $\overline{\text{TRST}}$	
		1	P61	TMCI2/ RxD4/ $\overline{\text{IRQ9}}$-B	$\overline{\text{TEND2}}$/ $\overline{\text{ETEND0}}$-B	TMCI2, $\overline{\text{IRQ9}}$-B	
		0	P60	TMRI2/ $\overline{\text{DREQ2}}$/ $\overline{\text{IRQ8}}$-B/ $\overline{\text{EDREQ0}}$-B	TxD4	TMRI2, $\overline{\text{IRQ8}}$-B	
Port A	General I/O port also functioning as system clock output and bus control I/Os	7	—	PA7	Bφ	—	—
		6	PA6	—	$\overline{\text{AS}}$/$\overline{\text{AH}}$/ $\overline{\text{BS}}$-B		
		5	PA5	—	$\overline{\text{RD}}$		
		4	PA4	—	$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$		
		3	PA3	—	$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$		
		2	PA2	$\overline{\text{BREQ}}$/ $\overline{\text{WAIT}}$	—		
		1	PA1	—	$\overline{\text{BACK}}$/ (RD/$\overline{\text{WR}}$)		
		0	PA0	—	$\overline{\text{BREQO}}$/ $\overline{\text{BS}}$-A		

					$\overline{CS7}$-B		
		0	PB0	—	$\overline{CS0}$/ $\overline{CS4}$/ $\overline{CS5}$-B		
Port D *3	General I/O port also functioning as address outputs	7	PD7	—	A7	—	O
		6	PD6	—	A6		
		5	PD5	—	A5		
		4	PD4	—	A4		
		3	PD3	—	A3		
		2	PD2	—	A2		
		1	PD1	—	A1		
		0	PD0	—	A0		
Port E *3	General I/O port also functioning as address outputs	7	PE7	—	A15	—	O
		6	PE6	—	A14		
		5	PE5	—	A13		
		4	PE4	—	A12		
		3	PE3	—	A11		
		2	PE2	—	A10		
		1	PE1	—	A9		
		0	PE0	—	A8		

	also functioning as bi-directional data bus	6	PH6/D6*2	—	—		
		5	PH5/D5*2	—	—		
		4	PH4/D4*2	—	—		
		3	PH3/D3*2	—	—		
		2	PH2/D2*2	—	—		
		1	PH1/D1*2	—	—		
		0	PH0/D0*2	—	—		
Port I	General I/O port also functioning as bi-directional data bus	7	PI7/D15*2	—	—	—	O
		6	PI6/D14*2	—	—		
		5	PI5/D13*2	—	—		
		4	PI4/D12*2	—	—		
		3	PI3/D11*2	—	—		
		2	PI2/D10*2	—	—		
		1	PI1/D9*2	—	—		
		0	PI0/D8*2	—	—		
Port J*4	General I/O port also functioning PPG I/Os and TPU I/Os	7	PJ7/TIOCB8	TIOCA8/TCLKH	PO23	All input functions	O
		6	PJ6/TIOCA8	—	PO22		
		5	PJ5/TIOCB7	TIOCA7/TCLKG	PO21		
		4	PJ4/TIOCA7	—	PO20		
		3	PJ3/TIOCD6	TIOCC6/TCLKF	PO19		
		2	PJ2/TIOCC6	TCLKE	PO18		
		1	PJ1/TIOCB6	TIOCA6	PO17		
		0	PJ0/TIOCA6	—	PO16		

		1	PK1/TIOCB9	TIOCA9	PO25		
		0	PK0/TIOCA9	—	PO24		
Port M	General I/O port also functioning as SCI I/Os	7	—	—	—	—	—
		6	—	—	—		
		5	—	—	—		
		4	PM4	—	—		
		3	PM3	—	—		
		2	PM2	—	—		
		1	PM1	RxD6	—		
		0	PM0	—	TxD6		

Notes: 1. Pins without Schmitt-trigger input have CMOS input functions.

2. Addresses are also output when accessing to the address/data multiplexed I/

3. Pins are disabled when PCJKE = 1.

4. Pins are disabled when PCJKE = 0.

Port 5	8	—	—	O	O	—	—
Port 6*[3]	6	O	O	O	O	—	—
Port A	8	O	O	O	O	—	—
Port B*[4]	4	O	O	O	O	—	—
Port D*[1]	8	O	O	O	O	O	—
Port E*[1]	8	O	O	O	O	O	—
Port F*[5]	5	O	O	O	O	O	O
Port H	8	O	O	O	O	O	—
Port I	8	O	O	O	O	O	—
Port J*[2]	8	O	O	O	O	O	—
Port K*[2]	8	O	O	O	O	O	—
Port M*[6]	5	O	O	O	O	—	—

[Legend]

O: Register exists

—: No register exists

Notes: 1. Do not access port D or E registers when PCJKE = 1.

2. Do not access port J or K registers when PCJKE = 0.

3. The lower six bits are valid and the upper two bits are reserved for port 6 regis write value should be the same as the initial value.

4. The lower four bits are valid and the upper four bits are reserved for port B reg The write value should be the same as the initial value.

5. The lower five bits are valid and the upper three bits are reserved for port F reg The write value should be the same as the initial value.

6. The lower five bits are valid and the upper three bits are reserved for port M re The write value should be the same as 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.
The lower five bits are valid and the upper three bits are reserved for port F registers.
The lower five bits are valid and the upper three bits are reserved for port M registers.
Do not access port J or K registers when PCJKE = 0.
Do not access port D or E registers when PCJKE = 1.

Table 13.3 Startup Mode and Initial Value

	Startup Mode	
Port	**External Extended Mode**	**Single-Chip Mode**
Port A	H'80	H'00
Other ports	H'00	H'00

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.
The lower five bits are valid and the upper three bits are reserved for port F registers.
The lower five bits are valid and the upper three bits are reserved for port M registers.
Do not access port J or K registers when PCJKE = 0.
Do not access port D or E registers when PCJKE = 1.

13.1.3 Port Register (PORTn) (n = 1, 2, 5, 6, A, B, D to F, H to K, and M)

PORT is an 8-bit read-only register that reflects the port pin state. A write to PORT is inv
When PORT is read, the DR bits that correspond to the respective DDR bits set to 1 are re
the status of each pin whose corresponding DDR bit is cleared to 0 is also read regardless
ICR value.

The initial value of PORT is undefined and is determined based on the port pin state.

Bit	7	6	5	4	3	2	1
Bit Name	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
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.
The lower five bits are valid and the upper three bits are reserved for port F registers.
The lower five bits are valid and the upper three bits are reserved for port M registers.
Do not access port J or K registers when PCJKE = 0.
Do not access port D or E registers when PCJKE = 1.

When PORT is read, the pin state is always read regardless of the ICR value. When the I
is cleared to 0 at this time, the read pin state is not reflected in a corresponding on-chip p
module.

If ICR is modified, an internal edge may occur depending on the pin state. Accordingly,
should be modified when the corresponding input pins are not used. For example, an IR
modify ICR while the corresponding interrupt is disabled, clear the IRQF flag in ISR of
interrupt controller to 0, and then enable the corresponding interrupt. If an edge occurs a
ICR setting, the edge should be cancelled.

The initial value of ICR is H'00.

Bit	7	6	5	4	3	2	1
Bit Name	Pn7ICR	Pn6ICR	Pn5ICR	Pn4ICR	Pn3ICR	Pn2ICR	Pn1ICR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower four bits are valid and the upper four bits are reserved for port B registers.
The lower five bits are valid and the upper three bits are reserved for port F registers.
The lower five bits are valid and the upper three bits are reserved for port M registers.
Do not access port J or K registers when PCJKE = 0.
Do not access port D or E registers when PCJKE = 1.

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

Note: The lower five bits are valid and the upper three bits are reserved for port F registers.

Table 13.4 Input Pull-Up MOS State

Port	Pin State	Reset	Hardware Standby Mode	Software Standby Mode	Ot Op
Port D	Address output			OFF	
	Port output			OFF	
	Port input		OFF	ON/OFF	
Port E	Address output			OFF	
	Port output			OFF	
	Port input		OFF	ON/OFF	
Port F	Address output			OFF	
	Port output			OFF	
	Port input		OFF	ON/OFF	
Port H	Data input/output			OFF	
	Port output			OFF	
	Port input		OFF	ON/OFF	
Port I	Data input/output			OFF	
	Port output			OFF	
	Port input		OFF	ON/OFF	

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 inpu
MOS is off.

13.1.6 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 NM
drain output. If a bit in ODR is cleared to 0, the pin corresponding to that bit in ODR fu
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

For a pin whose initial value changes according to the activation mode, "initial value E" i
the initial value when the LSI is started up in external extended mode and "initial value S
indicates the initial value when the LSI is started in single-chip mode.

13.2.1 Port 1

(1) P17/$\overline{\text{IRQ7}}$-A/TCLKD/SCL0/$\overline{\text{EDRAK1}}$/$\overline{\text{ADTRG1}}$

The pin function is switched as shown below according to the combination of the EXDM
IIC2 register settings and P17DDR bit setting.

		Setting		
		EXDMAC	IIC2	I/O Port
Module Name	Pin Function	$\overline{\text{EDRAK1}}$_OE	SCL0_OE	P17DDR
EXDMAC	$\overline{\text{EDRAK1}}$ output	1	—	—
IIC2	SCL0 input/output	0	1	—
I/O port	P17 output	0	0	1
	P17 input (initial value)	0	0	0

I/O port	P16 output	0	0	0	1
	P16 input (initial value)	0	0	0	0

(3) P15/RxD5/IrRxD/$\overline{\text{TEND1}}$/$\overline{\text{ETEND1}}$-A/$\overline{\text{IRQ5}}$-A/TCLKB/SCL1

The pin function is switched as shown below according to the combination of the EXDM
DMAC and IIC2 register settings and P15DDR bit setting.

		Setting			
		EXDMAC	DMAC	IIC2	I/
Module Name	Pin Function	$\overline{\text{ETEND1A}}$_OE	$\overline{\text{TEND1}}$_OE	SCL1_OE	P
EXDMAC	$\overline{\text{ETEND1}}$-A output	1	—	—	—
DMAC	$\overline{\text{TEND1}}$ output	0	1	—	—
IIC2	SCL1 input/output	0	0	1	—
I/O port	P15 output	0	0	0	1
	P15 input (initial value)	0	0	0	0

I/O port	P14 output	0	0	0	1
	P14 input (initial value)	0	0	0	0

(5) P13/$\overline{\text{ADTRG0}}$/$\overline{\text{IRQ3}}$-A/$\overline{\text{EDRAK0}}$

The pin function is switched as shown below according to the register setting of EXDMA
the P13DDR bit setting.

		Setting	
		EXDMAC	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{EDRAK0}}$_OE**	**P13DDR**
I/O port	$\overline{\text{EDRAK0}}$ output	1	—
	P13 output	0	1
	P13 input (initial value)	0	0

SCI	SCK2 output	0	0	1	—
I/O port	P12 output	0	0	0	1
	P12 input (initial value)	0	0	0	0

(7) P11/RxD2/$\overline{\text{TEND0}}$/$\overline{\text{IRQ1}}$-A/$\overline{\text{ETEND0}}$-A

The pin function is switched as shown below according to the combination of the EXDM
DMAC register settings and P11DDR bit setting.

		Setting		
		EXDMAC	**DMAC**	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{ETEND0A\_OE}}$**	**$\overline{\text{TEND0\_OE}}$**	**P11DDR**
EXDMAC	$\overline{\text{ETEND0}}$-A output	1	—	—
DMAC	$\overline{\text{TEND0}}$ output	0	1	—
I/O port	P11 output	0	0	1
	P11 input (initial value)	0	0	0

	P10 input (initial value)	0	0

13.2.2 Port 2

(1) P27/PO7/TIOCA5/TIOCB5

The pin function is switched as shown below according to the combination of the TPU an 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 value)	0	0	0

SCI	TxD1 output	0	0	1	—	-
PPG	PO6 output	0	0	0	1	-
I/O port	P26 output	0	0	0	0	
	P26 input (initial value)	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 value)	0	0	0

PPG	PO4 output	0	0	1	—
I/O port	P24 output	0	0	0	1
	P24 input (initial value)	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 ar
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 value)	0	0	0

SCI	TxD0 output	0	0	1	—
PPG	PO2 output	0	0	0	1
I/O port	P22 output	0	0	0	0
	P22 input (initial value)	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 value)	0	0	0

PPG	PO0 output	0	0	1	—
I/O port	P20 output	0	0	0	1
	P20 input (initial value)	0	0	0	0

13.2.3 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

EXDMAC	$\overline{EDACK1}$-B output	Except for boundary scan enabled mode*	1	—	—
DMAC	$\overline{DACK3}$ output		0	1	—
TMR	TMO3 output		0	0	1
I/O port	P65 output		0	0	0
	P65 input (initial value)		0	0	0

Note: * These pins are boundary scan dedicated input pins during boundary scan ena
mode.

(2) P64/TMCI3/$\overline{TEND3}$/$\overline{ETEND1}$-B/TDI

The pin function is switched as shown below according to the combination of the EXDM
DMAC register settings and P64DDR bit setting.

		Setting			
		MCU Operating Mode	EXDMAC	DMAC	I/(
Module Name	Pin Function		$\overline{ETEND1B}$_OE	$\overline{TEND3}$_OE	P6
EXDMAC	$\overline{ETEND1}$-B output	Except for boundary scan enabled mode*	1	—	—
DMAC	$\overline{TEND3}$ output		0	1	—
I/O port	P64 output		0	0	1
	P64 input (initial value)		0	0	0

Note: * These pins are boundary scan dedicated input pins during boundary scan ena
mode.

Note: * These pins are boundary scan dedicated input pins during boundary scan ena
mode.

(4) P62/TMO2/SCK4/$\overline{\text{DACK2}}$/$\overline{\text{EDACK0}}$-B/$\overline{\text{IRQ10}}$-B/$\overline{\text{TRST}}$

The pin function is switched as shown below according to the combination of the EXDM
DMAC, TMR, and SCI register settings and P62DDR bit setting.

		Setting				
		MCU Operating Mode	EXDMAC	DMAC	TMR	SCI
Module Name	Pin Function		$\overline{\text{EDACK0B}}$_OE	$\overline{\text{DACK2}}$_OE	TMO2_OE	SCK4_OE
EXDMAC	$\overline{\text{EDACK0}}$-B output	Except for boundary scan enabled mode*	1	—	—	—
DMAC	$\overline{\text{DACK2}}$ output		0	1	—	—
TMR	TMO2 output		0	0	1	—
SCI	SCK4 output		0	0	0	1
I/O port	P62 output		0	0	0	0
	P62 input (initial value)		0	0	0	0

Note: * These pins are boundary scan dedicated input pins during boundary scan enal
mode.

I/O port	P61 output	0	0	1
	P61 input (initial value)	0	0	0

(6) P60/TMRI2/TxD4/$\overline{DREQ2}$/$\overline{EDREQ0}$-B/$\overline{IRQ8}$-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 value)	0	0

	PA7 input (initial value S)	0

[Legend]

Initial value E: Initial value in external extended mode

Initial value S: Initial value in single-chip mode

(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 operating r
the EXPE bit, the bus controller register, the port function control register (PFCR), and th
PA6DDR bit settings.

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

[Legend]

Initial value E: Initial value in external extended mode

Initial value S: Initial value in single-chip mode

Note: * Valid in external extended mode (EXPE = 1)

	PA5 input (initial value S)	0	0

[Legend]

Initial value E: Initial value in external extended mode

Initial value S: Initial value in single-chip mode

Note: * Valid in external extended mode (EXPE = 1)

(4) PA4/$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$

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

		Setting		
		Bus Controller		I/O Port
Module Name	Pin Function	$\overline{\text{LUB}}$_OE*2	$\overline{\text{LHWR}}$_OE*2	PA4DDR
Bus controller	$\overline{\text{LUB}}$ output*1	1	—	—
	$\overline{\text{LHWR}}$ output*1 (initial value E)	—	1	—
I/O port	PA4 output	0	0	1
	PA4 input (initial value S)	0	0	0

[Legend]

Initial value E: Initial value in external extended mode

Initial value S: Initial value in single-chip mode

Notes: 1. Valid in external extended mode (EXPE = 1)

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

I/O port	PA3 output	0	0	1
	PA3 input (initial value S)	0	0	0

[Legend]

Initial value E: Initial value in external extended mode

Initial value S: Initial value in single-chip mode

Notes: 1. Valid in external extended mode (EXPE = 1)

2. If the byte control SRAM space is accessed, this pin functions as the $\overline{\text{LLB}}$ outp otherwise, the $\overline{\text{LLWR}}$.

(6) PA2/$\overline{\text{BREQ}}$/$\overline{\text{WAIT}}$

The pin function is switched as shown below according to the combination of the bus con register settings and the PA2DDR bit setting.

		Setting		
		Bus Controller		I/O Port
Module Name	Pin Function	BCR_BRLE	BCR_WAITE	PA2DDR
Bus controller	$\overline{\text{BREQ}}$ input	1	—	—
	$\overline{\text{WAIT}}$ input	0	1	—
I/O port	PA2 output	0	0	1
	PA2 input (initial value)	0	0	0

Bus controller	$\overline{\text{BACK}}$ output *	1	—	—	—
	RD/$\overline{\text{WR}}$ output *	0	1	—	—
		0	0	1	—
I/O port	PA1 output	0	0	0	1
	PA1 input (initial value)	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(8) PA0/$\overline{\text{BREQO}}$/$\overline{\text{BS}}$-A

The pin function is switched as shown below according to the combination of operating
the EXPE bit, the bus controller register, the port function control register (PFCR), and t
PA0DDR bit settings.

		Setting		
		I/O Port	Bus Controller	I/O Port
Module Name	**Pin Function**	**$\overline{\text{BS}}$-A_OE**	**$\overline{\text{BREQO}}$_OE**	**PA0DD**
Bus controller	$\overline{\text{BS}}$-A output*	1	—	—
	$\overline{\text{BREQO}}$ output*	0	1	—
I/O port	PA0 output	0	0	1
	PA0 input (initial value)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

Bus controller	$\overline{\text{CS3}}$ output*	1	—	—
	$\overline{\text{CS7}}$-A output*	—	1	—
I/O port	PB3 output	0	0	1
	PB3 input (initial value)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(2) PB2/$\overline{\text{CS2}}$-A/$\overline{\text{CS6}}$-A

The pin function is switched as shown below according to the combination of operating
the EXPE bit, the port function control register (PFCR), and the PB2DDR bit settings.

		Setting		
		I/O Port		
Module Name	**Pin Function**	**$\overline{\text{CS2A}}$_OE**	**$\overline{\text{CS6A}}$_OE**	**PB2DDR**
Bus controller	$\overline{\text{CS2}}$-A output*	1	—	—
	$\overline{\text{CS6}}$-A output*	—	1	—
I/O port	PB2 output	0	0	1
	PB2 input (initial value)	0	0	0

Note: * Valid in external extended mode (EXPE = 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 value)	0	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(4) PB0/$\overline{\text{CS0}}$/$\overline{\text{CS4}}$/$\overline{\text{CS5}}$-B

The pin function is switched as shown below according to the combination of operating
the EXPE bit, the bus controller register, the port function control register (PFCR), and t
PB0DDR bit settings.

		Setting			
		I/O Port			
Module Name	**Pin Function**	**$\overline{\text{CS0}}$_OE**	**$\overline{\text{CS4}}$_OE**	**$\overline{\text{CS5B}}$_OE**	**PB**
Bus controller	$\overline{\text{CS0}}$ output (initial value E)	1	—	—	—
	$\overline{\text{CS4}}$ output	—	1	—	—
	$\overline{\text{CS5}}$-B output	—	—	1	—
I/O port	PB0 output	0	0	0	1
	PB0 input (initial value S)	0	0	0	0

[Legend]
Initial value E: Initial value in on-chip ROM disabled external mode
Initial value S: Initial value in other modes
Note: * Valid in external extended mode (EXPE = 1)

EXPE bit, and the PDnDDR bit settings.

Module Name	Pin Function	MCU Operating Mode	I/O Port PDnDDF
		Setting	
Bus controller	Address output	On-chip ROM disabled extended mode	—
		On-chip ROM enabled extended mode	1
I/O port	PDn output	Single-chip mode*	1
	PDn input (initial value)	Modes other than on-chip ROM disabled extended mode	0

[Legend]

n: 0 to 7

Note: * Address output is enabled by setting PDnDDR = 1 in external extended mode (EXPE = 1)

EXPE bit, and the PEnDDR bit settings.

		Setting	
			I/O Por
Module Name	Pin Function	MCU Operating Mode	PEnDD
Bus controller	Address output	On-chip ROM disabled extended mode	—
		On-chip ROM enabled extended mode	1
I/O port	PEn output	Single-chip mode*	1
	PEn input (initial value)	Modes other than on-chip ROM disabled extended mode	0

[Legend]

n: 0 to 7

Note: * Address output is enabled by setting PDnDDR = 1 in external extended mode (EXPE = 1)

On-chip ROM disabled extended mode	Bus controller	A20 output	—	—
Modes other than on-chip ROM disabled extended mode	Bus controller	A20 output*	1	—
	I/O port	PF4 output	0	1
		PF4 input (initial value)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(2) PF3/A19

The pin function is switched as shown below according to the combination of operating m
EXPE bit, the port function control register (PFCR), and the PF3DDR bit settings.

MCU Operating Mode	Module Name	Pin Function	Setting I/O Port A19_OE	 I/O Port PF3DDR
On-chip ROM disabled extended mode	Bus controller	A19 output	—	—
Modes other than on-chip ROM disabled extended mode	Bus controller	A19 output*	1	—
	I/O port	PF3 output	0	1
		PF3 input (initial value)	0	0

Note: * Valid in external extended mode (EXPE = 1)

Modes other than on-chip ROM disabled extended mode	Bus controller	A18 output*	1	—
	I/O port	PF2 output	0	1
		PF2 input (initial value)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(4) PF1/A17

The pin function is switched as shown below according to the combination of operating
EXPE bit, the port function control register (PFCR), and the PF1DDR bit settings.

			Setting	
			I/O Port	I/O Port
MCU Operating Mode	Module Name	Pin Function	A17_OE	PF1DDR
On-chip ROM disabled extended mode	Bus controller	A17 output	—	—
Modes other than on-chip ROM disabled extended mode	Bus controller	A17 output*	1	—
	I/O port	PF1 output	0	1
		PF1 input (initial value)	0	0

Note: * Valid in external extended mode (EXPE = 1)

Modes other than on-chip ROM disabled extended mode	Bus controller	A16 output*	1	—
	I/O port	PF0 output	0	1
		PF0 input (initial value)	0	0

Note: * Valid in external extended mode (EXPE = 1)

13.2.10 Port H

(1) PH7/D7, PH6/D6, PH5/D5, PH4/D4, PH3/D3, PH2/D2, PH1/D1, PH0/D0

The pin function is switched as shown below according to the combination of operating r
EXPE bit, and the PHnDDR bit settings.

		Setting	
		MCU Operating Mode	I/O Port
Module Name	Pin Function	EXPE	PHnDDR
Bus controller	Data I/O* (initial value E)	1	—
I/O port	PHn output	0	1
	PHn input (initial value S)	0	0

[Legend]

Initial value E: Initial value in external extended mode

Initial value S: Initial value in single-chip mode

n: 0 to 7

Note: * Valid in external extended mode (EXPE = 1)

Bus controller	Data I/O* (initial value E)	1	—
I/O port	PIn output	0	1
	PIn input (initial value S)	0	0

[Legend]

Initial value E: Initial value in external extended mode

Initial value S: Initial value in single-chip mode

n: 0 to 7

Note: * Valid in external extended mode (EXPE = 1)

PPG and TPU, setting of the port function control register (PFCR), and the PJ7DDR bit s

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO23_OE	TIOCB8_OE	PJ7DDR
PPG	PO23 output*	1	—	—
TPU	TIOCB8 output*	0	1	—
I/O port	PJ7 output*	0	0	1
	PJ7 input*	0	0	0

Note: * Valid when PCJKE = 1.

(2) PJ6/TIOCA8/PO22

The pin function is switched as shown below according to the combination of register set
PPG and TPU, setting of the port function control register (PFCR), and the PJ6DDR bit s

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO22_OE	TIOCA8_OE	PJ6DDR
PPG	PO22 output*	1	—	—
TPU	TIOCA8 output*	0	1	—
I/O port	PJ6 output*	0	0	1
	PJ6 input*	0	0	0

Note: * Valid when PCJKE = 1.

I/O port	PJ5 output*	0	0	1
	PJ5 input*	0	0	0

Note: * Valid when PCJKE = 1.

(4) PJ4/TIOCA7/PO20

The pin function is switched as shown below according to the combination of register se
PPG and TPU, setting of the port function control register (PFCR), and the PJ4DDR bit

		Setting		
		PPG	**TPU**	**I/O Port**
Module Name	**Pin Function**	**PO20_OE**	**TIOCA7_OE**	**PJ4DDF**
PPG	PO20 output*	1	—	—
TPU	TIOCA7 output*	0	1	—
I/O port	PJ4 output*	0	0	1
	PJ4 input*	0	0	0

Note: * Valid when PCJKE = 1.

I/O port	PJ3 output*	0	0	1
	PJ3 input*	0	0	0

Note: * Valid when PCJKE = 1.

(6) PJ2/PO18/TIOCC6/TCLKE

The pin function is switched as shown below according to the combination of register set
PPG and TPU, setting of the port function control register (PFCR), and the PJ2DDR bit s

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO18_OE	TIOCC6_OE	PJ2DDR
PPG	PO18 output*	1	—	—
TPU	TIOCC6 output*	0	1	—
I/O port	PJ2 output*	0	0	1
	PJ2 input*	0	0	0

Note: * Valid when PCJKE = 1.

I/O port	PJ1 output*	0	0	1
	PJ1 input*	0	0	0

Note: * Valid when PCJKE = 1.

(8) PJ0/PO16/TIOCA6

The pin function is switched as shown below according to the combination of register se
PPG and TPU, setting of the port function control register (PFCR), and the PJ0DDR bit

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO16_OE	TIOCA6_OE	PJ0DDR
PPG	PO16 output*	1	—	—
TPU	TIOCA6 output*	0	1	—
I/O port	PJ0 output*	0	0	1
	PJ0 input*	0	0	0

Note: * Valid when PCJKE = 1.

PPG and TPU, setting of the port function control register (PFCR), and the PK7DDR bit

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO31_OE	TIOCB11_OE	PK7DDR
PPG	PO31 output*	1	—	—
TPU	TIOCB11 output*	0	1	—
I/O port	PK7 output*	0	0	1
	PK7 input*	0	0	0

Note: * Valid when PCJKE = 1.

(2) PK6/PO30/TIOCA11

The pin function is switched as shown below according to the combination of register set
PPG and TPU, setting of the port function control register (PFCR), and the PK6DDR bit

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO30_OE	TIOCA11_OE	PK6DDR
PPG	PO30 output*	1	—	—
TPU	TIOCA11 output*	0	1	—
I/O port	PK6 output*	0	0	1
	PK6 input*	0	0	0

Note: * Valid when PCJKE = 1.

I/O port	PK5 output*	0	0	1
	PK5 input*	0	0	0

Note: * Valid when PCJKE = 1.

(4) PK4/PO28/TIOCA10

The pin function is switched as shown below according to the combination of register se
PPG and TPU, setting of the port function control register (PFCR), and the PK4DDR bi

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO28_OE	TIOCA10_OE	PK4DDI
PPG	PO28 output*	1	—	—
TPU	TIOCA10 output*	0	1	—
I/O port	PK4 output*	0	0	1
	PK4 input*	0	0	0

Note: * Valid when PCJKE = 1.

I/O port	PK3 output*	0	0	1
	PK3 input*	0	0	0

Note: * Valid when PCJKE = 1.

(6) PK2/PO26/TIOCC9

The pin function is switched as shown below according to the combination of register set
PPG and TPU, setting of the port function control register (PFCR), and the PK2DDR bit

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO26_OE	TIOCC9_OE	PK2DDR
PPG	PO26 output*	1	—	—
TPU	TIOCC9 output*	0	1	—
I/O port	PK2 output*	0	0	1
	PK2 input*	0	0	0

Note: * Valid when PCJKE = 1.

I/O port	PK1 output*	0	0	1
	PK1 input*	0	0	0

Note: * Valid when PCJKE = 1.

(8) PK0/PO24/TIOCA9

The pin function is switched as shown below according to the combination of register se
PPG and TPU, setting of the port function control register (PFCR), and the PK0DDR bi

		Setting		
		PPG	TPU	I/O Port
Module Name	Pin Function	PO24_OE	TIOCA9_OE	PK0DD
PPG	PO24 output*	1	—	—
TPU	TIOCA9 output*	0	1	—
I/O port	PK0 output*	0	0	1
	PK0 input*	0	0	0

Note: * Valid when PCJKE = 1.

USB	PULLUP control output	1	—
I/O port	PM4 output	0	1
	PM4 input (initial value)	0	0

(2) PM3

The pin function is switched as shown below according to the combination of the PM3DI setting.

		Setting
		I/O Port
Module Name	Pin Function	PM3DDR
I/O port	PM3 output	1
	PM3 input (initial value)	0

(3) PM2

The pin function is switched as shown below according to the combination of the PM2DI setting.

		Setting
		I/O Port
Module Name	Pin Function	PM2DDR
I/O port	PM2 output	1
	PM2 input (initial value)	0

(5) PM0/TxD6

The pin function is switched as shown below according to the combination of the SCI re
setting and PM0DDR bit setting.

		Setting	
		SCI	**I/O Port**
Module Name	**Pin Function**	**TxD6_OE**	**PM0DDR**
SCI	TxD6 output	1	—
I/O port	PM0 output	0	1
	PM0 input (initial value)	0	0

	SDA0_OE	SDA0		ICCRA.ICE = 1
5	$\overline{\text{ETEND1A}}$_OE	$\overline{\text{ETEND1}}$	PFCR8.EDMAS1[A,B] = 00	SYSCR.EXPE = 1, EDMDR_1.ETEN
	$\overline{\text{TEND1}}$_OE	$\overline{\text{TEND1}}$	PFCR7.DMAS1[A,B] = 00	DMDR_1.TENDE = 1
	SCL1_OE	SCL1		ICCRA.ICE = 1
4	TxD5_OE	TxD5		SCR.TE = 1, IrCR.IrE = 0
	IrTxD_OE	IrTxD		SCR.TE = 1, IrCR.IrE = 1
	SDA1_OE	SDA1		ICCRA.ICE = 1
3	$\overline{\text{EDRAK0}}$_OE	$\overline{\text{EDRAK0}}$	PFCR8.EDMAS0[A,B] = 00	SYSCR.EXPE = 1, EDMDR_0.EDRA
2	$\overline{\text{EDACK0A}}$_OE	$\overline{\text{EDACK0}}$	PFCR8.EDMAS0[A,B] = 00	SYSCR.EXPE = 1, EDACR_0.AMS = EDMDR_0.EDACKE = 1
	$\overline{\text{DACK0}}$_OE	$\overline{\text{DACK0}}$	PFCR7.DMAS0[A,B] = 00	DMAC.DACR_0.AMS = 1, DMDR_0.
	SCK2_OE	SCK2		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE [1, 0] = 01 o SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = 01 o SMR.C/A = 1, SCR.CKE 1 = 0
1	$\overline{\text{ETEND0A}}$_OE	$\overline{\text{ETEND0}}$	PFCR8.EDMAS0[A,B] = 00	SYSCR.EXPE = 1, EDMDR_0.ETEN
	$\overline{\text{TEND0}}$_OE	$\overline{\text{TEND0}}$	PFCR7.DMAS0[A,B] = 00	DMDR_0.TENDE = 1
0	TxD2_OE	TxD2		SCR.TE = 1

		PO6_OE	PO6		NDERL.NDER6 = 1
	5	TIOCA4_OE	TIOCA4		TPU.TIOR_4.IOA3 = 0, TPU.TIOR_4.IOA[1,0] = 01/10/11
		PO5_OE	PO5		NDERL.NDER5 = 1
	4	TIOCB4_OE	TIOCB4		TPU.TIOR_4.IOB3 = 0, TPU.TIOR_4.IOB[1,0] = 01/10/11
		SCK1_OE	SCK1		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE [1, 0] = 01 SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = 01 SMR.C/A = 1, SCR.CKE 1 = 0
		PO4_OE	PO4		NDERL.NDER4 = 1
	3	TIOCD3_OE	TIOCD3		TPU.TMDR.BFB = 0, TPU.TIORL_3.IOD3 = 0, TPU.TIORL_3.IOD[1,0] = 01/10/11
		PO3_OE	PO3		NDERL.NDER3 = 1
	2	TIOCC3_OE	TIOCC3		TPU.TMDR.BFA = 0, TPU.TIORL_3.IOC3 = 0, TPU.TIORL_3.IOD[1,0] = 01/10/11
		TMO0_OE	TMO0		TMR.TCSR_0.OS[3,2] = 01/10/11 o TMR.TCSR_0.OS[1,0] = 01/10/11
		TxD0_OE	TxD0		SCR.TE = 1
		PO2_OE	PO2		NDERL.NDER2 = 1

					SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = 01 o SMR.C/A = 1, SCR.CKE 1 = 0
		PO0_OE	PO0		NDERL.NDER0 = 1
P6	5	$\overline{\text{EDACK1B}}$_OE	$\overline{\text{EDACK1}}$	PFCR8.EDMAS1[A,B] = 01	SYSCR.EXPE = 1, EDACR_1.AMS = EDMDR_1.EDACKE = 1
		$\overline{\text{DACK3}}$_OE	$\overline{\text{DACK3}}$	PFCR7.DMAS3[A,B] = 01	DMAC.DACR_3.AMS = 1, DMDR_3.DACKE = 1
		TMO3_OE	TMO3		TMR.TCSR_3.OS[3,2] = 01/10/11 or TMR.TCSR_3.OS[1,0] = 01/10/11
	4	$\overline{\text{ETEND1B}}$_OE	$\overline{\text{ETEND1}}$	PFCR8.EDMAS1[A,B] = 01	SYSCR.EXPE = 1, EDMDR_1.ETEN
		$\overline{\text{TEND3}}$_OE	$\overline{\text{TEND3}}$	PFCR7.DMAS3[A,B] = 01	DMDR_3.TENDE = 1
	2	$\overline{\text{EDACK0B}}$_OE	$\overline{\text{EDACK0}}$	PFCR8.EDMAS0[A,B] = 01	SYSCR.EXPE = 1, EDACR_0.AMS = EDMDR_0.EDACKE = 1
		$\overline{\text{DACK2}}$_OE	$\overline{\text{DACK2}}$	PFCR7.DMAS2[A,B] = 01	DMAC.DACR_2.AMS = 1, DMDR_2.DACKE = 1
		TMO2_OE	TMO2		TMR.TCSR_2.OS[3,2] = 01/10/11 or TMR.TCSR_2.OS[1,0] = 01/10/11
		SCK4_OE	SCK4		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE [1, 0] = 01 o SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = 01 o SMR.C/A = 1, SCR.CKE 1 = 0

		$\overline{AS}$_OE	$\overline{AS}$		SYSCR.EXPE = 1, PFCR2.ASOE =
	5	$\overline{RD}$_OE	$\overline{RD}$		SYSCR.EXPE = 1
	4	$\overline{LUB}$_OE	$\overline{LUB}$		SYSCR.EXPE = 1, PFCR6.LHWRO SRAMCR.BCSELn = 1
		$\overline{LHWR}$_OE	$\overline{LHWR}$		SYSCR.EXPE = 1, PFCR6.LHWRO
	3	$\overline{LLB}$_OE	$\overline{LLB}$		SYSCR.EXPE = 1, SRAMCR.BCSE
		$\overline{LLWR}$_OE	$\overline{LLWR}$		SYSCR.EXPE = 1
	1	$\overline{BACK}$_OE	$\overline{BACK}$		SYSCR.EXPE = 1,BCR1.BRLE = 1
		(RD/$\overline{WR}$)_OE	RD/$\overline{WR}$		SYSCR.EXPE = 1, PFCR2.RDWRE SRAMCR.BCSELn = 1
	0	$\overline{BSA}$_OE	$\overline{BS}$	PFCR2.BSS = 0	SYSCR.EXPE = 1, PFCR2.BSE =
		$\overline{BREQO}$_OE	$\overline{BREQO}$		SYSCR.EXPE = 1, BCR1.BRLE = BCR1.BREQOE = 1
PB	3	$\overline{CS3}$_OE	$\overline{CS3}$		SYSCR.EXPE = 1, PFCR0.CS3E =
		$\overline{CS7A}$_OE	$\overline{CS7}$	PFCR1.CS7S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS7E =
	2	$\overline{CS2A}$_OE	$\overline{CS2}$	PFCR2.CS2S = 0	SYSCR.EXPE = 1, PFCR0.CS2E =
		$\overline{CS6A}$_OE	$\overline{CS6}$	PFCR1.CS6S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS6E =
	1	$\overline{CS1}$_OE	$\overline{CS1}$		SYSCR.EXPE = 1, PFCR0.CS1E =
		$\overline{CS2B}$_OE	$\overline{CS2}$	PFCR2.CS2S = 1	SYSCR.EXPE = 1, PFCR0.CS2E =
		$\overline{CS5A}$_OE	$\overline{CS5}$	PFCR1.CS5S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS5E =
		$\overline{CS6B}$_OE	$\overline{CS6}$	PFCR1.CS6S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS6E =
		$\overline{CS7B}$_OE	$\overline{CS7}$	PFCR1.CS7S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS7E =
	0	$\overline{CS0}$_OE	$\overline{CS0}$		SYSCR.EXPE = 1, PFCR0.CS0E =
		$\overline{CS4}$_OE	$\overline{CS4}$		SYSCR.EXPE = 1, PFCR0.CS4E =
		$\overline{CS5B}$_OE	$\overline{CS5}$	PFCR1.CS5S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS5E =

	0	A0_OE	A0		SYSCR.EXPE = 1, PDDDR.PD0DDR
PE	7	A15_OE	A15		SYSCR.EXPE = 1, PEDDR.PE7DDR
	6	A14_OE	A14		SYSCR.EXPE = 1, PEDDR.PE6DDR
	5	A13_OE	A13		SYSCR.EXPE = 1, PEDR.PE5DDR
	4	A12_OE	A12		SYSCR.EXPE = 1, PEDDR.PE4DDR
	3	A11_OE	A11		SYSCR.EXPE = 1, PEDDR.PE3DDR
	2	A10_OE	A10		SYSCR.EXPE = 1, PEDDR.PE2DDR
	1	A9_OE	A9		SYSCR.EXPE = 1, PEDDR.PE1DDR
	0	A8_OE	A8		SYSCR.EXPE = 1, PEDDR.PE0DDR
PF	4	A20_OE	A20		SYSCR.EXPE = 1, PFCR4.A20E = 1
	3	A19_OE	A19		SYSCR.EXPE = 1, PFCR4.A19E = 1
	2	A18_OE	A18		SYSCR.EXPE = 1, PFCR4.A18E = 1
	1	A17_OE	A17		SYSCR.EXPE = 1, PFCR4.A17E = 1
	0	A16_OE	A16		SYSCR.EXPE = 1, PFCR4.A16E = 1
PH	7	D7_E	D7		SYSCR.EXPE = 1
	6	D6_E	D6		SYSCR.EXPE = 1
	5	D5_E	D5		SYSCR.EXPE = 1
	4	D4_E	D4		SYSCR.EXPE = 1
	3	D3_E	D3		SYSCR.EXPE = 1
	2	D2_E	D2		SYSCR.EXPE = 1
	1	D1_E	D1		SYSCR.EXPE = 1
	0	D0_E	D0		SYSCR.EXPE = 1

	0	D8_E	D8		SYSCR.EXPE = 1, ABWCR.ABW[
PJ	7	TIOCB8_OE	TIOCB8		TPU.TIOR_8.IOB3 = 0, TPU.TIOR_8.IOB[1,0] = 01/10/11
		PO 23_OE	PO23		NDERL_1.NDER23 = 1
	6	TIOCA8_OE	TIOCA8		TPU.TIOR_8.IOA3 = 0, TPU.TIOR_8.IOA[1,0] = 01/10/11
		PO 22_OE	PO22		NDERL_1.NDER22 = 1
	5	TIOCB7_OE	TIOCB7		TPU.TIOR_7.IOB3 = 0, TPU.TIOR_7.IOB[1,0] = 01/10/11
		PO 21_OE	PO21		NDERL_1.NDER21 = 1
	4	TIOCA7_OE	TIOCA7		TPU.TIOR_7.IOA3 = 0, TPU.TIOR_7.IOA[1,0] = 01/10/11
		PO 20_OE	PO20		NDERL_1.NDER20 = 1
	3	TIOCD6_OE	TIOCD6		TPU.TMDR_6.BFB = 0, TPU.TIORL_6.IOD3 =0, TPU.TIORL_6.IOD[1,0] = 01/10/11
		PO 19_OE	PO19		NDERL_1.NDER19 = 1
	2	TIOCC6_OE	TIOCC6		TPU.TMDR_6.BFA = 0, TPU.TIORL_6.IOC3 = 0, TPU.TIORL_6.IOC[1,0] = 01/10/11
		PO 18_OE	PO18		NDERL_1.NDER18 = 1
	1	TIOCB6_OE	TIOCB6		TPU.TIORH_6.IOB3 = 0, TPU.TIORH_6.IOB[1,0] = 01/10/11
		PO 17_OE	PO17		NDERL_1.NDER17 = 1
	0	TIOCA6_OE	TIOCA6		TPU.TIORH_6.IOA3 = 0, TPU.TIORH_6.IOA[1,0] = 01/10/11
		PO 16_OE	PO16		NDERL_1.NDER16 = 1

					TPU.TIOR_10.IOB[1,0] = 01/10/11
		PO29_OE	PO29		NDERH_1.NDER29 = 1
	4	TIOCA10_OE	TIOCA10		TPU.TIOR_10.IOA3 = 0, TPU.TIOR_10.IOA[1,0] = 01/10/11
		PO28_OE	PO28		NDERH_1.NDER28 = 1
	3	TIOCD9_OE	TIOCD9		TPU.TMDR_9.BFB = 0, TPU.TIORL_9. TPU.TIORL_9.IOD[1,0] = 01/10/11
		PO27_OE	PO27		NDERH_1.NDER27 = 1
	2	TIOCC9_OE	TIOCC9		TPU.TMDR_9.BFA = 0, TPU.TIORL_9. TPU.TIORL_9.IOC[1,0] = 01/10/11
		PO26_OE	PO26		NDERH_1.NDER26 = 1
	1	TIOCB9_OE	TIOCB9		TPU.TIORH_9.IOB3 = 0, TPU.TIORH_9.IOB[1,0] = 01/10/11
		PO25_OE	PO25		NDERH_1.NDER25 = 1
	0	TIOCA9_OE	TIOCA9		TPU.TIORH_9.IOA3 = 0, TPU.TIORH_9.IOA[1,0] = 01/10/11
		PO24_OE	PO24		NDERH_1.NDER24 = 1

- Port function control register 6 (PFCR6)
- Port function control register 7 (PFCR7)
- Port function control register 8 (PFCR8)
- Port function control register 9 (PFCR9)
- Port function control register A (PFCRA)
- Port function control register B (PFCRB)
- Port function control register C (PFCRC)
- Port function control register D (PFCRD)

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

Note: * 1 in external extended mode, 0 in other modes.

Bit	Bit Name	Value	R/W	Description
7	CS7SA*	0	R/W	$\overline{CS7}$ Output Pin Select
6	CS7SB*	0	R/W	Selects the output pin for $\overline{CS7}$ when $\overline{CS7}$ outp enabled (CS7E = 1) 00: Specifies pin PB3 as $\overline{CS7}$-A output 01: Specifies pin PB1 as $\overline{CS7}$-B output 10: Setting prohibited 11: Setting prohibited
5	CS6SA*	0	R/W	$\overline{CS6}$ Output Pin Select
4	CS6SB*	0	R/W	Selects the output pin for $\overline{CS6}$ when $\overline{CS6}$ outp enabled (CS6E = 1) 00: Specifies pin PB2 as $\overline{CS6}$-A output 01: Specifies pin PB1 as $\overline{CS6}$-B output 10: Setting prohibited 11: Setting prohibited
3	CS5SA*	0	R/W	$\overline{CS5}$ Output Pin Select
2	CS5SB*	0	R/W	Selects the output pin for $\overline{CS5}$ when $\overline{CS5}$ outp enabled (CS5E = 1) 00: Specifies pin PB1 as $\overline{CS5}$-A output 01: Specifies pin PB0 as $\overline{CS5}$-B output 10: Setting prohibited 11: Setting prohibited

section 9.5.3, Chip Select Signals.

13.3.3 Port Function Control Register 2 (PFCR2)

PFCR2 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	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
6	CS2S*[1]	0	R/W	$\overline{\text{CS2}}$ Output Pin Select Selects the output pin for $\overline{\text{CS2}}$ when $\overline{\text{CS2}}$ outpu 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	Reserved This bit is always read as 0. The write value s 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	Reserved This bit is always read as 0. The write value s always be 0.

Notes: 1. If multiple $\overline{\text{CS}}$ outputs are specified to a single pin according to the $\overline{\text{CS}}$n outpu select bit (n = 2), multiple $\overline{\text{CS}}$ signals are output from the pin. For details, see 9.5.3, Chip Select Signals.

2. If an area is specified as a byte control SDRAM space, the pin functions as R output regardless of the RDWRE bit value.

Bit	Bit Name	Value	R/W	Description
7 to 5	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
4	A20E	0/1*	R/W	Address A20 Enable Enables/disables the address output (A20) 0: Disables the A20 output 1: Enables the A20 output
3	A19E	0/1*	R/W	Address A19 Enable Enables/disables the address output (A19) 0: Disables the A19 output 1: Enables the A19 output
2	A18E	0/1*	R/W	Address A18 Enable Enables/disables the address output (A18) 0: Disables the A18 output 1: Enables the A18 output
1	A17E	0/1*	R/W	Address A17 Enable Enables/disables the address output (A17) 0: Disables the A17 output 1: Enables the A17 output
0	A16E	0/1*	R/W	Address A16 Enable Enables/disables the address output (A16) 0: Disables the A16 output 1: Enables the A16 output

Note: * Initial value is switched according to operating mode. 1 when on-chip ROM dis when enabled.

Bit	Bit Name	Value	R/W	Description
7	—	1	R/W	Reserved This bit is always read as 1. The write value s always be 1.
6	LHWROE	1	R/W	$\overline{\text{LHWR}}$ Output Enable Enables/disables $\overline{\text{LHWR}}$ output (valid in exter 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 s 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: The TPU external clock input pins cannot b 1: Specifies pins P14 to P17 as external clock pins.
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.

Bit	Bit Name	Value	R/W	Description
7	DMAS3A	0	R/W	DMAC Control Pin Select
6	DMAS3B	0	R/W	Selects the I/O port to control DMAC_3. 00: DMAC_3 control pins are disabled. 01: Specifies pins P63 to P65 as DMAC contro 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: DMAC_2 control pins are disabled. 01: Specifies pins P60 to P62 as DMAC contro 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 contro 01: Setting prohibited 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 contro 01: Setting prohibited 10: Setting prohibited 11: Setting prohibited

Bit	Bit Name	Value	R/W	Description
7 to 4	—	0	R/W	Reserved bit The write value should always be 0.
3 2	EDMAS1A EDMAS1B	0 0	R/W R/W	EXDMAC Control Pin Select Selects the I/O port to control EXDMAC_1. 00: Specifies pins P14 to P17 as EXDMAC co 01: Specifies pins P63 to P65 as EXDMAC co 10: Setting prohibited 11: Setting prohibited
1 0	EDMAS0A EDMAS0B	0 0	R/W R/W	EXDMAC Control Pin Select Selects the I/O port to control EXDMAC_0. 00: Specifies pins P10 to P13 as EXDMAC co 01: Specifies pins P60 to P62 as EXDMAC co 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 a capture 1: Specifies P27 as input capture input and P26 output compare
6	TPUMS4	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA4 function. 0: Specifies P25 as output compare output and capture 1: Specifies P24 as input capture input and P25 output compare
5	TPUMS3A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA3 function. 0: Specifies P21 as output compare output and capture 1: Specifies P20 as input capture input and P21 output compare
4	TPUMS3B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC3 function. 0: Specifies P22 as output compare output and capture 1: Specifies P23 as input capture input and P22 output compare
3 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.

Bit	Bit Name	Value	R/W	Description
7	TPUMS11	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA11 function. 0: Specifies pin PK6 as output compare outpu input capture 1: Specifies PK7 as input capture input and P output compare
6	TPUMS10	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA10 function. 0: Specifies PK4 as output compare output ar capture 1: Specifies PK5 as input capture input and P output compare
5	TPUMS9A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA9 function. 0: Specifies PK0 as output compare output ar capture 1: Specifies PK1 as input capture input and P output compare
4	TPUMS9B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC9 function. 0: Specifies PK2 as output compare output ar capture 1: Specifies PK3 as input capture input and P output compare

				capture
				1: Specifies PJ5 as input capture input and PJ4 output compare
1	TPUMS6A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA6 function. 0: Specifies PJ0 as output compare output and capture 1: Specifies PJ1 as input capture input and PJ0 output compare
0	TPUMS6B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC6 function. 0: Specifies PJ2 as output compare output and capture 1: Specifies PJ3 as input capture input and PJ2 output compare

• H8SX/1655 Group

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.

• H8SX/1655M Group

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved These bits are always read as 0. The write va always be 0.
6	ITS14	0	R/W	LVD Interrupt Select Enables/Disables the LVD interrupt select. 0: Disables the LVD interrupt 1: Enables the LVD interrupt
5 to 4	—	0	R/W	Reserved These bits are always read as 0. The write va 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

0	ITS8	0	R/W	$\overline{IRQ8}$ Pin Select Selects an input pin for $\overline{IRQ8}$. 0: Selects pin P20 as $\overline{IRQ8}$-A input 1: Selects pin P60 as $\overline{IRQ8}$-B input

Bit	Bit Name	Value	R/W	Description
7	ITS7	0	R/W	$\overline{\text{IRQ7}}$ Pin Select Selects an input pin for $\overline{\text{IRQ7}}$. 0: Selects pin P17 as $\overline{\text{IRQ7}}$-A input 1: Selects pin P57 as $\overline{\text{IRQ7}}$-B input
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 input
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 input
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 input
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 input

Bit	Bit Name	Initial Value	R/W	Description
0	ITS0	0	R/W	$\overline{IRQ0}$ Pin Select Selects an input pin for $\overline{IRQ0}$. 0: Selects pin P10 as $\overline{IRQ0}$-A input 1: Selects pin P50 as $\overline{IRQ0}$-B input

13.3.12 Port Function Control Register D (PFCRD)

PFCRD enables/disables the pin functions of ports J and K.

Bit	7	6	5	4	3	2	1
Bit Name	PCJKE*	—	—	—	—	—	—
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	PCJKE*	0	R/W	Ports J and K Enable Enables/disables ports J and K. 0: Ports J and K are disabled 1: Ports J and K are enabled
6 to 0	—	0	R/W	Reserved These bits are always read as 0 and cannot be modified. The initial values should not be chan

Note: * This bit is valid during single-chip mode. The initial value should not be change for the single-chip mode.

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

13.4.2 Notes on Port Function Control Register (PFCR) Settings

1. Port function controller controls the I/O port.

   Before enabling a port function, select the input/output destination.

2. When changing input pins, this LSI may malfunction due to the internal edge genera
pin level difference before and after the change.

- To change input pins, the following procedure must be performed.

   A. Disable the input function by the corresponding on-chip peripheral module settin

   B. Select another input pin by PFCR

   C. Enable its input function by the corresponding on-chip peripheral module setting

3. If a pin function has both a select bit that modifies the input/output destination and a
bit that enables the pin function, first specify the input/output destination by the sele
and then enable the pin function by the enable bit.

4. Modifying the PCJKE bit should be done in the initial setting right after activation. S
bits after setting the PCJKE bit.

5. Do not change the PCJKE bit setting once it is set.

- Maximum 16-pulse input/output
- Selection of eight counter input clocks for each channel
- The following operations can be set for each channel:
  - — Waveform output at compare match
  - — Input capture function
  - — Counter clear operation
  - — 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, 3, 6, and 9
- Phase counting mode settable independently for each of channels 1, 2, 4, 5, 7, 8, 10,
- 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 (unit 0 only)
- Module stop state can be set

(TGR)		TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TG
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		—	—	—	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	TGR compare match or input capture	TGR compare match or input capture	TC co ma inp ca
Compare match output	0 output	—	—	—	O	O	O
	1 output	—	—	—	O	O	O
	Toggle output	—	—	—	O	O	O
Input capture function		O	O	—	O	O	O
Synchronous operation		O	O	O	O	O	O
PWM mode		O	O	O	O	O	O
Phase counting mode		—	O	O	—	O	O
Buffer operation		O	—	—	O	—	—
DTC activation		TGR compare match or input capture	TGR compare match or input capture	TGR compare match	TGR compare match or input capture	TGR compare match or input capture	TC co ma inp ca

[Legend]

O: Possible

—: Not possible

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	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 2A Compare match 2B Overflow Underflow	5 sources Compare match or input capture 3A Compare match or input capture 3B Compare match or input capture 3C Compare match or input capture 3D Overflow	4 sources Compare match or input capture 4A Compare match or input capture 4B Overflow Underflow	4 C m in c C m in c O U

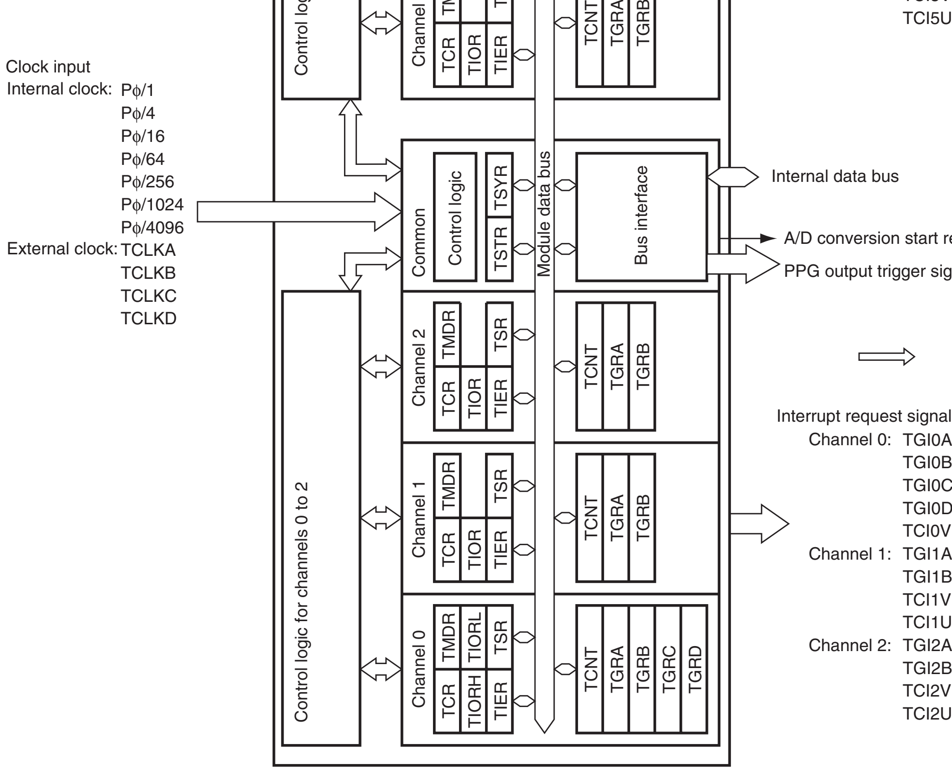
(TGR)		TGRB_6	TGRB_7	TGRB_8	TGRB_9	TGRB_10	T
General registers/ buffer registers		TGRC_6 TGRD_6	—	—	TGRC_9 TGRD_9	—	—
I/O pins		TIOCA6 TIOCB6 TIOCC6 TIOCD6	TIOCA7 TIOCB7	TIOCA8 TIOCB8	TIOCA9 TIOCB9 TIOCC9 TIOCD9	TIOCA10 TIOCB10	TI TI
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TG co ma inp ca
Compare match output	0 output	O	O	O	O	O	O
	1 output	O	O	O	O	O	O
	Toggle output	O	O	O	O	O	O
Input capture function		O	O	O	O	O	O
Synchronous operation		O	O	O	O	O	O
PWM mode		O	O	O	O	O	O
Phase counting mode		—	O	O	—	O	O
Buffer operation		O	—	—	O	—	—
DTC activation		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TG co ma inp ca

[Legend]

O: Possible

—: Not possible

	match	match	match	match		
Interrupt sources	5 sources Compare match or input capture 6A Compare match or input capture 6B Compare match or input capture 6C Compare match or input capture 6D Overflow	4 sources Compare match or input capture 7A Compare match or input capture 7B Overflow Underflow	4 sources Compare match or input capture 8A Compare match or input capture 8B Overflow Underflow	5 sources Compare match or input capture 9A Compare match or input capture 9B Compare match or input capture 9C Compare match or input capture 9D Overflow	4 sources Compare match or input capture 10A Compare match or input capture 10B Overflow Underflow	4 C m i c 1 C m i c 1 O U



[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 14.1 Block Diagram of TPU (Unit 0)

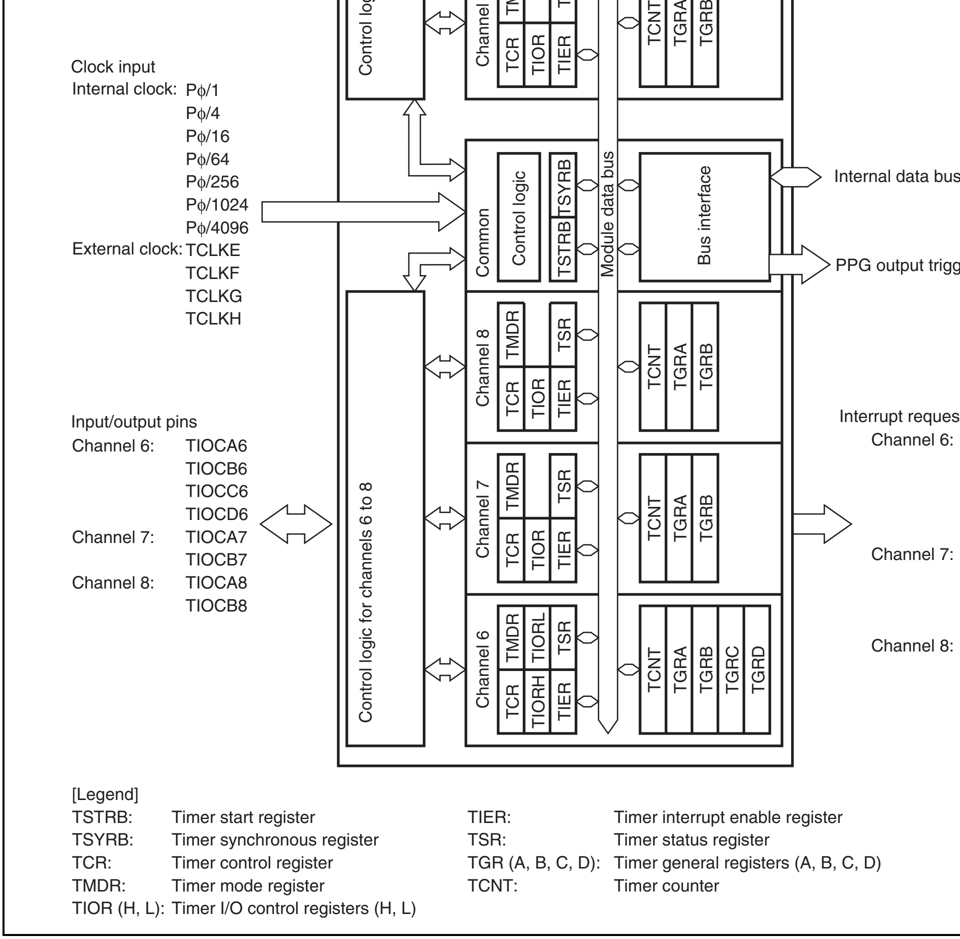


Figure 14.2 Block Diagram of TPU (Unit 1)

				(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)
	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

	6	TIOCA6	I/O	TGRA_6 input capture input/output compare output/PWM
		TIOCB6	I/O	TGRB_6 input capture input/output compare output/PWM
		TIOCC6	I/O	TGRC_6 input capture input/output compare output/PWM
		TIOCD6	I/O	TGRD_6 input capture input/output compare output/PWM
	7	TIOCA7	I/O	TGRA_7 input capture input/output compare output/PWM
		TIOCB7	I/O	TGRB_7 input capture input/output compare output/PWM
	8	TIOCA8	I/O	TGRA_8 input capture input/output compare output/PWM
		TIOCB8	I/O	TGRB_8 input capture input/output compare output/PWM
	9	TIOCA9	I/O	TGRA_9 input capture input/output compare output/PWM
		TIOCB9	I/O	TGRB_9 input capture input/output compare output/PWM
		TIOCC9	I/O	TGRC_9 input capture input/output compare output/PWM
		TIOCD9	I/O	TGRD_9 input capture input/output compare output/PWM
	10	TIOCA10	I/O	TGRA_10 input capture input/output compare output/PWM
		TIOCB10	I/O	TGRB_10 input capture input/output compare output/PWM
	11	TIOCA11	I/O	TGRA_11 input capture input/output compare output/PWM
		TIOCB11	I/O	TGRB_11 input capture input/output compare output/PWM

— Timer mode register_0 (TMDR_0)
— Timer I/O control register H_0 (TIORH_0)
— Timer I/O control register L_0 (TIORL_0)
— Timer interrupt enable register_0 (TIER_0)
— Timer status register_0 (TSR_0)
— Timer counter_0 (TCNT_0)
— Timer general register A_0 (TGRA_0)
— Timer general register B_0 (TGRB_0)
— Timer general register C_0 (TGRC_0)
— Timer general register D_0 (TGRD_0)

- 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)

Unit 1

- Channel 6
  - Timer control register_6 (TCR_6)
  - Timer mode register_6 (TMDR_6)
  - Timer I/O control register H_6 (TIORH_6)
  - Timer I/O control register L_6 (TIORL_6)
  - Timer interrupt enable register_6 (TIER_6)
  - Timer status register_6 (TSR_6)
  - Timer counter_6 (TCNT_6)
  - Timer general register A_6 (TGRA_6)
  - Timer general register B_6 (TGRB_6)
  - Timer general register C_6 (TGRC_6)
  - Timer general register D_6 (TGRD_6)

- Channel 8
  - Timer control register_8 (TCR_8)
  - Timer mode register_8 (TMDR_8)
  - Timer I/O control register_8 (TIOR_8)
  - Timer interrupt enable register_8 (TIER_8)
  - Timer status register_8 (TSR_8)
  - Timer counter_8 (TCNT_8)
  - Timer general register A_8 (TGRA_8)
  - Timer general register B_8 (TGRB_8)

- Channel 9
  - Timer control register_9 (TCR_9)
  - Timer mode register_9 (TMDR_9)
  - Timer I/O control register H_9 (TIORH_9)
  - Timer I/O control register L_9 (TIORL_9)
  - Timer interrupt enable register_9 (TIER_9)
  - Timer status register_9 (TSR_9)
  - Timer counter_9 (TCNT_9)
  - Timer general register A_9 (TGRA_9)
  - Timer general register B_9 (TGRB_9)
  - Timer general register C_9 (TGRC_9)
  - Timer general register D_9 (TGRD_9)

- Channel 11
  - Timer control register_11 (TCR_11)
  - Timer mode register_11 (TMDR_11)
  - Timer I/O control register_11 (TIOR_11)
  - Timer interrupt enable register_11 (TIER_11)
  - Timer status register_11 (TSR_11)
  - Timer counter_11 (TCNT_11)
  - Timer general register A_11 (TGRA_11)
  - Timer general register B_11 (TGRB_11)
- Common Registers
  - Timer start register (TSTRB)
  - Timer synchronous register (TSYRB)

Bit	Bit Name	Initial Value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	These bits select the TCNT counter clearing so
5	CCLR0	0	R/W	tables 14.4 and 14.5 for details.
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 14.6. 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 m used on channels 1, 2, 4, 5, 7, 8, 10, and 11, th is ignored and the phase counting mode setting priority. Internal clock edge selection is valid wh input clock is Pϕ/4 or slower. This setting is ign input clock is Pϕ/1, or when overflow/underflow channel is selected.
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The source can be selected independently for each
0	TPSC0	0	R/W	See tables 14.7 to 14.12 for details. To select t clock as the clock source, the DDR bit and ICR corresponding pin should be set to 0 and 1, res For details, see section 13, I/O Ports.

				synchronous operation*[1]
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare matc capture*[2]
	1	1	0	TCNT cleared by TGRD compare matc capture*[2]
	1	1	1	TCNT cleared by counter clearing for a channel performing synchronous cleari 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

Table 14.5 CCLR2 to CCLR0 (Channels 1, 2, 4, 5, 7, 8, 10, and 11)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5, 7, 8, 10, 11	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare matc capture
	0	1	0	TCNT cleared by TGRB compare matc capture
	0	1	1	TCNT cleared by counter clearing for a channel performing synchronous cleari 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, 5, 7, 8, 10, and 11. It is always read as 0 a cannot be modified.

Table 14.7 TPSC2 to TPSC0 (Channels 0 and 6)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0, 6	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

	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	Counts on TCNT2* overflow/underflow

Notes: This setting is ignored when channel 1 is in phase counting mode.

* Counts on TCNT8 overflow/underflow in the case of TCR_7.

Table 14.9 TPSC2 to TPSC0 (Channels 2 and 8)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2, 8	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin inp
	1	0	1	External clock: counts on TCLKB pin inp
	1	1	0	External clock: counts on TCLKC pin inp
	1	1	1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	Internal clock: counts on Pϕ/4096

Table 14.11 TPSC2 to TPSC0 (Channels 4 and 10)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4, 10	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/1024
	1	1	1	Counts on TCNT5* overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

* Counts on TCNT11 overflow/underflow in the case of TCR_10.

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

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

14.3.2 Timer Mode Register (TMDR)

TMDR sets the operating mode for each channel. The TPU has 12 TMDR registers, one f channel. TMDR register settings should be made only while TCNT operation is stopped.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	BFB	BFA	MD3	MD2	MD1
Initial Value	1	1	0	0	0	0	0
R/W	—	—	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be m
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to normally operate, o and TGRD are to be used together for buffer ope When TGRD is used as a buffer register, TGRD i capture/output compare is not generated. In channels 1, 2, 4, 5, 7, 8, 10, and 11, which hav TGRD, bit 5 is reserved. It is always read as 0 an cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer ope

				1: TGRA and TGRC used together for buffer op
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	Set the timer operating mode.
1	MD1	0	R/W	MD3 is a reserved bit. The write value should al
0	MD0	0	R/W	0. See table 14.13 for details.

Table 14.13 MD3 to MD0

Bit 3 MD3*[1]	Bit 2 MD2*[2]	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
0	0	0	1	Reserved
0	0	1	0	PWM mode 1
0	0	1	1	PWM mode 2
0	1	0	0	Phase counting mode 1
0	1	0	1	Phase counting mode 2
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, 3, 6, and 9. In this case, always be written to MD2.

operates as a buffer register.

To designate the input capture pin in TIOR, the DDR bit and ICR bit for the correspondin
should be set to 0 and 1, respectively. For details, see section 13, I/O Ports.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIOR_4, TIOR_5

Bit	7	6	5	4	3	2	1
Bit Name	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- TIORL_0, TORL_3

Bit	7	6	5	4	3	2	1
Bit Name	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

1	IOA1	0	R/W	For details, see tables 14.22, 14.24 to 14.26, 14
0	IOA0	0	R/W	14.29.

- TIORL_0, TIORL_3 (Unit 0)
- TIORL_6, TIORL_9 (Unit 1)

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 14.15, and 14.19.
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 14.23, and 14.27.
0	IOC0	0	R/W	

0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	x		
1	1	x	x	Input capture register	Capture input source is channel 1/coun Input capture at TCNT_1 count-up/coun

[Legend]

X: Don't care

Note: 1. In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA0 pin outp

2. 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.

0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	X		
1	1	X	X	Input capture register*2	Capture input source is channel 1/co Input capture at TCNT_1 count-up/c down*3

[Legend]

X: Don't care

Notes: 1. In PWM mode 1, the settings of bits IOD3 to IOD0 control the TIOCC0 pin ou

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

3. 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	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	X		
1	1	X	X	Input capture register	TGRC_0 compare match/input captur Input capture at generation of TGRC_ compare match/input capture

[Legend]

X: Don't care

Note: * In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA1 pin outp

0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	X	0	0		
1	X	0	1		
1	X	1	X	Input capture register	Capture input source is TIOCB2 pin Input capture at both edges

[Legend]

X: Don't care

Note: * In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA2 pin out

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/count Input capture at TCNT_4 count-up/count

[Legend]

X: Don't care

Notes: 1. In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA3 pin outp

2. 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.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCD3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCD3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/coun Input capture at TCNT_4 count-up/cour

[Legend]

X: Don't care

Notes: 1. In PWM mode 1, the settings of bits IOD3 to IOD0 control the TIOCC3 pin ou

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

3. When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and Pϕ/1 is used as tl TCNT_4 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	Capture input source is TIOCB4 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB4 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB4 pin Input capture at both edges
1	1	x	x		Capture input source is TGRC_3 compa match/input capture Input capture at generation of TGRC_3 c match/input capture

[Legend]

X: Don't care

Note: * In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA4 pin outp

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

Note: * In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA5 pin out

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 TIOCB6 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB6 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCB6 pin Input capture at both edges
1	1	X	X		Capture input source is channel 7/count Input capture at TCNT_7 count-up/count

[Legend]

X: Don't care

Notes: 1. In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA6 pin outp

2. When the bits TPSC2 to TPSC0 in TCR_7 are set to B'000 and Pϕ/1 is used a count clock of TCNT_7, this setting is invalid and input capture is not generate

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 TIOCD6 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCD6 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCD6 pin Input capture at both edges
1	1	X	X		Capture input source is channel 7/coun Input capture at TCNT_7 count-up/cour

[Legend]

X: Don't care

Notes: 1. In PWM mode 1, the settings of bits IOD3 to IOD0 control the TIOCC6 pin ou

2. When the BFB bit in TMDR_6 is set to 1 and TGRD_6 is used as a buffer re setting is invalid and input capture/output compare is not generated.

3. When the bits TPSC2 to TPSC0 in TCR_7 are set to B'000 and Pφ/1 is usec count clock of TCNT_7, this setting is invalid and input capture is not genera

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 TIOCB7 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB7 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCB7 pin Input capture at both edges
1	1	X	X		TGRC_6 compare match/input capture Input capture at generation of TGRC_6 c match/input capture

[Legend]

X: Don't care

Note: * In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA7 pin outp

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 TIOCB8 pin Input capture at rising edge
1	X	0	1		Capture input source is TIOCB8 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCB8 pin Input capture at both edges

[Legend]

X: Don't care

Note: * In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA8 pin out

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 TIOCB9 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB9 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCB9 pin Input capture at both edges
1	1	X	X		Capture input source is channel 10/coun Input capture at TCNT_10 count-up/cour

[Legend]

X: Don't care

Notes: 1. In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA9 pin outp

2. When the bits TPSC2 to TPSC0 in TCR_10 are set to B'000 and Pϕ/1 is used count clock of TCNT_10, this setting is invalid and input capture is not generat

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 TIOCD9 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCD9 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCD9 pin Input capture at both edges
1	1	X	X		Capture input source is channel 10/cou Input capture at TCNT_10 count-up/cou

[Legend]

X: Don't care

Notes: 1. In PWM mode 1, the settings of bits IOD3 to IOD0 control the TIOCC9 pin ou

2. When the BFB bit in TMDR_9 is set to 1 and TGRD_9 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

3. When the bits TPSC2 to TPSC0 in TCR_10 are set to B'000 and Pϕ/1 is used count clock of TCNT_10, this setting is invalid and input capture is not genera

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 TIOCB10 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB10 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCB10 pin Input capture at both edges
1	1	X	X		Capture input source is TGRC_9 compa match/input capture Input capture at generation of TGRC_9 c match/input capture

[Legend]

X: Don't care

Note: * In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA10 pin out

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 TIOCB11 pin Input capture at rising edge
1	X	0	1		Input capture source is TIOCB11 pin Input capture at falling edge
1	X	1	X		Input capture source is TIOCB11 pin Input capture at both edges

[Legend]

X: Don't care

Note: * In PWM mode 1, the settings of bits IOB3 to IOB0 control the TIOCA11 pin ou

0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	1		
1	0	0	0		
1	0	1	x		
1	1	x	x	Input capture register	Capture input source is channel 1/coun Input capture at TCNT_1 count-up/coun

[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 th TCNT_1 count clock, this setting is invalid and input capture is not generated.

0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	X		
1	1	X	X	Input capture register*2	Capture input source is channel 1/co Input capture at TCNT_1 count-up/c down*1

[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 BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	X		
1	1	X	X	Input capture register	Capture input source is TGRA_0 com match/input capture Input capture at generation of channe 0/TGRA_0 compare match/input captu

[Legend]

X: Don't care

0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	X	0	0		
1	X	0	1		
1	X	1	X	Input capture register	Capture input source is TIOCA2 pin Input capture at both edges

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCA3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/count Input capture at TCNT_4 count-up/count

[Legend]

X: Don't care

Note: * 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.

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 TIOCC3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCC3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCC3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/coun Input capture at TCNT_4 count-up/cour

[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 tl TCNT_4 count clock, this setting is invalid and input capture is not generated

2. When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA4 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCA4 pin Input capture at both edges
1	1	x	x		Capture input source is TGRC_3 compa match/input capture Input capture at generation of TGRC_3 c match/input capture

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA5 pin Input capture at rising edge
1	x	0	1		Capture input source is TIOCA5 pin Input capture at falling edge
1	x	1	x		Capture input source is TIOCA5 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	1	Input capture register	Capture input source is TIOCA6 pin Input capture at falling edge
1	0	0	0		Capture input source is TIOCA6 pin Input capture at rising edge
1	0	1	X		Capture input source is TIOCA6 pin Input capture at both edges
1	1	X	X		Capture input source is channel 7/count Input capture at TCNT_7 count-up/count

[Legend]

X: Don't care

Note: * When the bits TPSC2 to TPSC0 in TCR_7 are set to B'000 and Pϕ/1 is used a count clock of TCNT_7, this setting is invalid and input capture is not generate

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 TIOCC6 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCC6 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCC6 pin Input capture at both edges
1	1	X	X		Capture input source is channel 7/coun Input capture at TCNT_7 count-up/cour

[Legend]

X: Don't care

Notes: 1. When the bits TPSC2 to TPSC0 in TCR_7 are set to B'000 and Pφ/1 is used
count clock of TCNT_7, this setting is invalid and input capture is not generat

2. When the BFA bit in TMDR_6 is set to 1 and TGRC_6 is used as a buffer reg
setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA7 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA7 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA7 pin Input capture at both edges
1	1	X	X		Capture input source is TGRA_6 compa match/input capture Input capture at generation of channel 6/ 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 TIOCA8 pin Input capture at rising edge
1	X	0	1		Capture input source is TIOCA8 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCA8 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 TIOCA9 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA9 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA9 pin Input capture at both edges
1	1	X	X		Capture input source is channel 10/coun Input capture at TCNT_10 count-up/cour

[Legend]

X: Don't care

Note: * When the bits TPSC2 to TPSC0 in TCR_10 are set to B'000 and Pϕ/1 is used count clock of TCNT_10, this setting is invalid and input capture is not generat

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 TIOCC9 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCC9 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCC9 pin Input capture at both edges
1	1	X	X		Capture input source is channel 10/cou Input capture at TCNT_10 count-up/cou

[Legend]

X: Don't care

Notes: 1. When the bits TPSC2 to TPSC0 in TCR_10 are set to B'000 and Pϕ/1 is used count clock of TCNT_10, this setting is invalid and input capture is not genera

2. When the BFA bit in TMDR_9 is set to 1 and TGRC_9 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA10 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA10 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA10 pin Input capture at both edges
1	1	X	X		Capture input source is TGRA_9 compa match/input capture Input capture at generation of TGRA_9 c match/input capture

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	X	0	0	Input capture register	Input capture source is TIOCA11 pin Input capture at rising edge
1	X	0	1		Input capture source is TIOCA11 pin Input capture at falling edge
1	X	1	X		Input capture source is TIOCA11 pin Input capture at both edges

[Legend]
X: Don't care

This bit is always read as 0 and the initial value should not be changed.

Bit	Bit Name	Initial value	R/W	Description
7	TTGE*	0	R/W	A/D Conversion Start Request Enable Enables/disables generation of A/D conversion s requests by TGRA input capture/compare match 0: A/D conversion start request generation disab 1: A/D conversion start request generation enabl
6	—	1	—	Reserved This bit is always read as 1 and cannot be modif
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables/disables interrupt requests (TCIU) by th flag when the TCFU flag in TSR is set to 1 in cha 2, 4, 5, 7, 8, 10, and 11. In channels 0, 3, 6, and 9, bit 5 is reserved. It is read as 0 and cannot be modified. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables/disables interrupt requests (TCIV) by th flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled

				when the TGFC bit in TSR is set to 1 in channels 0, 3 In channels 1, 2, 4, 5, 7, 8, 10, and 11, bit 2 is reserv always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables/disables interrupt requests (TGIB) by the TG when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables/disables interrupt requests (TGIA) by the TG when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

Note: * The bit 7 in TIER of unit 1 is a reserved bit This bit is always read as 0 and th value should not be changed.

14.3.5 Timer Status Register (TSR)

TSR indicates the status of each channel. The TPU has 12 TSR registers, one for each c

Bit	7	6	5	4	3	2	1
Bit Name	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB
Initial Value	1	1	0	0	0	0	0
R/W	R	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to bits 5 to 0, to clear flags.

5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that a TCNT underflow has when channels 1, 2, 4, 5, 7, 8, 10, and 11 are set to p counting mode. In channels 0, 3, 6, and 9, bit 5 is reserved. It is alwa as 0 and cannot be modified. [Setting condition] When the TCNT value underflows (changes from H'0 H'FFFF) [Clearing condition] When a 0 is written to TCFU after reading TCFU = 1 (When the CPU is used to clear this flag by writing 0 corresponding interrupt is enabled, be sure to read th after writing 0 to it.)
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that a TCNT overflow has o [Setting condition] When the TCNT value overflows (changes from H'FF H'0000) [Clearing condition] When a 0 is written to TCFV after reading TCFV = 1 (When the CPU is used to clear this flag by writing 0 corresponding interrupt is enabled, be sure to read th after writing 0 to it.)

				• When TCNT value is transferred to TGRD b capture signal while TGRD is functioning as capture register [Clearing conditions] • When DTC is activated by a TGID interrupt DISEL bit in MRB of DTC is 0 • When 0 is written to TGFD after reading TG (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.)
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGF capture or compare match in channels 0, 3, 6, a In channels 1, 2, 4, 5, 7, 8, 10, and 11, bit 2 is r is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRC while TGRC is functio output compare register • 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.)

				capture register [Clearing conditions] • When DTC is activated by a TGIB interrupt w DISEL bit in MRB of DTC is 0 • When 0 is written to TGFB after reading TGF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGR capture or compare match. [Setting conditions] • When TCNT = TGRA while TGRA is function output compare register • When TCNT value is transferred to TGRA by capture signal while TGRA is functioning as i capture register [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 DTC 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

14.3.7 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 32 TGR registers, four each for channels 0, 3, 6, and 9 ar each for channels 1, 2, 4, 5, 7, 8, 10, and 11. TGRC and TGRD for channels 0, 3, 6, and be designated for operation as buffer registers. The TGR registers cannot be accessed in units; they must always be accessed in 16-bit units. TGR and buffer register combinatio buffer operations are TGRA–TGRC and TGRB–TGRD.

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	—	Reserved The write value should 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 wit
2	CST2	0	R/W	TIOC pin designated for output, the counter stop
1	CST1	0	R/W	TIOC pin output compare output level is retaine is written to when the CST bit is cleared to 0, the
0	CST0	0	R/W	output level will be changed to the set initial outp 0: TCNT_5 to TCNT_0* count operation is stopp 1: TCNT_5 to TCNT_0* performs count operatio

Note: * In the case of unit 1, these bits select operation or stoppage for TCNT_11 to T

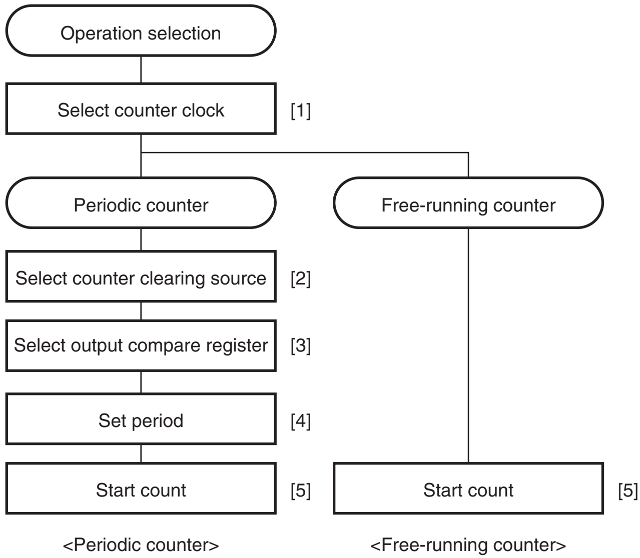
Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved The write value should always be 0.
5	SYNC5	0	R/W	Timer Synchronization 5 to 0
4	SYNC4	0	R/W	These bits select whether operation is indepenc synchronized with other channels.
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	When synchronous operation is selected, syncl presetting of multiple channels, and synchronou through counter clearing on another channel ar
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	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 c (TCNT synchronous presetting/synchronous is possible)

Note: * In the case of unit 1, these bits select independent or synchronous operation TCNT_11 to TCNT_6.

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 14.3 shows an example of the count operation setting procedure.



[1] Select the counter clock with l
TPSC2 to TPSC0 in TCR. At t
same time, select the input clo
with bits CKEG1 and CKEG0

[2] For periodic counter operation
the TGR to be used as the TC
clearing source with bits CCLI
CCLR0 in TCR.

[3] Designate the TGR selected i
an output compare register by
of TIOR.

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

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

Figure 14.3 Example of Counter Operation Setting Procedure

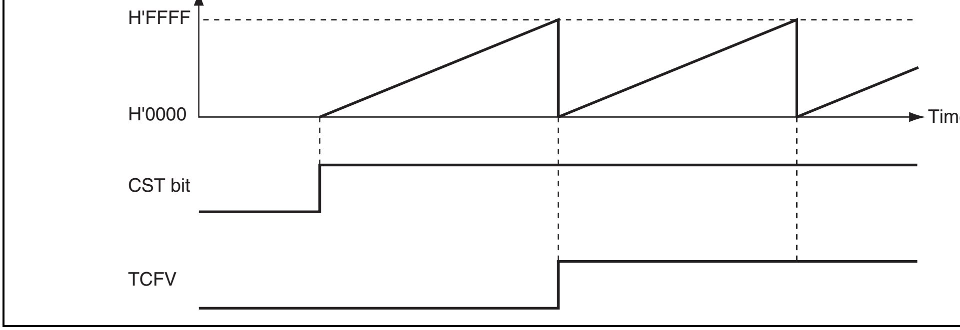


Figure 14.4 Free-Running Counter Operation

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 14.5 illustrates periodic counter operation.

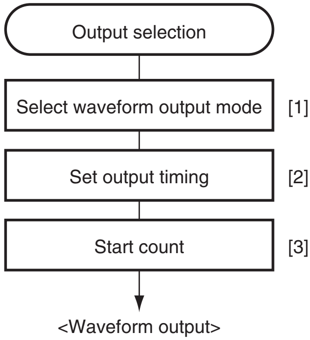
Figure 14.5 Periodic Counter Operation

(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 14.6 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 14.6 Example of Setting Procedure for Waveform Output by Compare M

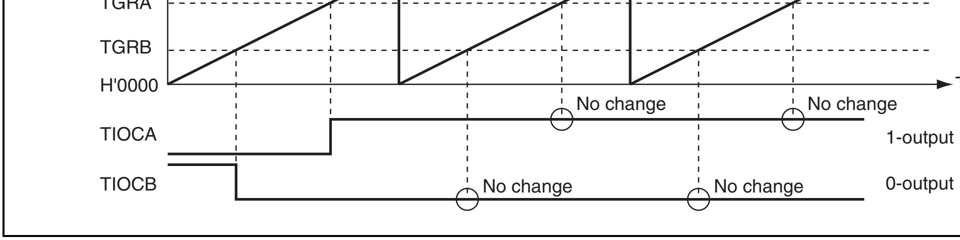


Figure 14.7 Example of 0-Output/1-Output Operation

Figure 14.8 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.

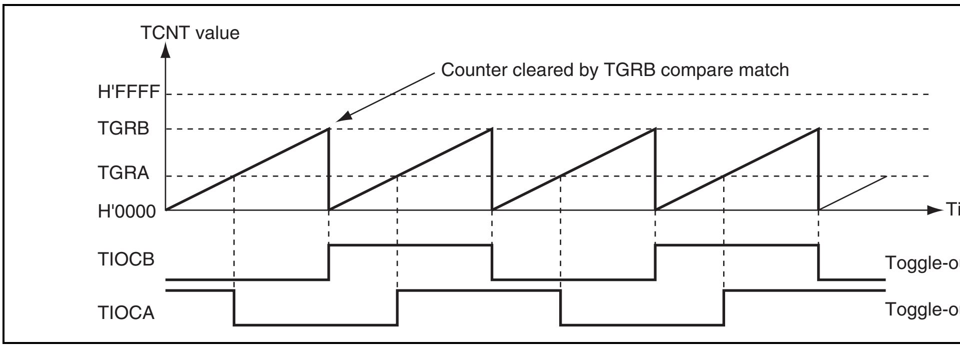
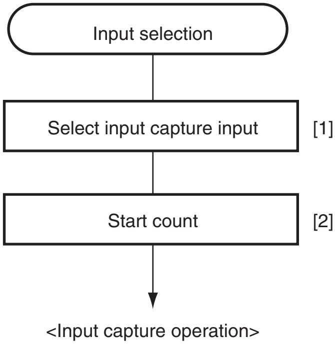


Figure 14.8 Example of Toggle Output Operation

(a) Example of setting procedure for input capture operation

Figure 14.9 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 14.9 Example of Setting Procedure for Input Capture Operation

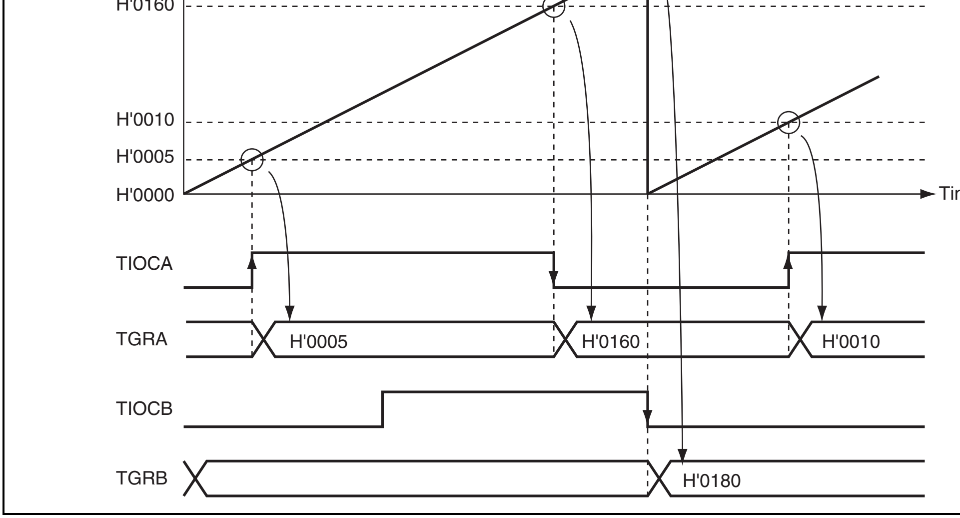
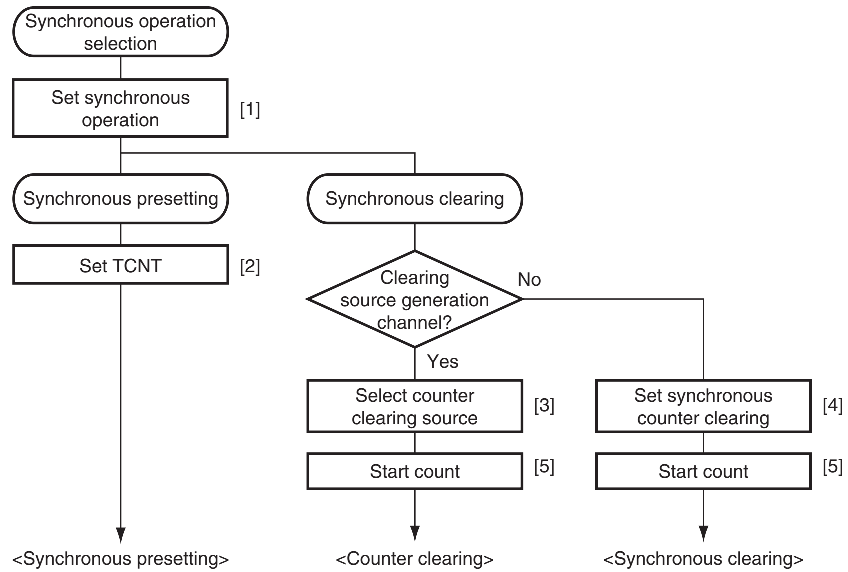


Figure 14.10   Example of Input Capture Operation

Figure 14.11 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 14.11 Example of Synchronous Operation Setting Procedure

For details on PWM modes, see section 14.4.5, PWM Modes.

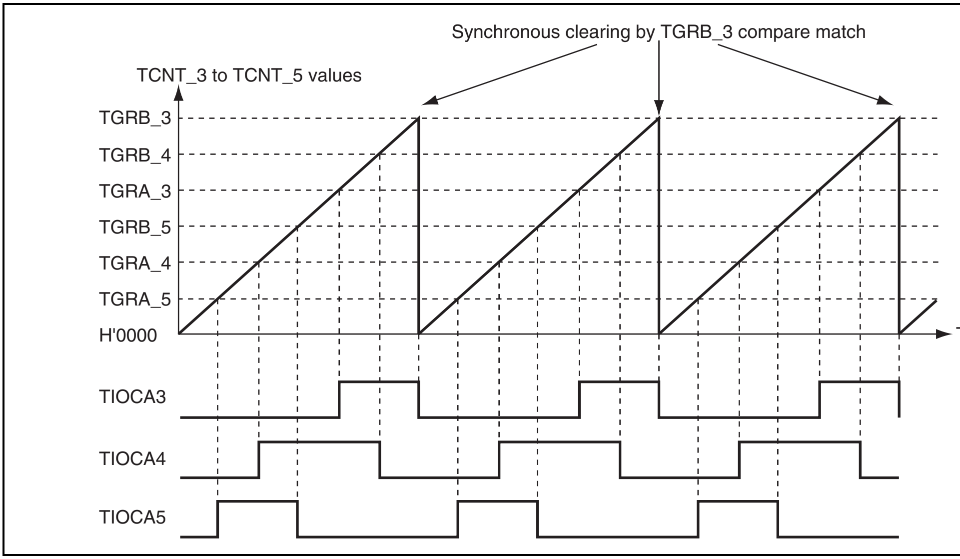


Figure 14.12 Example of Synchronous Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
6	TGRA_6	TGRC_6
	TGRB_6	TGRD_6
9	TGRA_9	TGRC_9
	TGRB_9	TGRD_9

- 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 14.13.

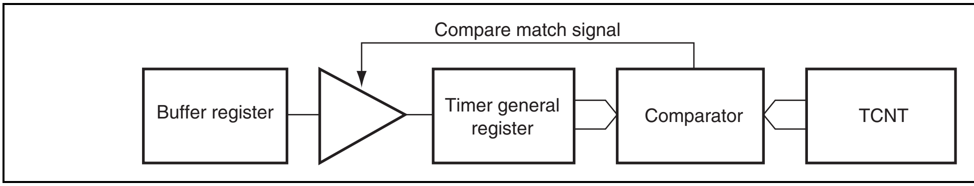


Figure 14.13 Compare Match Buffer Operation

Figure 14.14 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 14.15 shows an example of the buffer operation setting procedure.

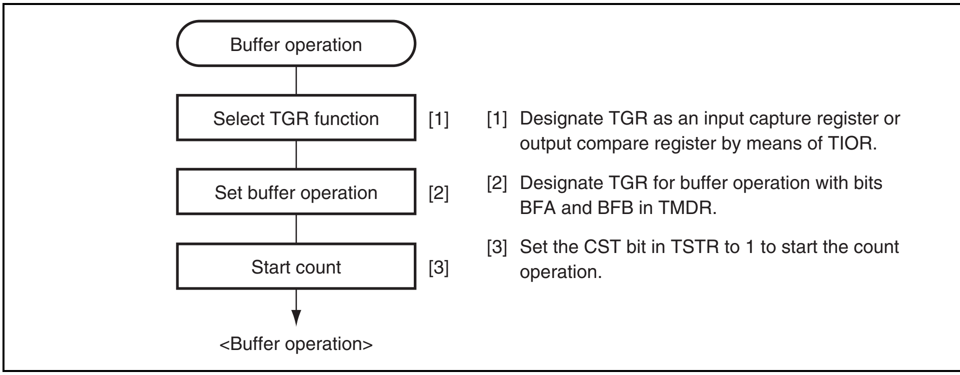


Figure 14.15 Example of Buffer Operation Setting Procedure

For details on PWM modes, see section 14.4.5, PWM Modes.

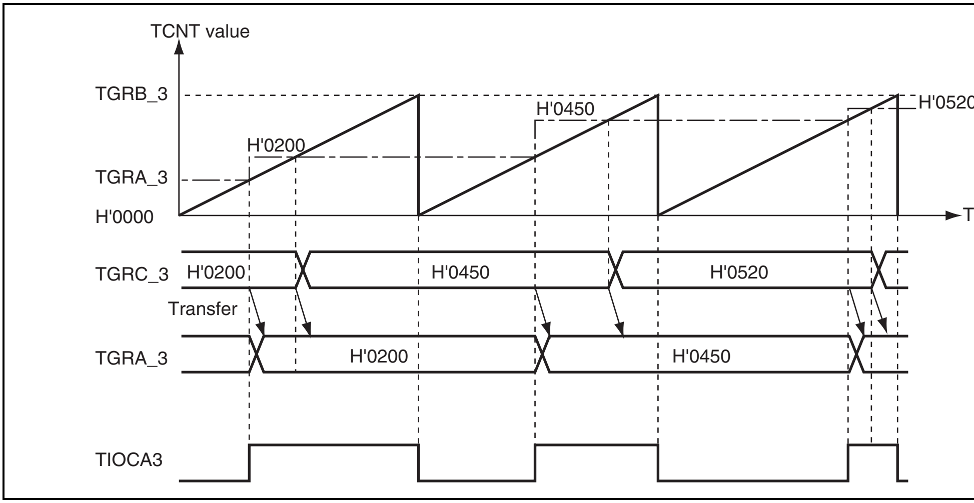


Figure 14.16 Example of Buffer Operation (1)

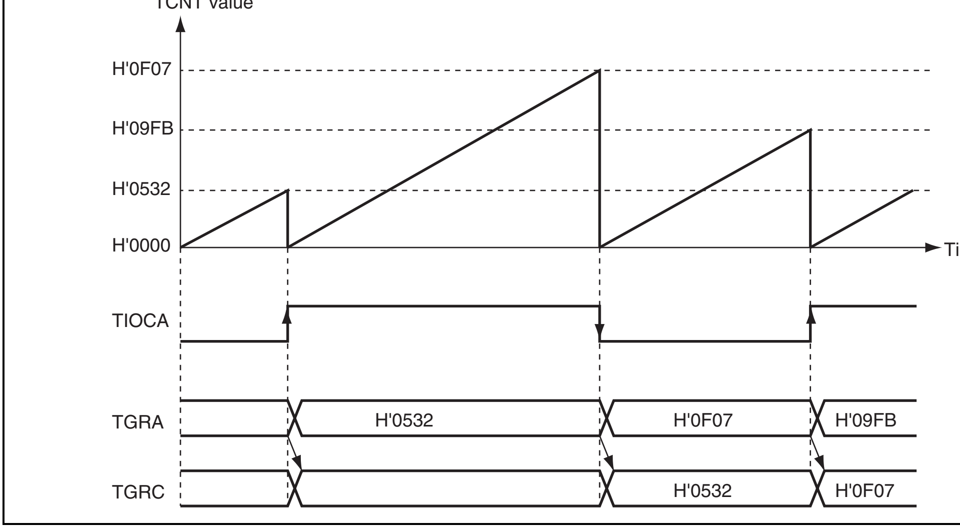


Figure 14.17 Example of Buffer Operation (2)

Table 14.47 shows the register combinations used in cascaded operation.

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

Table 14.47 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
Channels 7 and 8	TCNT_7	TCNT_8
Channels 10 and 11	TCNT_10	TCNT_11

(1) Example of Cascaded Operation Setting Procedure

Figure 14.18 shows an example of the setting procedure for cascaded operation.

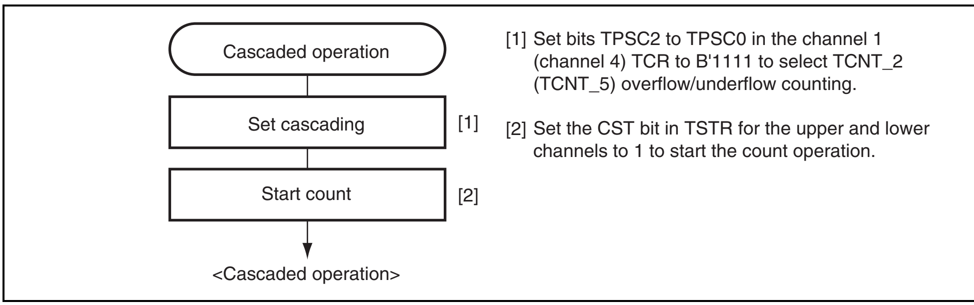


Figure 14.18 Cascaded Operation Setting Procedure

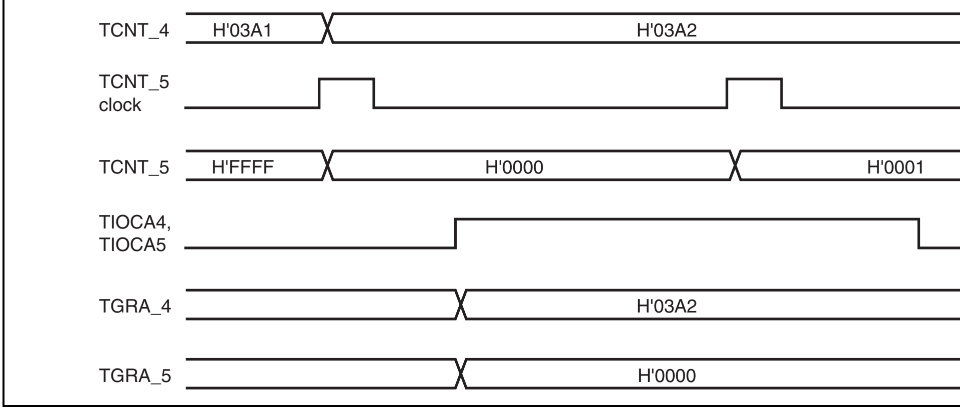


Figure 14.19 Example of Cascaded Operation (1)

Figure 14.20 illustrates the operation when counting upon TCNT_5 overflow/underflow
set for TCNT_4, and phase counting mode has been designated for channel 5.

TCNT_4 is incremented by TCNT_5 overflow and decremented by TCNT_5 underflow

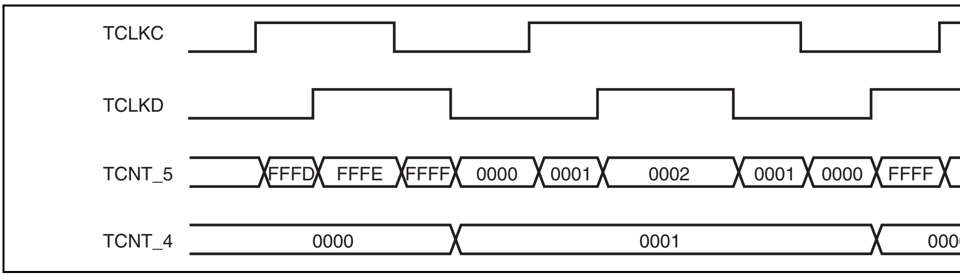


Figure 14.20 Example of Cascaded Operation (2)

There are two PWM modes, as described below.

1. PWM mode 1

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

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

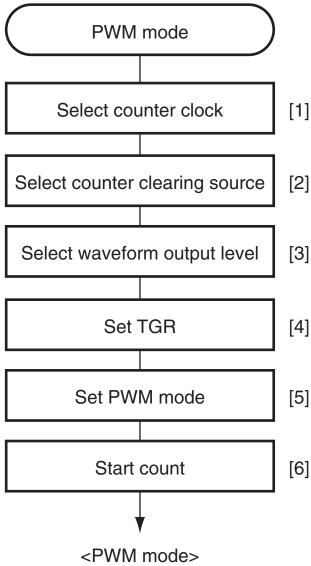
2. PWM mode 2

   PWM output is generated using one TGR as the cycle register and the others as duty c
   registers. The output specified in TIOR is performed by means of compare matches. U
   counter clearing by a synchronous register compare match, the output value of each p
   initial value set in TIOR. If the set values of the cycle and duty cycle registers are ider
   output value does not change when a compare match occurs.

   In PWM mode 2, a maximum 15-phase PWM output is possible by combined use wit
   synchronous operation.

   The correspondence between PWM output pins and registers is shown in table 14.48.

2	TGRA_2	—	—
	TGRB_2		
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
6	TGRA_6	TIOCA6	TIOCA6
	TGRB_6		TIOCB6
	TGRC_6	TIOCC6	TIOCC6
	TGRD_6		TIOCD6
7	TGRA_7	TIOCA7	TIOCA7
	TGRB_7		TIOCB7
8	TGRA_8	TIOCA8	TIOCA8
	TGRB_8		TIOCB8
9	TGRA_9	TIOCA9	TIOCA9
	TGRB_9		TIOCB9
	TGRC_9	TIOCC9	TIOCC9
	TGRD_9		TIOCD9
10	TGRA_10	TIOCA10	TIOCA10
	TGRB_10		TIOCB10



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] Use bits CCLR2 to CCLR0 in TCR to select the TGR to be used as the TCNT clearing source.

[3] Use TIOR to designate 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 14.21 Example of PWM Mode Setting Procedure

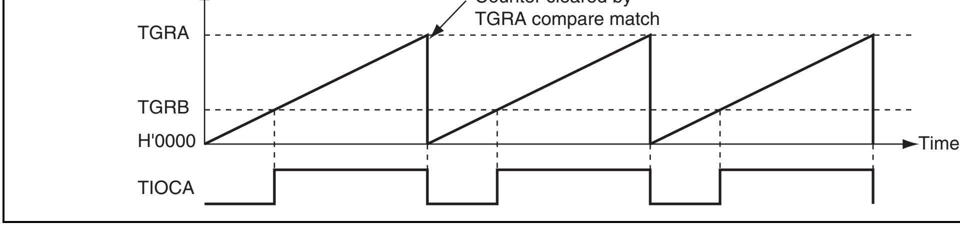


Figure 14.22 Example of PWM Mode Operation (1)

Figure 14.23 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 3 and 4, TGRB_4 cor match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 f output value of the other TGR registers (TGRA_3 to TGRD_3, TGRA_4), to output a 5- PWM waveform.

In this case, the value set in TGRB_4 is used as the cycle, and the values set in the other the duty cycle.

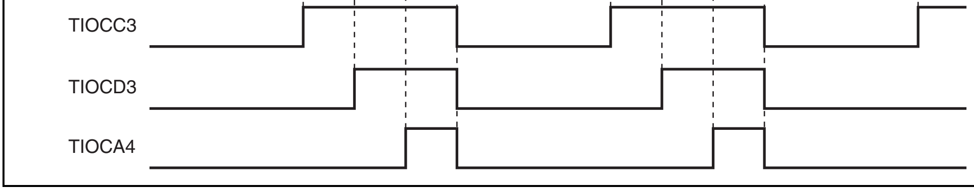


Figure 14.23 Example of PWM Mode Operation (2)

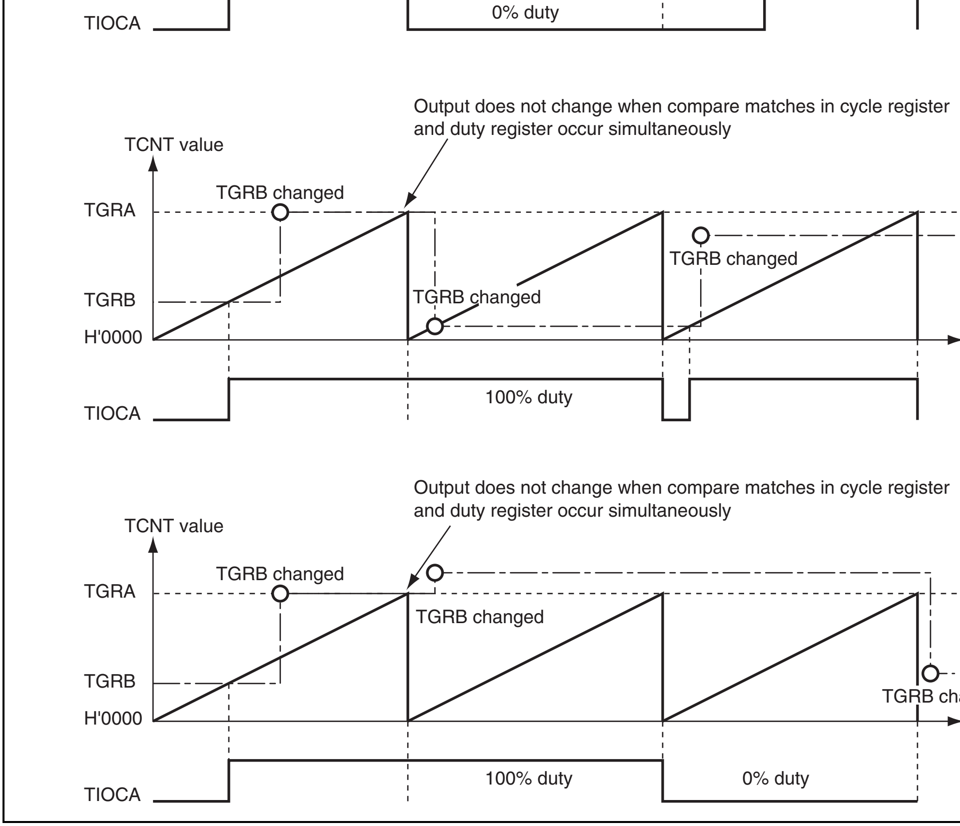


Figure 14.24 Example of PWM Mode Operation (3)

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 14.49 shows the correspondence between external clock pins and channels.

Table 14.49 Clock Input Pins in Phase Counting Mode

	External Clock Pin	
Channels	**A-Phase**	**B-Phase**
When channel 1, 5, 7, or 11 is set to phase counting mode	TCLKA	TCLKB
When channel 2, 4, 8, or 10 is set to phase counting mode	TCLKC	TCLKD

Figure 14.25 Example of Phase Counting Mode Setting Procedure

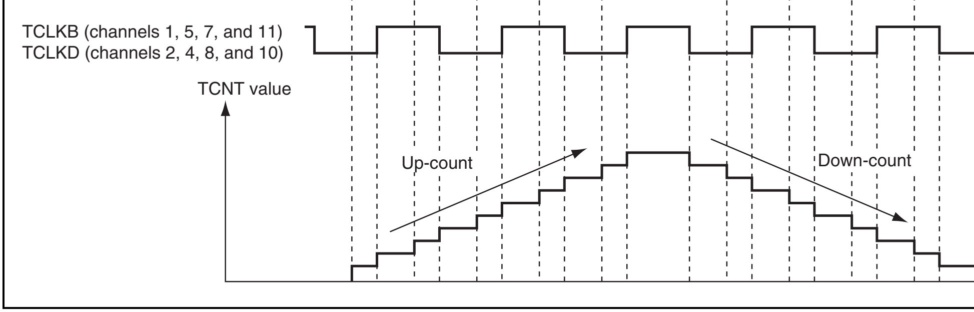


Figure 14.26 Example of Phase Counting Mode 1 Operation

Table 14.50 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1, 5, 7, and 11) TCLKC (Channels 2, 4, 8, and 10)	TCLKB (Channels 1, 5, 7, and 11) TCLKD (Channels 2, 4, 8, and 10)	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

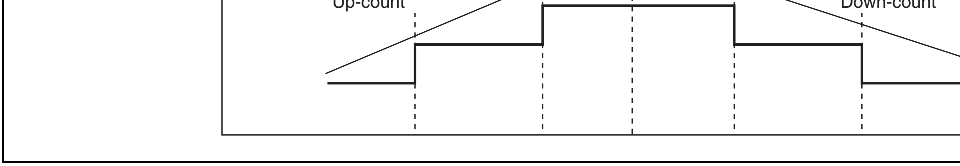


Figure 14.27 Example of Phase Counting Mode 2 Operation

Table 14.51 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1, 5, 7, and 11) TCLKC (Channels 2, 4, 8, and 10)	TCLKB (Channels 1, 5, 7, and 11) TCLKD (Channels 2, 4, 8, and 10)	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

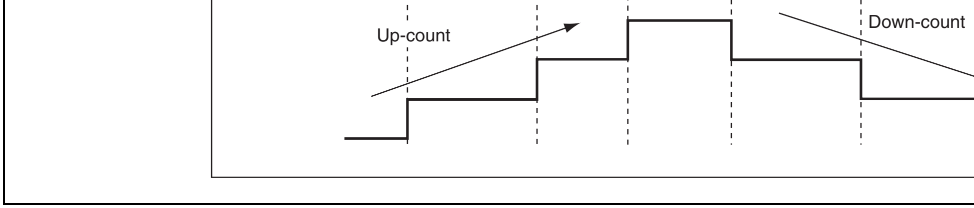


Figure 14.28 Example of Phase Counting Mode 3 Operation

Table 14.52 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1, 5, 7, and 11) TCLKC (Channels 2, 4, 8, and 10)	TCLKB (Channels 1, 5, 7, and 11) TCLKD (Channels 2, 4, 8, and 10)	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

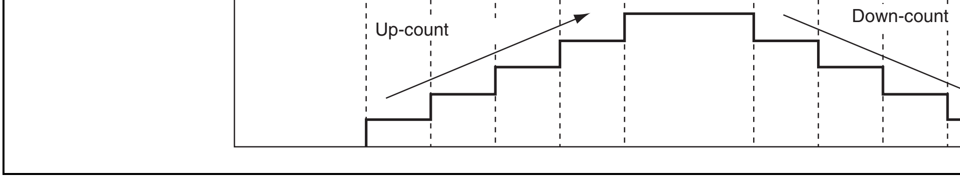


Figure 14.29 Example of Phase Counting Mode 4 Operation

Table 14.53 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1, 5, 7, and 11) TCLKC (Channels 2, 4, 8, and 10)	TCLKB (Channels 1, 5, 7, and 11) TCLKD (Channels 2, 4, 8, and 10)	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 op
in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input cap
source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, channel 0 TGRA_
TGRC_0 compare matches are selected as the input capture source, and the up/down-cou
values for the control cycles are stored.

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

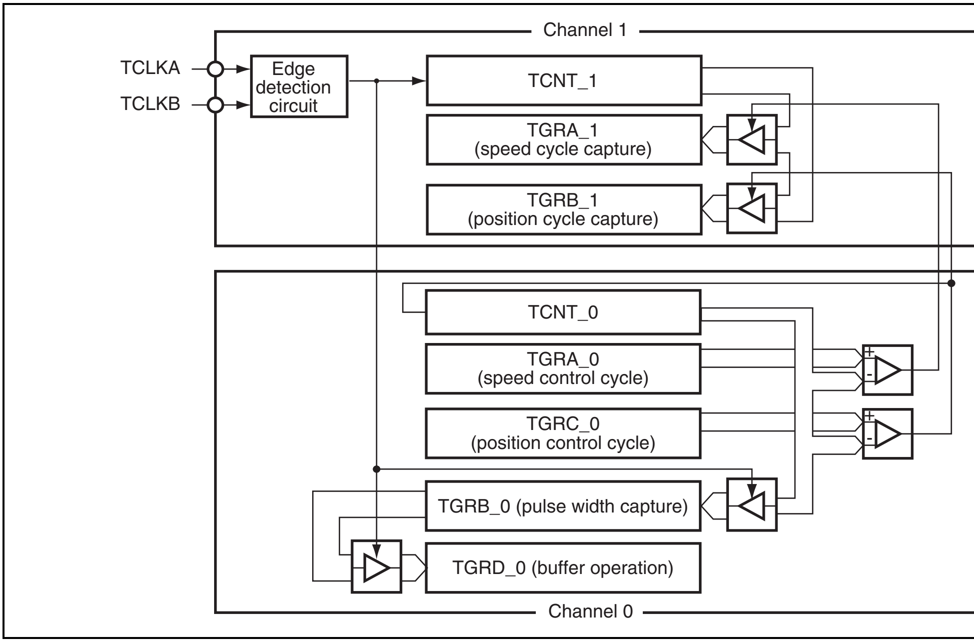


Figure 14.30 Phase Counting Mode Application Example

channel is fixed. For details, see section 7, Interrupt Controller.

Table 14.54 lists the TPU interrupt sources.

Table 14.54 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMA Acti
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	Possible	Poss
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	Possible	Not
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	Possible	Not
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	Possible	Not
	TCI0V	TCNT_0 overflow	TCFV_0	Not possible	Not
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Possible	Poss
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Possible	Not
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible	Not
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible	Not
2	TGI2A	TGRA_2 compare match	TGFA_2	Possible	Poss
	TGI2B	TGRB_2 compare match	TGFB_2	Possible	Not
	TCI2V	TCNT_2 overflow	TCFV_2	Not possible	Not
	TCI2U	TCNT_2 underflow	TCFU_2	Not possible	Not

	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	Possi
	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
6	TGI6A	TGRA_6 input capture/compare match	TGFA_6	Possible	Possi
	TGI6B	TGRB_6 input capture/compare match	TGFB_6	Possible	Not po
	TGI6C	TCRC_6 input capture/compare match	TGFC_6	Possible	Not po
	TGI6D	TCRD_6 input capture/compare match	TGFD_6	Possible	Not po
7	TGI7A	TGRA_7 input capture/compare match	TGFA_7	Possible	Possi
	TGI7B	TGRB_7 input capture/compare match	TGFB_7	Possible	Not po
	TCI7V	TCNT_7 overflow	TCFV_7	Not possible	Not po
	TCI7U	TCNT_7 underflow	TCFU_7	Not possible	Not po
8	TGI8A	TGRA_8 input capture/compare match	TGFA_8	Possible	Possi
	TGI8B	TGRB_8 input capture/compare match	TGFB_8	Possible	Not po
	TCI8V	TCNT_8 overflow	TCFV_8	Not possible	Not po
	TCI8U	TCNT_8 underflow	TCFU_8	Not possible	Not po
9	TGI9A	TGRA_9 input capture/compare match	TGFA_9	Possible	Possi
	TGI9B	TGRB_9 input capture/compare match	TGFB_9	Possible	Not po
	TGI9C	TGRC_9 input capture/compare match	TGFC_9	Possible	Not po
	TGI9D	TGRD_9 input capture/compare match	TGFD_9	Possible	Not po
	TCI9V	TCNT_9 overflow	TCFV_9	Not possible	Not po

TCI11U	TCNT_11 underflow	TCFU_11	Not possible	Not

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

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is by the occurrence of a TGR input capture/compare match on a channel. The interrupt re cancelled by clearing the TGF flag to 0. The TPU has 32 input capture/compare match i four each for channels 0, 3, 6, and 9, and two each for channels 1, 2, 4, 5, 7, 8, 10, and 1

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TS 1 by the occurrence of a TCNT overflow on a channel. The interrupt request is cancelle clearing the TCFV flag to 0. The TPU has s12 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 TS 1 by the occurrence of a TCNT underflow on a channel. The interrupt request is cancelle clearing the TCFU flag to 0. The TPU has eight underflow interrupts, one each for chan 4, 5, 7, 8, 10, and 11.

The DMAC can be activated by the TGRA input capture/compare match interrupt for a c
For details, see section 10, DMA Controller (DMAC).

In TPU, one in each channel, totally 12 TGRA input capture/compare match interrupts ca
as DMAC activation sources.

14.8 A/D Converter Activation

Concerning the unit 0 in TPU, the TGRA input capture/compare match for each channel
activate the A/D converter. (However, the A/D converter cannot be activated in unit 1.)

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurre
TGRA input capture/compare match on a particular channel, a request to start A/D conve
sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/
converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A
converter conversion start sources, one for each channel of unit 0.

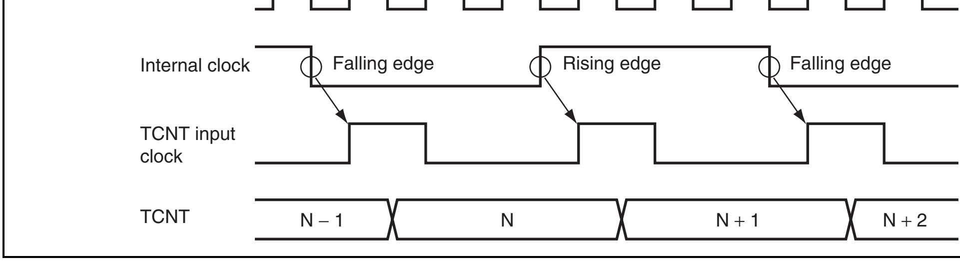


Figure 14.31 Count Timing in Internal Clock Operation

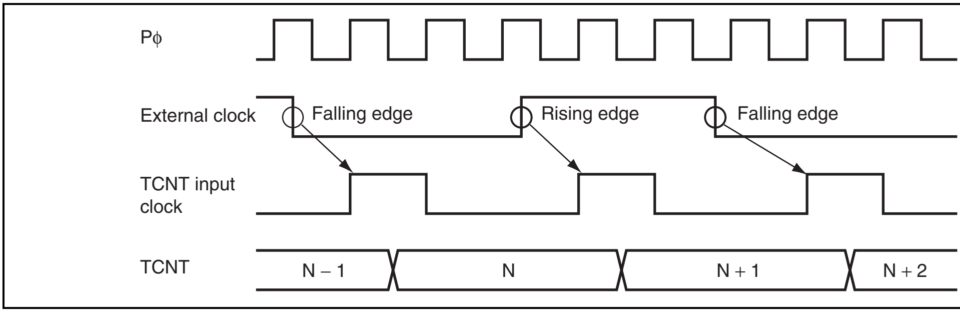


Figure 14.32 Count Timing in External Clock Operation

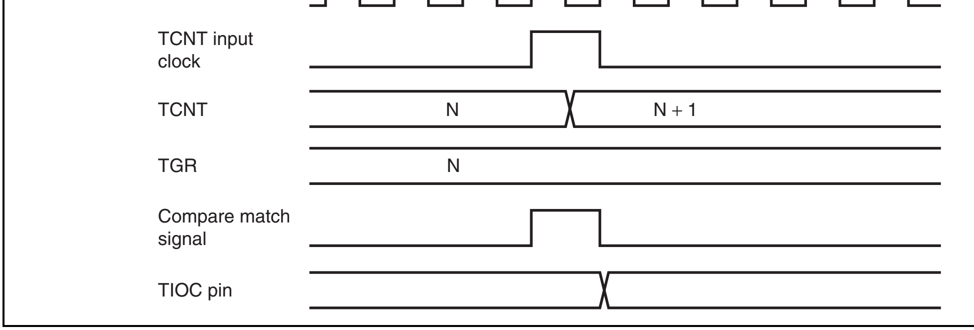


Figure 14.33 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 14.34 shows input capture signal timing.

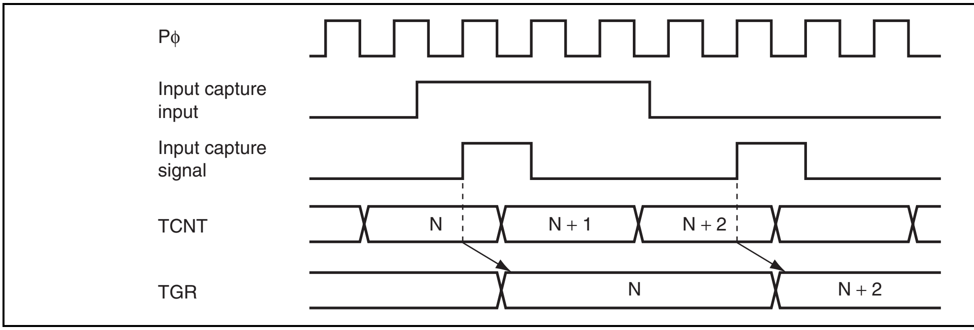


Figure 14.34 Input Capture Input Signal Timing

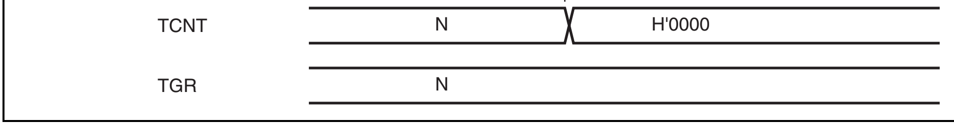


Figure 14.35 Counter Clear Timing (Compare Match)

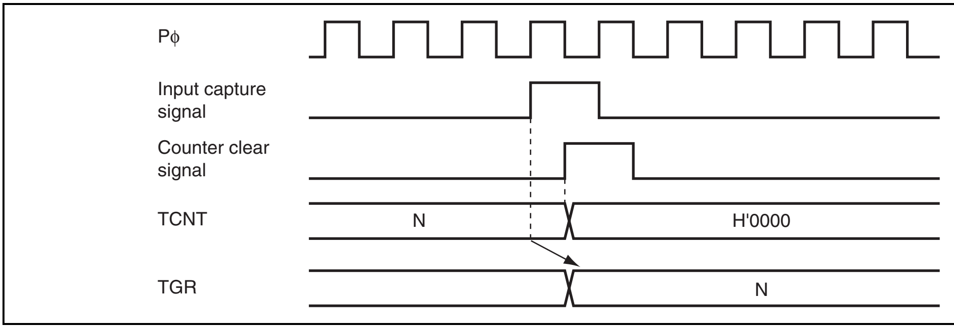


Figure 14.36 Counter Clear Timing (Input Capture)

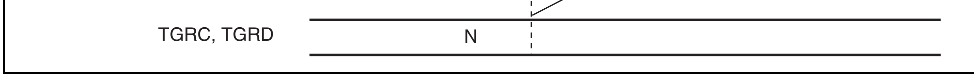


Figure 14.37 Buffer Operation Timing (Compare Match)

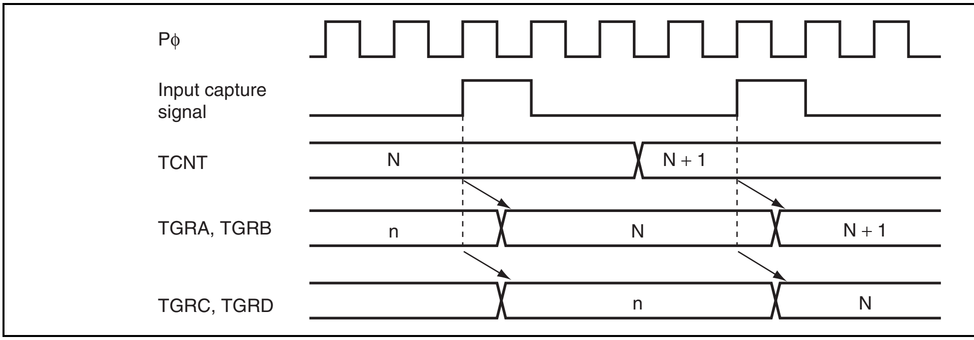


Figure 14.38 Buffer Operation Timing (Input Capture)

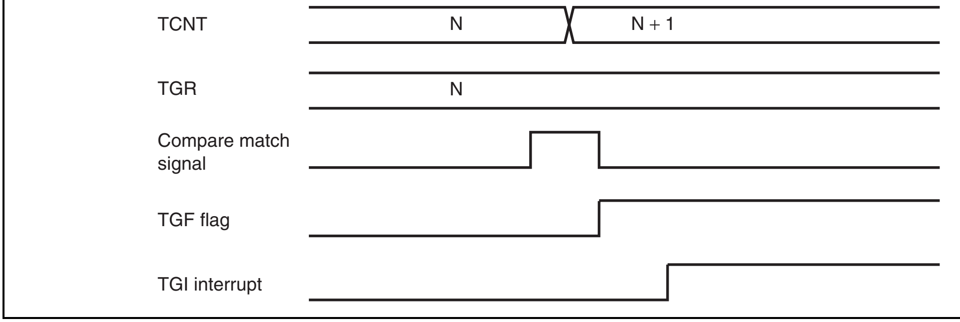


Figure 14.39 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 14.40 shows the timing for setting of the TGF flag in TSR by input capture occur
the TGI interrupt request signal timing.

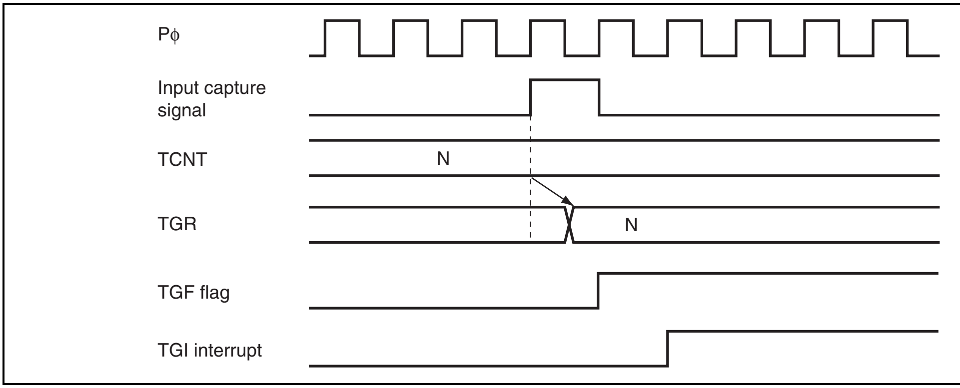


Figure 14.40 TGI Interrupt Timing (Input Capture)

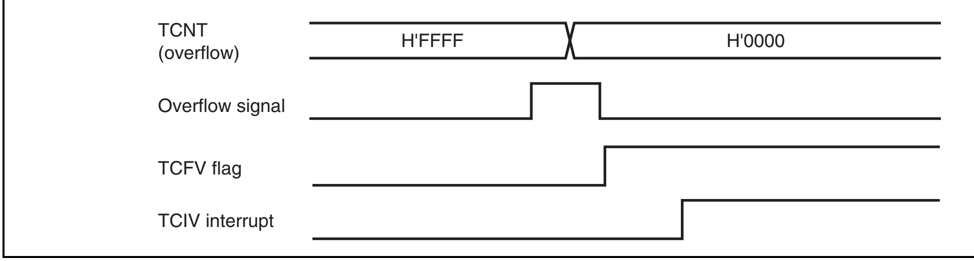


Figure 14.41 TCIV Interrupt Setting Timing

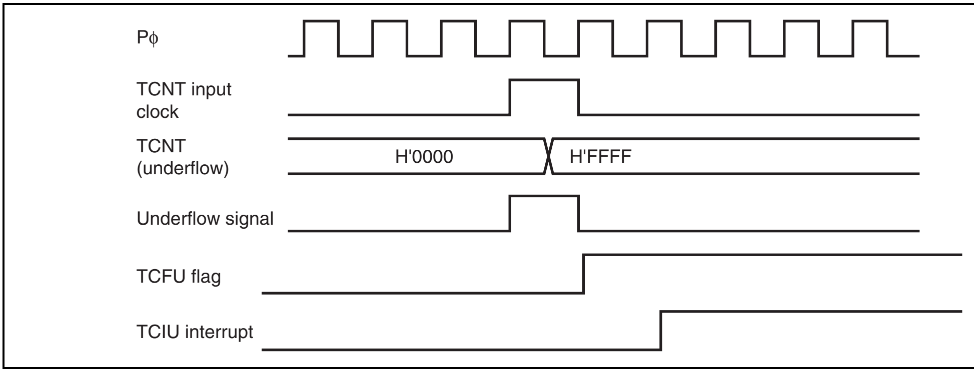


Figure 14.42 TCIU Interrupt Setting Timing

Figure 14.43 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 14.44. 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 14.45. 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 14.44 Timing for Status Flag Clearing by DTC/DMAC Activation (1

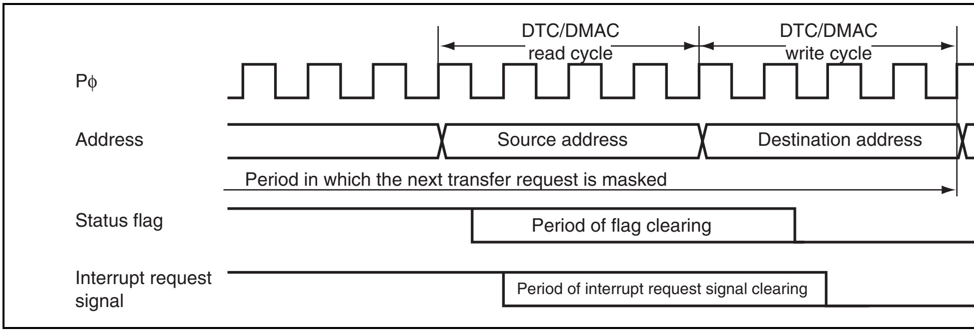


Figure 14.45 Timing for Status Flag Clearing by DTC/DMAC Activation (2

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 14.46 shows the in conditions in phase counting mode.

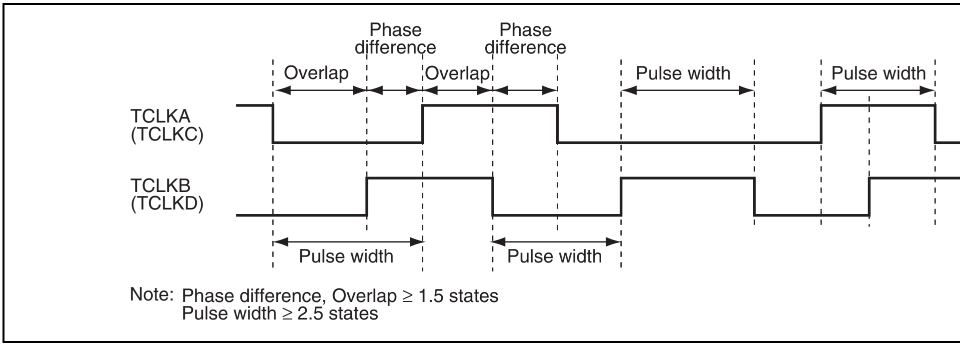


Figure 14.46 Phase Difference, Overlap, and Pulse Width in Phase Counting

N: TGR set value

14.10.4 Conflict between TCNT Write and Clear Operations

If the counter-clearing signal is generated in the T2 state of a TCNT write cycle, TCNT c
takes precedence and the TCNT write is not performed. Figure 14.47 shows the timing in
case.

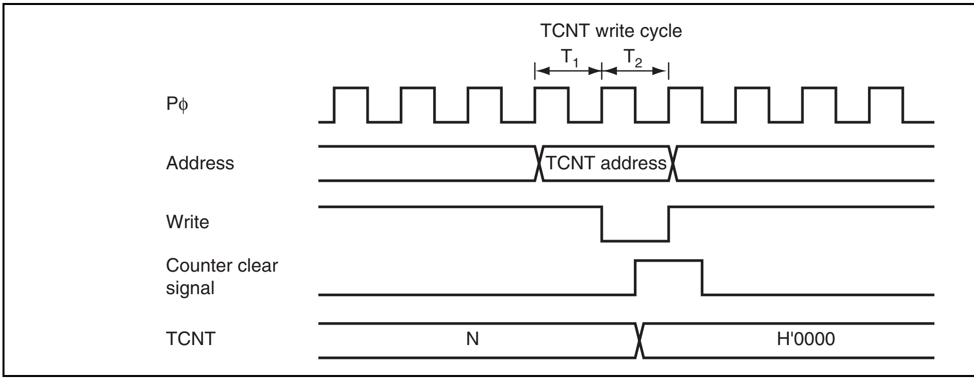


Figure 14.47 Conflict between TCNT Write and Clear Operations

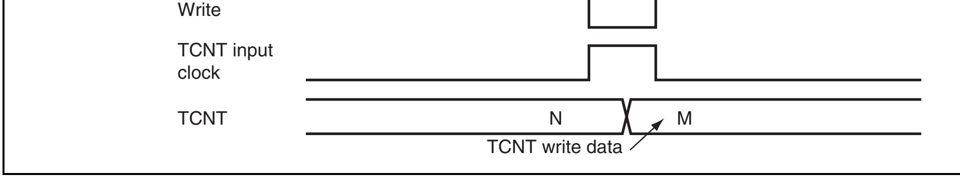


Figure 14.48 Conflict between TCNT Write and Increment Operations

14.10.6 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 14.49 shows the timing in this case.

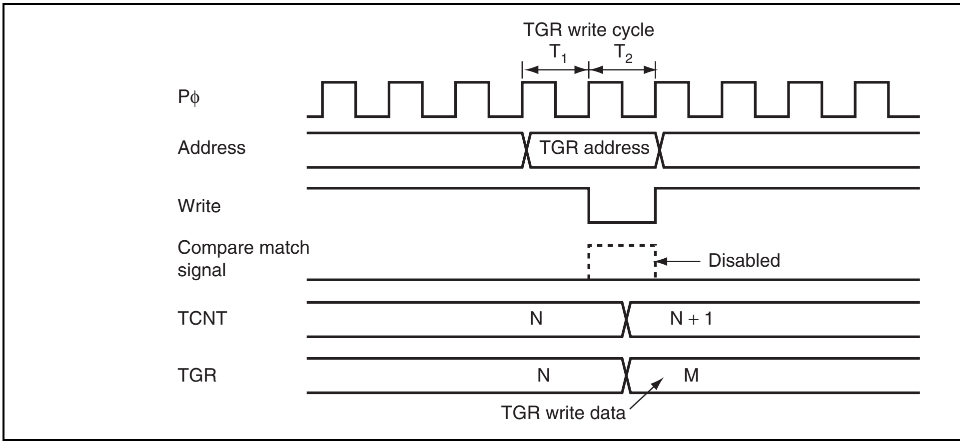


Figure 14.49 Conflict between TGR Write and Compare Match

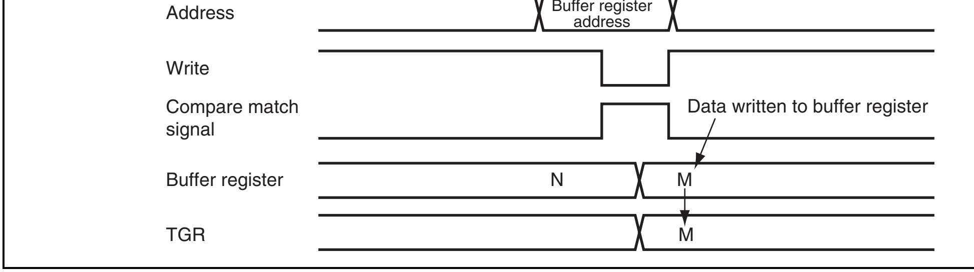


Figure 14.50 Conflict between Buffer Register Write and Compare Match

14.10.8 Conflict between TGR Read and Input Capture

If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is
will be the data after input capture transfer.

Figure 14.51 shows the timing in this case.

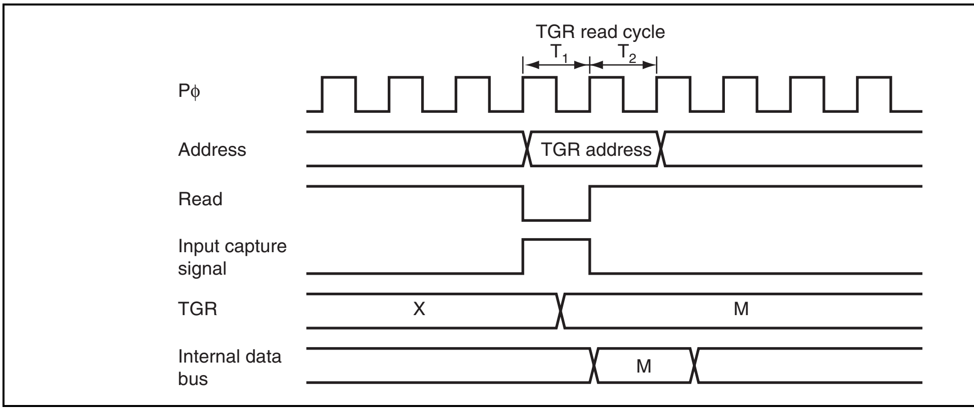


Figure 14.51 Conflict between TGR Read and Input Capture

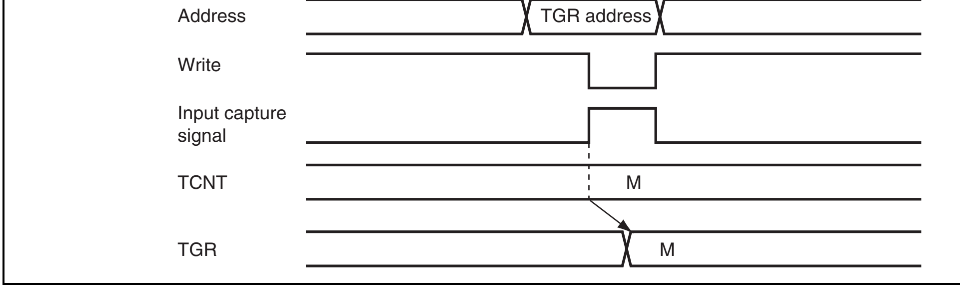


Figure 14.52 Conflict between TGR Write and Input Capture

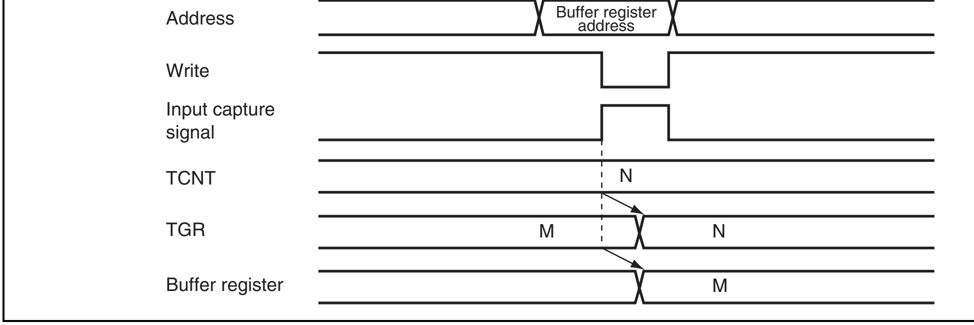


Figure 14.53 Conflict between Buffer Register Write and Input Capture

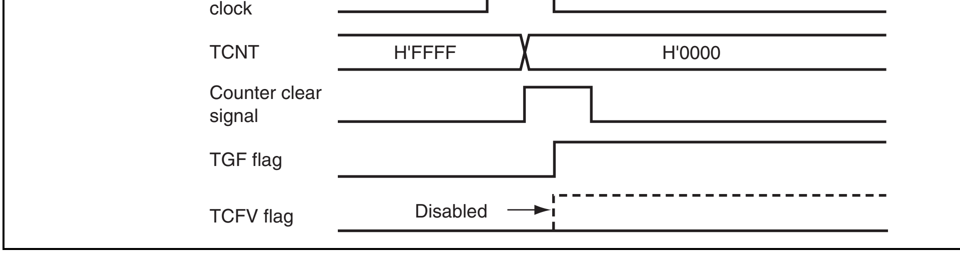


Figure 14.54 Conflict between Overflow and Counter Clearing

14.10.12 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 14.55 shows the operation timing when there is conflict between TCNT write anc overflow.

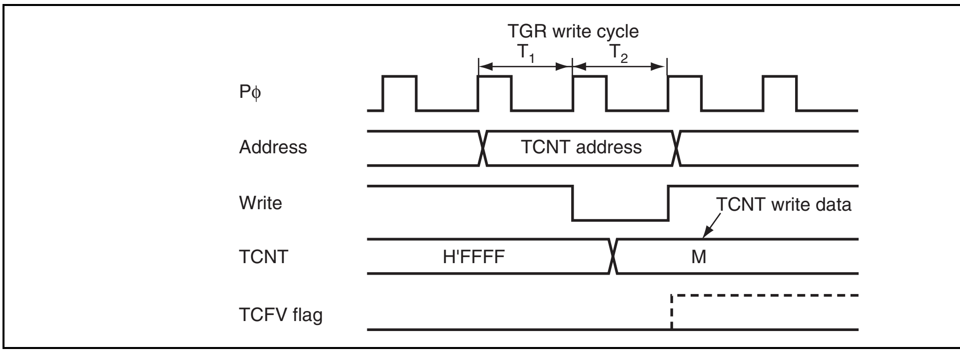


Figure 14.55 Conflict between TCNT Write and Overflow

- Four output groups
- Selectable output trigger signals
- Non-overlapping mode
- Can operate together with the data transfer controller (DTC) and DMA controller (D
- Inverted output can be set
- Module stop state specifiable

Table 15.1 List of PPG Functions

Function			PPG0	PPG1
PPG output trigger	TPU0	Compare match	Possible	Not possibl
		Input capture	Possible	Not possibl
	TPU1	Compare match	Not possible	Possible
		Input capture	Not possible	Not possibl
Non-overlapping mode			Possible	Possible
Output data transfer		DTC	Possible	Possible
		DMAC	Possible	Possible
Inverted output			Possible	Possible

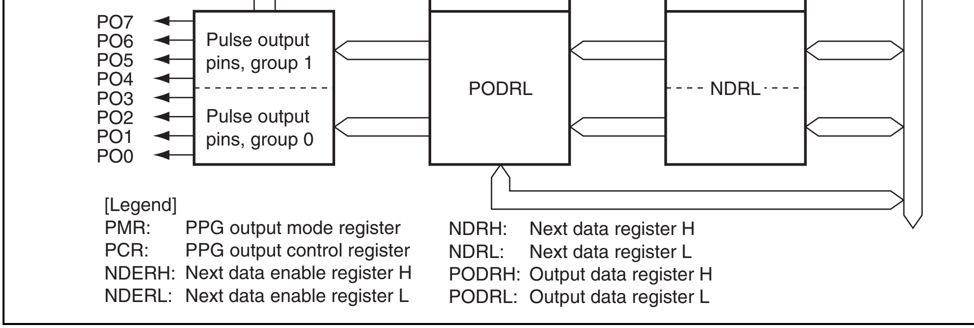


Figure 15.1 Block Diagram of PPG (Unit 0)

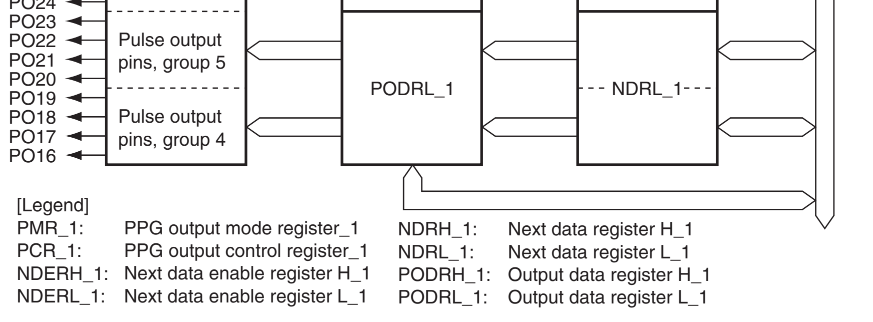


Figure 15.2 Block Diagram of PPG (Unit 1)

	PO3	Output	
	PO4	Output	Group 1 pulse output
	PO5	Output	
	PO6	Output	
	PO7	Output	
1	PO16	Output	Group 4 pulse output
	PO17	Output	
	PO18	Output	
	PO19	Output	
	PO20	Output	Group 5 pulse output
	PO21	Output	
	PO22	Output	
	PO23	Output	
	PO24	Output	Group 6 pulse output
	PO25	Output	
	PO26	Output	
	PO27	Output	
	PO28	Output	Group 7 pulse output
	PO29	Output	
	PO30	Output	
	PO31	Output	

- Next data register H (NDRH)
- Next data register L (NDRL)
- PPG output control register (PCR)
- PPG output mode register (PMR)

Unit 1:

- Next data enable register H_1 (NDERH_1)
- Next data enable register L_1 (NDERL_1)
- Output data register H_1 (PODRH_1)
- Output data register L_1 (PODRL_1)
- Next data register H_1 (NDRH_1)
- Next data register L_1 (NDRL_1)
- PPG output control register_1 (PCR_1)
- PPG output mode register_1 (PMR_1)

• NDERL

Bit	7	6	5	4	3	2	1	
Bit Name	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	N
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

• NDERH

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 15 to 8
6	NDER14	0	R/W	These are read-only bits and cannot be modified
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

1	NDER1	0	R/W	
0	NDER0	0	R/W	

- NDERH_1

Bit	Bit Name	Initial Value	R/W	Description
7	NDER31	0	R/W	Next Data Enable 31 to 24
6	NDER30	0	R/W	When a bit is set to 1, the value in the correspo NDRH_1 bit is transferred to the PODRH_1 bit selected output trigger. Values are not transferr NDRH_1 to PODRH_1 for cleared bits.
5	NDER29	0	R/W	
4	NDER28	0	R/W	
3	NDER27	0	R/W	
2	NDER26	0	R/W	
1	NDER25	0	R/W	
0	NDER24	0	R/W	

1	NDER17	0	R/W
0	NDER16	0	R/W

15.3.2 Output Data Registers H, L (PODRH, PODRL)

PODRH and PODRL store output data for use in pulse output. A bit that has been set for output by NDER is read-only and cannot be modified.

- PODRH

Bit	7	6	5	4	3	2	1
Bit Name	POD15	POD14	POD13	POD12	POD11	POD10	POD9
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- PODRL

Bit	7	6	5	4	3	2	1
Bit Name	POD7	POD6	POD5	POD4	POD3	POD2	POD1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

1	POD9	0	R/W	
0	POD8	0	R/W	

- PODRL

Bit	Bit Name	Initial Value	R/W	Description
7	POD7	0	R/W	Output Data Register 7 to 0
6	POD6	0	R/W	For bits which have been set to pulse output by
5	POD5	0	R/W	the output trigger transfers NDRL values to this
4	POD4	0	R/W	during PPG operation. While NDERL is set to 1
3	POD3	0	R/W	cannot write to this register. While NDERL is cl
2	POD2	0	R/W	initial output value of the pulse can be set.
1	POD1	0	R/W	
0	POD0	0	R/W	

1	POD25	0	R/W	
0	POD24	0	R/W	

- PODRL_1

Bit	Bit Name	Initial Value	R/W	Description
7	POD23	0	R/W	Output Data Register 23 to 16
6	POD22	0	R/W	For bits which have been set to pulse output by NDERL_1, the output trigger transfers NDRL_1 v this register during PPG operation. While NDERI to 1, the CPU cannot write to this register. While NDERL_1 is cleared, the initial output value of th can be set.
5	POD21	0	R/W	
4	POD20	0	R/W	
3	POD19	0	R/W	
2	POD18	0	R/W	
1	POD17	0	R/W	
0	POD16	0	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

- NDRH

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mappe same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 8
6	NDR14	0	R/W	These are read-only bits and cannot be modifie
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
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 lo
bits are mapped to different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 4
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be r

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be r
3	NDR3	0	R/W	Next Data Register 3 to 0
2	NDR2	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

3	NDR27	0	R/W	
2	NDR26	0	R/W	
1	NDR25	0	R/W	
0	NDR24	0	R/W	

If pulse output groups 6 and 7 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	NDR31	0	R/W	Next Data Register 31 to 28
6	NDR30	0	R/W	The register contents are transferred to the corresponding PODRH_1 bits by the output trig specified with PCR_1.
5	NDR29	0	R/W	
4	NDR28	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	NDR27	0	R/W	Next Data Register 27 to 24
2	NDR26	0	R/W	The register contents are transferred to the corresponding PODRH_1 bits by the output trig specified with PCR_1.
1	NDR25	0	R/W	
0	NDR24	0	R/W	

3	NDR19	0	R/W	
2	NDR18	0	R/W	
1	NDR17	0	R/W	
0	NDR16	0	R/W	

If pulse output groups 4 and 5 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	NDR23	0	R/W	Next Data Register 23 to 20
6	NDR22	0	R/W	The register contents are transferred to the corresponding PODRL_1 bits by the output trigg specified with PCR_1.
5	NDR21	0	R/W	
4	NDR20	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	NDR19	0	R/W	Next Data Register 19 to 16
2	NDR18	0	R/W	The register contents are transferred to the corresponding PODRL_1 bits by the output trigg specified with PCR_1.
1	NDR17	0	R/W	
0	NDR16	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 1 and 0
6	G3CMS0	1	R/W	These are read-only bits and cannot be modifie
5	G2CMS1	1	R/W	Group 2 Compare Match Select 1 and 0
4	G2CMS0	1	R/W	These are read-only bits and cannot be modifie
3	G1CMS1	1	R/W	Group 1 Compare Match Select 1 and 0
2	G1CMS0	1	R/W	These bits select output trigger of pulse output 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
1	G0CMS1	1	R/W	Group 0 Compare Match Select 1 and 0
0	G0CMS0	1	R/W	These bits select output trigger of pulse output 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3

4	G2CMS0	1	R/W	These bits select output trigger of pulse output g 00: Compare match in TPU channel 6 01: Compare match in TPU channel 7 10: Compare match in TPU channel 8 11: Compare match in TPU channel 9
3 2	G1CMS1 G1CMS0	1 1	R/W R/W	Group 5 Compare Match Select 1 and 0 These bits select output trigger of pulse output g 00: Compare match in TPU channel 6 01: Compare match in TPU channel 7 10: Compare match in TPU channel 8 11: Compare match in TPU channel 9
1 0	G0CMS1 G0CMS0	1 1	R/W R/W	Group 4 Compare Match Select 1 and 0 These bits select output trigger of pulse output g 00: Compare match in TPU channel 6 01: Compare match in TPU channel 7 10: Compare match in TPU channel 8 11: Compare match in TPU channel 9

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

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion These are read-only bits and cannot be modifie
6	G2INV	1	R/W	Group 2 Inversion These are read-only bits and cannot be modifie
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

				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

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

				Selects normal or non-overlapping operation for output group 6. 0: Normal operation (output values updated by c match A on the selected TPU channel) 1: Non-overlapping operation (output values upd compare match A or B on the selected TPU ch
1	G1NOV	0	R/W	Group 5 Non-Overlap Selects normal or non-overlapping operation for output group 5. 0: Normal operation (output values updated by c match A on the selected TPU channel) 1: Non-overlapping operation (output values upd compare match A or B on the selected TPU ch
0	G0NOV	0	R/W	Group 4 Non-Overlap Selects normal or non-overlapping operation for output group 4. 0: Normal operation (output values updated by c match A on the selected TPU channel) 1: Non-overlapping operation (output values upd compare match A or B on the selected TPU ch

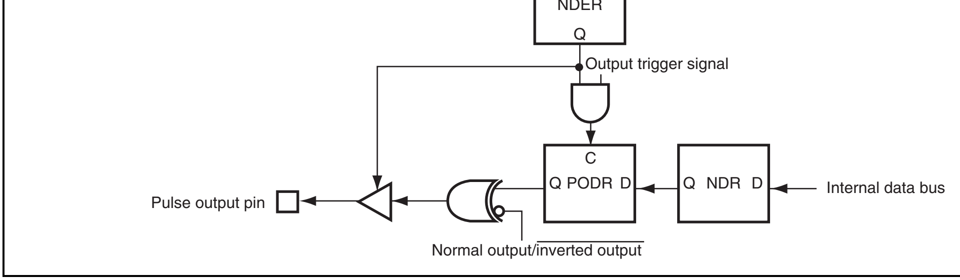


Figure 15.3 Schematic Diagram of PPG

15.4.1 Output Timing

If pulse output is enabled, the NDR contents are transferred to PODR and output when t
specified compare match event occurs. Figure 15.4 shows the timing of these operations
case of normal output in groups 2 and 3, triggered by compare match A.

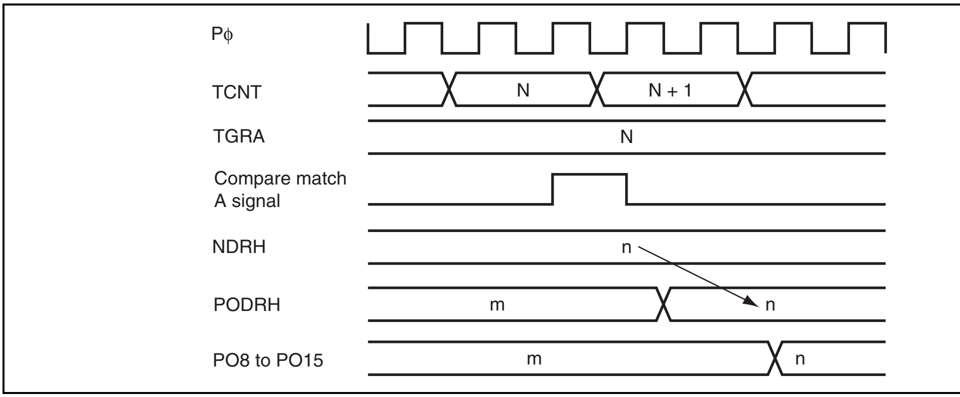
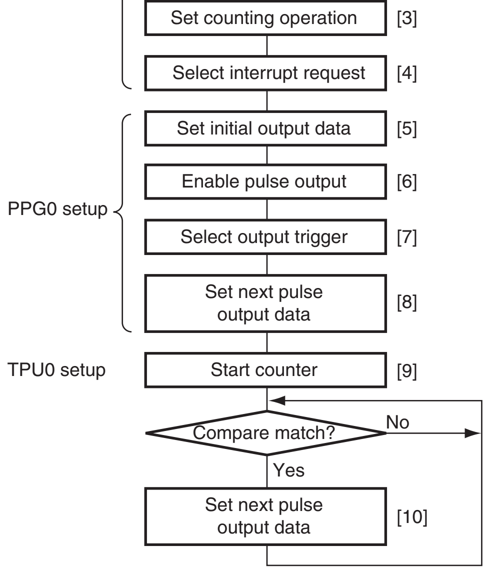


Figure 15.4 Timing of Transfer and Output of NDR Contents (Example)



counter clear source with bits CCLR1 a
CCLR0.

[4] Enable the TGIA interrupt in TIER. The
DTC or DMAC can also be set up to
transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the bits in NDER for the pins to be
used for pulse output to 1.

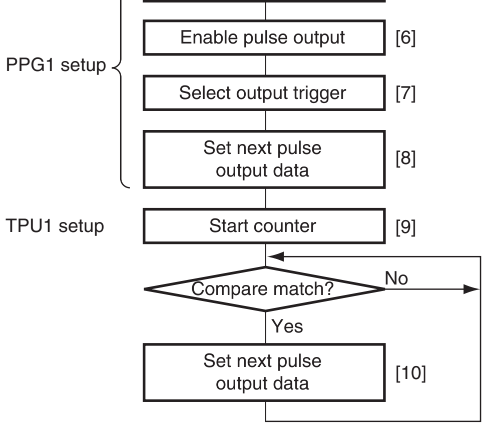
[7] Select the TPU compare match event to
be used as the output trigger in PCR.

[8] Set the next pulse output values in NDF

[9] Set the CST bit in TSTR to 1 to start the
TCNT counter.

[10] At each TGIA interrupt, set the next

Figure 15.5 Setup Procedure for Normal Pulse Output (PPG0)



[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 th TCNT counter.

[10] At each TGIA interrupt, set the next output values in NDR.

Figure 15.6 Setup Procedure for Normal Pulse Output (PPG1)

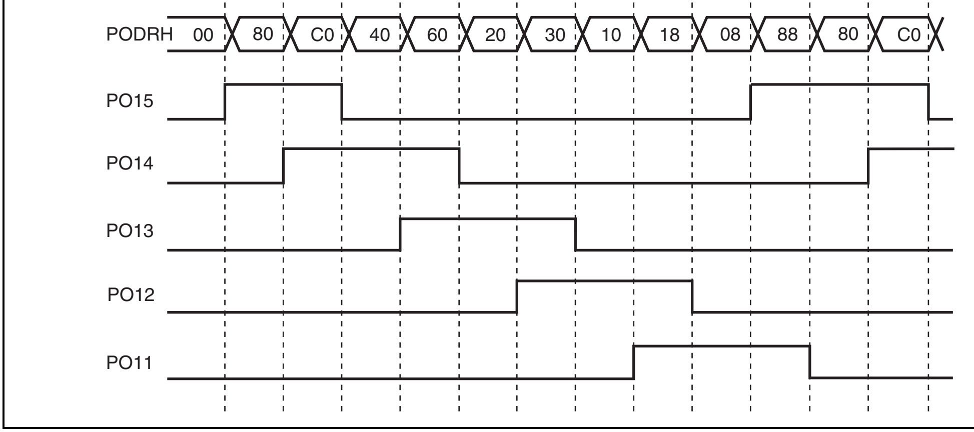


Figure 15.7 Normal Pulse Output Example (5-Phase Pulse Output)

1. Set up TGRA in TPU which is used as the output trigger to be an output compare regi a cycle in TGRA so the counter will be cleared by compare match A. Set the TGIEA TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 to NDERH, and set bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 in select compare match in the TPU channel set up in the previous step to be the output t Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the NDI contents are transferred to PODRH and output. The TGIA interrupt handling routine v next output data (H'C0) in NDRH.
4. 5-phase pulse output (one or two phases active at a time) can be obtained subsequentl writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupt

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

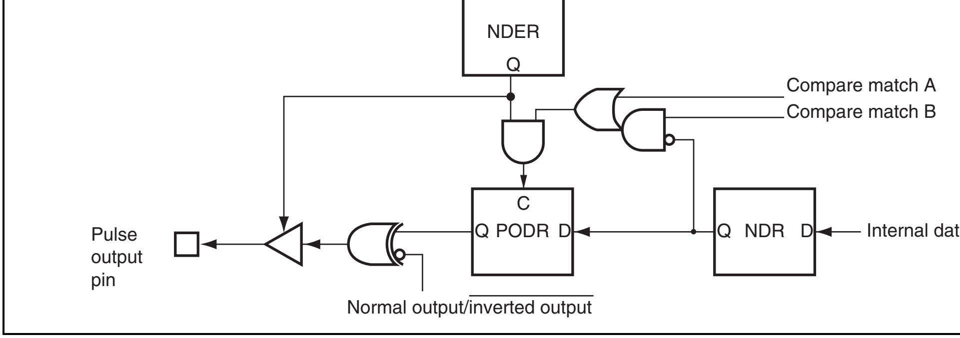


Figure 15.8 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur compare match A.

The NDR contents should not be altered during the interval from compare match B to co match A (the non-overlapping margin).

This can be accomplished by having the TGIA interrupt handling routine write the next NDR, or by having the TGIA interrupt activate the DTC or DMAC. Note, however, that data must be written before the next compare match B occurs.

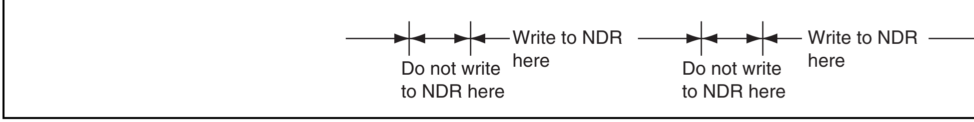
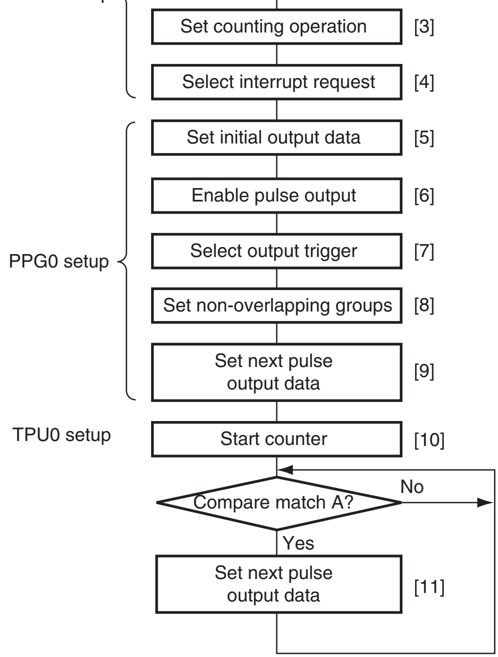


Figure 15.9 Non-Overlapping Operation and NDR Write Timing



[3] Select the counter clock source with b
TPSC2 to TPSC0 in TCR. Select the
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 be
used for pulse output to 1.

[7] Select the TPU compare match event
be used as the pulse output trigger in
PCR.

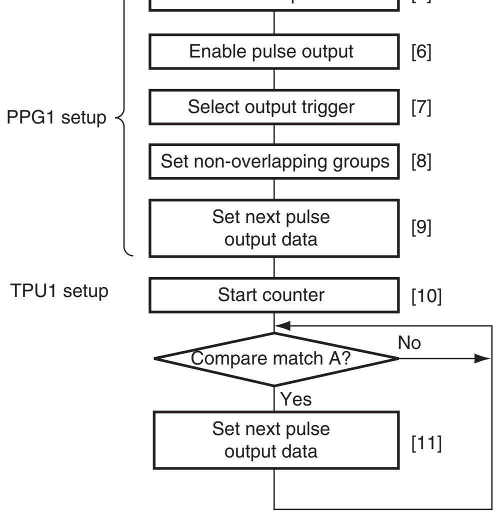
[8] In PMR, select the groups that will
operate in non-overlapping mode.

[9] Set the next pulse output values in NI

[10] Set the CST bit in TSTR to 1 to start t
TCNT counter.

[11] At each TGIA interrupt, set the next
output values in NDR.

Figure 15.10 Setup Procedure for Non-Overlapping Pulse Output (PPG0



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 t 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 ND

[10] Set the CST bit in TSTR to 1 to start th TCNT counter.

[11] At each TGIA interrupt, set the next output values in NDR.

Figure 15.11 Setup Procedure for Non-Overlapping Pulse Output (PPG1)

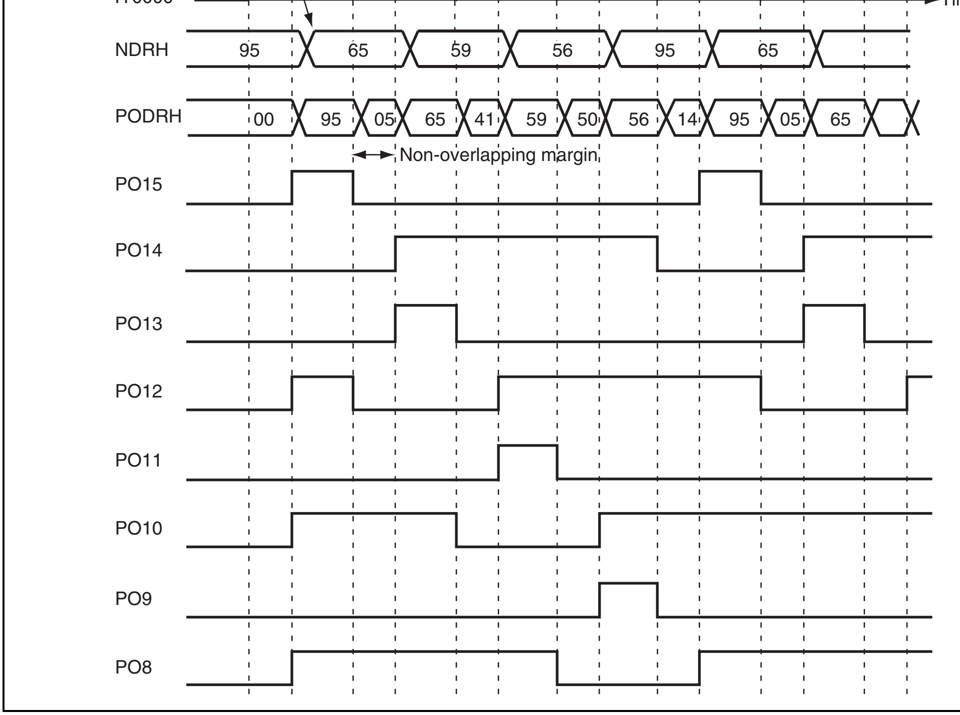


Figure 15.12 Non-Overlapping Pulse Output Example (4-Phase Complemen

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 b
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 wi
imposing a load on the CPU.

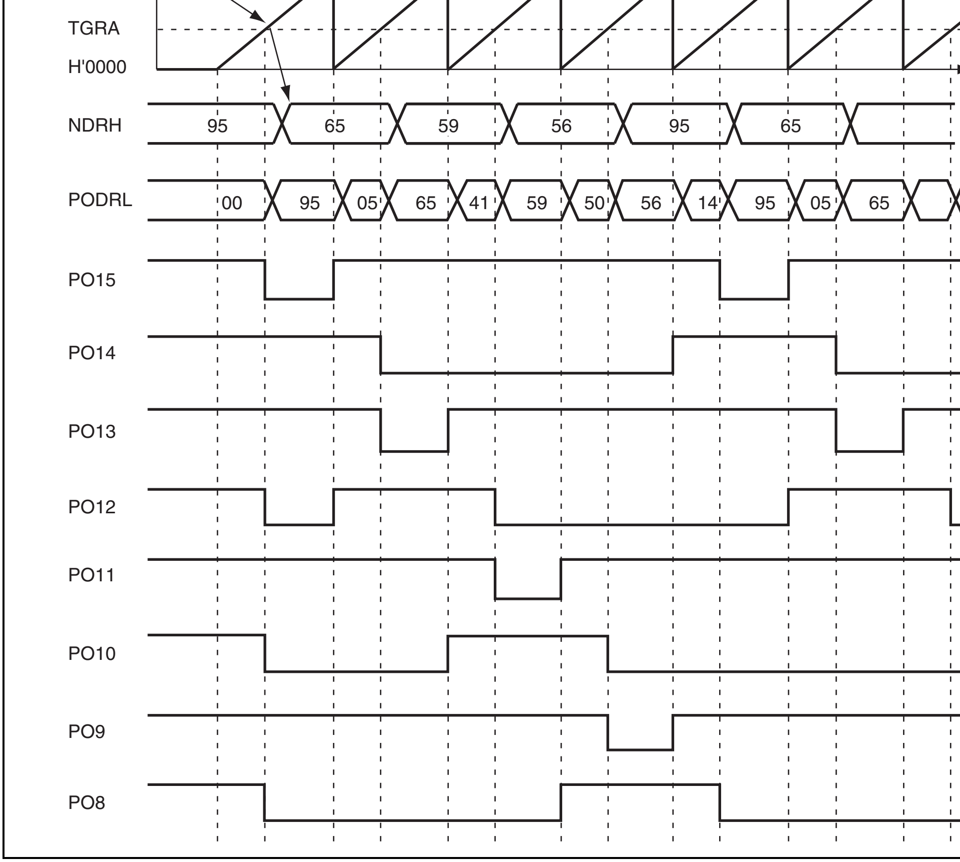


Figure 15.13 Inverted Pulse Output (Example)

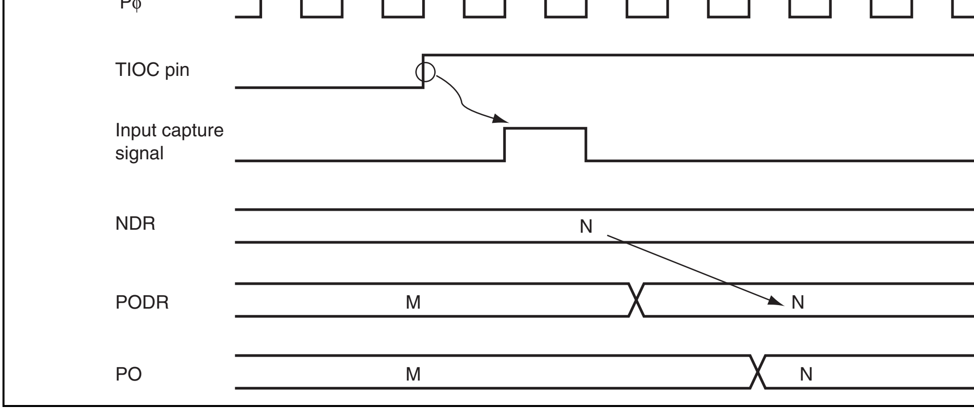


Figure 15.14 Pulse Output Triggered by Input Capture (Example)

Pins PO0 to PO7 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
Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of
of the pins.

Pin functions should be changed only under conditions in which the output trigger event
occur.

15.5.3 TPU Setting when PPG1 is in Use

When using PPG1, output toggling on compare-matches must be specified in the TIOR
the TPU that acts as the activation source and output must be selected as the PPG1 funct

the same functions. Unit 2 and unit 3 can generate baud rate clock for SCI and have the
functions.

16.1 Features

- Selection of seven clock sources

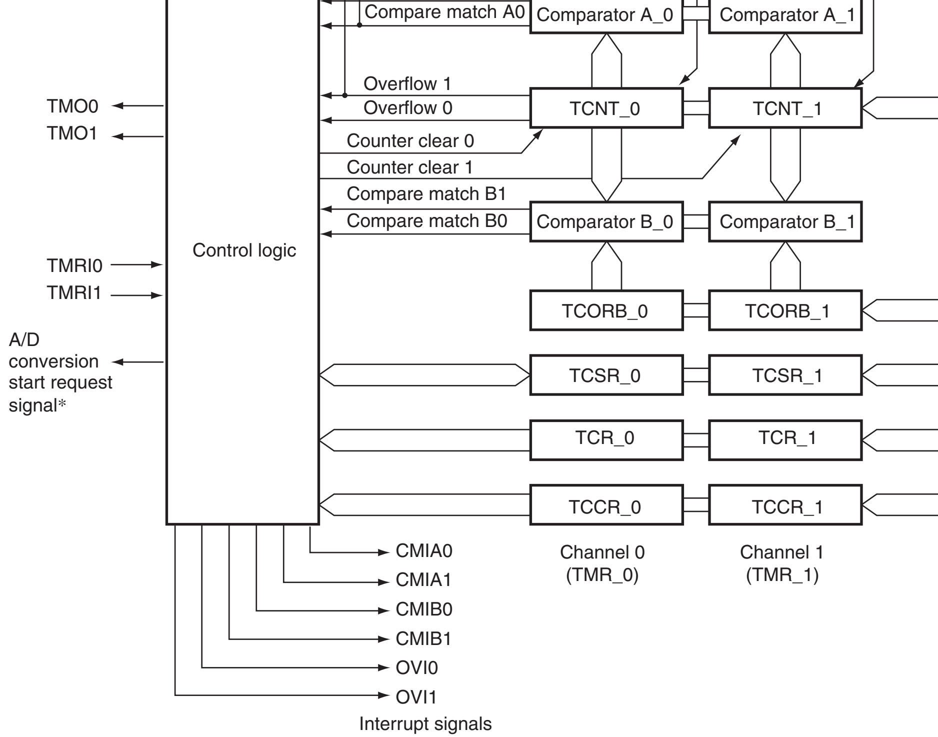
  The counters can be driven by one of six internal clock signals (Pϕ/2, Pϕ/8, Pϕ/32, P
  Pϕ/1024, or Pϕ/8192) or an external clock input (only internal clock available in unit
  Pϕ, Pϕ/2, Pϕ/8, Pϕ/32, Pϕ/64, Pϕ/1024, and Pϕ/8192).
- Selection of three ways to clear the counters

  The counters can be cleared on compare match A or B, or by an external reset signal
  available only in unit 0 and unit 1.)
- 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

  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
  (This is available only in unit 0 and unit 1.)
- Generation of trigger to start A/D converter conversion (available in unit 0 to unit 3)
- Capable of generating baud rate clock for SCI_5 and SCI_6. (This is available only i
  and unit 3). For details, see section 18, Serial Communication Interface (SCI, IrDA,
- Module stop state specifiable

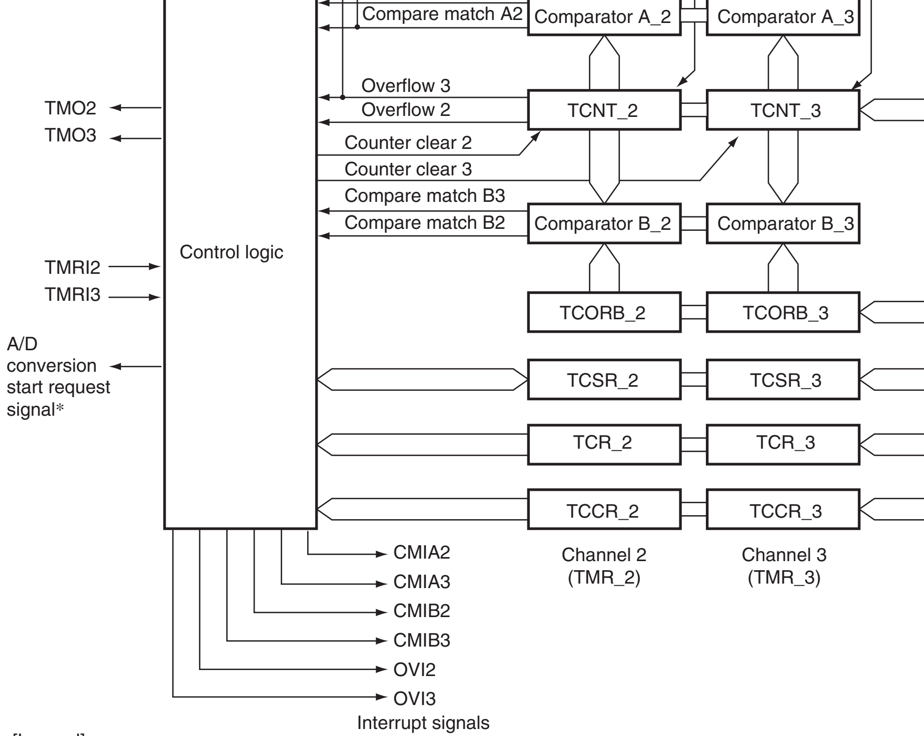


[Legend]

TCORA_0:	Time constant register A_0	TCORA_1:	Time constant register A_1
TCNT_0:	Timer counter_0	TCNT_1:	Timer counter_1
TCORB_0:	Time constant register B_0	TCORB_1:	Time constant register B_1
TCSR_0:	Timer control/status register_0	TCSR_1:	Timer control/status register_1
TCR_0:	Timer control register_0	TCR_1:	Timer control register_1
TCCR_0:	Timer counter control register_0	TCCR_1:	Timer counter control register_1

Note: * For the corresponding A/D converter channels, see section 21, A/D Converter.

Figure 16.1 Block Diagram of 8-Bit Timer Module (Unit 0)

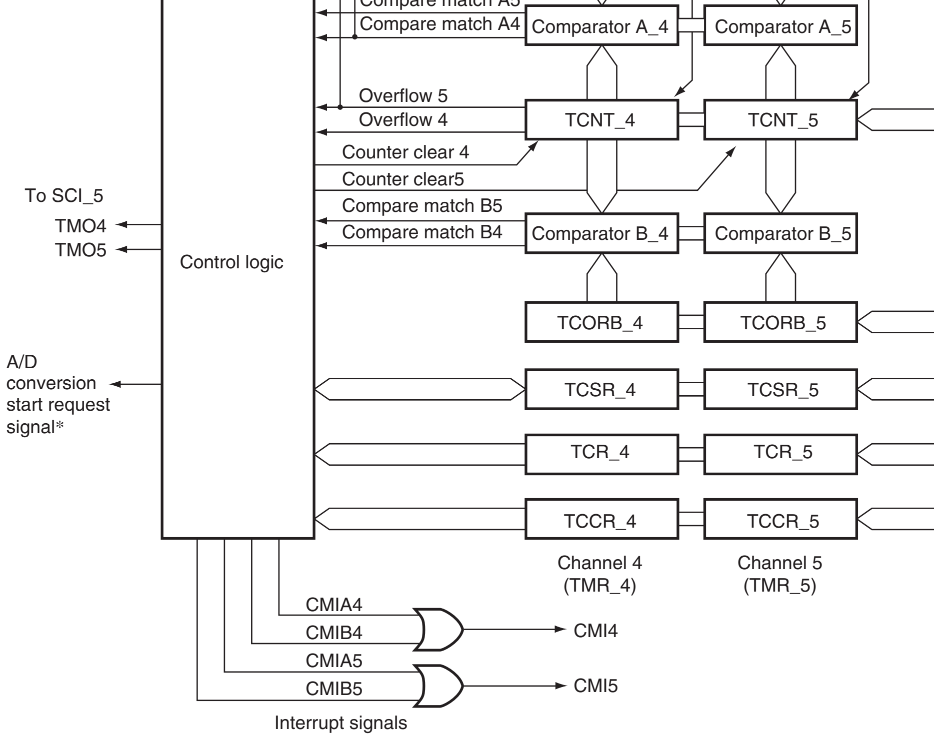


[Legend]

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

Note: * For the corresponding A/D converter channels, see section 21, A/D Converter.

Figure 16.2 Block Diagram of 8-Bit Timer Module (Unit 1)

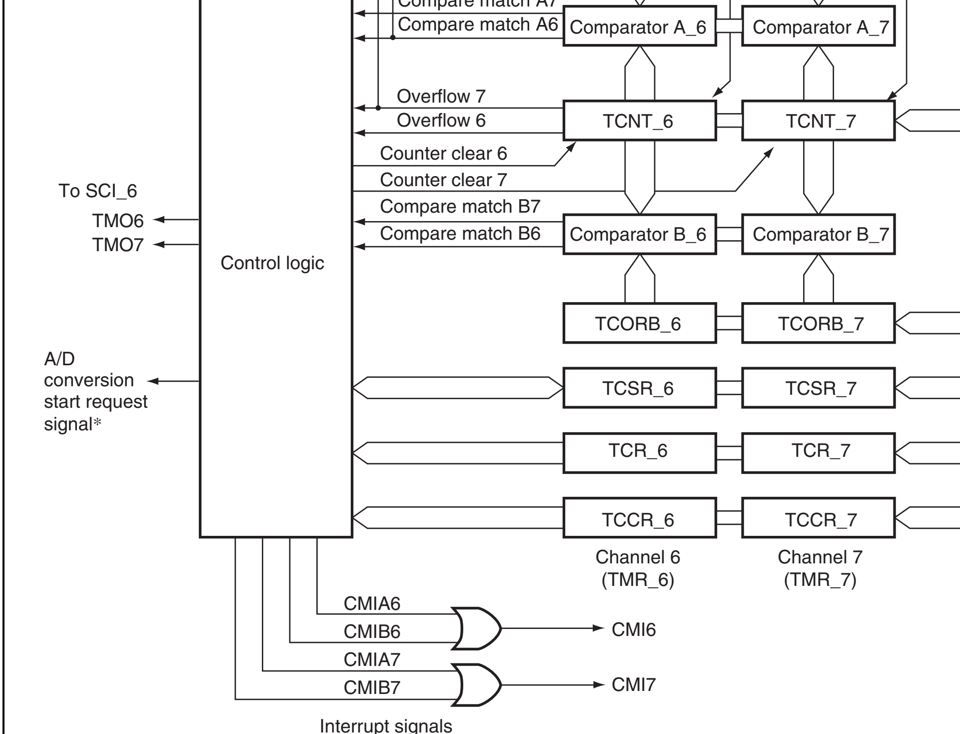


[Legend]

TCORA_4:	Time constant register A_4	TCORA_5:	Time constant register A_5
TCNT_4:	Timer counter_4	TCNT_5:	Timer counter_5
TCORB_4:	Time constant register B_4	TCORB_5:	Time constant register B_5
TCSR_4:	Timer control/status register_4	TCSR_5:	Timer control/status register_5
TCR_4:	Timer control register_4	TCR_5:	Timer control register_5
TCCR_4:	Timer counter control register_4	TCCR_5:	Timer counter control register_5

Note: * For the corresponding A/D converter channels, see section 21, A/D Converter.

Figure 16.3 Block Diagram of 8-Bit Timer Module (Unit 2)



[Legend]

TCORA_6:	Time constant register A_6	TCORA_7:	Time constant register A_7
TCNT_6:	Timer counter_6	TCNT_7:	Timer counter_7
TCORB_6:	Time constant register B_6	TCORB_7:	Time constant register B_7
TCSR_6:	Timer control/status register_6	TCSR_7:	Timer control/status register_7
TCR_6:	Timer control register_6	TCR_7:	Timer control register_7
TCCR_6:	Timer counter control register_6	TCCR_7:	Timer counter control register_7

Note: * For the corresponding A/D converter channels, see section 21, A/D Converter.

Figure 16.4 Block Diagram of 8-Bit Timer Module (Unit 3)

	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
2	4	—	—	—	—
	5				
3	6				
	7				

— Timer counter control register_0 (TCCR_0)
— Timer control/status register_0 (TCSR_0)

- Channel 1 (TMR_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 (TMR_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 (TMR_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)

— Timer counter_5 (TCNT_5)
— Time constant register A_5 (TCORA_5)
— Time constant register B_5 (TCORB_5)
— Timer control register_5 (TCR_5)
— Timer counter control register_5 (TCCR_5)
— Timer control/status register_5 (TCSR_5)

Unit 3:

- Channel 6 (TMR_6):
  - Timer counter_6 (TCNT_6)
  - Time constant register A_6 (TCORA_6)
  - Time constant register B_6 (TCORB_6)
  - Timer control register_6 (TCR_6)
  - Timer counter control register_6 (TCCR_6)
  - Timer control/status register_6 (TCSR_6)
- Channel 7 (TMR_7):
  - Timer counter_7 (TCNT_7)
  - Time constant register A_7 (TCORA_7)
  - Time constant register B_7 (TCORB_7)
  - Timer control register_7 (TCR_7)
  - Timer counter control register_7 (TCCR_7)
  - Timer control/status register_7 (TCSR_7)

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

16.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 comprise a sin
register so they can be accessed together by a word transfer instruction. The value in TC
continually compared with the value in TCNT. When a match is detected, the correspon
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 thi
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

Bit Name															
Initial Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the condition for clearing TCNT, and enables/di
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 (CMIB enabled or disabled when the CMFB flag in TCS to 1. *[2] 0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled

				enabled or disabled when the OVF flag in TCS to 1. 0: OVF interrupt requests (OVI) are disabled 1: OVF interrupt requests (OVI) are enabled
4 3	CCLR1 CCLR0	0 0	R/W R/W	Counter Clear 1 and 0*[1] These bits select the method by which TCNT is 00: Clearing is disabled 01: Cleared by compare match A 10: Cleared by compare match B 11: Cleared at rising edge (TMRIS in TCCR is c 0) of the external reset input or when the ex reset input is high (TMRIS in TCCR is set t
2 1 0	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	Clock Select 2 to 0*[1] These bits select the clock input to TCNT and c condition. See table 16.2.

Notes: 1. To use an external reset or external clock, the DDR and ICR bits in the corres
pin should be set to 0 and 1, respectively. For details, see section 13, I/O Por
2. In unit 2 and unit 3, one interrupt signal is used for CMIEB or CMIEA. For det
section 16.7, Interrupt Sources.
3. Available only in unit 0 and unit 1.

Bit	Bit Name	Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. It should not be
3	TMRIS	0	R/W	Timer Reset Input Select* Selects an external reset input when the CCLR1 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	Reserved This bit is always read as 0. It should not be set
1	ICKS1	0	R/W	Internal Clock Select 1 and 0
0	ICKS0	0	R/W	These bits in combination with bits CKS2 to CKS select the internal clock. See table 16.2.

Note: * Available only in unit 0 and unit 1. The write value should always be 0 in unit 2 3.

				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.
All	1	0	1	—	—	Uses external clock. Counts at rising edge*2.
	1	1	0	—	—	Uses external clock. Counts at falling edge*2.
	1	1	1	—	—	Uses external clock. Counts at both rising an edges*2.

Notes: 1. If the clock input of channel 0 is the TCNT_1 overflow signal and that of chan TCNT_0 compare match signal, no incrementing clock is generated. Do not u setting.

2. To use the external clock, the DDR and ICR bits in the corresponding pin sho to 0 and 1, respectively. For details, see section 13, I/O Ports.

				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	0	1	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	1	0	0	—	—	Counts at TCNT_3 overflow signal*1.
TMR_3	0	0	0	—	—	Clock input prohibited
	0	0	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	0	1	0	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	0	1	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	1	0	0	—	—	Counts at TCNT_2 compare match A*1.
All	1	0	1	—	—	Uses external clock. Counts at rising edge*2.
	1	1	0	—	—	Uses external clock. Counts at falling edge*2.
	1	1	1	—	—	Uses external clock. Counts at both rising and edges*2.

Notes: 1. If the clock input of channel 2 is the TCNT_3 overflow signal and that of chann TCNT_2 compare match signal, no incrementing clock is generated. Do not us setting.

2. To use the external clock, the DDR and ICR bits in the corresponding pin shou to 0 and 1, respectively. For details, see section 13, I/O Ports.

				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 rising edge of
				1	1	Uses internal clock. Counts at falling edge of
	1	0	0	—	—	Counts at TCNT_5 overflow signal*.
TMR_5	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 rising edge of
				1	1	Uses internal clock. Counts at falling edge of
	1	0	0	—	—	Counts at TCNT_4 compare match A*.
All	1	0	1	—	—	Setting prohibited
	1	1	0	—	—	Setting prohibited
	1	1	1	—	—	Setting prohibited

Note: * If the clock input of channel 4 is the TCNT_5 overflow signal and that of chan TCNT_4 compare match signal, no incrementing clock is generated. Do not u setting.

				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	0	1	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at rising edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	1	0	0	—	—	Counts at TCNT_7 overflow signal*.
TMR_7	0	0	0	—	—	Clock input prohibited
	0	0	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	0	1	0	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	0	1	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at rising edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	1	0	0	—	—	Counts at TCNT_6 compare match A*.
All	1	0	1	—	—	Setting prohibited
	1	1	0	—	—	Setting prohibited
	1	1	1	—	—	Setting prohibited

Note: * If the clock input of channel 6 is the TCNT_7 overflow signal and that of chann TCNT_6 compare match signal, no incrementing clock is generated. Do not us setting.

• 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.

- TCSR_0, TCSR_4

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*1	Compare Match Flag B [Setting condition] • When TCNT matches TCORB [Clearing conditions] • When writing 0 after reading CMFB = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enabled to read the flag after writing 0 to it.) • When the DTC is activated by a CMIB interr the DISEL bit in MRB of the DTC is 0

				• When the DTC is activated by a CMIA interru 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 writin while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
4	ADTE	0	R/W	A/D Trigger Enable Selects enabling or disabling of A/D converter st requests by compare match A. 0: A/D converter start requests by compare matc disabled 1: A/D converter start requests by compare matc enabled
3	OS3	0	R/W	Output Select 3 and 2*[2]
2	OS2	0	R/W	These bits select a method of TMO pin output w compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B oc (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 a reset.

- TCSR_1, TCSR_5

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*[1]	Compare Match Flag B [Setting condition] • When TCNT matches TCORB [Clearing conditions] • When writing 0 after reading CMFB = 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 CMIB interr the DISEL bit in MRB of the DTC is 0*[3]

				• When the DTC is activated by a CMIA interru the DISEL bit in MRB of the DTC is 0*[3]
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 wr OVF (When the CPU is used to clear this flag by writi while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
4	—	1	R	Reserved This bit is always read as 1 and cannot be modif
3	OS3	0	R/W	Output Select 3 and 2*[2]
2	OS2	0	R/W	These bits select a method of TMO pin output w compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B o (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 u
compare match occurs after a reset.

3. Available only in unit 0 and unit 1.

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 timer outp
until the first compare match occurs after a reset.

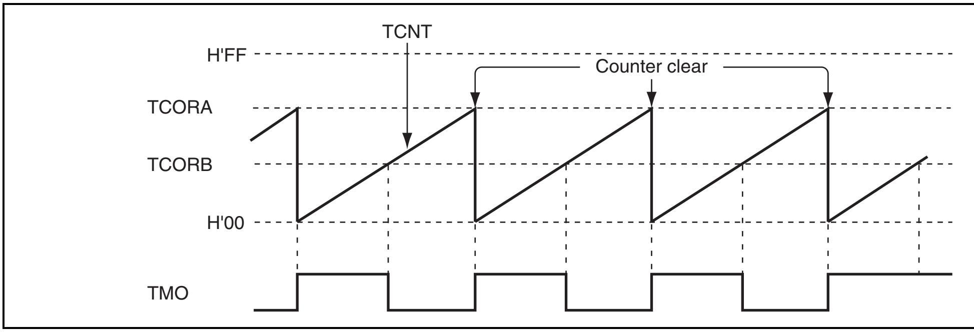


Figure 16.5 Example of Pulse Output

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

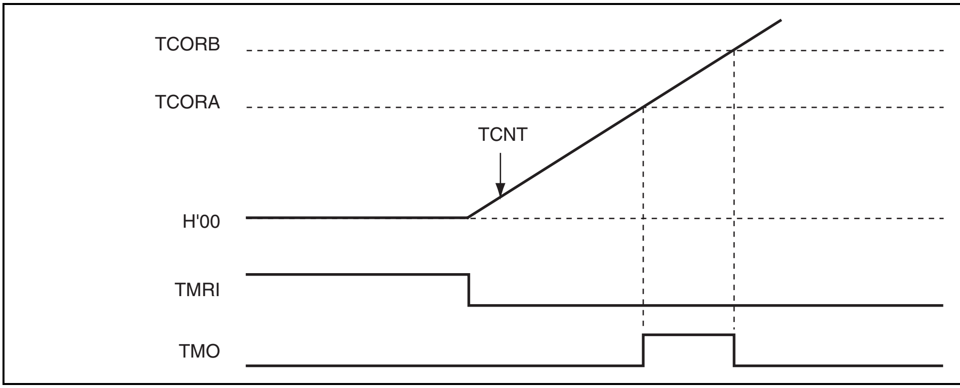


Figure 16.6 Example of Reset Input

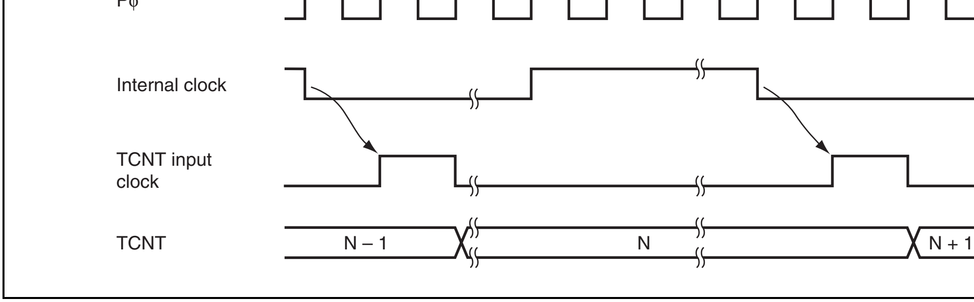


Figure 16.7 Count Timing for Internal Clock Input

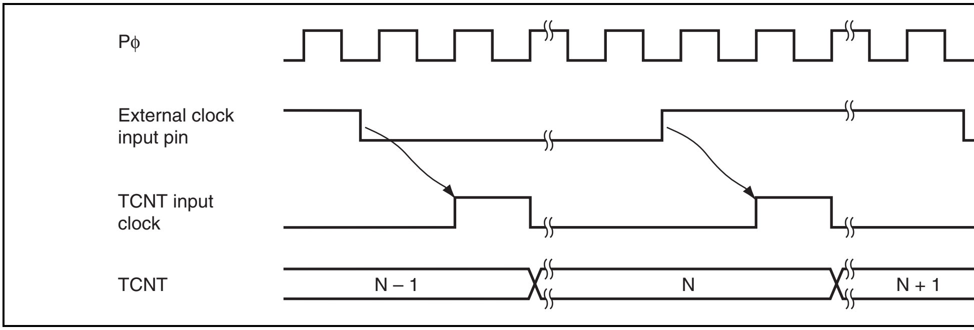


Figure 16.8 Count Timing for External Clock Input

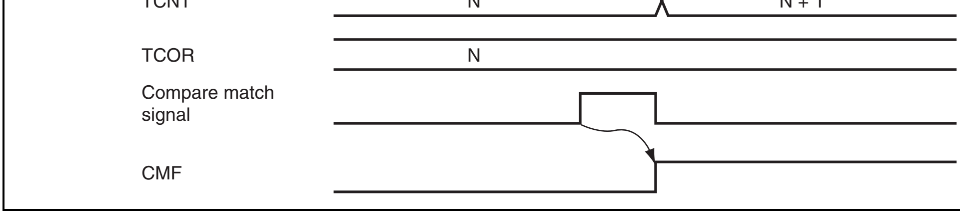


Figure 16.9 Timing of CMF Setting at Compare Match

16.5.3 Timing of Timer Output at Compare Match

When a compare match signal is generated, the timer output changes as specified by the
to OS0 in TCSR. Figure 16.10 shows the timing when the timer output is toggled by the
match A signal.

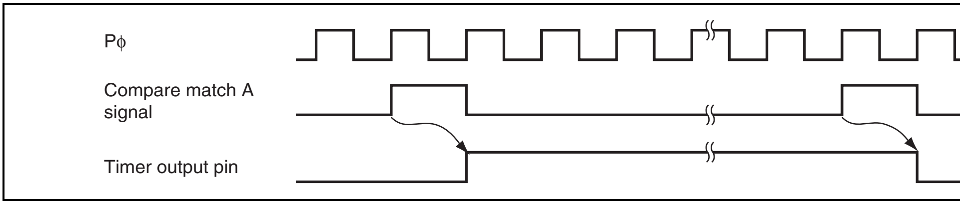


Figure 16.10 Timing of Toggled Timer Output at Compare Match A

Figure 16.11   Timing of Counter Clear by Compare Match

16.5.5 Timing of TCNT External Reset*

TCNT is cleared at the rising edge or high level of an external reset input, depending on t
settings of bits CCLR1 and CCLR0 in TCR. The clear pulse width must be at least 2 state
Figures 16.12 and 16.13 shows the timing of this operation.

Note: * Clearing by an external reset is available only in units 0 and 1.

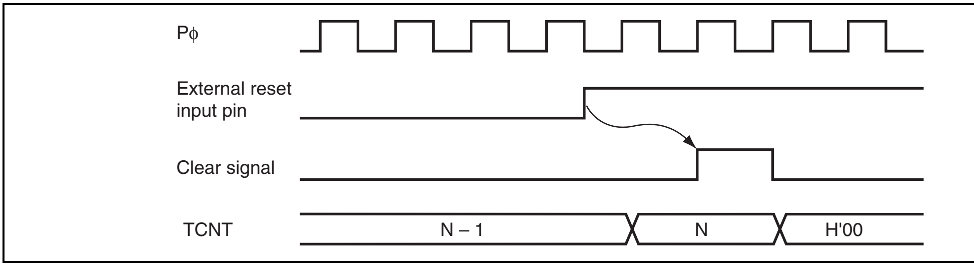


Figure 16.12   Timing of Clearance by External Reset (Rising Edge)

16.5.6 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). F
16.14 shows the timing of this operation.

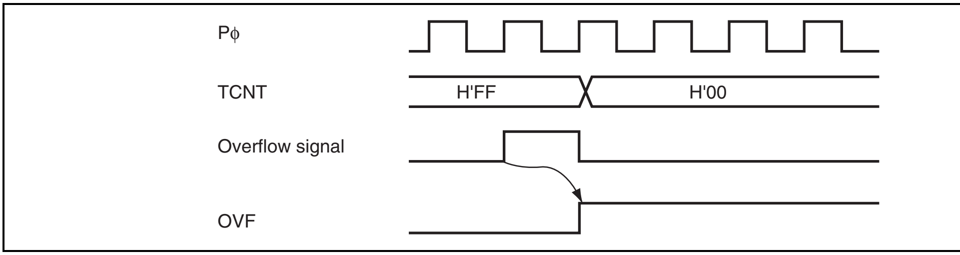


Figure 16.14 Timing of OVF Setting

timer 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 m 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare ma occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter cl the TMRI0 pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits c cleared independently.

(3) Pin Output

- Control of output from the TMO0 pin by the bits OS3 to OS0 in TCSR_0 is in accord the 16-bit compare match conditions.
- Control of output from the TMO1 pin by the bits OS3 to OS0 in TCSR_1 is in accord the lower 8-bit compare match conditions.

16.6.2 Compare Match Count Mode

When the bits CKS2 to CKS0 in TCR_1 are set to B'100, TCNT_1 counts compare matcl channel 0. Channels 0 and 1 are controlled independently. Conditions such as setting of tl flag, generation of interrupts, output from the TMO pin, and counter clear are in accordar the settings for each channel.

Table 16.6 8-Bit Timer (TMR_0 or TMR_1) Interrupt Sources (in Unit 0 and Uni

Signal Name	Name	Interrupt Source	Interrupt Flag	DTC Activation	Pr
CMIA0	CMIA0	TCORA_0 compare match	CMFA	Possible	Hi
CMIB0	CMIB0	TCORB_0 compare match	CMFB	Possible	↑
OVI0	OVI0	TCNT_0 overflow	OVF	Not possible	Lo
CMIA1	CMIA1	TCORA_1 compare match	CMFA	Possible	Hi
CMIB1	CMIB1	TCORB_1 compare match	CMFB	Possible	↑
OVI1	OVI1	TCNT_1 overflow	OVF	Not possible	Lo

- Interrupt in unit 2 and unit 3

There are two interrupt sources for the 8-bit timer (TMR_4 or TMR_5): CMIA, CMIB. interrupt signal is CMI only. The interrupt sources are shown in table 16.7. When enabli disabling is set by the interrupt enable bit in TCR or TCSR, and when either CMIA or C interrupt source is generated, CMI is sent to the interrupt controller. To verify which int source is generated, confirm by checking each flag in TCSR. No overflow-related interr exists. DTC cannot be activated by this interrupt.

The A/D converter can be activated by a compare match A for the even channels of each
unit. *

If the ADTE bit in TCSR is set to 1 when the CMFA flag in TCSR is set to 1 by the occu
a compare match A, a request to start A/D conversion is sent to the A/D converter. If the
timer conversion start trigger has been selected on the A/D converter side at this time, A/
conversion is started.

Note: * For the corresponding A/D converter channels, see section 21, A/D Converter

f: Counter frequency
ϕ: Operating frequency
N: TCOR value

16.8.2 Conflict between TCNT Write and Counter 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 16.15.

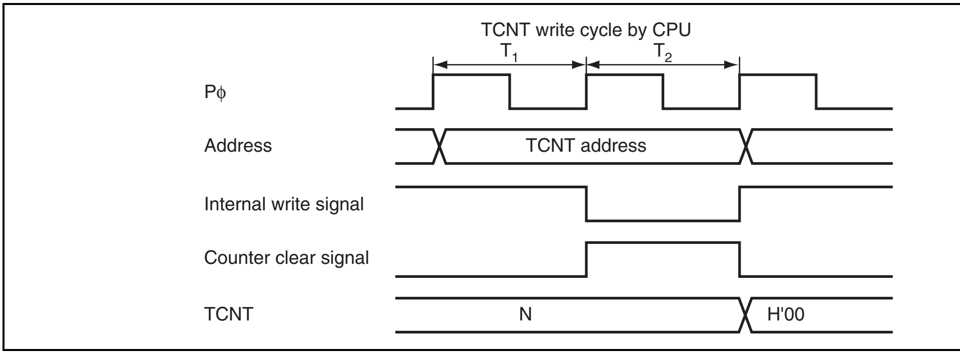


Figure 16.15 Conflict between TCNT Write and Clear

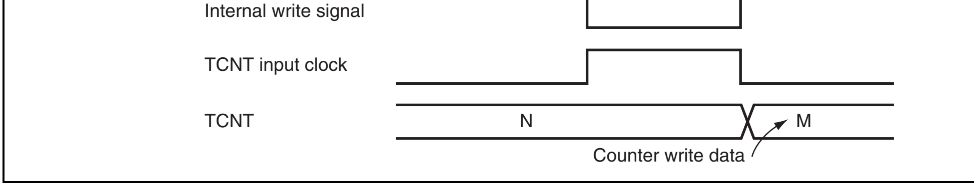


Figure 16.16 Conflict between TCNT Write and Increment

16.8.4 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 16.17.

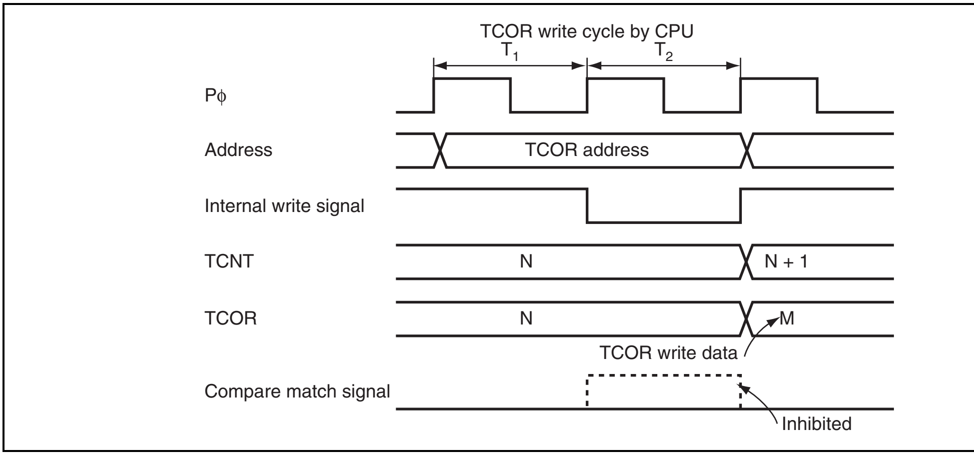


Figure 16.17 Conflict between TCOR Write and Compare Match

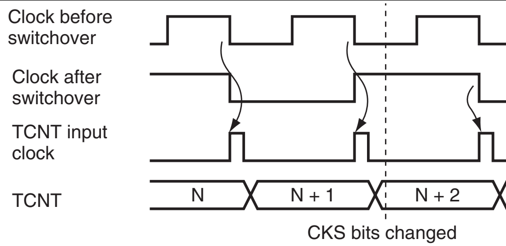
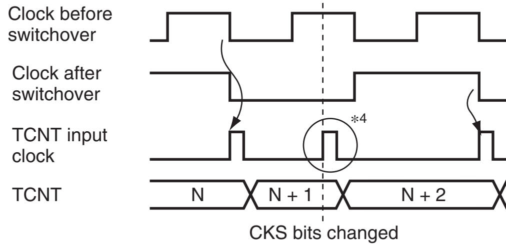
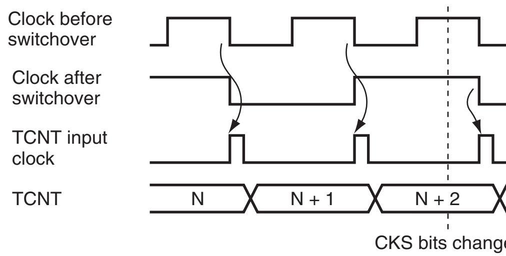
0-output	
No change	Lo

16.8.6 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switche
16.9 shows the relationship between the timing at which the internal clock is switched (b
to the 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 16.9 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 increment of TCNT can also happen when switching between rising and
edges of the internal clock, and when switching between internal and external clocks.

2	Switching from low to high*[2]	
3	Switching from high to low*[3]	
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated because the change of the signal levels is considered as a falling e
TCNT is incremented.

module stop state. For details, see section 27, Power-Down Modes.

16.8.9 Interrupts in Module Stop State

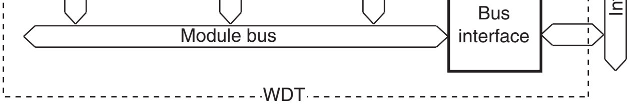
If the module stop state is entered when an interrupt has been requested, it will not be po
clear the CPU interrupt source or the DTC activation source. Interrupts should therefore
disabled before entering the module stop state.

17.1 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 17.1 Block Diagram of WDT

17.2 Input/Output Pin

Table 17.1 shows the WDT pin configuration.

Table 17.1 Pin Configuration

Name	Symbol	I/O	Function
Watchdog timer overflow*	$\overline{\text{WDTOVF}}$	Output	Outputs a counter overflow signal in watchdog timer mode

Note: * In boundary scan valid mode, counter overflow signal output cannot be used.

17.3.1 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

17.3.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

Bit	7	6	5	4	3	2	1
Bit Name	OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1
Initial Value	0	0	0	1	1	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, to clear the flag.

				Cleared by reading TCSR when OVF = 1, then w to OVF. (When the CPU is used to clear this flag while th corresponding interrupt is enabled, be sure to re 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 watchdog interval timer. 0: Interval timer mode When TCNT overflows, an interval timer interr (WOVI) is requested. 1: Watchdog timer mode When TCNT overflows, the $\overline{\text{WDTOVF}}$ signal is
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. W bit is cleared, TCNT stops counting and is initiali H'00.
4, 3	—	All 1	R	Reserved These are read-only bits and cannot be modified
2 1 0	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	Clock Select 2 to 0 Select the clock source to be input to TCNT. The cycle for Pϕ = 20 MHz is indicated in parenthese 000: Clock Pϕ/2 (cycle: 25.6 μs) 001: Clock Pϕ/64 (cycle: 819.2 μs) 010: Clock Pϕ/128 (cycle: 1.6 ms) 011: Clock Pϕ/512 (cycle: 6.6 ms) 100: Clock Pϕ/2048 (cycle: 26.2 ms) 101: Clock Pϕ/8192 (cycle: 104.9 ms) 110: Clock Pϕ/32768 (cycle: 419.4 ms) 111: Clock Pϕ/131072 (cycle: 1.68 s)

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Timer Overflow Flag This bit is set when TCNT overflows in watchdo mode. This bit cannot be set in interval timer m only 0 can be written. [Setting condition] When TCNT overflows (changed from H'FF to H watchdog timer mode [Clearing condition] Reading RSTCSR when WOVF = 1, and then w WOVF
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.

If TCNT overflows when the RSTE bit in RSTCSR is set to 1, a signal that resets this LS internally is generated at the same time as the $\overline{\text{WDTOVF}}$ signal. If a reset caused by a sig to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{\text{RES}}$ pi has priority and the WOVF bit in RSTCSR is cleared to 0.

The $\overline{\text{WDTOVF}}$ signal is output for 133 cycles of Pϕ when RSTE = 1 in RSTCSR, and fo cycles of Pϕ when RSTE = 0 in RSTCSR. The internal reset signal is output for 519 cycl

When RSTE = 1, an internal reset signal is generated. Since the system clock control regi (SCKCR) is initialized, the multiplication ratio of Pϕ becomes the initial value.

When RSTE = 0, an internal reset signal is not generated. Neither SCKCR nor the multip ratio of Pϕ is changed.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. I overflows when the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated entire LSI.

Notes: 1. If TCNT overflows when the RSTE bit is set to 1, an internal reset signal is generated.
2. 130 states when the RSTE bit is cleared to 0.

Figure 17.2 Operation in Watchdog Timer Mode

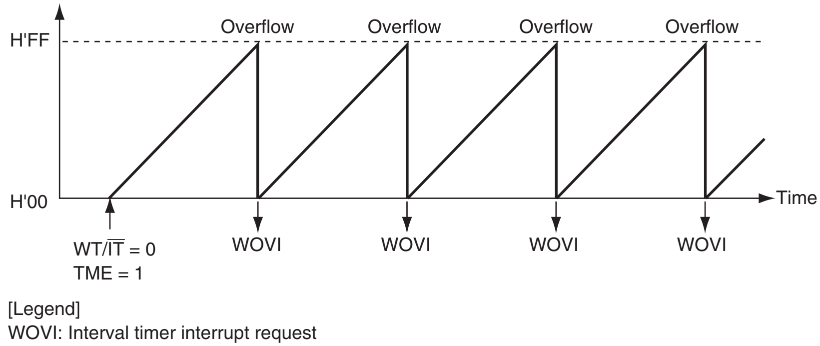


Figure 17.3 Operation in Interval Timer Mode

17.5 Interrupt Source

During interval timer mode operation, an overflow generates an interval timer interrupt (V
The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. The
must be cleared to 0 in the interrupt handling routine.

Table 17.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activatio
WOVI	TCNT overflow	OVF	Impossible

TCNT and TCSR must be written to by a word transfer instruction. They cannot be writ
byte transfer instruction.

For writing, TCNT and TCSR are assigned to the same address. Accordingly, perform d
transfer as shown in figure 17.4. The transfer instruction writes the lower byte data to T
TCSR.

To write to RSTCSR, execute a word transfer instruction for address H'FFA6. A byte tr
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 17.4.

At data transfer, the transfer instruction clears the WOVF bit to 0, but has no effect on t
bit. To write to the RSTE bit, perform data transfer as shown in figure 17.4. In this case,
transfer instruction writes the value in bit 6 of the lower byte to the RSTE bit, but has n
the WOVF bit.

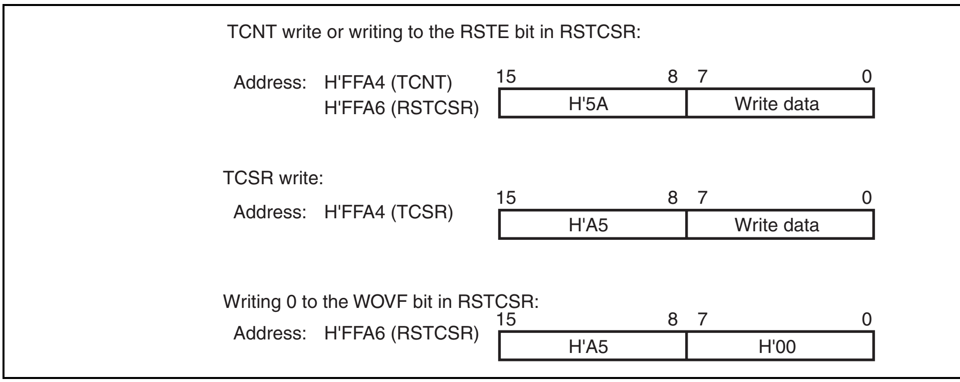


Figure 17.4 Writing to TCNT, TCSR, and RSTCSR

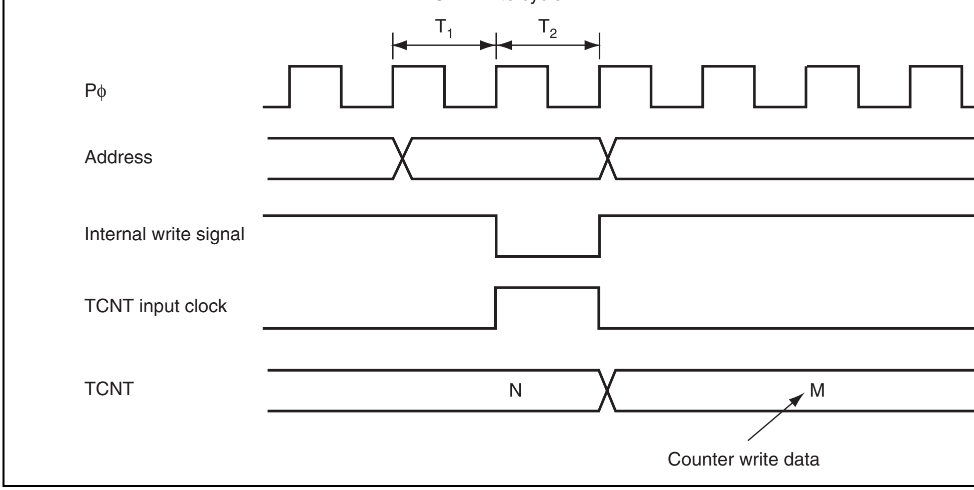


Figure 17.5 Conflict between TCNT Write and Increment

17.6.3 Changing Values of Bits CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could oc the incrementation. The watchdog timer must be stopped (by clearing the TME bit to 0) b values of bits CKS2 to CKS0 are changed.

17.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the timer mode is switched from watchdog timer mode to interval timer mode while the operating, errors could occur in the incrementation. The watchdog timer must be stopped clearing the TME bit to 0) before switching the timer mode.

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin, this LSI will not be initialized correctly. sure that the $\overline{\text{WDTOVF}}$ signal is not input logically to the $\overline{\text{RES}}$ pin. To reset the entire sy means of the $\overline{\text{WDTOVF}}$ signal, use a circuit like that shown in figure 17.6.

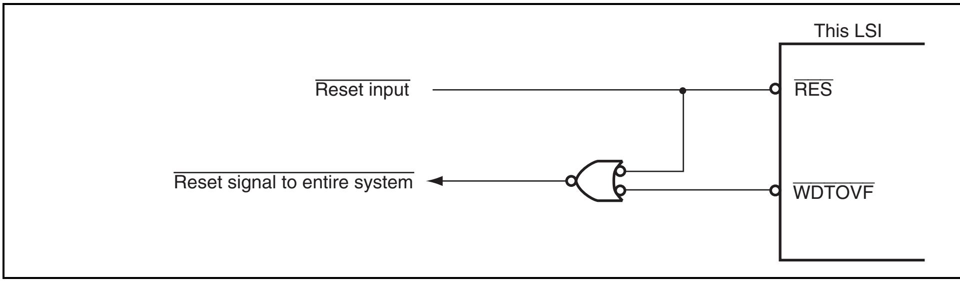


Figure 17.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

17.6.7 Transition to Watchdog Timer Mode or Software Standby Mode

When the WDT operates in watchdog timer mode, a transition to software standby mode made even when the SLEEP instruction is executed when the SSBY bit in SBYCR is se Instead, a transition to sleep mode is made.

To transit to software standby mode, the SLEEP instruction must be executed after halti WDT (clearing the TME bit to 0).

When the WDT operates in interval timer mode, a transition to software standby mode i through execution of the SLEEP instruction when the SSBY bit in SBYCR is set to 1.

communication mode. SCI_5 enables transmitting and receiving IrDA communication w
based on the IrDA Specifications version 1.0. This LSI incorporates the on-chip CRC (C
Redundancy Check) computing unit that realizes high reliability of high-speed data tran
the CRC computing unit is not connected to SCI, operation is executed by writing data t
registers.

Figure 18.1 shows a block diagram of the SCI_0 to SCI_4. Figure 18.2 shows a block di
the SCI_5 and SCI_6.

18.1 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 state specifiable

16-MHz operation: 115.192 kbps, 460.784 kbps, or 720 kbps can be selected
32-MHz operation: 720 kbps

- Average transfer rate generator (SCI_5, SCI_6)
  8-MHz operation: 460.784 kbps can be selected
  10.667-MHz operation: 115.152 kbps or 460.606 kbps can be selected
  12-MHz operation: 230.263 kbps or 460.526 kbps can be selected
  16-MHz operation: 115.196 kbps, 460.784 kbps, 720 kbps, or 921.569 kbps can be se
  24-MHz operation: 115.132 kbps, 460.526 kbps, 720 kbps, or 921.053 kbps can be se
  32-MHz operation: 720 kbps can be selected

Clocked Synchronous Mode (SCI_0, 1, 2, and 4):

- Data length: 8 bits
- Receive error detection: Overrun errors

Smart Card Interface:

- 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

			115.192 kbps	115.1
	Pφ = 12 MHz	—	—	460.5 230.2
	Pφ = 16 MHz	—	720 kbps 460 784kbps 115.192 kbps	921.5 720 k 460.7 115.1
	Pφ = 24 MHz	—	—	921.0 720 k 460.5 115.1
	Pφ = 32 MHz	—	720 kbps	720 k

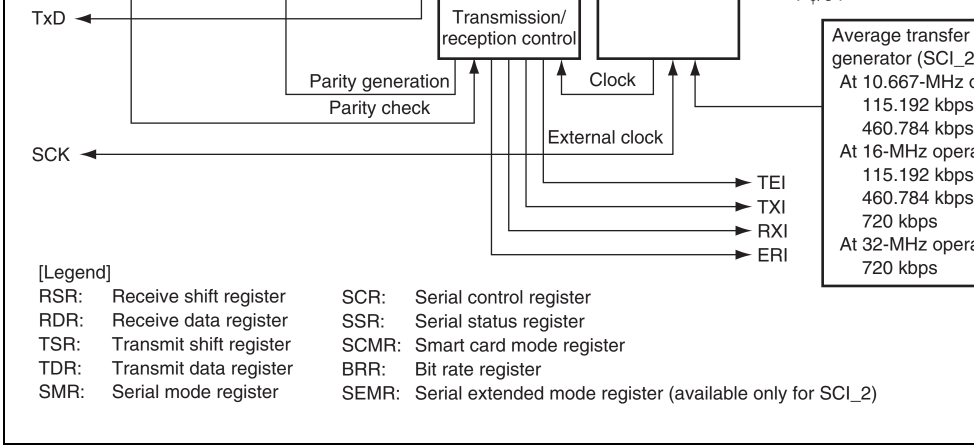


Figure 18.1 Block Diagram of SCI_0, 1, 2, and 4

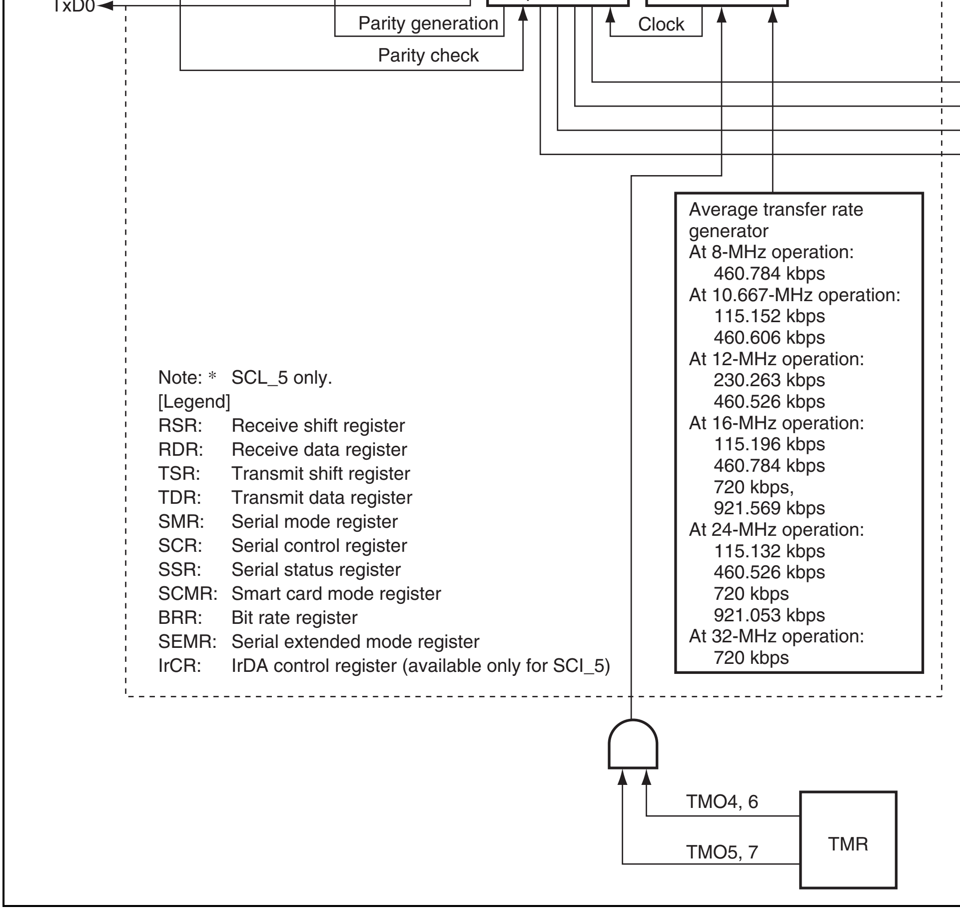


Figure 18.2 Block Diagram of SCI_5 and SCI_6

1	SCK1	I/O	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	I/O	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output
4	SCK4	I/O	Channel 4 clock input/output
	RxD4	Input	Channel 4 receive data input
	TxD4	Output	Channel 4 transmit data output
5	RxD5/IrRxD	Input	Channel 5 receive data input
	TxD5/IrTxD	Output	Channel 5 transmit data output
6	RxD6	Input	Channel 6 receive data input
	TxD6	Output	Channel 6 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting th 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)

- Serial extended mode register_2 (SEMR_2)

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)

Channel 5:

- Receive shift register_5 (RSR_5)
- Transmit shift register_5 (TSR_5)
- Receive data register_5 (RDR_5)
- Transmit data register_5 (TDR_5)
- Serial mode register_5 (SMR_5)
- Serial control register_5 (SCR_5)
- Serial status register_5 (SSR_5)
- Smart card mode register_5 (SCMR_5)
- Bit rate register_5 (BRR_5)
- Serial extended mode register_5 (SEMR_5)
- IrDA control register_5 (IrCR)

- Bit rate register_6 (BRR_6)
- Serial extended mode register_6 (SEMR_6)

18.3.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RxD pin and co
into parallel data. When one frame of data has been received, it is transferred to RDR
automatically. RSR cannot be directly accessed by the CPU.

18.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one frame o
data, it transfers the received serial data from RSR to RDR where it is stored. This allow
receive the next data. Since RSR and RDR function as a double buffer in this way, conti
receive operations can be performed. After confirming that the RDRF bit in SSR is set t
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

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

18.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the S
automatically transfers transmit data from TDR to TSR, and then sends the data to the Tx
TSR cannot be directly accessed by the CPU.

18.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator cloc
Some bits in SMR have different functions in normal mode and smart card interface mod

- 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

				the MSB (bit 7) in TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data leng bits is used.
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode When this bit is set to 1, the parity bit is added t data before transmission, and the parity bit is cl reception. For a multiprocessor format, parity bi and checking are not performed regardless of t setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
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. If second stop bit is 0, it is treated as the start bit transmit frame.
2	MP	0	R/W	Multiprocessor Mode (valid only in asynchronou When this bit is set to 1, the multiprocessor fun enabled. The PE bit and O/$\overline{E}$ bit settings are inv multiprocessor mode.

				baud rate, see section 18.3.9, Bit Rate Register is the decimal display of the value of n in BRR (s section 18.3.9, Bit Rate Register (BRR)).

Note: * Available in SCI_0, 1, 2, and 4 only. Setting is prohibited in SCI_5 and SCI_6.

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

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode Setting this bit to 1 allows GSM mode operation. mode, the TEND set timing is put forward to 11.0 the start and the clock output control function is appended. For details, see sections 18.7.6, Data Transmission (Except in Block Transfer Mode) a 18.7.8, Clock Output Control (Only SCI_0, 1, 2, a
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode op For details, see section 18.7.3, Block Transfer M
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode) When this bit is set to 1, the parity bit is added to data before transmission, and the parity bit is che reception. Set this bit to 1 in smart card interface
4	O/$\overline{E}$	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 card mode, see section 18.7.2, Data Format (Except i Transfer Mode).

				18.3.9, Bit Rate Register (BRR).
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0	R/W	These bits select the clock source for the baud 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 baud rate, see section 18.3.9, Bit Rate Register is the decimal display of the value of n in BRR (section 18.3.9, Bit Rate Register (BRR)).

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

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

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1
Bit Name	TIE	RIE	TE	RE	MPIE	TEIE	CKE1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit Functions in Normal Serial Communication Interface Mode (When SMIF in SCM

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request i enabled. A TXI interrupt request can be cancelled by read from the TDRE flag and then clearing the flag to clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt re are enabled. RXI and ERI interrupt requests can be cancelled reading 1 from the RDRF, FER, PER, or ORER then clearing the flag to 0, or by clearing the RIE

				When this bit is set to 1, reception is enabled. U condition, serial reception is started by detectin bit in asynchronous mode or the synchronous c in clocked synchronous mode. Note that SMR s set prior to setting the RE bit to 1 in order to de the reception format. Even if reception is halted by clearing this bit to RDRF, FER, PER, and ORER flags are not affe the previous value is retained.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (valid only whe bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which t multiprocessor bit is 0 is skipped, and setting o RDRF, FER, and ORER status flags in SSR is On receiving data in which the multiprocessor b bit is automatically cleared and normal receptio resumed. For details, see section 18.5, Multipro Communication Function. When receive data including MPB = 0 in SSR is received, transfer of the received data from RS detection of reception errors, and the settings o FER, and ORER flags in SSR are not performe receive data including MPB = 1 is received, the in SSR is set to 1, the MPIE bit is automatically 0, and RXI and ERI interrupt requests (in the ca the TIE and RIE bits in SCR are set to 1) and s the FER and ORER flags are enabled.

00: On-chip baud rate generator

The SCK pin functions as I/O port.

01: On-chip baud rate generator

The clock with the same frequency as the bi
output from the SCK pin.

1X: External clock

The clock with a frequency 16 times the bit r
should be input from the SCK pin.

- Clocked synchronous mode

0X: Internal clock

The SCK pin functions as the clock output p

1X: External clock

The SCK pin functions as the clock input pin

				1X: External clock or average transfer rate gen When an external clock is used, the clock w frequency 16 times the bit rate should be in the SCK pin. When an average transfer rate generator is • Clocked synchronous mode 0X: Internal clock The SCK pin functions as the clock output p 1X: External clock The SCK pin functions as the clock input pi
1	CKE1	0	R/W	Clock Enable 1, 0 (for SCI_5 and SCI_6)
0	CKE0	0	R/W	These bits select the clock source. • Asynchronous mode 00: On-chip baud rate generator 1X: TMR clock input or average transfer rate ge When an average transfer rate generator is When TMR clock input is used. • Clocked synchronous mode Not available

[Legend]

X: Don't care

				When this bit is set to 1, RXI and ERI interrupt re are enabled. RXI and ERI interrupt requests can be cancelled reading 1 from the RDRF, FER, PER, or ORER f then clearing the flag to 0, or by clearing the RIE
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled this condition, serial transmission is started by w transmit data to TDR, and clearing the TDRE flag to 0. Note that SMR should be set prior to setting bit to 1 in order to designate the transmission for 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. Un condition, serial reception is started by detecting bit in asynchronous mode or the synchronous cl in clocked synchronous mode. Note that SMR sh set prior to setting the RE bit to 1 in order to desi the reception format. Even if reception is halted by clearing this bit to 0 RDRF, FER, PER, and ORER flags are not affec the previous value is retained.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (valid only when 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.

1X: Reserved

- When GM in SMR = 1

00: Output fixed low

01: Clock output

10: Output fixed high

11: Clock output

Note: * No SCK pins exist in SCI_5 and SCI_6.

18.3.7 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 func
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	FER	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.

Bit	Bit Name	Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR [Clearing conditions] • When 0 is written to TDRE after reading TDF (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 a TXI interrupt request is issued allowi DMAC or DTC to write data to TDR

				while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When an RXI interrupt request is issued allo DMAC or DTC to read data from RDR The RDRF flag is not affected and retains its pr value when the RE bit in SCR is cleared to 0. Note that when the next serial reception is com while the RDRF flag is being set to 1, an overru occurs and the received data is lost.
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error has occurred du reception and the reception ends abnormally. [Setting condition] • When the next serial reception is completec RDRF = 1 In RDR, receive data prior to an overrun err occurrence is retained, but data received af overrun error occurrence is lost. When the (is set to 1, subsequent serial reception canr performed. Note that, in clocked synchronou serial transmission also cannot continue. [Clearing condition] • When 0 is written to ORER after reading OF (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) Even when the RE bit in SCR is cleared, the flag is not affected and retains its previous v

is transferred to RDR, however, the RDRF fla
set. In addition, when the FER flag is being s
the subsequent serial reception cannot be pe
In clocked synchronous mode, serial transmi
also cannot continue.

[Clearing condition]

- When 0 is written to FER after reading FER =

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

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

				subsequent serial reception cannot be perfo clocked synchronous mode, serial transmis cannot continue. [Clearing condition] • When 0 is written to PER after reading PER (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) Even when the RE bit in SCR is cleared, the is not affected and retains its previous value
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 receiv When the RE bit in SCR is cleared to 0 its prev is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Sets the multiprocessor bit value to be added to transmit frame.

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

				• When 0 is written to TDRE after reading TDF (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 a TXI interrupt request is issued allowi DMAC or DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RDR. [Setting condition] • When serial reception ends normally and rec is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDF (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 an RXI interrupt request is issued allov DMAC or DTC to read data from RDR The RDRF flag is not affected and retains its pre value even when the RE bit in SCR is cleared to Note that when the next reception is completed v RDRF flag is being set to 1, an overrun error occ the received data is lost.

				is set to 1, subsequent serial reception cann performed. Note that, in clocked synchronou serial transmission also cannot continue. [Clearing condition] • When 0 is written to ORER after reading OR (When the CPU is used to clear this flag by while the corresponding interrupt is enabled to read the flag after writing 0 to it.) Even when the RE bit in SCR is cleared, the flag is not affected and retains its previous v
4	ERS	0	R/(W)*	Error Signal Status [Setting condition] • When a low error signal is sampled [Clearing condition] • When 0 is written to ERS after reading ERS

subsequent serial reception cannot be perfor
clocked synchronous mode, serial transmissi
cannot continue.

[Clearing condition]

- When 0 is written to PER after reading PER =

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

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

				set timing depends on the register setting a When GM = 0 and BLK = 0, 2.5 etu after tra start When GM = 0 and BLK = 1, 1.5 etu after tra start When GM = 1 and BLK = 0, 1.0 etu after tra start When GM = 1 and BLK = 1, 1.0 etu after tra start [Clearing conditions] • When 0 is written to TEND after reading TE • When a TXI interrupt request is issued allov DMAC or DTC to write the next data to TDF
1	MPB	0	R	Multiprocessor Bit Not used in smart card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in smart card interface mode.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: Transfer with LSB-first 1: Transfer with MSB-first This bit is valid only when the 8-bit data format is used for transmission/reception; when the 7-bit data format is used, data is always transmitted/received with LSB-first.
2	SINV	0	R/W	Smart Card Data Invert Inverts the transmit/receive data logic level. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.
1	—	1	—	Reserved This bit is always read as 1.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface mode is selected. 0: Normal asynchronous or clocked synchronous mode 1: Smart card interface mode

Asynchronous mode	0	$N = \frac{P\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{P\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)}$
	1	$N = \frac{P\phi \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{P\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)}$
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 ($0 \le N \le 255$)

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	3
0	1	6
1	0	3
1	1	2

Table 18.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

	Operating Frequency Pϕ (MHz)										
	8			9.8304			10			12	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155
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

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

Note: In SCI_2, 5, and 6, this is an example when the ABCS bit in SEMR_2, 5, and 6 is
When the ABCS bit is set to 1, the bit rate is two times.

Table 18.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode

	Operating Frequency Pϕ (MHz)										
	17.2032			18			19.6608			2	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	3	75	0.48	3	79	–0.12	3	86	0.31	3	88
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64
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

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	27

Note: In SCI_2, 5, and 6, this is an example when the ABCS bit in SEMR_2, 5, and 6 is 0. When the ABCS bit is set to 1, the bit rate is two times.

Table 18.5 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mod

Pφ (MHz)	Maximum Bit Rate (bit/s)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0

Pφ (MHz)	Maximum Bit Rate (bit/s)	n
17.2032	537600	0
18	562500	0
19.6608	614400	0
20	625000	0
25	781250	0
30	937500	0
33	1031250	0
35	1093750	0

14.7456	3.6864	230400	33	8.2500	5156
16	4.0000	250000	35	8.7500	5468

Note: In SCI_2, this is an example when the ABCS bit in SEMR_2 is 0.
When the ABCS bit is set to 1, the bit rate is two times.

5k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2
10k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1
25k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1
50k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	0
100k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0
250k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0
500k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	—
1M	0	1			0	3	0	4	—	—	—	—	—	—	—
2.5M			0	0*[1]			0	1	—	—	0	2	—	—	—
5M							0	0*[1]	—	—	—	—	—	—	—

[Legend]

Space: Setting prohibited.

—: Can be set, but there will be error.

Notes: 1. Continuous transmission or reception is not possible.

2. No clocked synchronous mode exists in SCI_5 and SCI_6.

Table 18.8 Maximum Bit Rate with External Clock Input (Clocked Synchronous M

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	Pϕ (MHz)	External Input Clock (MHz)	Maximur Rate (bit
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			

Note * No clocked synchronous mode exists in SCI_5 and SCI_6.

(bit/sec)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2

Bit Rate	Operating Frequency Pϕ (MHz)										
	25.00			30.00			33.00			35	
(bit/sec)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
9600	0	3	12.49	0	3	5.01	0	4	7.59	0	4

Table 18.10 Maximum Bit Rate for Each Operating Frequency (Smart Card Inter Mode, S = 372)

Pϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	Pϕ (MHz)	Maximum Bit Rate (bit/s)	n
7.1424	9600	0	0	18.00	24194	0
10.00	13441	0	0	20.00	26882	0
10.7136	14400	0	0	25.00	33602	0
13.00	17473	0	0	30.00	40323	0
14.2848	19200	0	0	33.00	44355	0
16.00	21505	0	0	35.00	47043	0

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	R	Reserved These bits are always read as undefined and c modified.
3	ABCS	0	R/W	Asynchronous Mode Base clock Select (valid c asynchronous mode) Selects the base clock for a 1-bit period. 0: The base clock has a frequency 16 times the rate 1: The base clock has a frequency 8 times the rate

base clock with a frequency 16 times the
rate)

010: 460.784 kbps of average transfer rate sp
Pϕ = 10.667 MHz is selected (operated u
base clock with a frequency 8 times the t
rate)

011: 720 kbps of average transfer rate specifi
32 MHz is selected (operated using the b
with a frequency 16 times the transfer ra

100: Setting prohibited

101: 115.192 kbps of average transfer rate sp
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the trans

110: 460.784 kbps of average transfer rate sp
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the trans

111: 720 kbps of average transfer rate specifi
16 MHz is selected (operated using the b
with a frequency 8 times the transfer rate

The average transfer rate only supports opera
frequencies of 10.667 MHz, 16 MHz, and 32 I

Bit Name	—	—	—	ABCS	ACS3	ACS2	ACS1
Initial Value	Undefined	Undefined	Undefined	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	R	Reserved These bits are always read as undefined and c modified.
4	ABCS	0	R/W	Asynchronous Mode Base Clock Select (valid asynchronous mode) Selects the base clock for a 1-bit period. 0: The base clock has a frequency 16 times the rate 1: The base clock has a frequency 8 times the rate
3	ACS3	0	R/W	Asynchronous Mode Clock Source Select
2	ACS2	0	R/W	These bits select the clock source for the avera
1	ACS1	0	R/W	transfer rate function in the asynchronous mod
0	ACS0	0	R/W	the average transfer rate function is enabled, t clock is automatically specified regardless of th bit value. The average transfer rate only corres 8MHz, 10.667MHz, 12MHz, 16MHz, 24MHz, a 32MHz. No other clock is available. Setting of ACS0 must be done in the asynchronous mode C/$\overline{A}$ bit in SMR = 0) and the external clock inpu (the CKE bit I SCR = 1). The setting examples figures 18.3 and 18.4. (Each number in the four-digit number below corresponds to the value in the bits ACS3 to A left to right respectively.)

average transfer rate specific to Pϕ = 8M
selected (operated using the base clock
frequency 8 times the transfer rate)

0100: TMR clock input
This setting allows the TMR compare ma
to be used as the base clock. The table
shows the correspondence between the
channels and the compare match output

SCI Channel	TMR Unit	Compare M Output
SCI_5	Unit 2	TMO4, TMO
SCI_6	Unit 3	TMO6, TMO

0101: 115.196 kbps of average transfer rate sp
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the trans

0110: 460.784 kbps of average transfer rate sp
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the trans

0111: 720 kbps of average transfer rate specifi
16 MHz is selected (operated using the b
with a frequency 8 times the transfer rate

1011: 921.053 kbps of average transfer rate spe
Pϕ = 24 MHz is selected or 460.526 kbps
average transfer rate specific to Pϕ = 12M
selected (operated using the base clock w
frequency 8 times the transfer rate)

1100: 720 kbps of average transfer rate specific
32 MHz is selected (operated using the ba
with a frequency 16 times the transfer rate

1101: Reserved (setting prohibited)

111x: Reserved (setting prohibited)

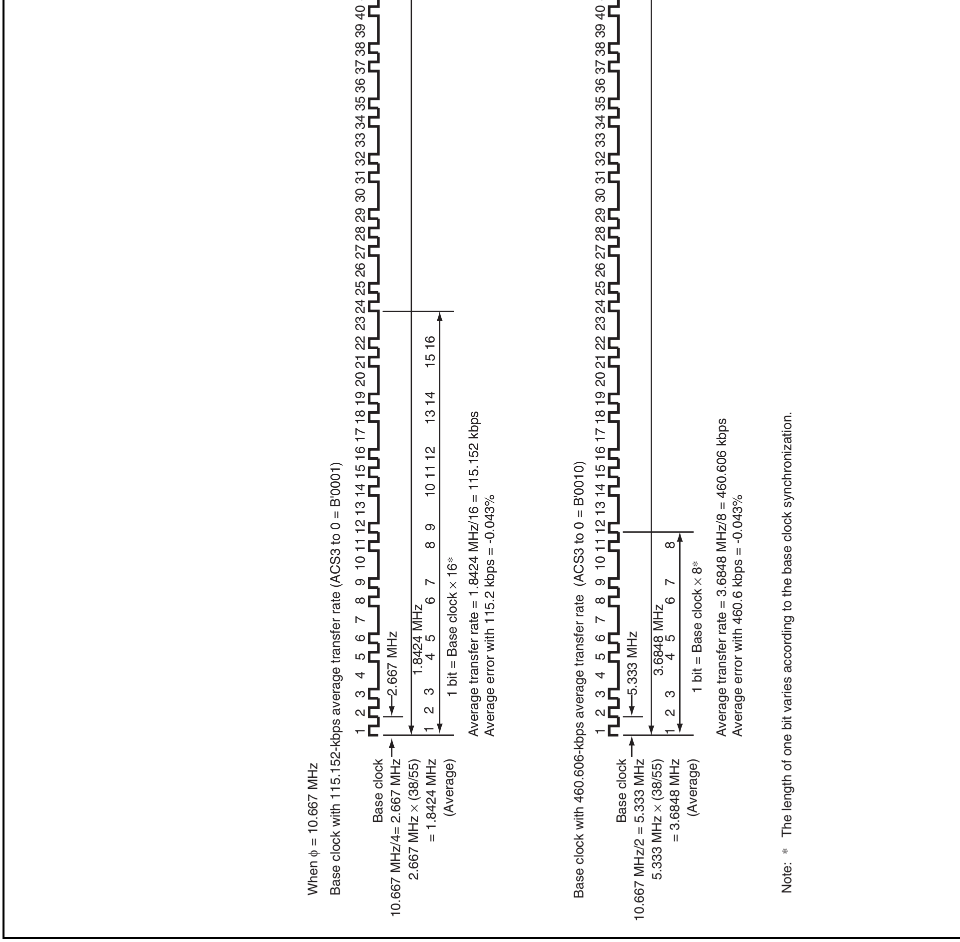


Figure 18.3 Examples of Base Clock when Average Transfer Rate Is Selecte

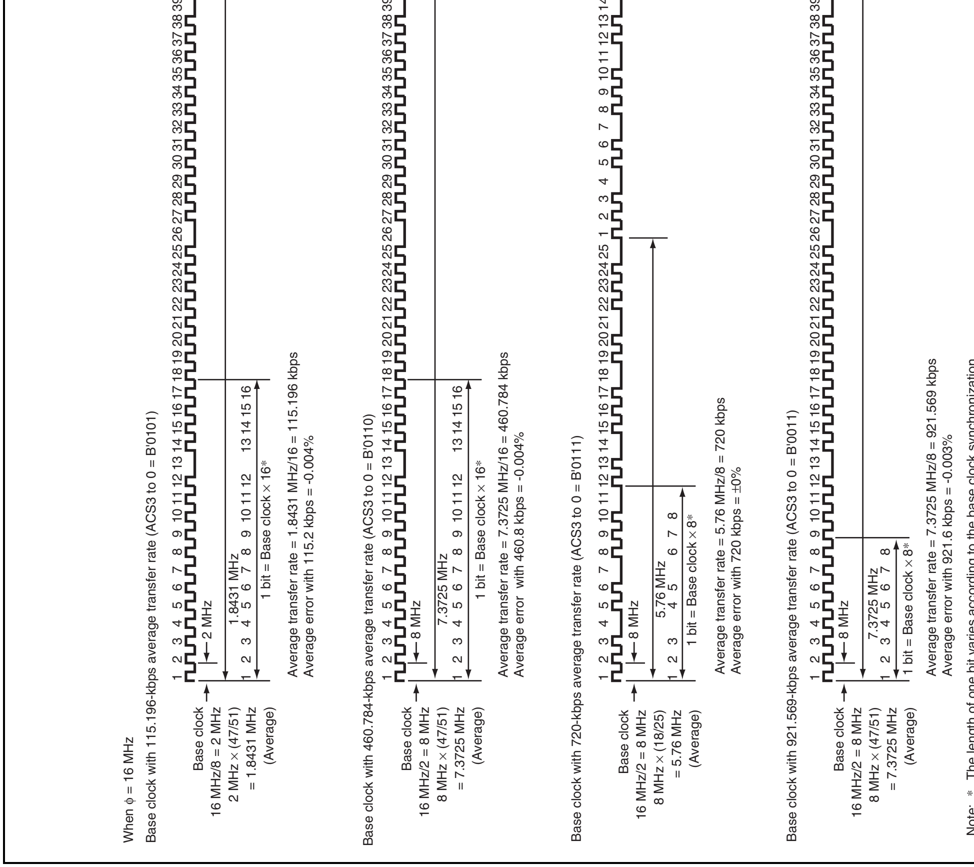


Figure 18.3 Examples of Base Clock when Average Transfer Rate Is Selected

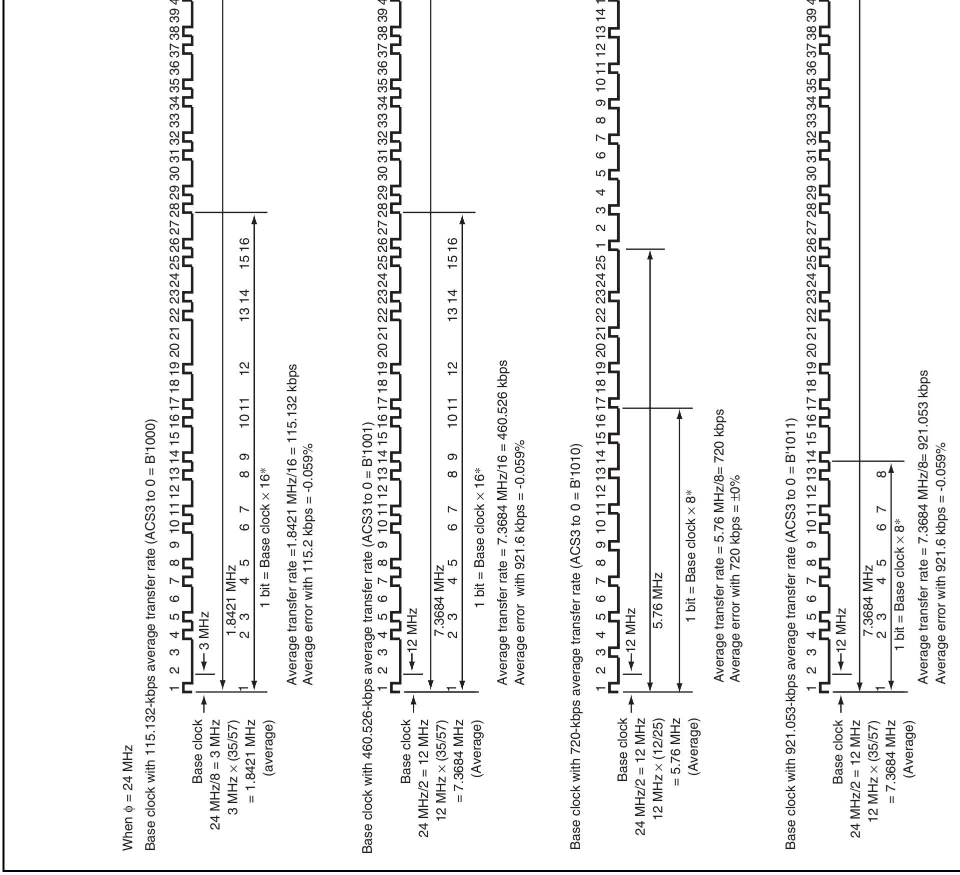


Figure 18.3 Examples of Base Clock when Average Transfer Rate Is Selecte

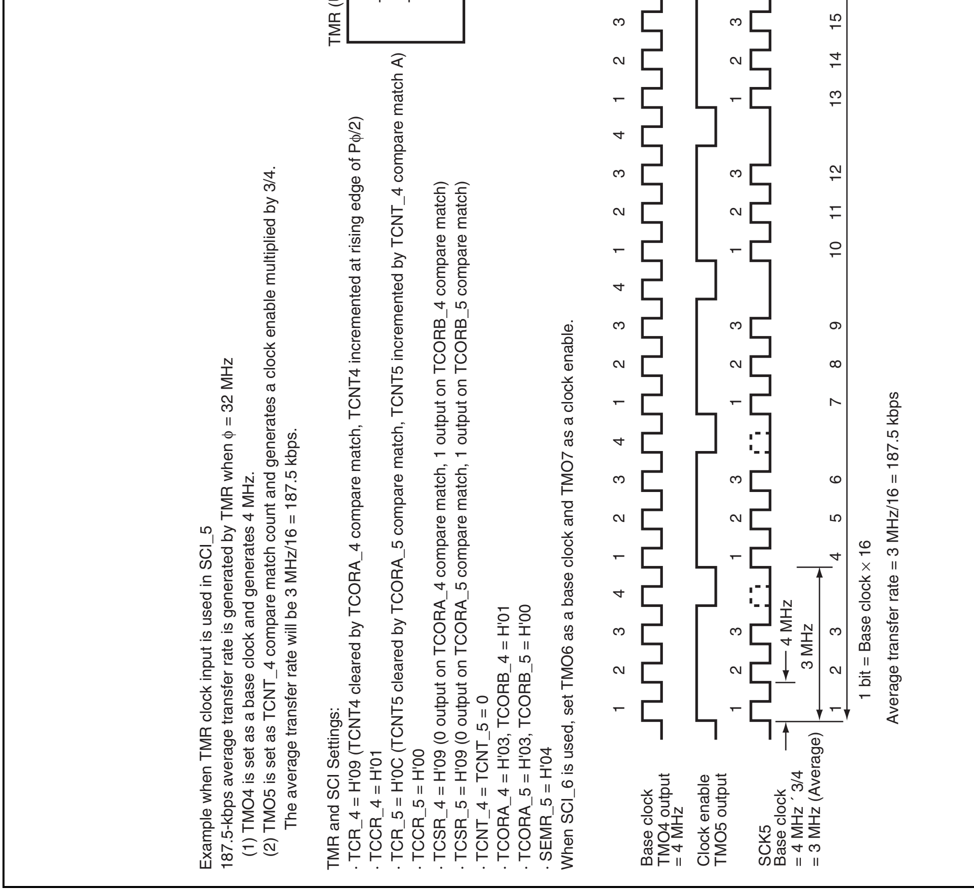


Figure 18.4 Example of Average Transfer Rate Setting when TMR Clock Is In

Bit	Bit Name	Value	R/W	Description
7	IrE	0	R/W	IrDA Enable* Sets the SCI_5 I/O to normal SCI or IrDA. 0: TxD5/IrTxD and RxD5/IrRxD pins operate a TxD5 and RxD5. 1: TxD5/IrTxD and RxD5/IrRxD pins are opera IrTxD and IrRxD.
6	IrCK2	0	R/W	IrDA Clock Select 2 to 0
5	IrCK1	0	R/W	Sets the pulse width of high state at encoding
4	IrCK0	0	R/W	output pulse when the IrDA function is enable 000: Pulse-width = B × 3/16 (Bit rate × 3/16) 001: Pulse-width = Pϕ/2 010: Pulse-width = Pϕ/4 011: Pulse-width = Pϕ/8 100: Pulse-width = Pϕ/16 101: Pulse-width = Pϕ/32 110: Pulse-width = Pϕ/64 111: Pulse-width = Pϕ/128
3	IrTxINV	0	R/W	IrTx Data Invert This bit specifies the inversion of the logic lev output. When inversion is done, the pulse wid state specified by the bits 6 to 4 becomes the width in low state. 0: Outputs the transmission data as it is as IrT 1: Outputs the inverted transmission data as I output

0.

Note: * The IrDA function should be used when the ABCS bit in SEMR_5 is set to 0 ar ACS3 to ACS0 bits in SEMR_5 are set to B'0000.

18.4 Operation in Asynchronous Mode

Figure 18.5 shows the general format for asynchronous serial communication. One frame of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level). In asynchronous serial communication, the communication line is usually held in the mark s (high level). The SCI monitors the communication line, and when it goes to the space stat level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitt receiver are independent units, enabling full-duplex communication. Both the transmitter receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

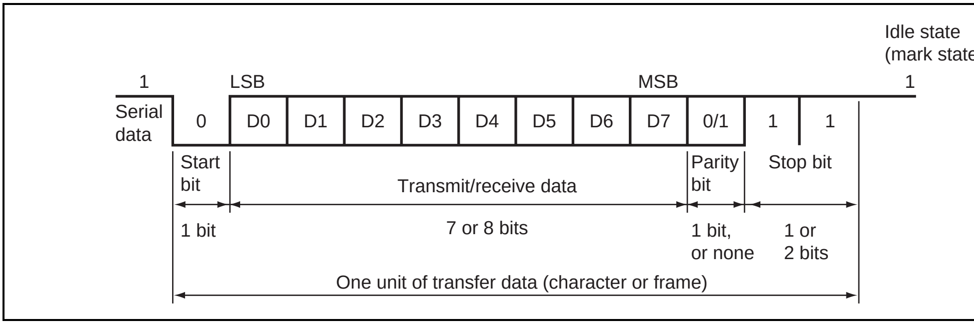


Figure 18.5 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

0	0	0	0	S / 8-bit data / STOP
0	0	0	1	S / 8-bit data / STOP / ST
0	1	0	0	S / 8-bit data / P / STO
0	1	0	1	S / 8-bit data / P / 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 / STO
0	–	1	0	S / 8-bit data / MPB / ST
0	–	1	1	S / 8-bit data / MPB / STO
1	–	1	0	S / 7-bit data / MPB / STOP
1	–	1	1	S / 7-bit data / MPB / STOP / STO

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

M: Reception margin
N: Ratio of bit rate to clock (When ABCS = 0, N = 16. When ABCS = 1, N = 8.)
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 determined formula below.

$$M = \left(0.5 - \frac{1}{2 \times 16} \right) \times 100 \quad [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowe system design.

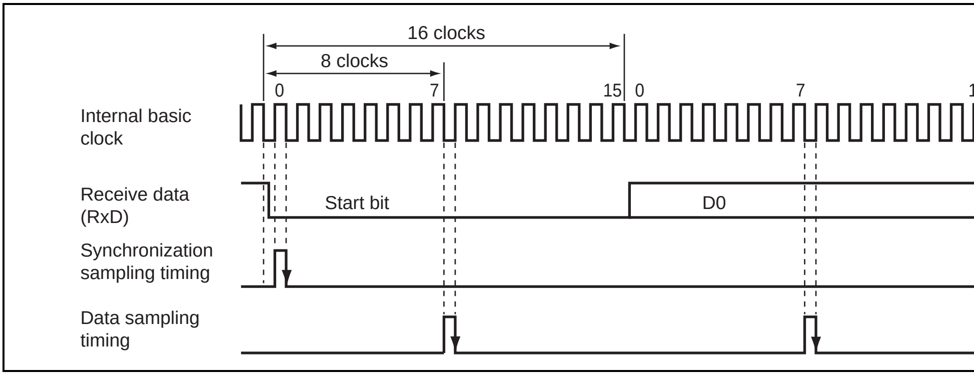


Figure 18.6 Receive Data Sampling Timing in Asynchronous Mode

Note: * This is an example when the ABCS bit in SEMR_2, 5, and 6 is 0. When the A is 1, a frequency of 8 times the bit rate is used as a base clock and receive data sampled at the rising edge of the 4th pulse of the base clock.

When the SCI is operated on an internal clock, the clock can be output from the SCK pi
frequency of the clock output in this case is equal to the bit rate, and the phase is such th
rising edge of the clock is in the middle of the transmit data, as shown in figure 18.7.

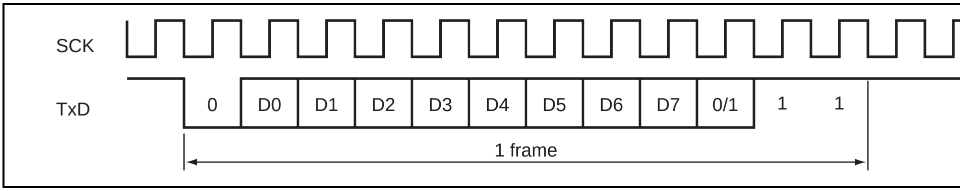
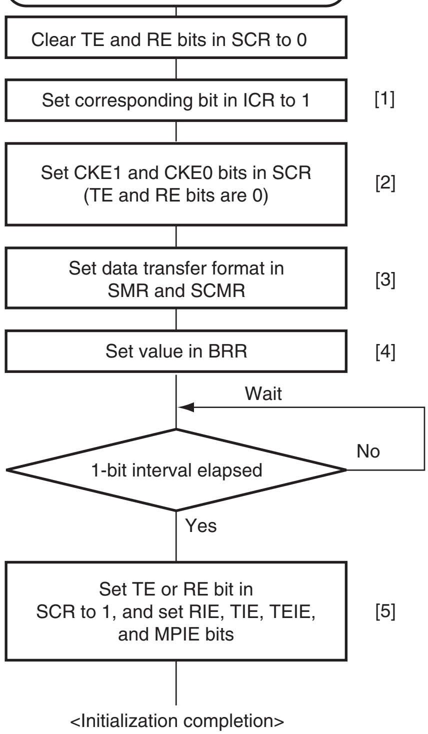


Figure 18.7 Phase Relation between Output Clock and Transmit Data (Asynchronous Mode)



[1] Set the bit in ICR for the corresponding pin when receiving data or using an external clock.

[2] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set the data transfer format in SMR and SCMR.

[4] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[5] Wait at least one bit interval, then set 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.

Figure 18.8 Sample SCI Initialization Flowchart

transmission of the current transmit data has finished, continuous transmission can b

3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity b 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 sto 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 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 18.10 shows a sample flowchart for transmission in asynchronous mode.

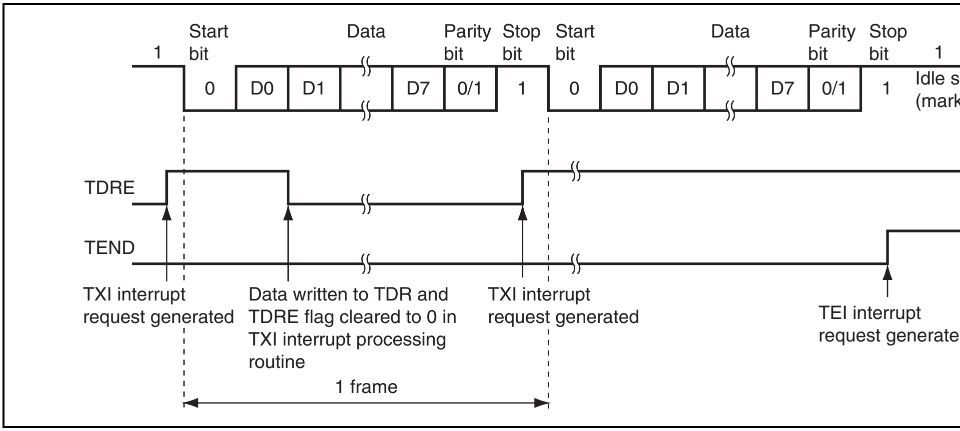
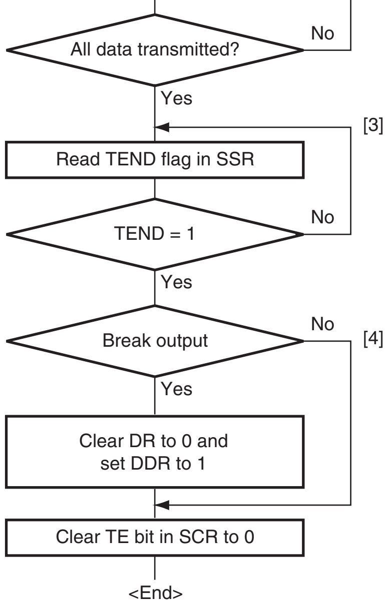


Figure 18.9 Example of Operation for Transmission in Asynchronous Mo (Example with 8-Bit Data, Parity, One Stop Bit)



To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 18.10 Example of 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 transfer RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is genera

4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 a data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI inter request is generated.

5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data i transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt re generated. Because the RXI interrupt processing routine reads the receive data trans RDR before reception of the next receive data has finished, continuous reception ca enabled.

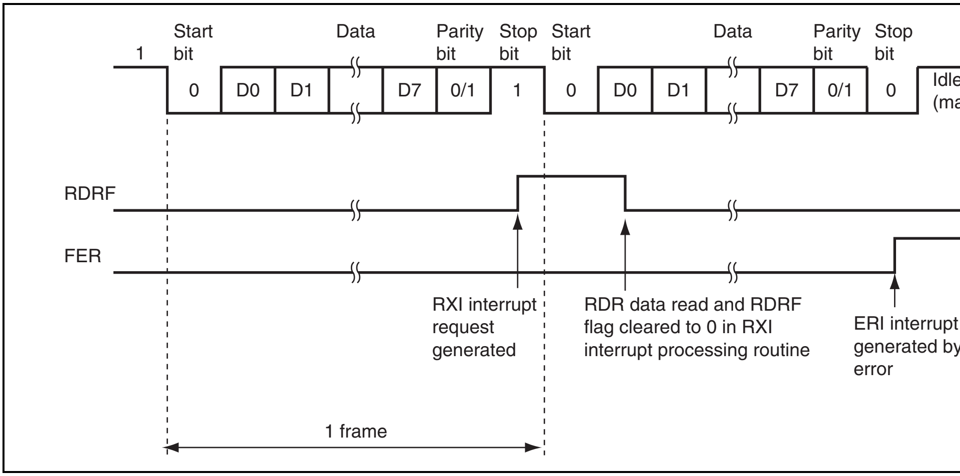
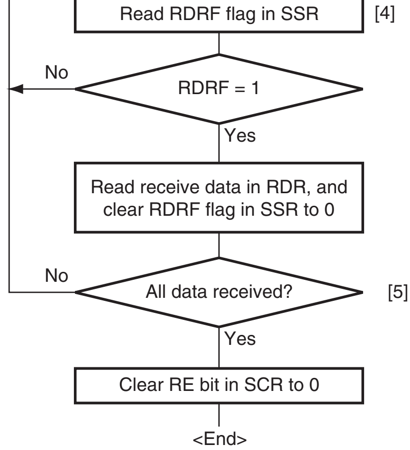


Figure 18.11 Example of SCI Operation for Reception (Example with 8-Bit Data, Parity, One Stop Bit)

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 + framing
1	1	0	1	Lost	Overrun error + parity e
0	0	1	1	Transferred to RDR	Framing error + parity e
1	1	1	1	Lost	Overrun error + framing parity error

Note: * The RDRF flag retains the state it had before data reception.



value of the input port corresponding to the RxD pin.

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

[5] Serial reception continuation procedure: To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag and RDR, and clear the RDRF flag to 0. However, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by an RXI interrupt and reads data from RDR.

Figure 18.12 Sample Serial Reception Flowchart (1)

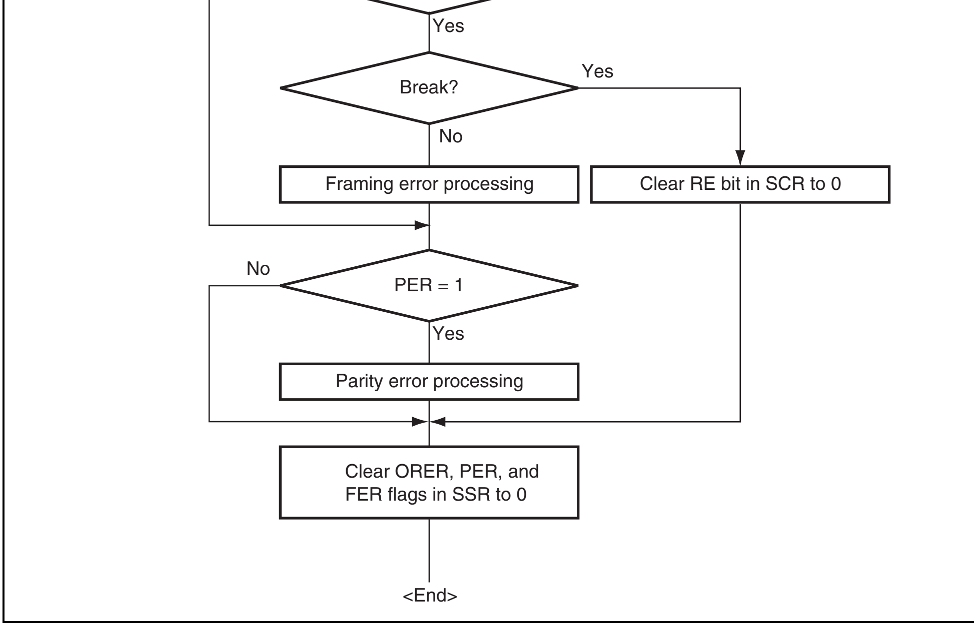


Figure 18.12 Sample Serial Reception Flowchart (2)

18.13 shows an example of inter-processor communication using the multiprocessor for
transmitting station first sends data which includes the ID code of the receiving station a
multiprocessor bit set to 1. It then transmits transmit data added with a multiprocessor bi
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 I
station whose ID matches then receives the data sent next. Stations whose ID does not m
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
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR statu
RDRF, FER, and ORER in SSR to 1 are prohibited until data with a 1 multiprocessor bi
received. On reception of a receive character with a 1 multiprocessor bit, the MPBR bit
set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If t
in SCR is set to 1 at this time, an RXI interrupt is generated.

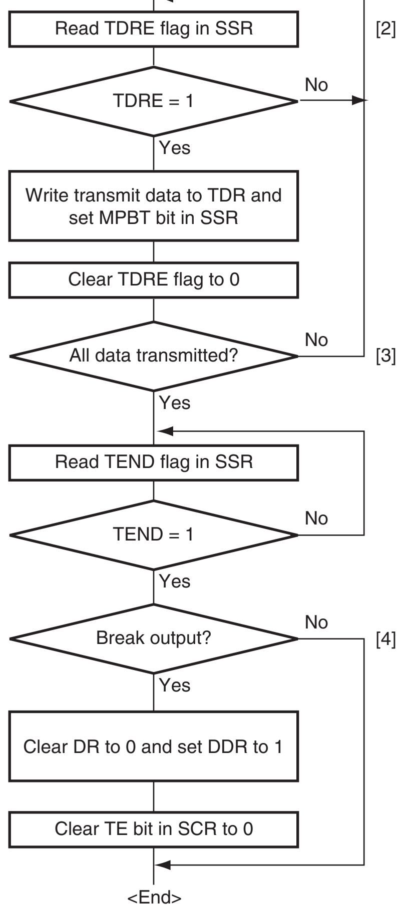
When the multiprocessor format is selected, the parity bit setting is invalid. All other bit
are the same as those in normal asynchronous mode. The clock used for multiprocessor
communication is the same as that in normal asynchronous mode.

ID transmission cycle = receiving station specification

Data transmission cycle = Data transmission to receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

Figure 18.13 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



to 1, a 1 is output for one frame, and transmission is enabled.

[2] SCI status check and transmit data write: Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure: To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission: To output a break in serial transmission, set DDR for the port to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Figure 18.14 Sample Multiprocessor Serial Transmission Flowchart

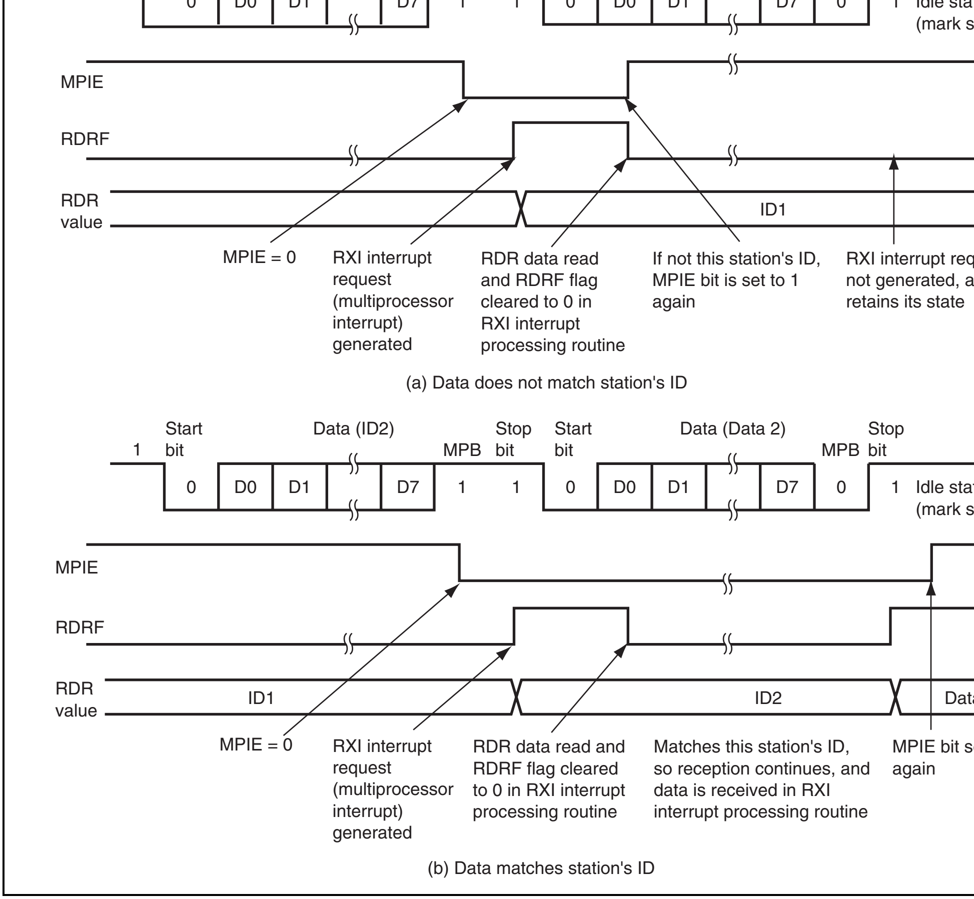
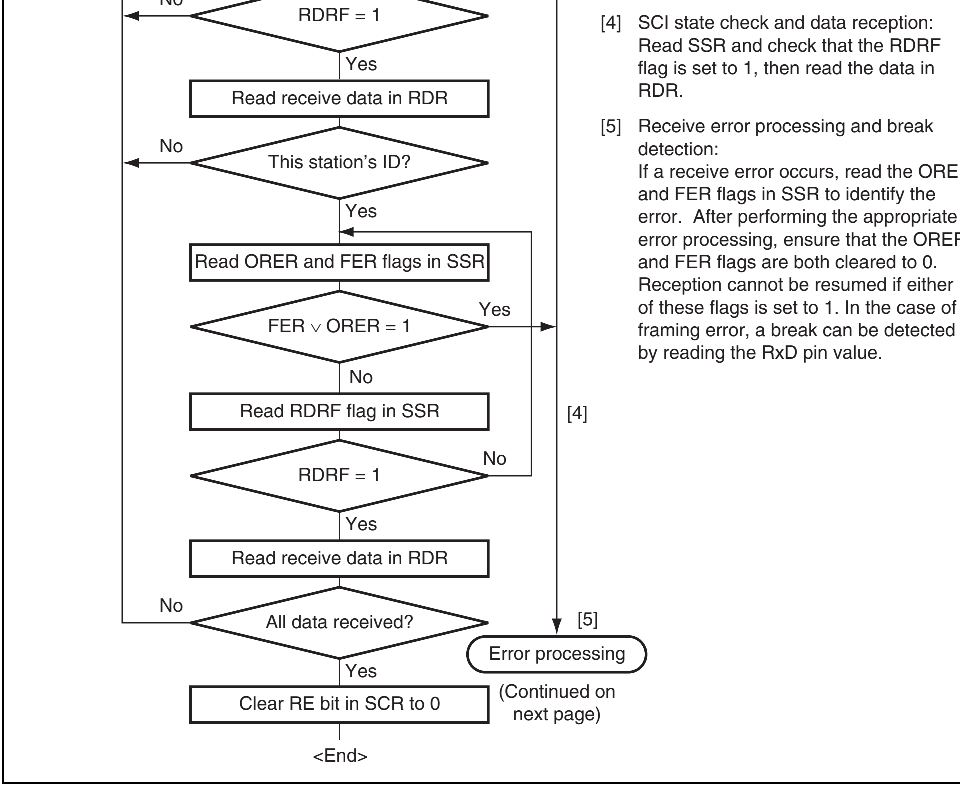


Figure 18.15 Example of SCI Operation for Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[4] SCI state check and data reception: Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error processing and break detection: If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FER flags are both cleared to 0. Reception cannot be resumed if either of these flags is set to 1. In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 18.16 Sample Multiprocessor Serial Reception Flowchart (1)

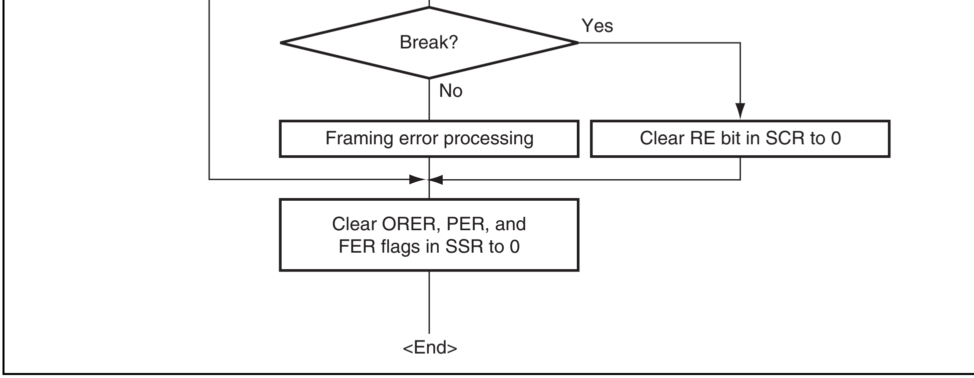
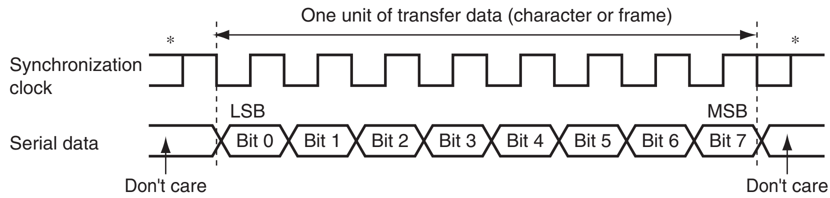


Figure 18.16 Sample Multiprocessor Serial Reception Flowchart (2)

receiver also have a double-buffered structure, so that the next transmit data can be writ
transmission or the previous receive data can be read during reception, enabling continu
transfer. (Setting is prohibited in SCI_5 and SCI_6.)

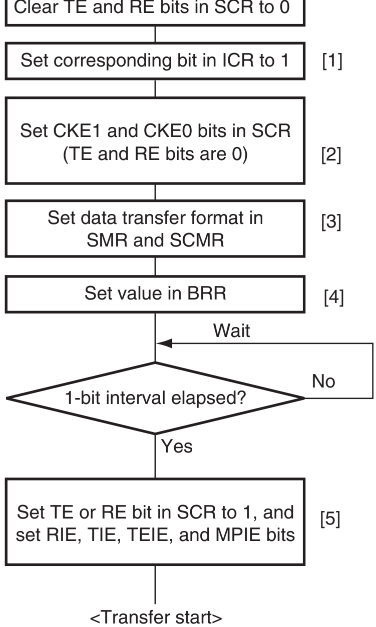


Note: * Holds a high level except during continuous transfer.

Figure 18.17 Data Format in Clocked Synchronous Communication (LSB-F

18.6.1 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 th and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronizati 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 cas reception only, the synchronization clock is output until an overrun error occurs or until is cleared to 0. (Setting is prohibited in SCI_5 and SCI_6.)



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 18.18 Sample SCI Initialization Flowchart

transmission of the current transmit data has finished, continuous transmission can b

3. 8-bit data is sent from the TxD pin synchronized with the output clock when clock o mode has been specified and synchronized with the input clock when use of an exter 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 retain 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 18.20 shows a sample flowchart for serial data transmission. Even if the TDRE fl cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) Make sure to clear the receive error flags to 0 before starting transmission. Note that cle RE bit to 0 does not clear the receive error flags.

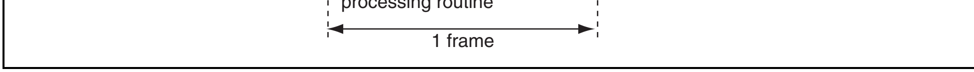


Figure 18.19 Example of Operation for Transmission in Clocked Synchronous

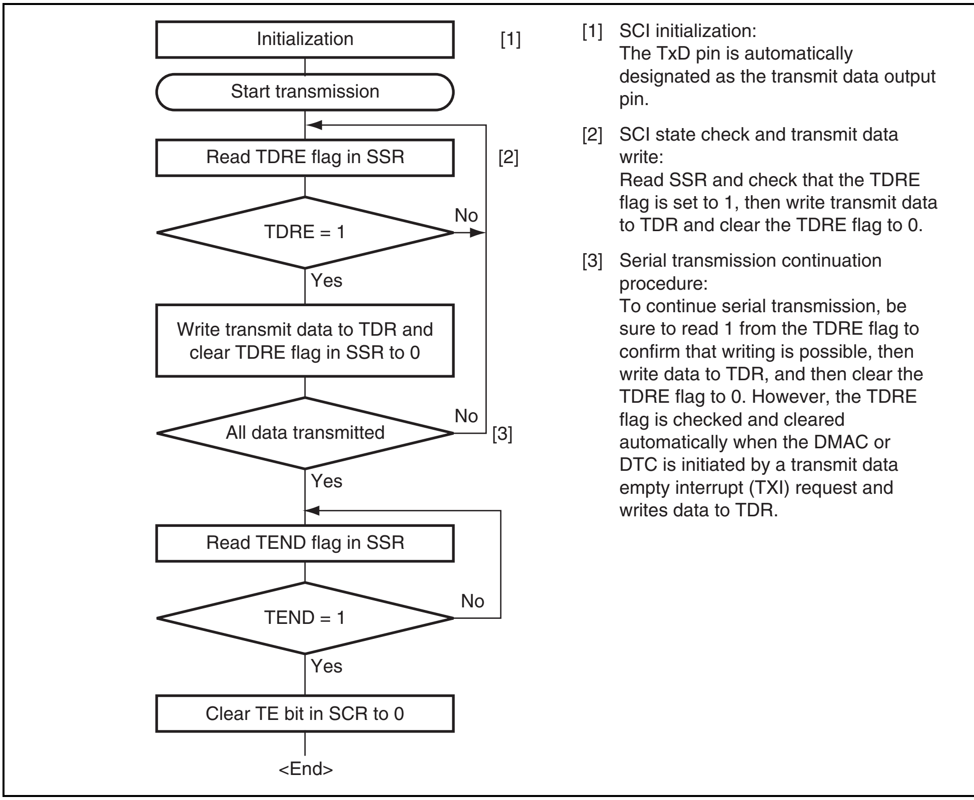


Figure 18.20 Sample Serial Transmission Flowchart

flag remains to be set to 1.

3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data i transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt rec generated. Because the RXI interrupt processing routine reads the receive data transf RDR before reception of the next receive data has finished, continuous reception car enabled.

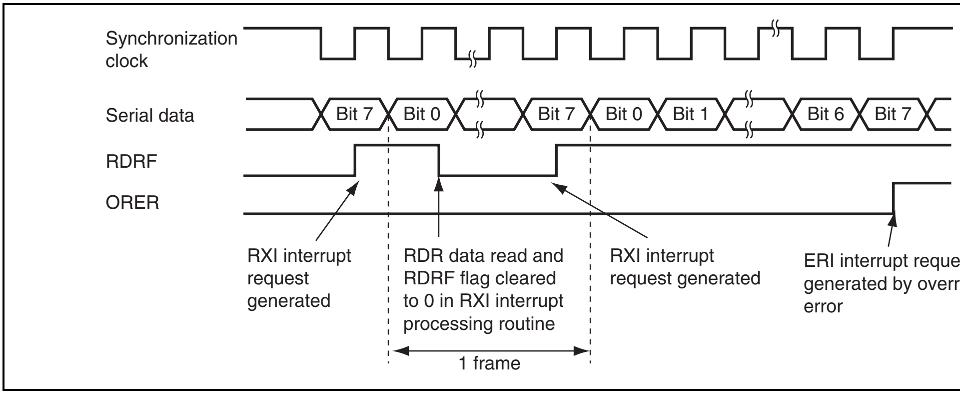
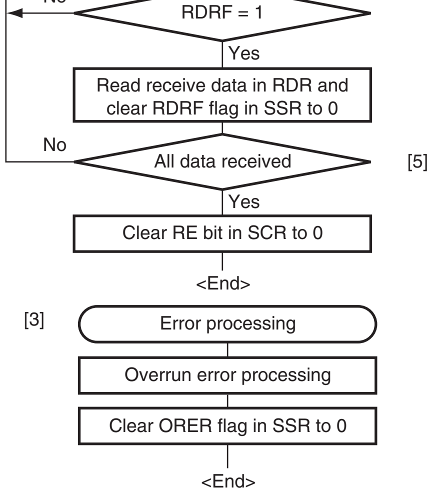


Figure 18.21 Example of Operation for Reception in Clocked Synchronous M

Transfer cannot be resumed while a receive error flag is set to 1. Accordingly, clear the FER, PER, and RDRF bits to 0 before resuming reception. Figure 18.22 shows a sample for serial data reception.



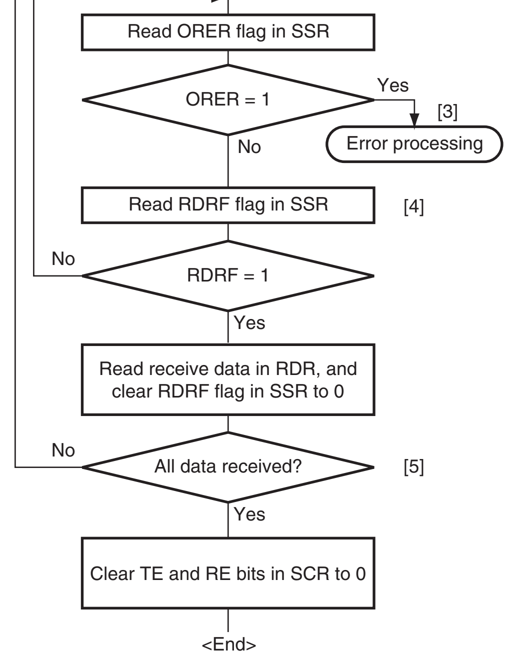
data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0 should be finished. However, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

Figure 18.22 Sample Serial Reception Flowchart

18.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchron Mode) (SCI_0, 1, 2, and 4 only)

Figure 18.23 shows a sample flowchart for simultaneous serial transmit and receive opera After initializing the SCI, the following procedure should be used for simultaneous serial transmit and receive operations. To switch from transmit mode to simultaneous transmit a receive mode, after checking that the SCI has finished transmission and the TDRE and TI flags are set to 1, clear the TE bit to 0. Then simultaneously set both the TE and RE bits t a single instruction. To switch from receive mode to simultaneous transmit and receive m after checking that the SCI has finished reception, clear the RE bit to 0. Then after checki 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.



ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Reception cannot be resumed i 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 th receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/ reception, before the MSB (bit 7) o the current frame is received, finis reading the RDRF flag, reading RDR, and clearing the RDRF flag t 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
However, the TDRE flag is checke and cleared automatically when th DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR. Similarly, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 18.23 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 smart card not connected enables closed transmission/reception
self diagnosis. To supply the smart card with the clock pulses generated by the SCI, input
pin output to the CLK pin of the smart card. A reset signal can be supplied via the output
this LSI. (In SCI_5 and SCI-6, the clock generated in SCI cannot be provided to smart ca

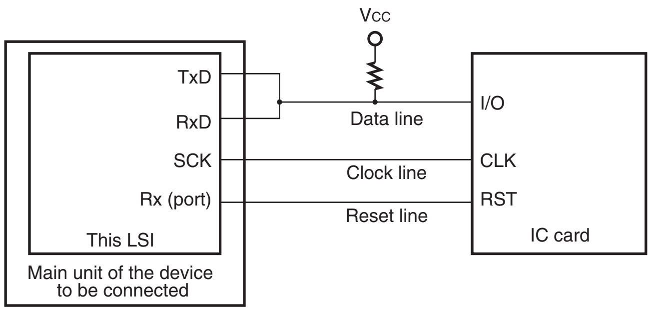


Figure 18.24 Pin Connection for Smart Card Interface

after at least 2 etu.

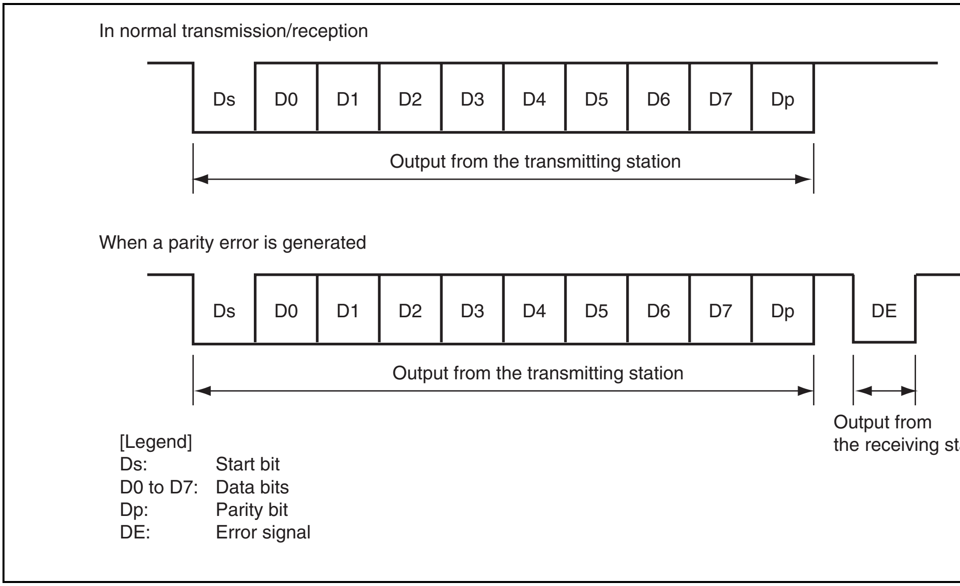


Figure 18.25 Data Formats in Normal Smart Card Interface Mode

For communication with the smart cards of the direct convention and inverse conventio follow the procedure below.

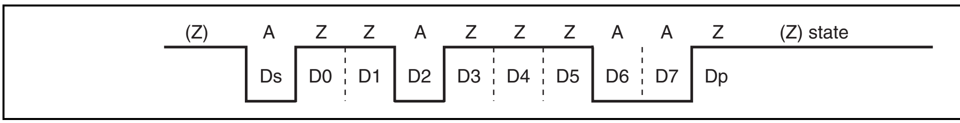


Figure 18.26 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

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 18.27. Th data in the start character in the figure is H'3F. When using the inverse convention type, both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even 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/$\overline{E}$ bit in SMR to invert the parity both transmission and reception.

18.7.3 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 no smart card interface mode, the flag is always read as 0 because no error signal is trans

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

[Legend]

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.

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

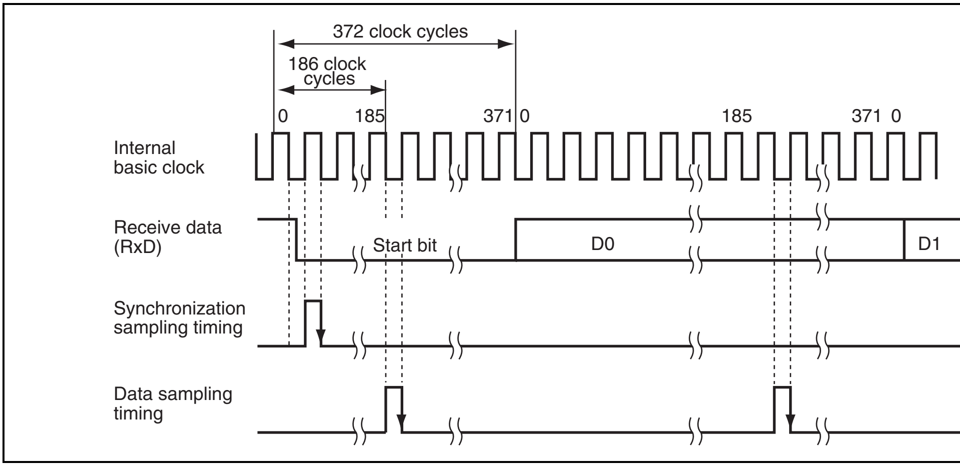


Figure 18.28 Receive Data Sampling Timing in Smart Card Interface Mo
(When Clock Frequency is 372 Times the Bit Rate)

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 18.31 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 12, Data Controller (DTC) and section 10, DMA Controller (DMAC).

Figure 18.29 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 18.30 shows the TEND flag set timing.

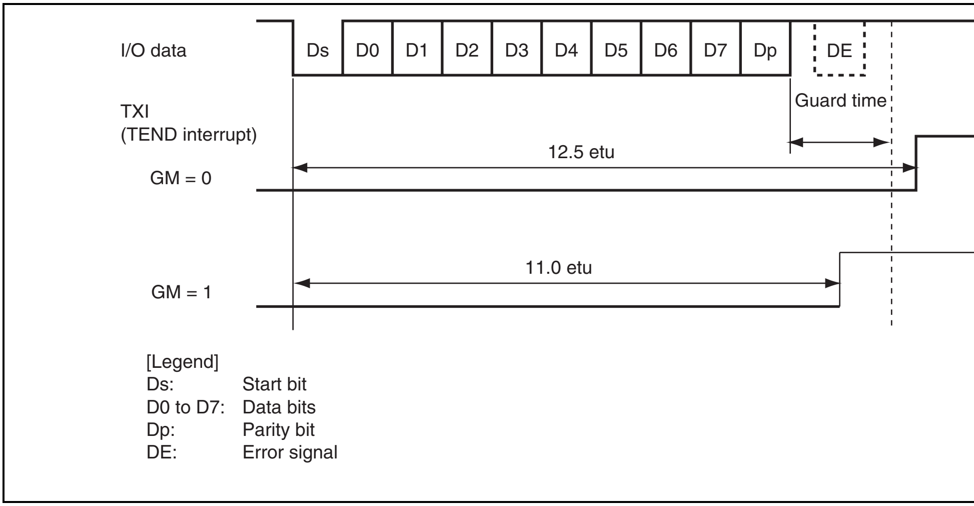


Figure 18.30 TEND Flag Set Timing during Transmission

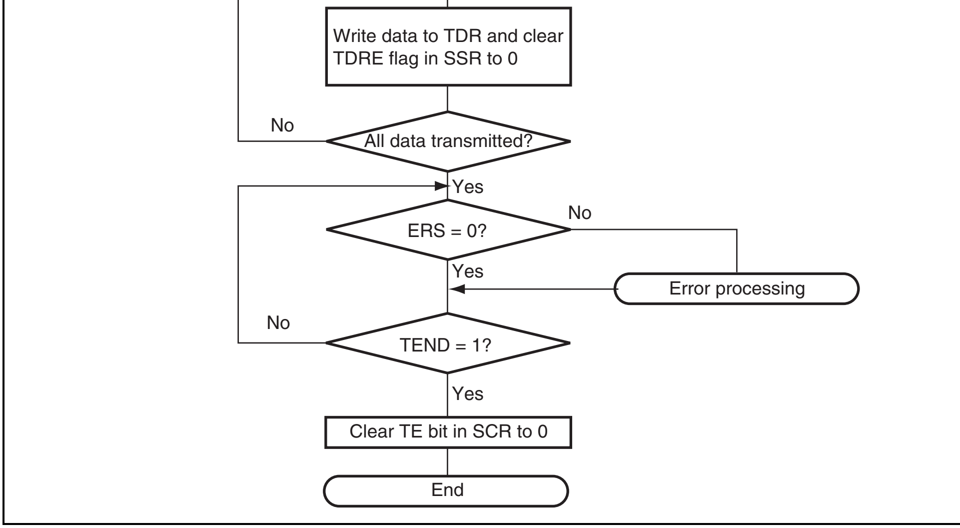


Figure 18.31 Sample Transmission Flowchart

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 18.33 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 du reception, i.e., either the ORER or PER flag is set to 1, a transmit/receive error interrupt (request is generated and the error flag must be cleared. If an error occurs, the 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 18.4, Operation in Asynchrono

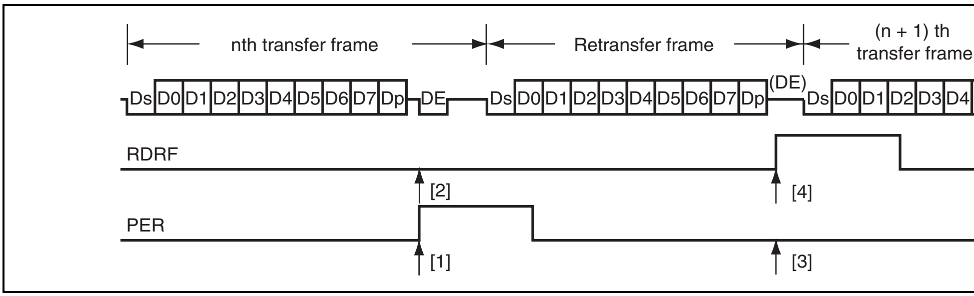


Figure 18.32 Data Re-Transfer Operation in SCI Reception Mode

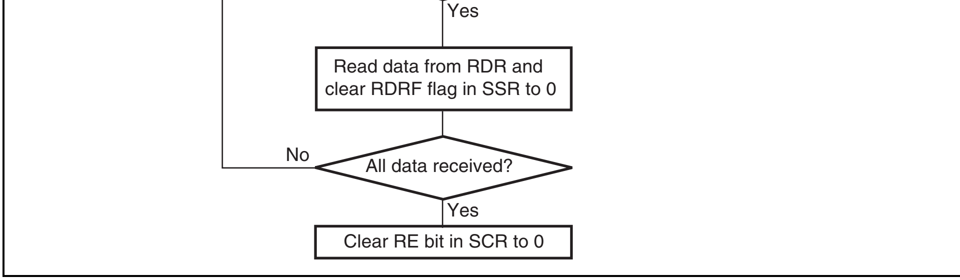


Figure 18.33 Sample Reception Flowchart

18.7.8 Clock Output Control (Only SCI_0, 1, 2, and 4)

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 18.34 shows an example of clock output fixing timing when the CKE0 bit is cont
with GM = 1 and CKE1 = 0.

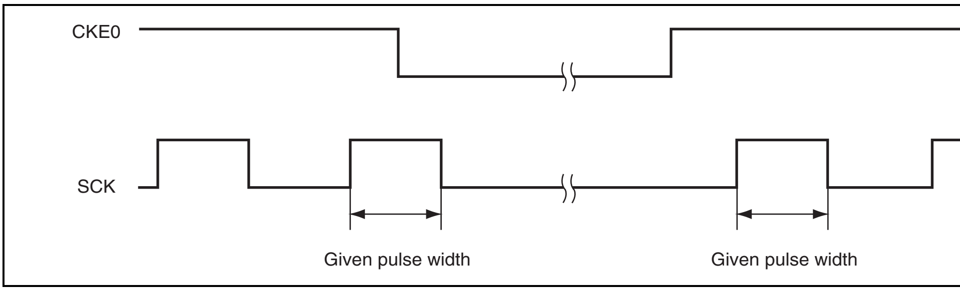


Figure 18.34 Clock Output Fixing Timing

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

- At mode switching

a) 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. (SCI_0, 1 4 only)
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.

b) At transition from software standby mode to smart card interface 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.

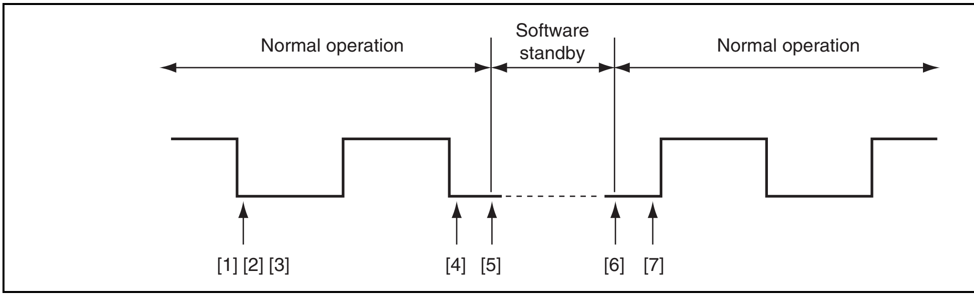


Figure 18.35 Clock Stop and Restart Procedure

rate, the transfer rate must be modified through programming.

Figure 18.36 is the IrDA block diagram.

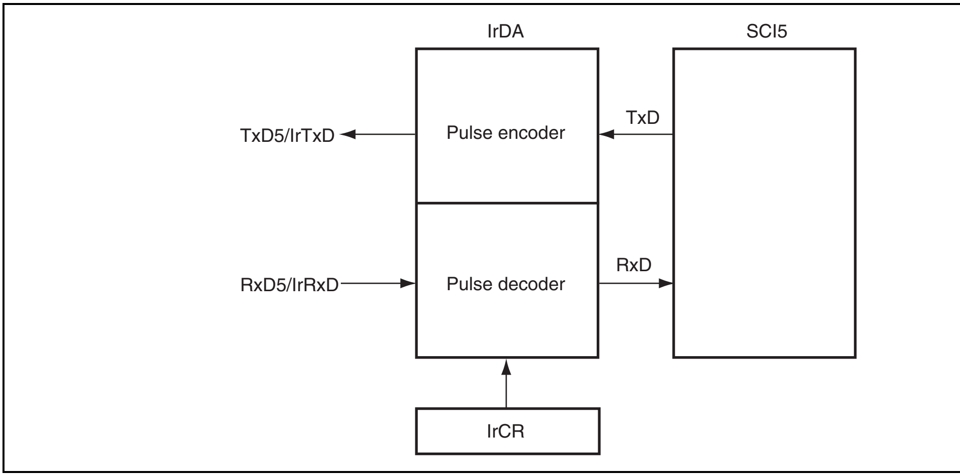


Figure 18.36 IrDA Block Diagram

Note: * The IrDA function should be used when the ABCS bit in SEMR_5 is set to 0
ACS3 to ACS0 bits in SEMR_5 are set to B'0000.

MHz, a high-level pulse width of 1.6 μs can be specified because it is the smallest value i
range greater than 1.41 μs.

For serial data of level 1, no pulses are output.

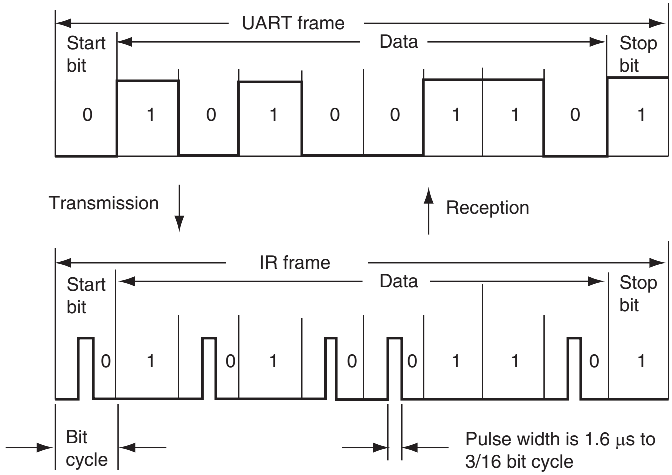


Figure 18.37 IrDA Transmission and Reception

(2) Reception

During reception, IR frames are converted to UART frames using the IrDA interface befo
inputting to SCI. 0 is output when the high level pulse is detected while 1 is output when
is detected during one bit period. Note that a pulse shorter than the minimum pulse width
μs is also regarded as a 0 signal.

7.3728	100	100	100	100	100	10
8	100	100	100	100	100	10
9.8304	100	100	100	100	100	10
10	100	100	100	100	100	10
12	101	101	101	101	101	10
12.288	101	101	101	101	101	10
14	101	101	101	101	101	10
14.7456	101	101	101	101	101	10
16	101	101	101	101	101	10
17.2032	101	101	101	101	101	10
18	101	101	101	101	101	10
19.6608	101	101	101	101	101	10
20	101	101	101	101	101	10
25	110	110	110	110	110	11
30	110	110	110	110	110	11
33	110	110	110	110	110	11
35	110	110	110	110	110	11

DTC or DMAC to allow data transfer. The TDRE flag is automatically cleared to 0 at dat
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 interru
activate the DTC or DMAC to allow data transfer. The RDRF flag is automatically cleare
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 1
interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority
acceptance. However, note that if the TDRE and TEND flags are cleared to 0 simultaneou
the TXI interrupt processing routine, the SCI cannot branch to the TEI interrupt processin
later.

Note that the priority order for interrupts is different between the group of SCI_0, 1, 2, an
the group of SCI_5 and SCI_6.

Table 18.14 SCI Interrupt Sources (SCI_0, 1, 2, and 4)

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
TXI	Transmit data empty	TDRE	Possible	Possible
TEI	Transmit end	TEND	Not possible	Not possible

Table 18.16 shows the interrupt sources in smart card interface mode. A transmit end (T
interrupt request cannot be used in this mode.

Note that the priority order for interrupts is different between the group of SCI_0, 1, 2, a
the group of SCI_5 and SCI_6.

Table 18.16 SCI Interrupt Sources (SCI_0, 1, 2, and 4)

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
TXI	Transmit data empty	TEND	Possible	Possible

Table 18.17 SCI Interrupt Sources (SCI_5 and SCI_6)

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
RXI	Receive data full	RDRF	Not possible	Possible
TXI	Transmit data empty	TDRE	Not possible	Possible
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible

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 12, Data T
Controller (DTC) and section 10, DMA Controller (DMAC).

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.

18.10.3 Mark State and Break Detection

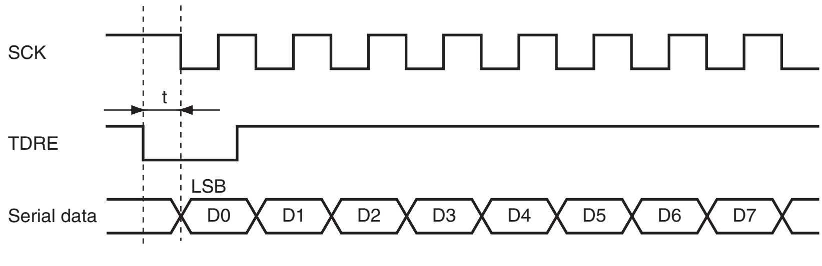
When the TE bit is 0, the TxD pin is used as an I/O port whose direction (input or outpu
level are determined by DR and DDR. This can be used to set the TxD pin to mark state
level) or send a break during serial data transmission. To maintain the communication li
state (the state of 1) until TE is set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

18.10.4 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.

18.10.6 Restrictions on Using DTC or DMAC

- When the external clock source is used as a synchronization clock, update TDR by the or DTC 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 (see figure 18.38).
- When using the DMAC or DTC 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 18.38 Sample Transmission using DTC in Clocked Synchronous Mo

- The DTC is not activated by the RXI or TXI request by SCI_5 or SCI6.

For using the IrDA function, set the IrE bit in addition to setting the TE bit.

Figure 18.39 shows a sample flowchart for transition to software standby mode during transmission. Figures 18.40 and 18.41 show the port pin states during transition to softw standby mode.

Before specifying the module stop state or making a transition to software standby mode transmission mode using DTC transfer, stop all transmit operations (TE = TIE = TEIE = Setting the TE and TIE bits to 1 after cancellation sets the TXI flag to start transmission DTC.

Reception: Before specifying the module stop state or making a transition to software st mode, stop the receive operations (RE = 0). RSR, RDR, and SSR are reset. If transition during data reception, the data being received will be invalid.

To receive data in the same reception mode after cancellation of the power-down state, s bit to 1, and then start reception. To receive data in a different reception mode, initialize first.

For using the IrDA function, set the IrE bit in addition to setting the RE bit.

Figure 18.42 shows a sample flowchart for mode transition during reception.

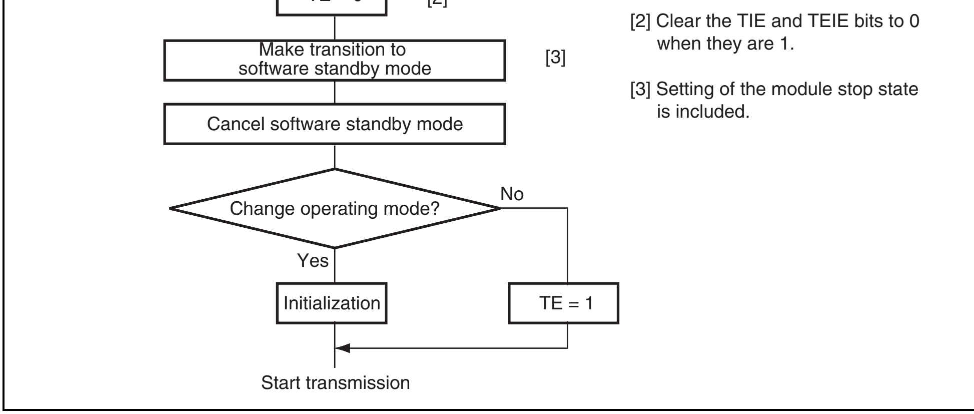


Figure 18.39 Sample Flowchart for Software Standby Mode Transition duri Transmission

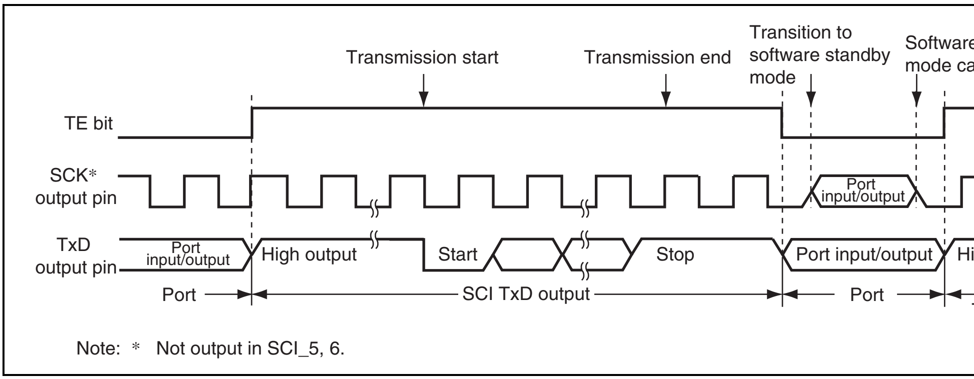


Figure 18.40 Port Pin States during Software Standby Mode Transition (Internal Clock, Asynchronous Transmission)

**Figure 18.41 Port Pin States during Software Standby Mode Transition
(Internal Clock, Clocked Synchronous Transmission)
(Setting is Prohibited in SCI_5 and SCI_6)**

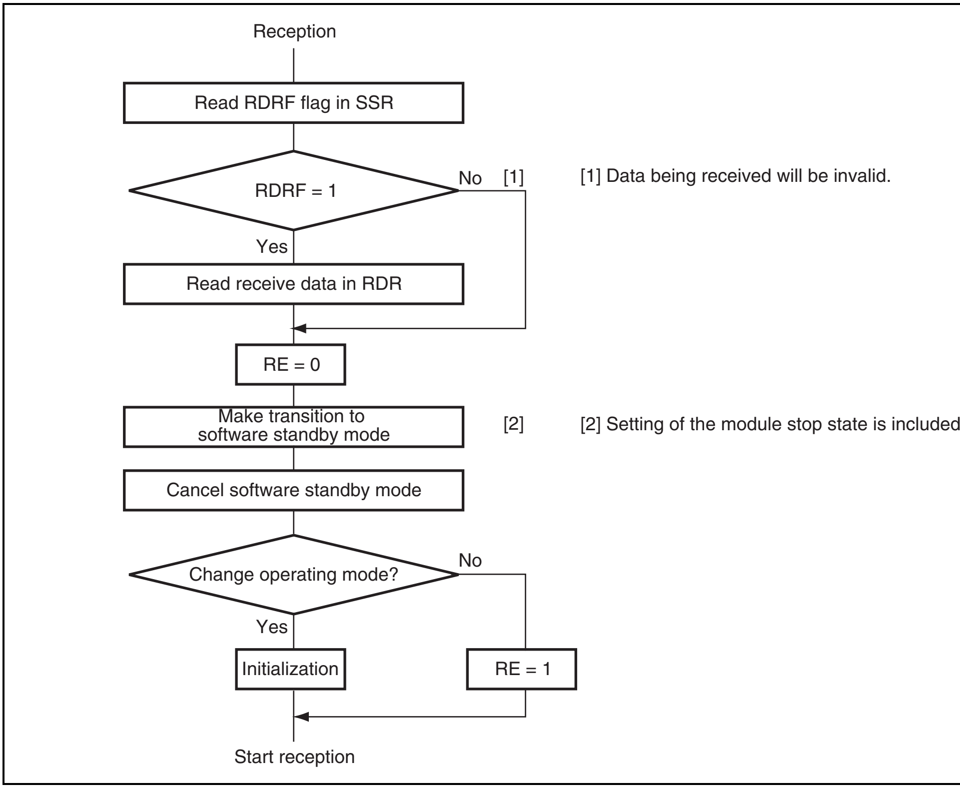


Figure 18.42 Sample Flowchart for Software Standby Mode Transition during R

- One of three generating polynomials selectable
- CRC code generation for LSB-first or MSB-first communication selectable

Figure 18.43 shows a block diagram of the CRC operation circuit.

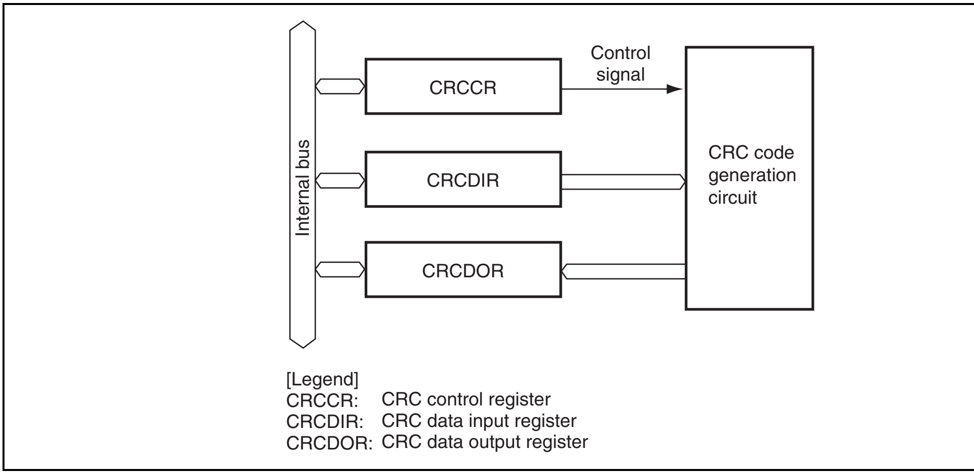


Figure 18.43 Block Diagram of CRC Operation Circuit

generating polynomial.

Bit	7	6	5	4	3	2	1
Bit Name	DORCLR	—	—	—	—	LMS	G1
Initial Value	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	DORCLR	0	W	CRCDOR Clear Setting this bit to 1 clears CRCDOR to H'0000
6 to 3	—	All 0	R	Reserved The initial value should not be changed.
2	LMS	0	R/W	CRC Operation Switch Selects CRC code generation for LSB-first or communication. 0: Performs CRC operation for LSB-first communication. The lower byte (bits 7 to 0) transmitted when CRCDOR contents (CRC are divided into two bytes to be transmitted parts. 1: Performs CRC operation for MSB-first communication. The upper byte (bits 15 to transmitted when CRCDOR contents (CRC are divided into two bytes to be transmitted parts.

CRCDIR is an 8-bit readable/writable register, to which the bytes to be CRC-operated are
The result is obtained in CRCDOR.

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

(3) CRC Data Output Register (CRCDOR)

CRCDOR is a 16-bit readable/writable register that contains the result of CRC operation bytes to be CRC-operated are written to CRCDIR after CRCDOR is cleared. When the C operation result is additionally written to the bytes to which CRC operation is to be perfo CRC operation result will be H'0000 if the data contains no CRC error. When bits 1 and 0 CRCCR (G1 and G0 bits) are set to 0 and 1, respectively, the lower byte of this register c the result.

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	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

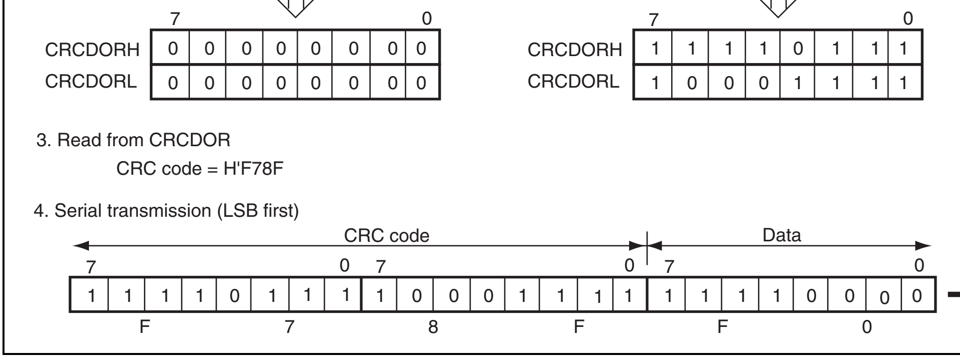


Figure 18.44 LSB-First Data Transmission

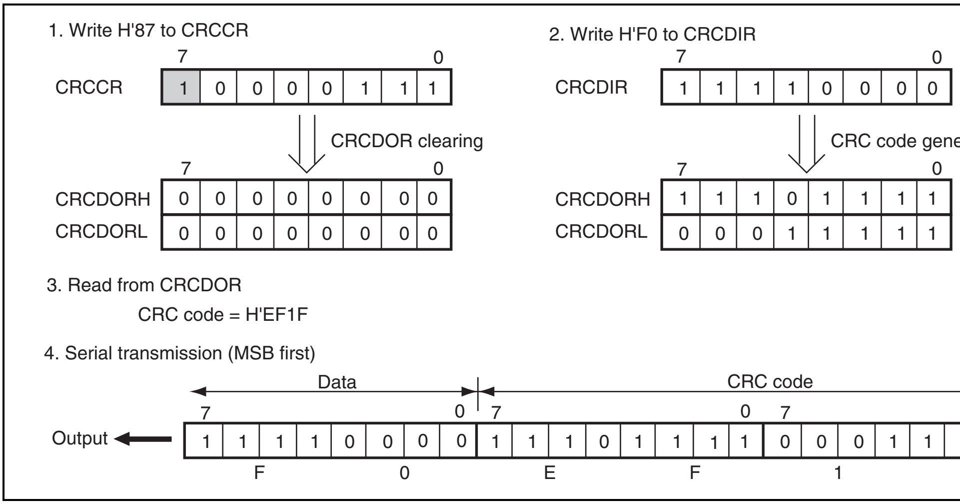


Figure 18.45 MSB-First Data Transmission

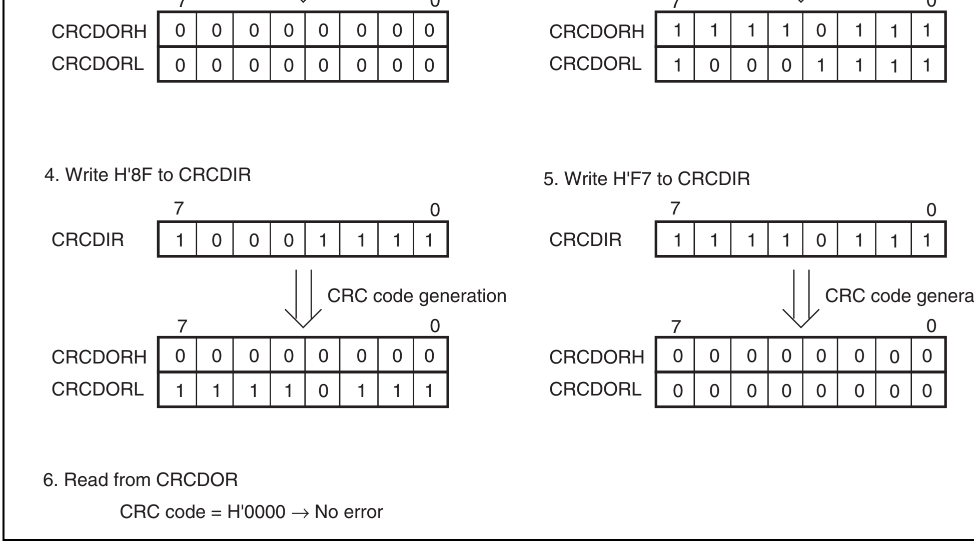


Figure 18.46 LSB-First Data Reception

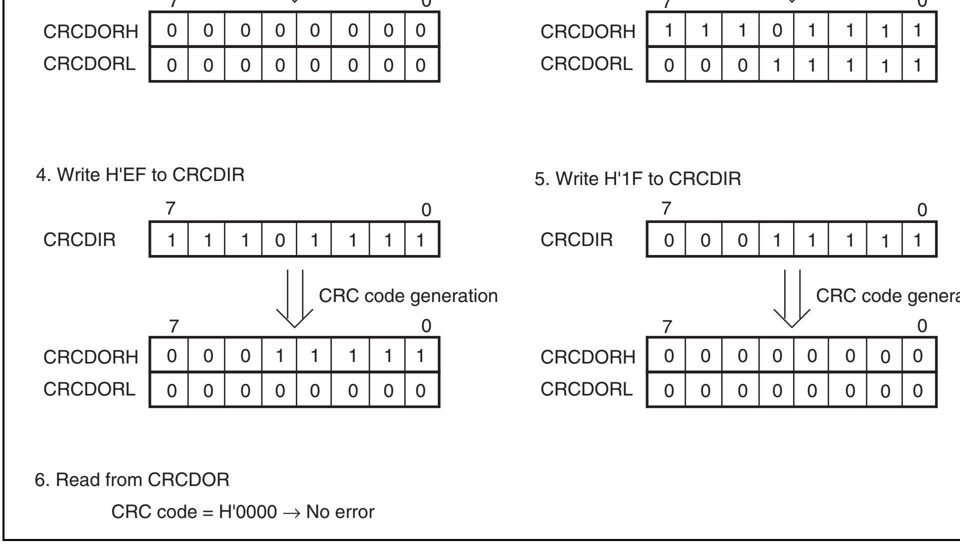


Figure 18.47 MSB-First Data Reception

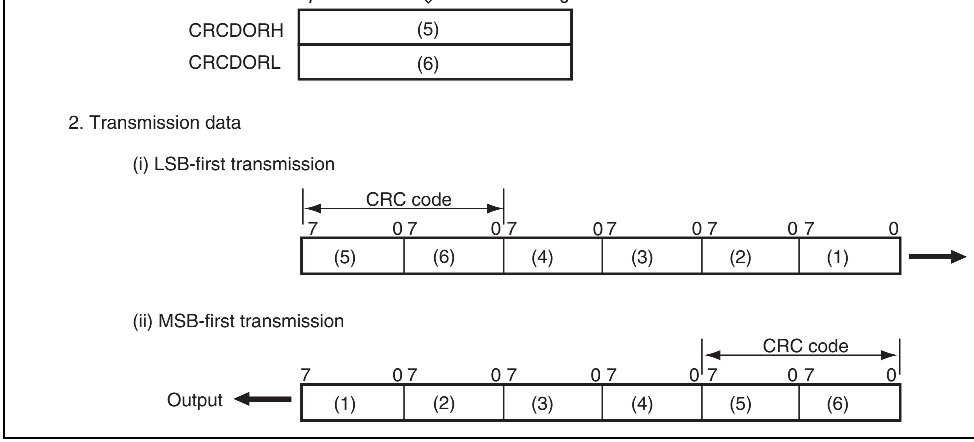


Figure 18.48 LSB-First and MSB-First Transmit Data

- Transfer speed: Supports full-speed (12 Mbps)
- Endpoint configuration:

Endpoint Name	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DMA Tr
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	8	8	—
	EP0o	Control-out	8	8	—
Endpoint 1	EP1	Bulk-out	64	128	Possibl
Endpoint 2	EP2	Bulk-in	64	128	Possibl
Endpoint 3	EP3	Interrupt-in	8	8	—

Configuration1-Interface0-AlternateSetting0
- EndPoint1
- EndPoint2
- EndPoint3

- Interrupt requests: Generates various interrupt signals necessary for USB transmission/reception
- Power mode: Self power mode or bus power mode can be selected by the power mo (PWMD) in the control register (CTLR).

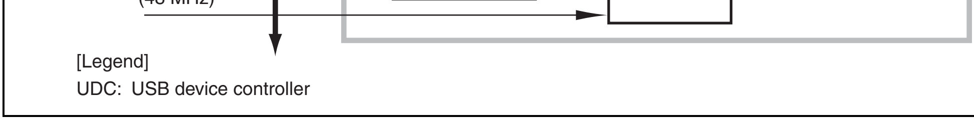


Figure 19.1 Block Diagram of USB

19.2 Input/Output Pins

Table 19.1 shows the USB pin configuration.

Table 19.1 Pin Configuration

Pin Name	I/O	Function
VBUS	Input	USB cable connection monitor pin
USD+	I/O	USB data I/O pin
USD-	I/O	USB data I/O pin
DrVcc	Input	Power supply pin for USB on-chip transceiver
DrVss	Input	Ground pin for USB on-chip transceiver

- Interrupt select register 2 (ISR2)
- Interrupt enable register 0 (IER0)
- Interrupt enable register 1 (IER1)
- Interrupt enable register 2 (IER2)
- EP0i data register (EPDR0i)
- EP0o data register (EPDR0o)
- EP0s data register (EPDR0s)
- EP1 data register (EPDR1)
- EP2 data register (EPDR2)
- EP3 data register (EPDR3)
- EP0o receive data size register (EPSZ0o)
- EP1 receive data size register (EPSZ1)
- Trigger register (TRG)
- Data status register (DASTS)
- FIFO clear register (FCLR)
- DMA transfer setting register (DMA)
- Endpoint stall register (EPSTL)
- Configuration value register (CVR)
- Control register (CTLR)
- Endpoint information register (EPIR)
- Transceiver test register 0 (TRNTREG0)
- Transceiver test register 1 (TRNTREG1)

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

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus Reset This bit is set to 1 when a bus reset signal is det the USB bus. (When the CPU is used to clear this flag by writi while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
6	EP1 FULL	0	R	EP1 FIFO Full This bit is set when endpoint 1 receives one pac data successfully from the host, and holds a valu as long as there is valid data in the FIFO buffer. This is a status bit, and cannot be cleared.
5	EP2 TR	0	R/W	EP2 Transfer Request This bit is set if there is no valid transmit data in buffer when an IN token for endpoint 2 is receive the host. A NACK handshake is returned to the data is written to the FIFO buffer and packet transmission is enabled. (When the CPU is used to clear this flag by writi while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)

				(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.)
2	EP0o TS	0	R/W	EP0o Receive Complete This bit is set to 1 when endpoint 0 receives da the host successfully, stores the data in the FIF and returns an ACK handshake to the host. (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.)
1	EP0i TR	0	R/W	EP0i Transfer Request This bit is set if there is no valid transmit data i FIFO buffer when an IN token for endpoint 0 is from the host. A NACK handshake is returned host until data is written to the FIFO buffer and transmission is enabled. (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.)
0	EP0i TS	0	R/W	EP0i Transmit Complete This bit is set when data is transmitted to the h endpoint 0 and an ACK handshake is returned (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.)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write valu should always be 0.
5	—	0	R	
4	—	0	R	
3	VBUS MN	0	R	This is a status bit which monitors the state of t VBUS pin. This bit reflects the state of the VBUS pin and generates no interrupt request. This bit is alwa when the PULLUP_E bit in DMA is 0.
2	EP3 TR	0	R/W	EP3 Transfer Request This bit is set if there is no valid transmit data i FIFO buffer when an IN token for endpoint 3 is received from the host. A NACK handshake is to the host until data is written to the FIFO buff packet transmission is enabled. (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.)
1	EP3 TS	0	R/W	EP3 Transmit Complete This bit is set when data is transmitted to the h endpoint 3 and an ACK handshake is returned (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.)

19.3.3 Interrupt Flag Register 2 (IFR2)

IFR2, together with interrupt flag registers 0 and 1 (IFR0 and IFR1), indicates interrupt information required by the application. When an interrupt source is generated, the corre bit is set to 1. And then this bit, in combination with interrupt enable register 2 (IER2), g an interrupt request to the CPU. To clear, write 0 to the bit to be cleared and 1 to the oth

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va should always be 0.
5	SURSS	0	R	Suspend/Resume Status This is a status bit that describes bus state. 0: Normal state 1: Suspended state This bit is a status bit and generates no interr request.

				information register to the EPIR register ends (end). This module starts the USB operation aft endpoint information is completely set. (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	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
1	SETC	0	R/W	Set_Configuration Command Detection When the Set_Configuration command is detec bit is set to 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.)
0	SETI	0	R/W	Set_Interface Command Detection When the Set_Interface command is detected, is set to 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.)

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	BRST	0	R/W	Bus Reset
6	EP1 FULL	0	R/W	EP1 FIFO Full
5	EP2 TR	0	R/W	EP2 Transfer Request
4	EP2 EMPTY	0	R/W	EP2 FIFO Empty
3	SETUP TS	0	R/W	Setup Command Receive Complete
2	EP0o TS	0	R/W	EP0o Receive Complete
1	EP0i TR	0	R/W	EP0i Transfer Request
0	EP0i TS	0	R/W	EP0i Transmission Complete

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write valu
5	—	0	R	should always be 0.
4	—	0	R	
3	—	0	R	
2	EP3 TR	1	R/W	EP3 Transfer Request
1	EP3 TS	1	R/W	EP3 Transmission Complete
0	VBUSF	1	R/W	USB Bus Connect

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va
5	—	0	R	should always be 0.
4	SURSE	1	R/W	Suspend/Resume Detection
3	CFDN	1	R/W	End Point Information Load End
2	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.
1	SETCE	1	R/W	Set_Configuration Command Detection
0	SETIE	1	R/W	Set_Interface Command Detection

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	BRST	0	R/W	Bus Reset
6	EP1 FULL	0	R/W	EP1 FIFO Full
5	EP2 TR	0	R/W	EP2 Transfer Request
4	EP2 EMPTY	0	R/W	EP2 FIFO Empty
3	SETUP TS	0	R/W	Setup Command Receive Complete
2	EP0o TS	0	R/W	EP0o Receive Complete
1	EP0i TR	0	R/W	EP0i Transfer Request
0	EP0i TS	0	R/W	EP0i Transmission Complete

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va should always be 0.
5	—	0	R	
4	—	0	R	
3	—	0	R	
2	EP3 TR	0	R/W	EP3 Transfer Request
1	EP3 TS	0	R/W	EP3 Transmission Complete
0	VBUSF	0	R/W	USB Bus Connect

19.3.9 Interrupt Enable Register 2 (IER2)

IER2 enables the interrupt requests of interrupt flag register 2 (IFR2). When an interrupt to 1 while the corresponding bit of each interrupt is set to 1, an interrupt request is sent t CPU. The interrupt vector number is determined by the contents of interrupt select regis (ISR2).

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

				Suspend and Resume Operations.
3	CFDN	0	R/W	End Point Information Load End
2	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
1	SETCE	0	R/W	Set_Configuration Command Detection
0	SETIE	0	R/W	Set_Interface Command Detection

19.3.10 EP0i Data Register (EPDR0i)

EPDR0i is an 8-byte transmit FIFO buffer for endpoint 0. EPDR0i holds one packet of tra data for control-in. Transmit data is fixed by writing one packet of data and setting EP0iP the trigger register. When an ACK handshake is returned from the host after the data has transmitted, EP0iTS in interrupt flag register 0 is set. This FIFO buffer can be initialized of EP0iCLR in the FCLR register.

Bit	7	6	5	4	3	2	1	
Bit Name	D7	D6	D5	D4	D3	D2	D1	
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Ur
R/W	W	W	W	W	W	W	W	

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for control-in transfer

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
7 to 0	D7 to D0	All 0	R	Data register for control-out transfer

19.3.12 EP0s Data Register (EPDR0s)

EPDR0s is an 8-byte FIFO buffer specifically for receiving endpoint 0 setup commands setup command to be processed by the application is received. When command data is r successfully, the SETUPTS bit in interrupt flag register 0 is set.

As a latest setup command must be received in high priority, if data is left in this buffer, overwritten with new data. If reception of the next command is started while the current is being read, command reception has priority, the read by the application is forcibly sto the read data is invalid.

Bit	7	6	5	4	3	2	1
Bit Name	D7	D6	D5	D4	D3	D2	D1
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
7 to 0	D7 to D0	All 0	R	Data register for storing the setup command a control-out transfer

Bit Name	D7	D6	D5	D4	D3	D2	D1
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
7 to 0	D7 to D0	All 0	R	Data register for endpoint 1 transfer

19.3.14 EP2 Data Register (EPDR2)

EPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. EPDR2 has a dual-buffer conf and has a capacity of twice the maximum packet size. When transmit data is written to th buffer and EP2PKTE in the trigger register is set, one packet of transmit data is fixed, anc dual-FIFO buffer is switched over. The transmit data for this FIFO buffer can be transferr DMA. This FIFO buffer can be initialized by means of EP2CLR in the FCLR register.

Bit	7	6	5	4	3	2	1	
Bit Name	D7	D6	D5	D4	D3	D2	D1	
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Ur
R/W	W	W	W	W	W	W	W	

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 2 transfer

Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	U
R/W	W	W	W	W	W	W	W	

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 3 transfer

19.3.16 EP0o Receive Data Size Register (EPSZ0o)

EPSZ0o indicates the number of bytes received at endpoint 0 from the host.

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
7 to 0	—	All 0	R	Number of receive data for endpoint 0

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of received bytes for endpoint 1

19.3.18 Trigger Register (TRG)

TRG generates one-shot triggers to control the transfer sequence for each endpoint.

Bit	7	6	5	4	3	2	1	
Bit Name	—	EP3 PKTE	EP1 RDFN	EP2 PKTE	—	EP0s RDFN	EP0o RDFN	EP
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Ur
R/W	—	W	W	W	—	W	W	

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved The write value should always be 0.
6	EP3 PKTE	Undefined	W	EP3 Packet Enable After one packet of data has been written to tl endpoint 3 transmit FIFO buffer, the transmit fixed by writing 1 to this bit.

				fixed by writing 1 to this bit.
3	—	Undefined	—	Reserved The write value should always be 0.
2	EP0s RDFN	Undefined	W	EP0s Read Complete Write 1 to this bit after data for the EP0s con FIFO has been read. Writing 1 to this bit ena transfer of data in the following data stage. A handshake is returned in response to transfe requests from the host in the data stage unti written to this bit.
1	EP0o RDFN	Undefined	W	EP0o Read Complete Writing 1 to this bit after one packet of data h read from the endpoint 0 transmit FIFO buffe initializes the FIFO buffer, enabling the next be received.
0	EP0i PKTE	Undefined	W	EP0i Packet Enable After one packet of data has been written to endpoint 0 transmit FIFO buffer, the transmi fixed by writing 1 to this bit.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va should always be 0.
5	EP3 DE	0	R	EP3 Data Present This bit is set when the endpoint 3 FIFO buffe contains valid data.
4	EP2 DE	0	R	EP2 Data Present This bit is set when the endpoint 2 FIFO buffe contains valid data.
3	—	0	R	Reserved
2	—	0	R	These bits are always read as 0.
1	—	0	R	
0	EP0i DE	0	R	EP0i Data Present This bit is set when the endpoint 0 FIFO buffe contains valid data.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved The write value should always be 0.
6	EP3 CLR	Undefined	W	EP3 Clear Writing 1 to this bit initializes the endpoint 3 FIFO buffer.
5	EP1 CLR	Undefined	W	EP1 Clear Writing 1 to this bit initializes both sides of th endpoint 1 receive FIFO buffer.
4	EP2 CLR	Undefined	W	EP2 Clear Writing 1 to this bit initializes both sides of th endpoint 2 transmit FIFO buffer.
3	—	Undefined	—	Reserved
2	—		—	The write value should always be 0.
1	EP0o CLR	Undefined	W	EP0o Clear Writing 1 to this bit initializes the endpoint 0 FIFO buffer.
0	EP0i CLR	Undefined	W	EP0i Clear Writing 1 to this bit initializes the endpoint 0 FIFO buffer.

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va
5	—	0	R	should always be 0.
4	—	0	R	
3	—	0	R	
2	PULLUP_E	0	R/W	PULLUP Enable This pin performs the pull-up control for the D with using PM4 as the pull-up control pin. 0: D+ is not pulled up. 1: D+ is pulled up.

(USBINTN1) is asserted again. However, if t
the data packet to be transmitted is less than
bytes, the EP2 packet enable bit is not set
automatically, and so should be set by the C
DMA transfer end interrupt.

As EP2-related interrupt requests to the CPU
automatically masked, interrupt requests sho
masked as necessary in the interrupt enable

- Operating procedure

1. Write of 1 to the EP2 DMAE bit in DMAR
2. Set the DMAC to activate through USBIN
3. Transfer count setting in the DMAC
4. DMAC activation
5. DMA transfer
6. DMA transfer end interrupt generated

See section 19.8.3, DMA Transfer for Endpo

EP1-related interrupt requests to the CPU are automatically masked.

- Operating procedure:

1. Write of 1 to the EP1 DMAE bit in DMA
2. Set the DMAC to activate through USBIN
3. Transfer count setting in the DMAC
4. DMAC activation
5. DMA transfer
6. DMA transfer end interrupt generated

See section 19.8.2, DMA Transfer for Endpoi

Bit Name	—	—	—	—	EP3STL	EP2STL	EP1STL
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write v should always be 0.
5	—	0	R	
4	—	0	R	
3	EP3STL	0	R/W	EP3 Stall When this bit is set to 1, endpoint 3 is placec stall state.
2	EP2STL	0	R/W	EP2 Stall When this bit is set to 1, endpoint 2 is placec stall state.
1	EP1STL	0	R/W	EP1 Stall When this bit is set to 1, endpoint 1 is placec stall state.
0	EP0STL	0	R/W	EP0 Stall When this bit is set to 1, endpoint 0 is placec stall state.

Bit	Bit Name	Initial Value	R/W	Description
7 6	CNFV1 CNFV0	All 0	R	These bits store Configuration Setting value w they receive Set Configuration command. CN updated when the SETC bit in IFR2 is set to 1
5 4	INTV1 INTV0	All 0	R	These bits store Interface Setting value when receive Set Interface command. INTV is upda when the SETI bit in IFR2 is set to 1.
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2	ALTV2	0	R	These bits store Alternate Setting value when receive Set Interface command. ALTV2 to AL updated when the SETI bit in IFR2 is set to 1.
1	ALTV1	0	R	
0	ALTV0	0	R	

19.3.24 Control Register (CTLR)

This register sets functions for bits ASCE, PWMD, RSME, and, PWUPS.

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

				Feature request. This bit is set to 1 when rer wakeup command is enabled.
3	RSME	0	R/W	Resume Enable This bit releases the suspend state (or execu remote wakeup). When RSME is set to 1, re request starts. If RSME is once set to 1, clea to 0 again afterwards. In this case, the value RSME must be kept for at least one clock pe 12-MHz clock.
2	PWMD	0	R/W	Bus Power Mode This bit specifies the USB power mode. Whe is set to 0, the self-power mode is selected f module. When set to 1, the bus-power mode selected.
1	ASCE	0	R/W	Automatic Stall Clear Enable Setting the ASCE bit to 1 automatically clear setting bit (the EPxSTL (x = 0, 1, 2, or 3) bit EPSTLR0 or EPSTR1) of the end point that returned the stall handshake to the host. The automatic stall clear enable is common to th points. Thus the individual control of the end not possible. When the ASCE bit is set to 0, the stall settin not automatically cleared. This bit must be re by the users. To enable this bit, make sure th ASCE bit should be set to 1 before the EPxS 1, 2, or 3) bit in EPSTL is set to 1.
0	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

Bit	7	6	5	4	3	2	1
Bit Name	D7	D6	D5	D4	D3	D2	D1
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	W	W	W	W	W	W	W

- EPIR00

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	D7 to D4	Undefined	W	Endpoint Number [Enable setting range] 0 to 3
3, 2	D3, D2	Undefined	W	Endpoint Configuration Number [Enable setting range] 0 or 1
1, 0	D1, D0	Undefined	W	Endpoint Interface Number [Enable setting range] 0 to 3

Bit	Bit Name	Initial Value	R/W	Description
				2: Bulk 3: Interrupt
3	D3	Undefined	W	Endpoint Transmission Direction [Possible setting range] 0: Out 1: In
2 to 0	D2 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR02

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	D7 to D1	Undefined	W	Endpoint Maximum Packet Size [Possible setting range] 0 to 64
0	D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR03

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

described below.

Since each endpoint FIFO number is optimized by the exclusive software that correspond transfer system, direction, and the maximum packet size, make sure to set the endpoint FI number to the data described in table 19.2.

1. The endpoint FIFO number 1 cannot designate other than the maximum packet size o bytes, bulk transfer method, and out transfer direction.
2. endpoint number 0 and the endpoint FIFO number must have one-on one relationshi
3. The maximum packet size for the endpoint FIFO number 0 is 8 bytes only.
4. The endpoint FIFO number 0 can specify only the maximum packet size and the data rest should be all 0.
5. The maximum packet size for the endpoint FIFO numbers 1 and 2 is limited to 64 byt
6. The maximum packet size for the endpoint FIFO numbers 3 is limited to 8 bytes.
7. The maximum number of endpoint information setting is ten.
8. Up to ten endpoint information setting should be made.
9. Write 0 to the endpoints not in use.

Table 19.2 shows the example of limitations for the maximum packet size, the transfer m and the transfer direction.

Table 19.2 Example of Limitations for Setting Values

Endpoint FIFO Number	Maximum Packet Size	Transfer Method	Transfer Dire
0	8 bytes	Control	—
1	64 bytes	Bulk	Out
2	64 bytes	Bulk	In
3	8 bytes	Interrupt	In

—	1	1	1	—	—	—	—

N	EPIR[N]0	EPIR[N]1	EPIR[N]2	EPIR[N]3	EPIR[
0	00	00	10	00	00
1	14	20	80	00	01
2	24	28	80	00	02
3	34	38	10	00	03
4	00	00	00	00	00
5	00	00	00	00	00
6	00	00	00	00	00
7	00	00	00	00	00
8	00	00	00	00	00
9	00	00	00	00	00

Configuration	Interface	Alternate Setting	Endpoint Number	Endpoint FIFO Number	A
—	—	—	0	0	C
1	0	0	1	1	B
			2	2	B
			3	3	Ir

Bit	Bit Name	Initial Value	R/W	Description
7	PTSTE	0	R/W	Pin Test Enable Enables the test control for the on-chip transc output pins (USD+ and USD-).
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
3	SUSPEND	0	R/W	On-Chip Transceiver Output Signal Setting SUSPEND: Sets the (SUSPEND) signal of th transceiver. txenl: Sets the output enable (txenl) sig on-chip transceiver. txse0: Sets the Signal-ended 0 (txse0) s the on-chip transceiver. txdata: Sets the (txdata) signal of the on- transceiver.
2	txenl	0	R/W	
1	txse0	0	R/W	
0	txdata	0	R/W	

1	1	1	X	X	Hi-Z	Hi-Z

[Legend]

X: Don't care.

—: Cannot be controlled. Indicates state in normal operation according to the USB o and port settings.

R/W R R R R R R R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
2	xver_data	—*	R	On-Chip Transceiver Input Signal Monitor
1	dpls	—*	R	xver_data: Monitors the differential input level (xver_data) signal of the on-chip transceiver.
0	dmns	—*	R	dpls: Monitors the USD+ (dpls) signal of t chip transceiver. dmns: Monitors the USD- (dmns) signal of chip transceiver.

Note: * Determined by the state of pins, VBUS, USD+, and USD-

1	0	1	1	0	1	1	0	
1	0	1	1	1	X	1	1	
1	1	1	0	0	0	0	0	
1	1	1	0	1	0	0	1	
1	1	1	1	0	0	1	0	
1	1	1	1	1	0	1	1	
1	X	0	X	X	0	1	1	Can be m when VBU

[Legend]

X: Don't care.

		transfer (EP0)		complete	USBINTN3		
	1		EP0i_TR*	EP0i transfer request	USBINTN2 or USBINTN3	×	×
	2		EP0o_TS*	EP0o receive complete	USBINTN2 or USBINTN3	×	×
	3		SETUP_TS*	Setup command receive complete	USBINTN2 or USBINTN3	×	×
	4	Bulk_in transfer (EP2)	EP2_EMPTY	EP2 FIFO empty	USBINTN2 or USBINTN3	×	US
	5		EP2_TR	EP2 transfer request	USBINTN2 or USBINTN3	×	×
	6	Bulk_out transfer (EP1)	EP1_FULL	EP1 FIFO Full	USBINTN2 or USBINTN3	×	US
	7	Status	BRST	Bus reset	USBINTN2 or USBINTN3	×	×
IFR1	0	Status	VBUSF	USB disconnection detection	USBINTN2 or USBINTN3	×	×
	1	Interrupt_in transfer (EP3)	EP3_TS	EP3 transfer complete	USBINTN2 or USBINTN3	×	×
	2		EP3_TR	EP3 transfer request	USBINTN2 or USBINTN3	×	×
	3	Status	VBUSMN	VBUS connection status	—	×	×
	4	—	Reserved	—	—	—	—
	5						
	6						
	7						

					USBINTN3, or RESUME		
	5		SURSS	Suspend/resume status	—	×	×
	6	—	Reserved	—	—	—	—
	7						

Note: * EP0 interrupts must be assigned to the same interrupt request signal.

- USBINTN0 signal

  DMAC start interrupt signal only EP1. See section 19.8, DMA Transfer.
- USBINTN1 signal

  DMAC start interrupt signal only EP2. See section 19.8, DMA Transfer.
- USBINTN2 signal

  The USBINTN2 signal requests interrupt sources for which the corresponding bits in select registers 0 to 2 (ISR0 to ISR2) are cleared to 0. The USBINTN2 is driven low corresponding bit in the interrupt flag register is set to 1.
- USBINTN3 signal

  The USBINTN3 signal requests interrupt sources for which the corresponding bits in select registers 0 to 2 (ISR0 to ISR2) are cleared to 0. The USBINTN3 is driven low corresponding bit in the interrupt flag register is set to 1.
- RESUME signal

  The RESUME signal is a resume interrupt signal for canceling software standby mo deep software standby mode. The RESUME signal is driven low at the transition to t state for canceling software standby mode and deep software standby mode.

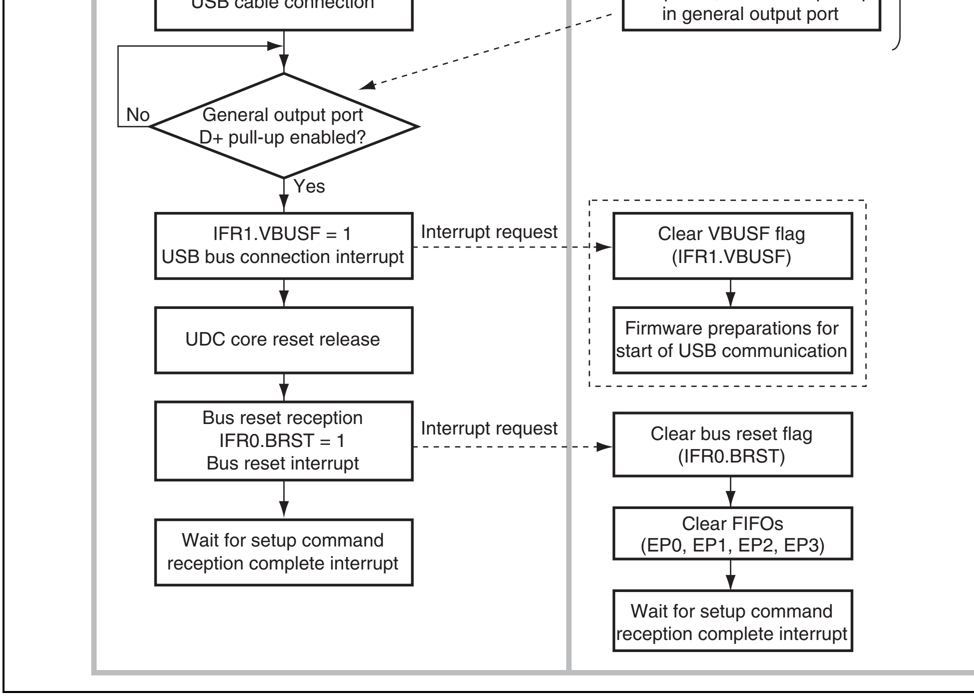


Figure 19.2 Cable Connection Operation

The above flowchart shows the operation in the case of in section 19.9, Example of USB
Circuitry.

In applications that do not require USB cable connection to be detected, processing by th
bus connection interrupt is not necessary. Preparations should be made with the bus-reset
interrupt.

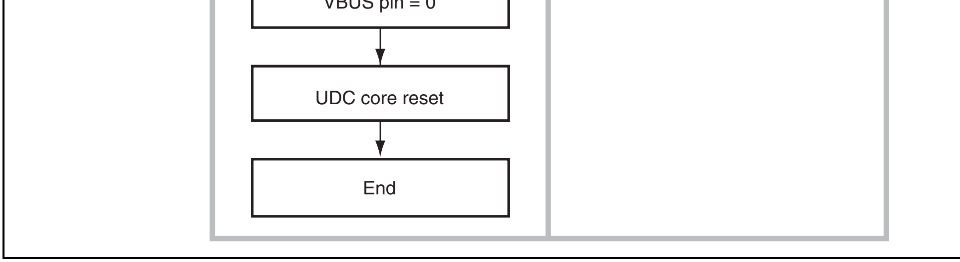


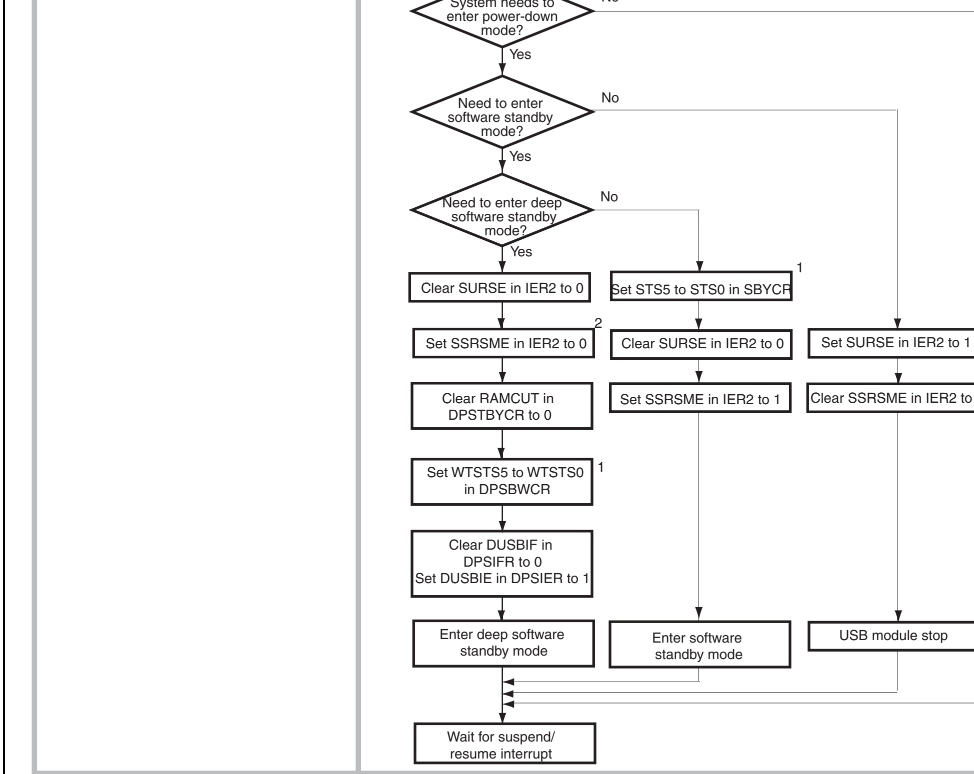
Figure 19.3 Cable Disconnection Operation

The above flowchart shows the operation in section 19.9, Example of USB External Cir

19.5.3 Suspend and Resume Operations

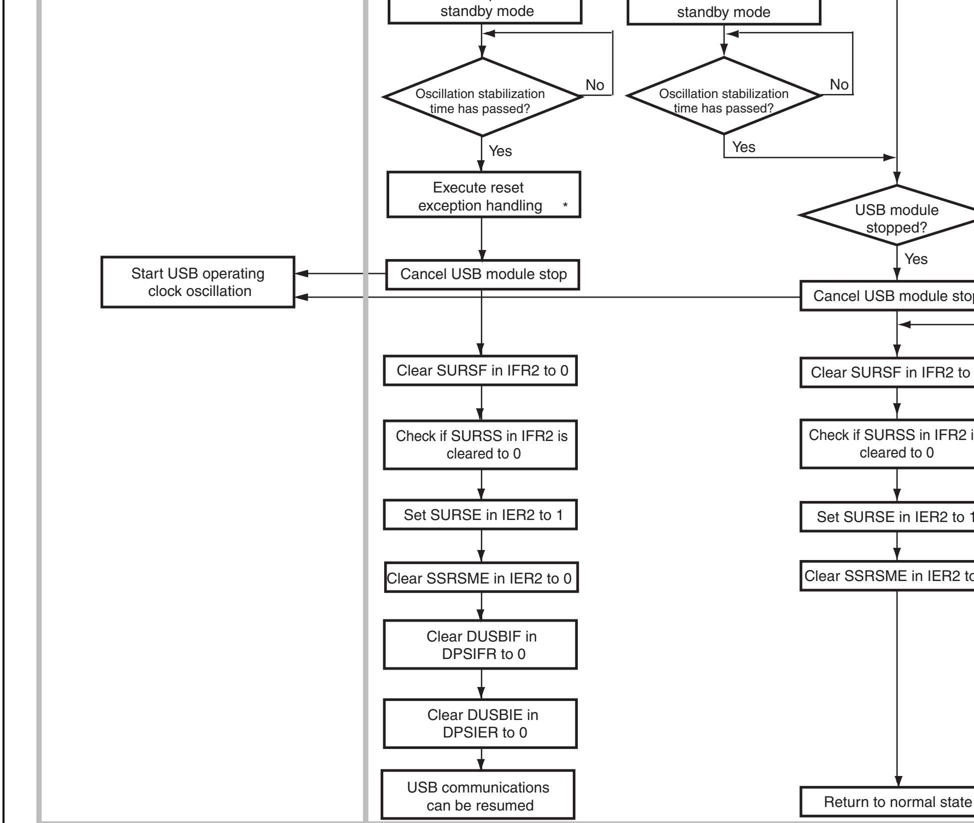
(1) Suspend Operation

If the USB bus enters the suspend state from the non-suspend state, perform the operatio
shown in figure 19.4.



Notes: 1. For details, see section 27, Power-Down Modes.
2. When the USB enters deep software standby mode, the sources to cancel software standby mode may be conflicted. In this figure, the operation to cancel software standby mode is not performed. For details, see section 27.12, Usage Notes.

Figure 19.4 Suspend Operation



Note: * For details, see section 27.8, Deep Software Standby Mode.

Figure 19.5 Resume Operation from Up-Stream

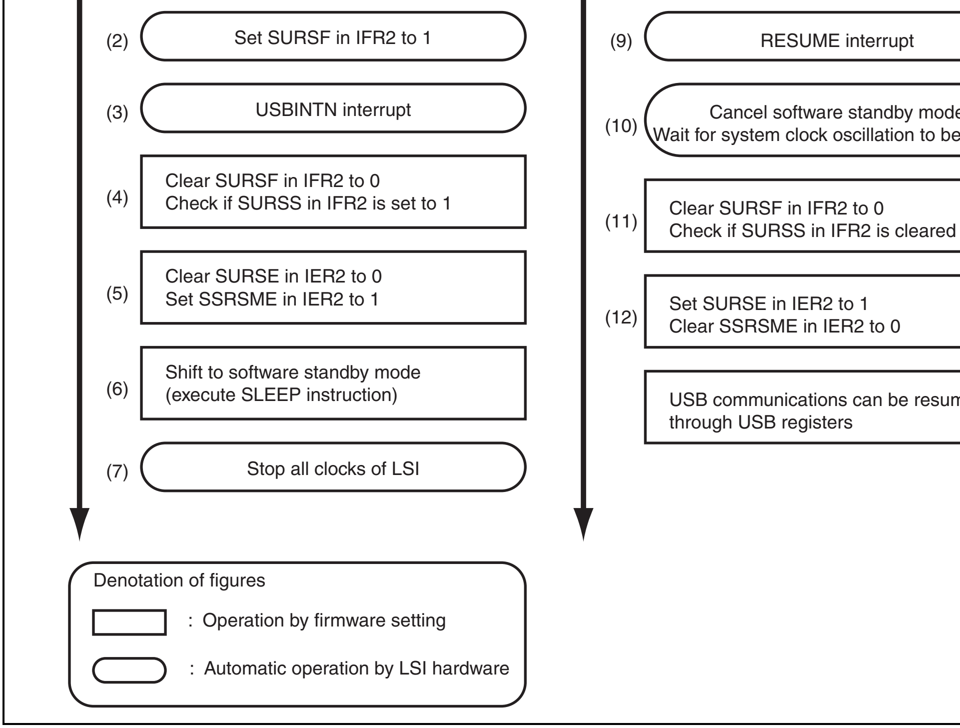


Figure 19.6 Flow of Transition to and Canceling Software Standby Mod

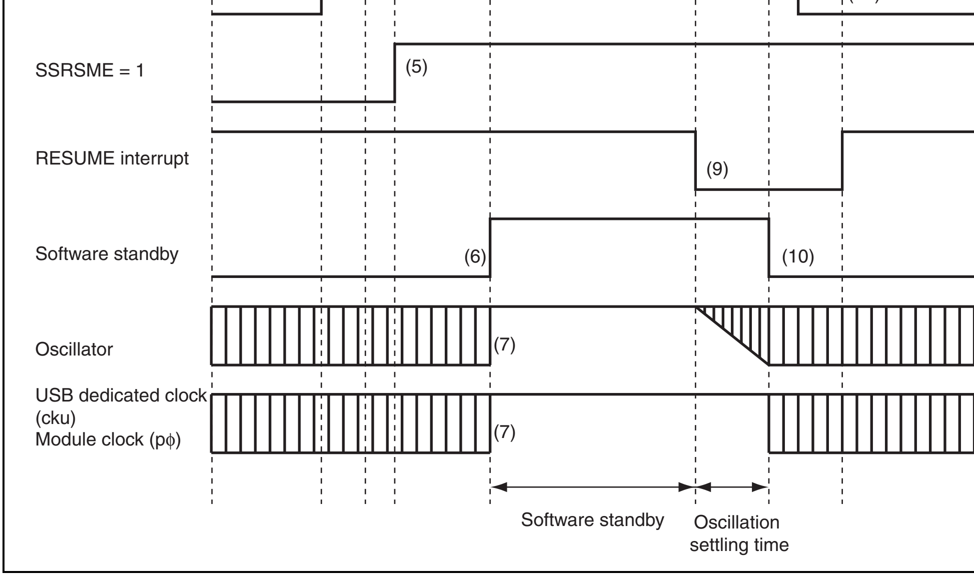
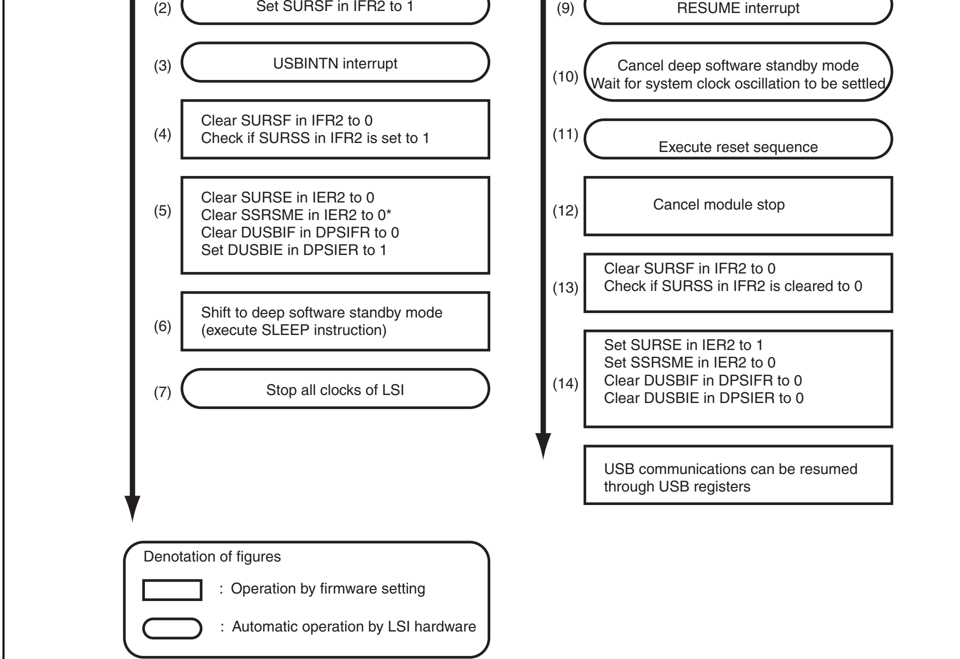


Figure 19.7 Timing of Transition to and Canceling Software Standby Mod



Note: * When the USB enters deep software standby mode, the sources to cancel software standby mode may be conflicted. In this figure, the operation to cancel software standby mode is not performed. For details, see section 27.12, Usage Notes.

Figure 19.8 Flow of Transition to and Canceling Deep Software Standby M

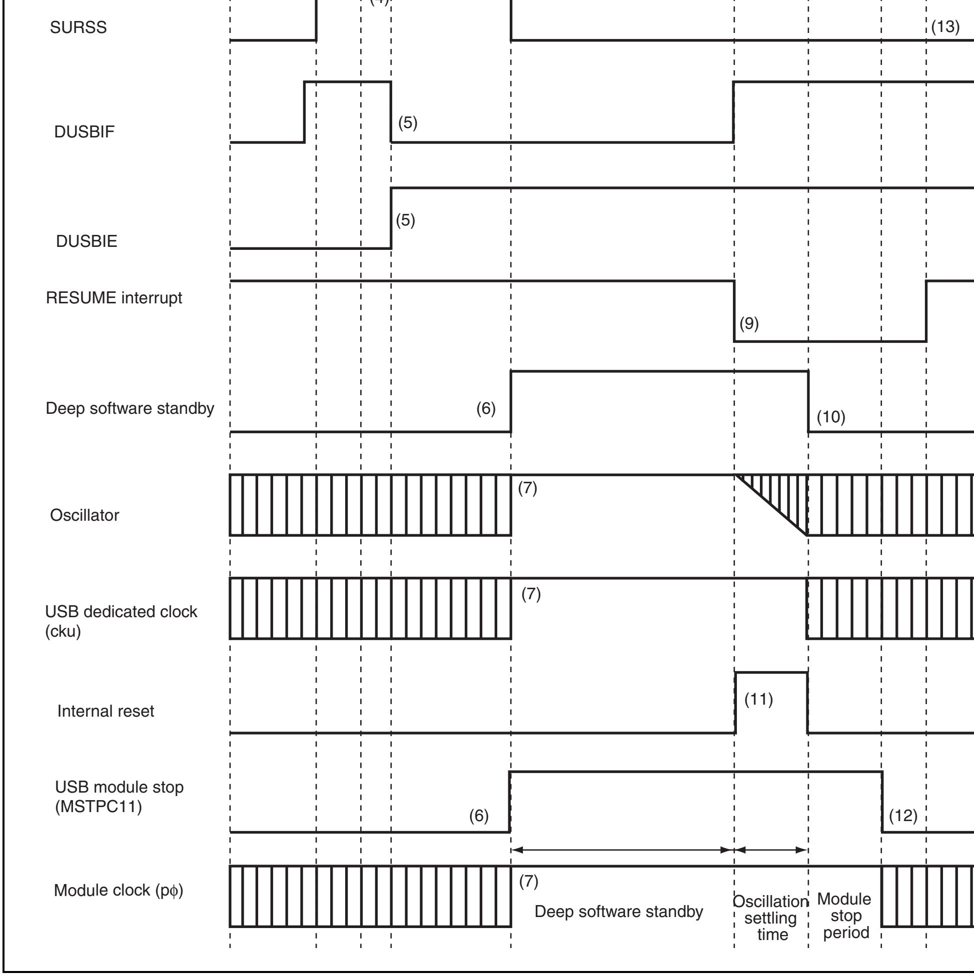


Figure 19.9 Timing of Transition to and Canceling Deep Software Standby M

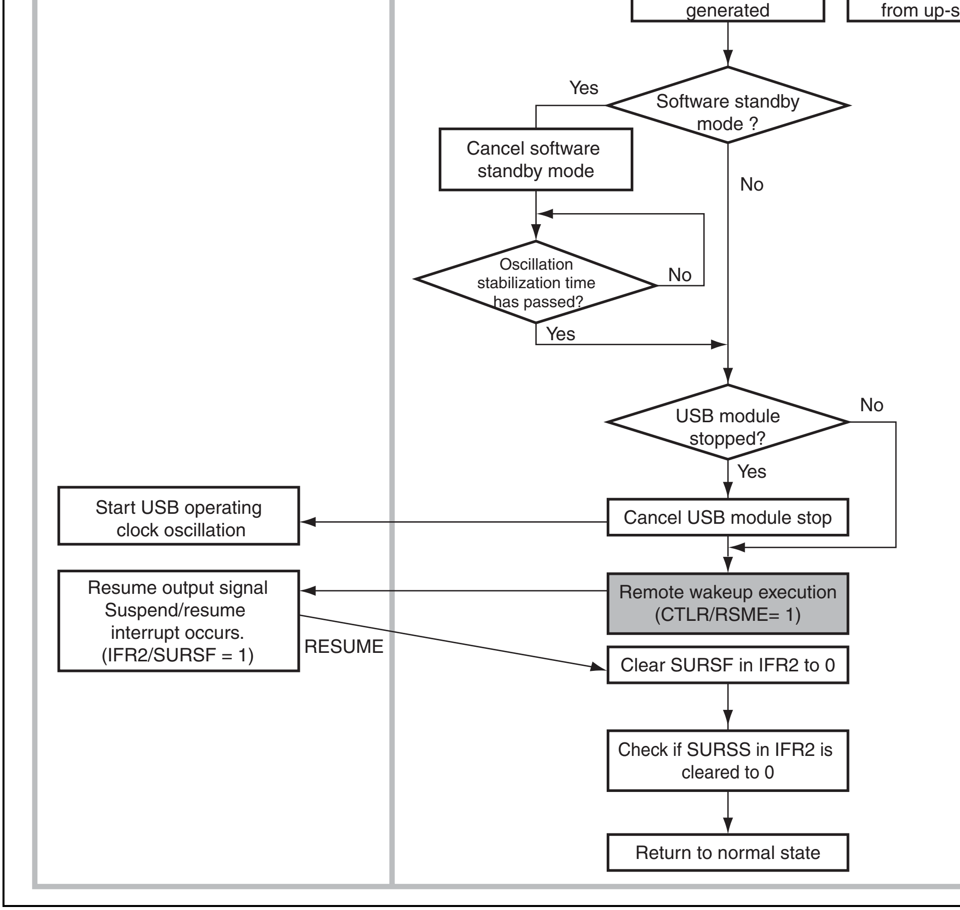


Figure 19.10 Remote-Wakeup

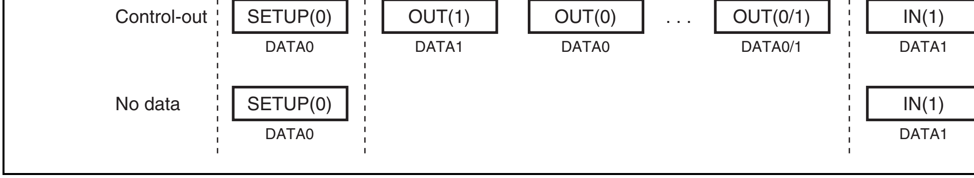
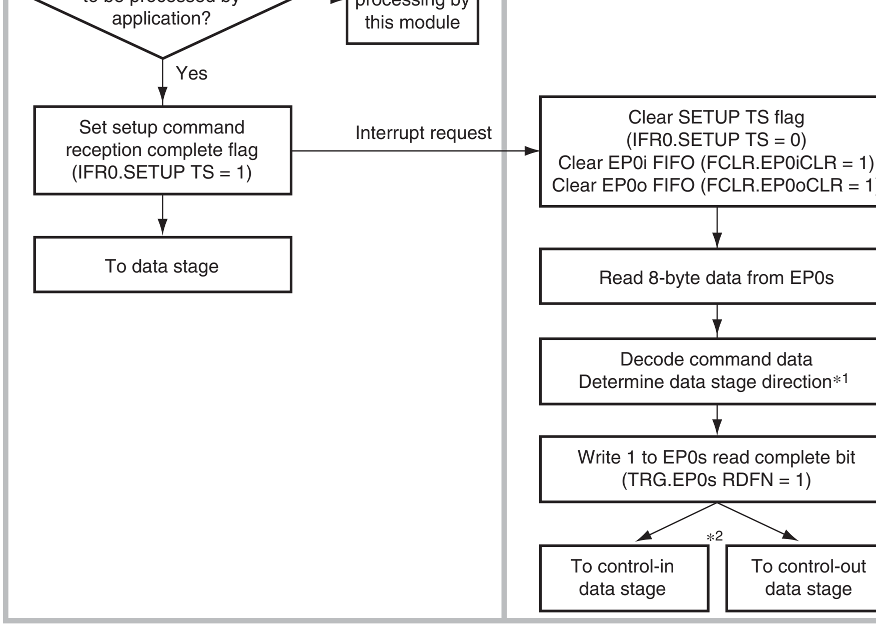


Figure 19.11 Transfer Stages in Control Transfer



Notes: 1. In the setup stage, the application analyzes command data from the host requiring processing the application, and determines the subsequent processing (for example, data stage direction,

2. When the transfer direction is control-out, the EP0i transfer request interrupt required in the sta stage should be enabled here. When the transfer direction is control-in, this interrupt is not requ and should be disabled.

Figure 19.12 Setup Stage Operation

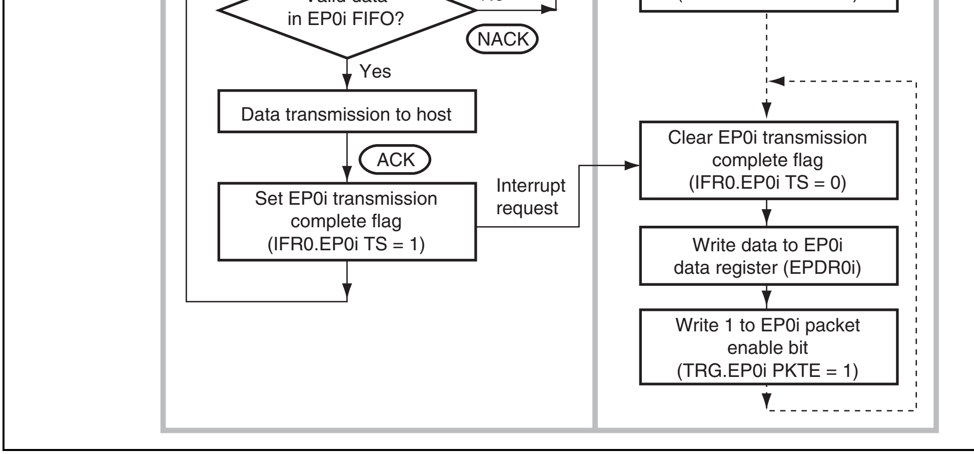


Figure 19.13 Data Stage (Control-In) Operation

The application first analyzes command data from the host in the setup stage, and determ subsequent data stage direction. If the result of command data analysis is that the data sta transfer, one packet of data to be sent to the host is written to the FIFO. If there is more d sent, this data is written to the FIFO after the data written first has been sent to the host (E bit in IFR0 = 1).

The end of the data stage is identified when the host transmits an OUT token and the statu is entered.

Note: If the size of the data transmitted by the function is smaller than the data size requ the host, the function indicates the end of the data stage by returning to the host a shorter than the maximum packet size. If the size of the data transmitted by the fu an integral multiple of the maximum packet size, the function indicates the end o stage by transmitting a zero-length packet.

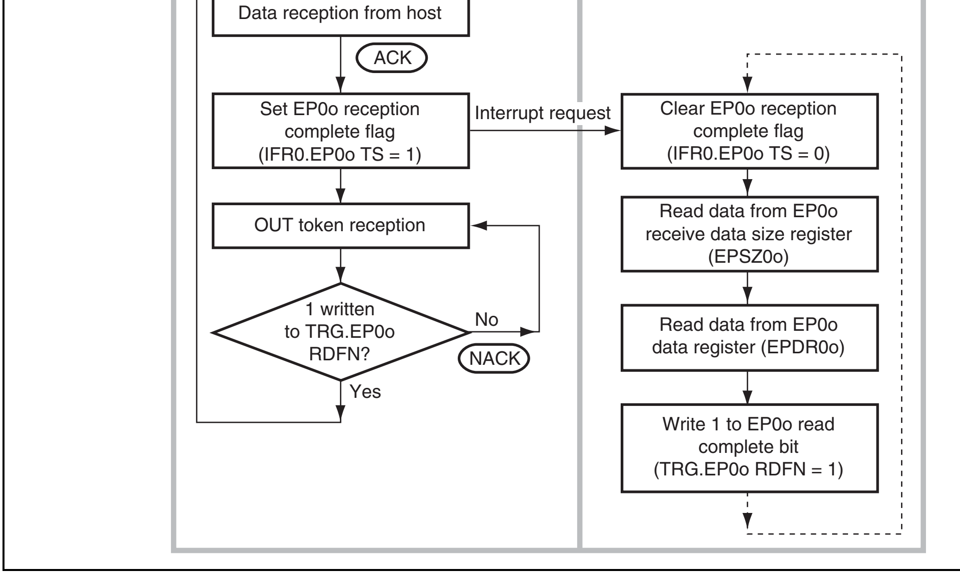


Figure 19.14 Data Stage (Control-Out) Operation

The application first analyzes command data from the host in the setup stage, and deterr subsequent data stage direction. If the result of command data analysis is that the data st transfer, the application waits for data from the host, and after data is received (EP0oTS IFR0 = 1), reads data from the FIFO. Next, the application writes 1 to the EP0o read cor empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the status entered.

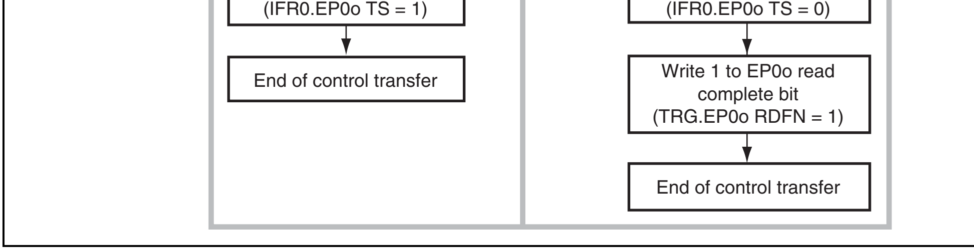


Figure 19.15 Status Stage (Control-In) Operation

The control-in status stage starts with an OUT token from the host. The application receiv
byte data from the host, and ends control transfer.

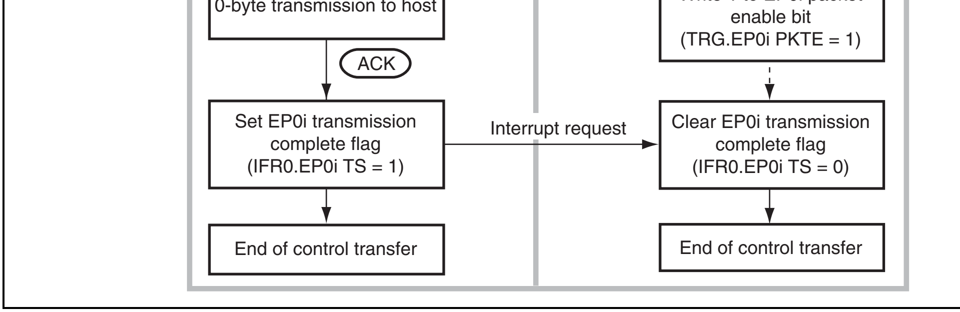


Figure 19.16 Status Stage (Control-Out) Operation

The control-out status stage starts with an IN token from the host. When an IN-token is r the start of the status stage, there is not yet any data in the EP0i FIFO, and so an EP0i tr request interrupt is generated. The application recognizes from this interrupt that the stat has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i pac bit but no data is written to the EP0i FIFO. As a result, the next IN token causes 0-byte c transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be w the EP0i packet enable bit.

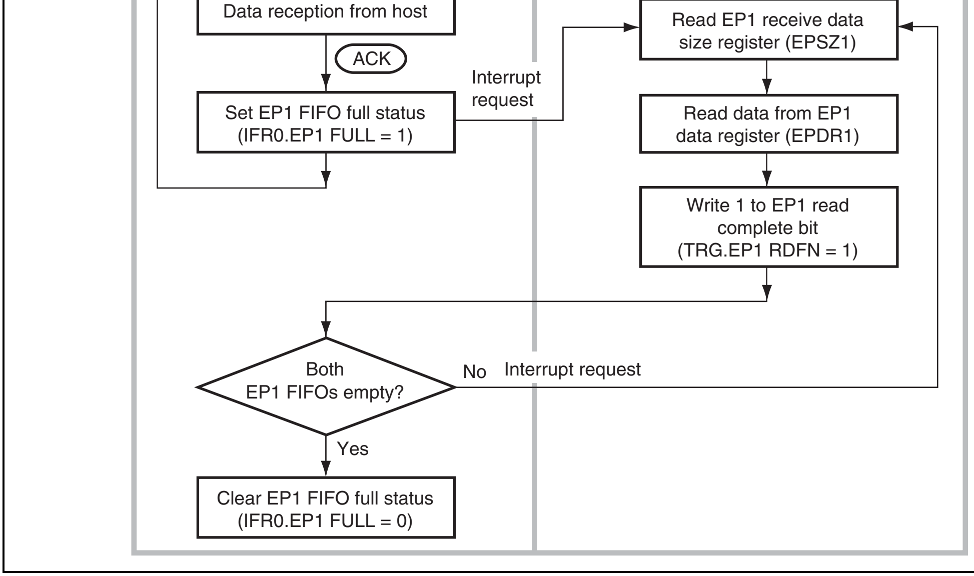


Figure 19.17 EP1 Bulk-Out Transfer Operation

EP1 has two 64-byte FIFOs, but the user can receive data and read receive data without b aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the EP1FULL bit in IFR0 is set. Afte receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is en so the next packet can be received immediately. When both FIFOs are full, NACK is retu the host automatically. When reading of the receive data is completed following data rece is written to the EP1RDFN bit in TRG. This operation empties the FIFO that has just bee and makes it ready to receive the next packet.

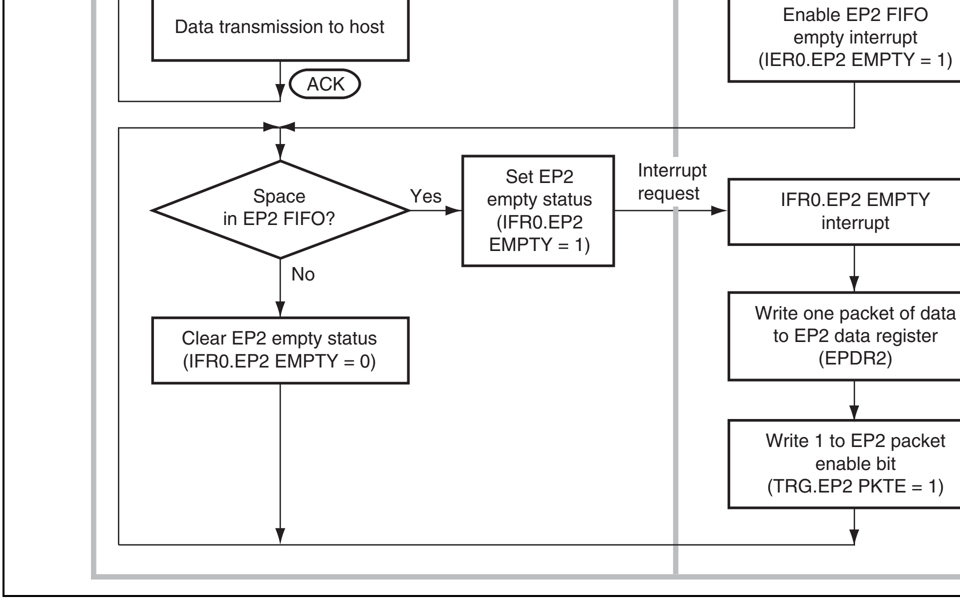
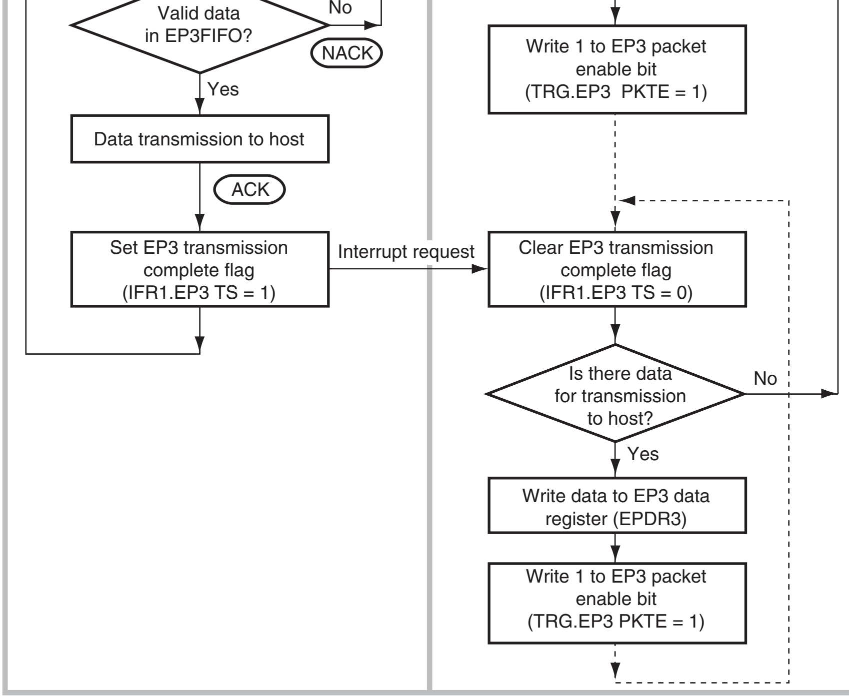


Figure 19.18 EP2 Bulk-In Transfer Operation

EP2 has two 64-byte FIFOs, but the user can transmit data and write transmit data witho aware of this dual-FIFO configuration. However, one data write is performed for one FI example, even if both FIFOs are empty, it is not possible to perform EP2PKTE at one ti consecutively writing 128 bytes of data. EP2PKTE must be performed for each 64-byte

When performing bulk-in transfer, as there is no valid data in the FIFOs on reception of IN token, an EP2TR bit interrupt in IFR0 is requested. With this interrupt, 1 is written to EP2EMPTY bit in IER0, and the EP2 FIFO empty interrupt is enabled. At first, both EP are empty, and so an EP2 FIFO empty interrupt is generated immediately.



Note: This flowchart shows just one example of interrupt transfer processing. Other possibilities include operation flow in which, if there is data to be transferred, the EP3 DE bit in the data status register referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 19.19 Operation of EP3 Interrupt-In Transfer

Decoding not Necessary on Application Side	Decoding Necessary on Applicatio
Clear Feature	Get Descriptor
Get Configuration	Class/Vendor command
Get Interface	Set Descriptor
Get Status	Sync Frame
Set Address	
Set Configuration	
Set Feature	
Set Interface	

If decoding is not necessary on the application side, command decoding and data stage ar stage processing are performed automatically. No processing is necessary by the user. An is not generated in this case.

If decoding is necessary on the application side, this module stores the command in the E FIFO. After reception is completed successfully, the IFR0/SETUP TS flag is set and an in request is generated. In the interrupt routine, eight bytes of data must be read from the EP register (EPDR0s) and decoded by firmware. The necessary data stage and status stage pr should then be carried out according to the result of the decoding operation.

The USB function module has internal status bits that hold the status (stall or non-stall) of each endpoint. When a transaction is sent from the host, the module references these internal status bits and determines whether to return a stall to the host. These bits cannot be cleared by the application; they must be cleared with a Clear Feature command from the host.

However, the internal status bit for EP0 is automatically cleared only when the setup command is received.

19.7.2 Forcible Stall by Application

The application uses the EPSTL register to issue a stall request for the USB function module. When the application wishes to stall a specific endpoint, it sets the corresponding bit in EPSTL (1-1 in figure 19.20). The internal status bits are not changed at this time. When a transaction is sent from the host for the endpoint for which the EPSTL bit was set, the USB function module references the internal status bit, and if this is not set, references the corresponding bit in EPSTL (1-2 in figure 19.20). If the corresponding bit in EPSTL is set, the USB function module sets the internal status bit and returns a stall handshake to the host (1-3 in figure 19.20). If the corresponding bit in EPSTL is not set, the internal status bit is not changed and the transaction is accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the EPSTL register. Even after a bit is cleared by the Clear Feature command (3-1 in figure 19.20), the USB function module continues to return a stall handshake while the bit in EPSTL is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 19.20). To clear a stall, therefore, it is necessary for the corresponding bit in EPSTL to be cleared by the application, and also for the internal status bit to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 19.20).

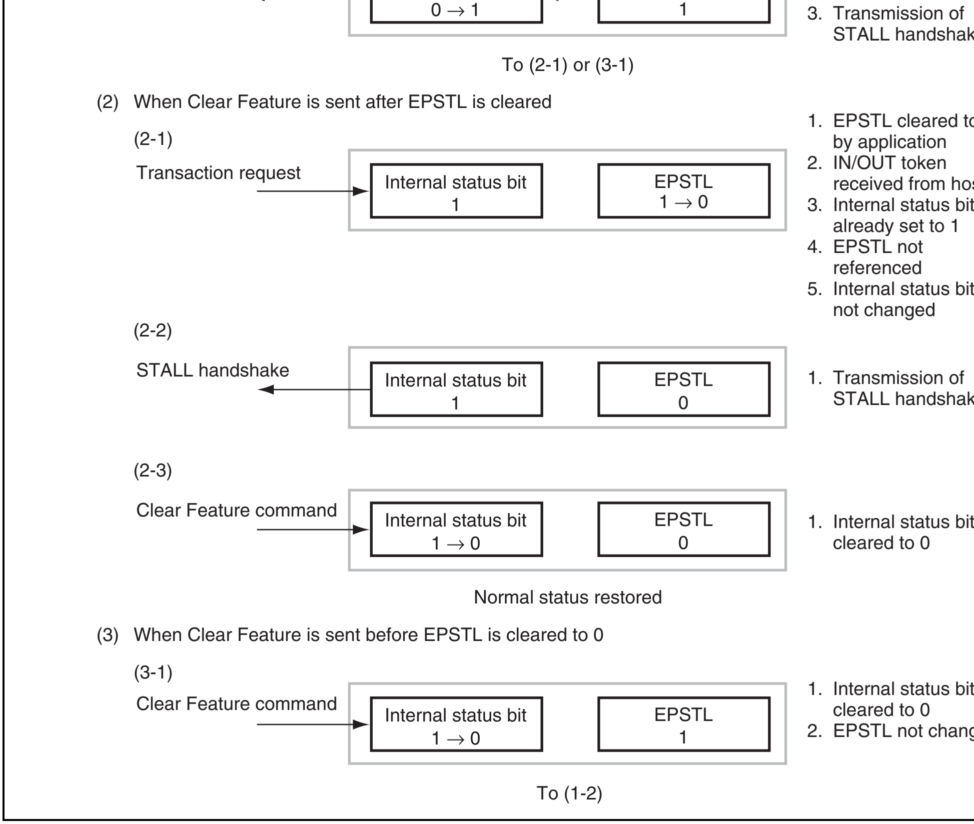


Figure 19.20 Forcible Stall by Application

the internal status bit must be cleared with a Clear Feature command (3-1 in figure 19.2
by the application, EPSTL should also be cleared (2-1 in figure 19.21).

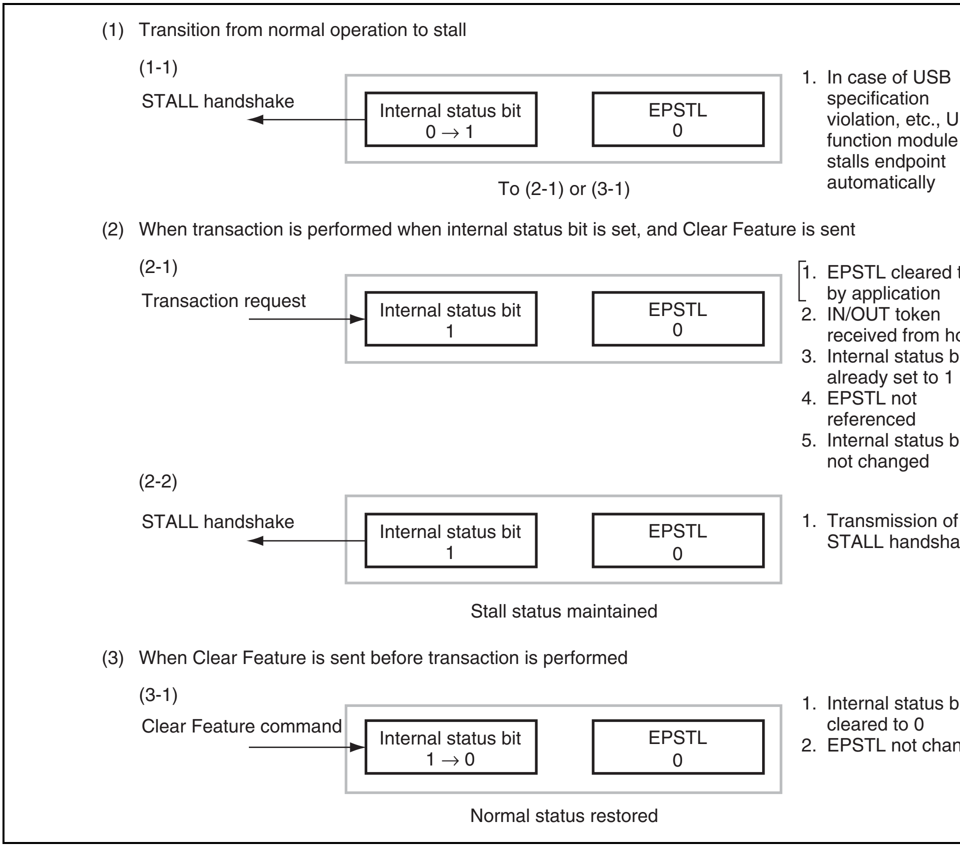


Figure 19.21 Automatic Stall by USB Function Module

If the DMA transfer is enabled by setting the EP1DMAE bit in the DMA transfer setting
to 1, zero-length data reception at endpoint 1 is ignored. When the DMA transfer is enabl
RDFN bit for EP1 and PKTE bit for EP2 do not need to be set to 1 in TRG (note that the
must be set to 1 when the transfer data is less than the maximum number of bytes). When
data received at EP1 is read, the FIFO automatically enters the EMPTY state. When the m
number of bytes (64 bytes) are written to the EP2 FIFO, the FIFO automatically enters th
state, and the data in the FIFO can be transmitted (see figures 19.22 and 19.23).

19.8.2 DMA Transfer for Endpoint 1

When the data received at EP1 is transferred by the DMA, the USB function module auto
performs the same processing as writing 1 to the RDFN bit in TRG if the currently select
becomes empty. Accordingly, in DMA transfer, do not write 1 to the RDFN bit for EP1 i
the user writes 1 to the RDFN bit in DMA transfer, correct operation cannot be guarantee

Figure 19.22 shows an example of receiving 150 bytes of data from the host. In this case,
processing which is the same as writing 1 to the RDFN bit in TRG is automatically perfo
three times. This internal processing is performed when the currently selected data FIFO
empty. Accordingly, this processing is automatically performed both when 64-byte data i
and when data less than 64 bytes is sent.

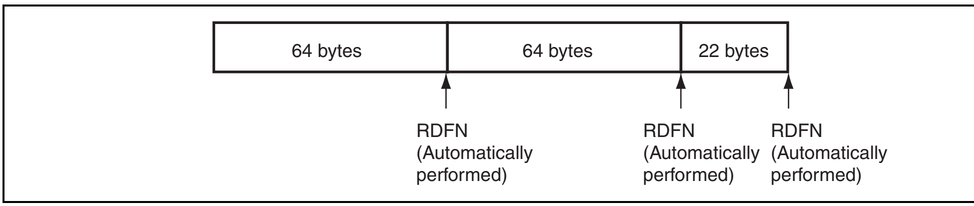


Figure 19.22 RDFN Bit Operation for EP1

processing which is the same as writing 1 to the PKTE bit in TRG is automatically perfo
twice. This internal processing is performed when the currently selected data FIFO beco
Accordingly, this processing is automatically performed only when 64-byte data is sent.

When the last 22 bytes are sent, the internal processing for writing 1 to the PKTE bit in
performed, and the user must write 1 to the PKTE bit by software. In this case, the appli
no more data to transfer but the USB function module continues to output DMA request
as long as the FIFO has an empty space. When all data has been transferred, write 0 to t
EP2DMAE bit in DMAR to cancel DMA requests for EP2.

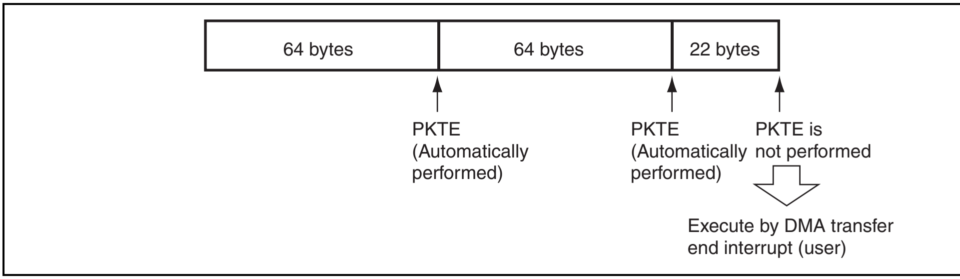
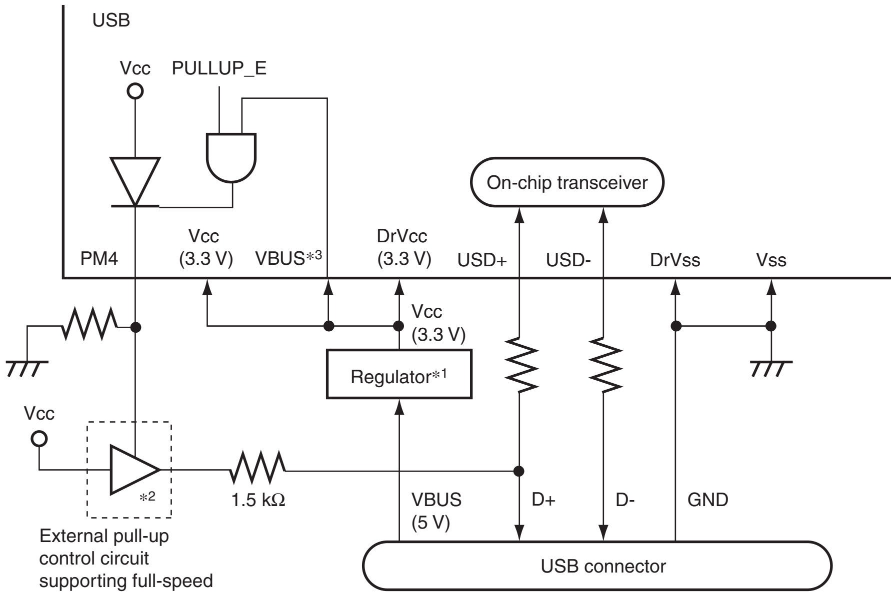


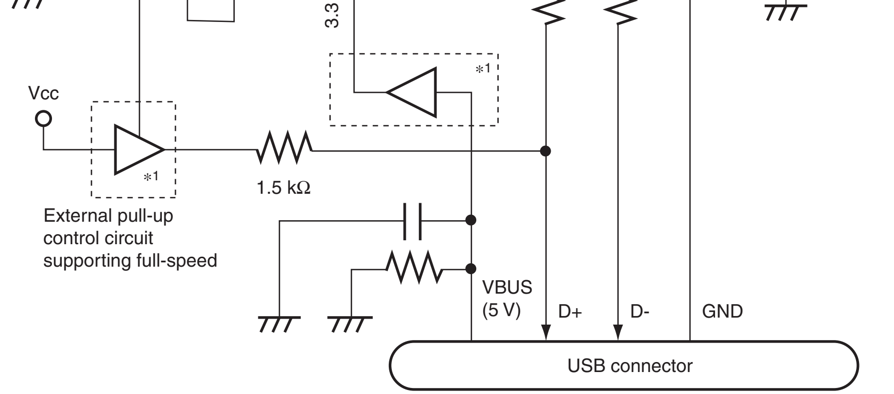
Figure 19.23 PKTE Bit Operation for EP2

As USB states, etc., are managed by hardware in this module, a VBUS signal that rec
connection/disconnection is necessary. The power supply signal (VBUS) in the USB c
used for this purpose. However, if the cable is connected to the USB host/hub when th
function (system installing this LSI) power is off, a voltage (5 V) will be applied from
host/hub. Therefore, an IC (such as an HD74LV1G08A or 2G08A) that allows voltag
application when the system power is off should be connected externally.



Notes: 1. Reduce voltage to the operating voltage (Vcc) of this LSI (3.3 V).
2. To protect this LSI from being damaged, use the IC (such as HD74LV-A Series) which ca
be applied voltage even when the system power is turned off.
3. Prevent noise from the VBUS pin while the USB is performing communication.

Figure 19.24 Example of Circuitry in Bus Power Mode



Notes:
1. To protect this LSI from being damaged, use the IC (such as HD74LV-A Series) which ca be applied voltage even when the system power is turned off.
2. Prevent noise from the VBUS pin while the USB is performing communication.

Figure 19.25 Example of Circuitry in Self Power Mode

2. EPDR0s must always be read in 8-byte units. If the read is terminated at a midpoint, t received at the next setup cannot be read correctly.

19.10.2 Clearing the FIFO

If a USB cable is disconnected during data transfer, the data being received or transmitted remain in the FIFO. When disconnecting a USB cable, clear the FIFO.

While a FIFO is transferring data, it must not be cleared.

19.10.3 Overreading and Overwriting the Data Registers

Note the following when reading or writing to a data register of this module.

(1) Receive data registers

The receive data registers must not be read exceeding the valid amount of receive data, th number of bytes indicated by the receive data size register. Even for EPDR1 which has d FIFO buffers, the maximum data to be read at one time is 64 bytes. After the data is read current valid FIFO buffer, be sure to write 1 to EP1RDFN in TRG, which switches the va buffer, updates the receive data size to the new number of bytes, and enables the next dat received.

(2) Transmit data registers

The transmit data registers must not be written to exceeding the maximum packet size. Ev EPDR2 which has double FIFO buffers, write data within the maximum packet size at on After the data is written, write 1 to PKTE in TRG to switch the valid buffer and enable th data to be written. Data must not be continuously written to the two FIFO buffers.

19.10.6 Notes on TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for IN transf
EP2, or EP3.

The TR interrupt flag is set if the FIFO for the target EP has no data when the IN token r
from the USB host. However, at the timing shown in figure 19.26, multiple TR interrupt
successively. Take appropriate measures against malfunction in such a case.

Note: This module determines whether to return NAKC if the FIFO of the target EP ha
when receiving the IN token, but the TR interrupt flag is set after a NAKC hand
sent. If the next IN token is sent before PKTE of TRG is written to, the TR inter
set again.

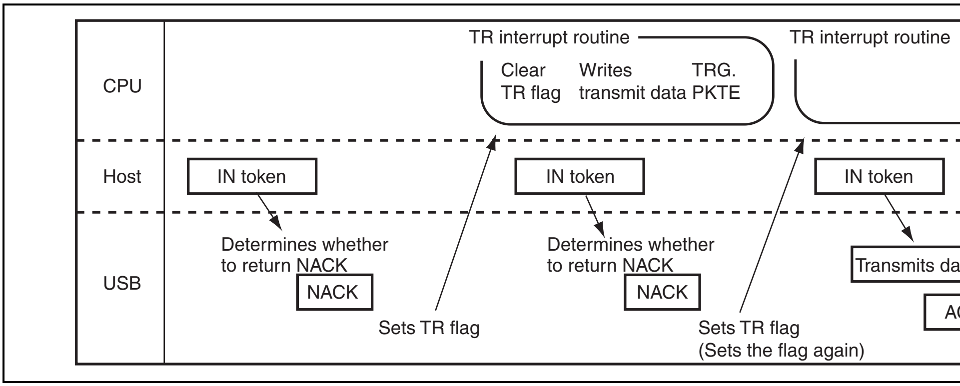


Figure 19.26 TR Interrupt Flag Set Timing

1	16 MHz	EXTAL × 3	EXTAL × 1 (16 M
			EXTAL × 2 (32 M

19.10.8 Notes on Deep Software Standby Mode when USB is Used

1. Unlike software standby mode, deep standby software mode is canceled from the rese
   For details, see section 27.8, Deep Software Standby Mode.
2. If the RAMCUT bit is set to 1 when the USB enters deep software standby mode, the
   states of the USB cannot be retained. When USB is used, set the RAMCUT bit to 1, a
   make the USB enter deep software standby mode.
3. Set the USB module stop (MSTPC11) bit to 0 after canceling deep software standby r
4. If the DUSBIE bit is set to 0 when the USB enters deep software standby mode, softw
   standby mode cannot be canceled through USB RESUME interrupt. Set the DUSBIE
   and then, make the USB enter deep software standby mode.

20.1 Features

- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independe each other, the continuous transmission/reception can be performed.
- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronize automatically. If transmission or reception is not yet possible, drive the SCL signal l preparations are completed
- Six interrupt sources

  Transmit-data-empty (including slave-address match), transmit-end, receive-data-ful (including slave-address match), arbitration lost, NACK detection, and stop conditio detection
- Direct bus drive

  Two pins, the SCL and SDA pins function as NMOS open-drain outputs.
- Module stop state can be set.

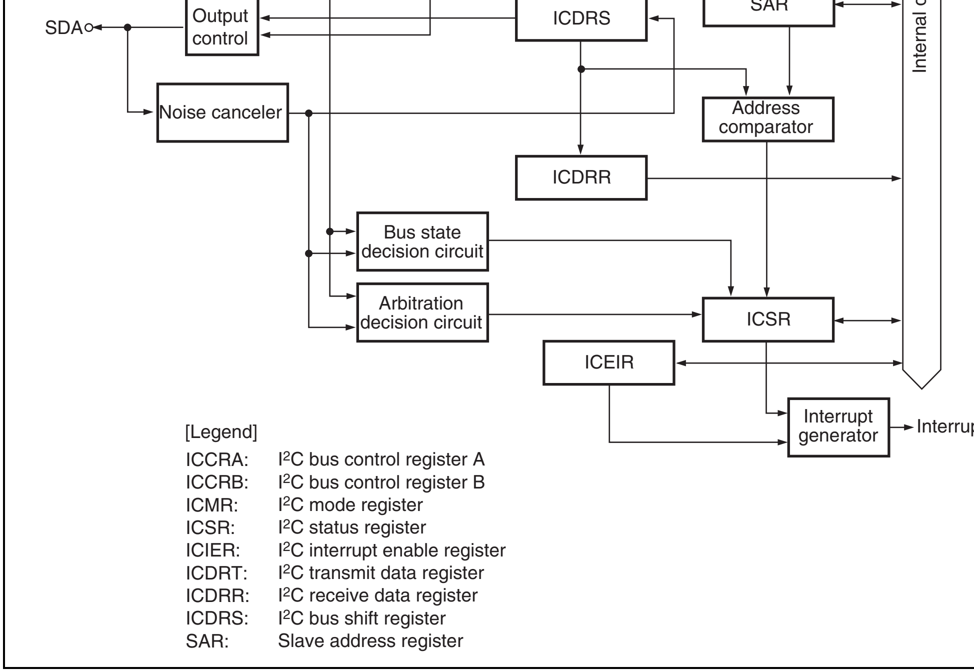


Figure 20.1 Block Diagram of I²C Bus Interface 2

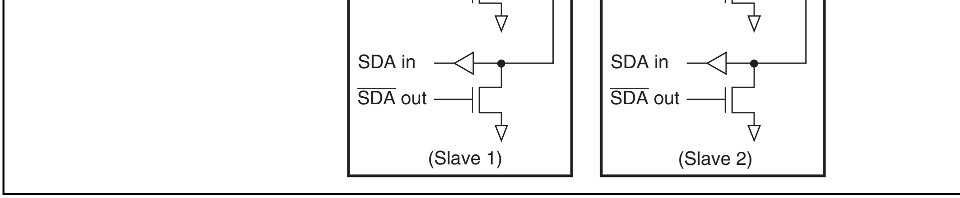


Figure 20.2 Connections to the External Circuit by the I/O Pins

20.2 Input/Output Pins

Table 20.1 shows the pin configuration of the I^2C bus interface 2.

Table 20.1 Pin Configuration of the I^2C Bus Interface 2

Channel	Abbreviation	I/O	Function
0	SCL0	I/O	Channel 0 serial clock I/O pin
	SDA0	I/O	Channel 0 serial data I/O pin
1	SCL1	I/O	Channel 1 serial clock I/O pin
	SDA1	I/O	Channel 1 serial data I/O pin

Note: The pin symbols are represented as SCL and SDA; channel numbers are omitted manual.

- I^{2}C bus status register_0 (ICSR_0)
- Slave address register_0 (SAR_0)
- I^{2}C bus transmit data register_0 (ICDRT_0)
- I^{2}C bus receive data register_0 (ICDRR_0)
- I^{2}C bus shift register_0 (ICDRS_0)

Channel 1:

- I^{2}C bus control register A_1 (ICCRA_1)
- I^{2}C bus control register B_1 (ICCRB_1)
- I^{2}C bus mode register_1 (ICMR_1)
- I^{2}C bus interrupt enable register_1 (ICIER_1)
- I^{2}C bus status register_1 (ICSR_1)
- Slave address register_1 (SAR_1)
- I^{2}C bus transmit data register_1 (ICDRT_1)
- I^{2}C bus receive data register_1 (ICDRR_1)
- I^{2}C bus shift register_1 (ICDRS_1)

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: This module is halted 1: This bit is enabled for transfer operations SDA pins are bus drive state)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operatio TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select When arbitration is lost in master mode, MS TRS are both reset by hardware, causing a t to slave receive mode. Modification of the TI should be made between transfer frames. Operating modes are described below accor MST and TRS combination. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3	CKS3	0	R/W	Transfer Clock Select 3 to 0
2	CKS2	0	R/W	These bits are valid only in master mode. Ma setting according to the required transfer rat details on the transfer rate, see table 20.2.
1	CKS1	0	R/W	
0	CKS0	0	R/W	

			1	Pφ/100	80.0 kHz	100 kHz	200 kHz	250 kHz	330 kHz
		1	0	Pφ/112	71.4 kHz	89.3 kHz	179 kHz	223 kHz	295 kHz
			1	Pφ/128	62.5 kHz	78.1 kHz	156 kHz	195 kHz	258 kHz
1	0	0	0	Pφ/56	143 kHz	179 kHz	357 kHz	446 kHz	589 kHz
			1	Pφ/80	100 kHz	125 kHz	250 kHz	313 kHz	413 kHz
		1	0	Pφ/96	83.3 kHz	104 kHz	208 kHz	260 kHz	344 kHz
			1	Pφ/128	62.5 kHz	78.1 kHz	156 kHz	195 kHz	258 kHz
	1	0	0	Pφ/336	23.8 kHz	29.8 kHz	59.5 kHz	74.4 kHz	98.2 kHz
			1	Pφ/200	40.0 kHz	50.0 kHz	100 kHz	125 kHz	165 kHz
		1	0	Pφ/224	35.7 kHz	44.6 kHz	89.3 kHz	112 kHz	147 kHz
			1	Pφ/256	31.3 kHz	39.1 kHz	78.1 kHz	97.7 kHz	129 kHz

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy This bit indicates whether the I^2C bus is occup released and to issue start and stop condition master mode. This bit is set to 1 when the SD changes from high to low under the condition high, assuming that the start condition has be issued. This bit is cleared to 0 when the SDA changes from low to high under the condition high, assuming that the stop condition has be issued. Follow this procedure also when re-tra a start condition. To issue a start or stop cond the MOV instruction.
6	SCP	1	R/W	Start/Stop Condition Issue This bit controls the issuance of start or stop c in master mode. To issue a start condition, write 1 to BBSY an SCP. A re-transmit start condition is issued in way. To issue a stop condition, write 0 to BBS to SCP. This bit is always read as 1. If 1 is wr data is not stored.
5	SDAO	1	R	This bit monitors the output level of SDA. 0: When reading, the SDA pin outputs a low le 1: When reading the SDA pin outputs a high l

1	IICRST	0	R/W	IIC Control Module Reset This bit reset the IIC control module except the registers. If hang-up occurs because of commu failure during I2C operation, by setting this bit t I2C control module can be reset without setting and initializing the registers.
0	—	1	—	Reserved This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The write value should always be 0.
6	WAIT	0	R/W	Wait Insertion This bit selects whether to insert a wait after d transfer except for the acknowledge bit. Wher set to 1, after the falling of the clock for the las the low period is extended for two transfer clo When this bit is cleared to 0, data and the ack bit are transferred consecutively with no waits The setting of this bit is invalid in slave mode.
5	—	1	—	Reserved
4	—	1	—	These bits are always read as 1.
3	BCWP	1	R/W	BC Write Protect This bit controls the modification of the BC2 to bits. When modifying, this bit should be cleare and the MOV instruction should be used. 0: When writing, the values of BC2 to BC0 are 1: When reading, 1 is always read When writing, the settings of BC2 to BC0 are

001: 2

010: 3

011: 4

100: 5

101: 6

110: 7

111: 8

20.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and the acknowledge bits, sets the acknowled
be transferred, and confirms the acknowledge bit to be received.

Bit	7	6	5	4	3	2	1
Bit Name	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

				This bit enables or disables the transmit end i (TEI) request at the rising of the ninth clock w TDRE bit in ICSR is set to 1. The TEI request canceled by clearing the TEND bit or the TEIE 0: Transmit end interrupt (TEI) request is disa 1: Transmit end interrupt (TEI) request is ena
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive full int (RXI) request when receive data is transferred ICDRS to ICDRR and the RDRF bit in ICSR is The RXI request can be canceled by clearing RDRF or RIE bit to 0. 0: Receive data full interrupt (RXI) request is 1: Receive data full interrupt (RXI) request is
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK receive (NAKI) request when the NACKF and AL bits are set to 1. The NAKI request can be cancel clearing the NACKF or AL bit, or the NAKIE b 0: NACK receive interrupt (NAKI) request is d 1: NACK receive interrupt (NAKI) request is e

				1: If the acknowledge bit is 1, continuous trans suspended
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowled that are returned by the receive device. This bi be modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be the acknowledge timing. 0: 0 is sent at the acknowledge timing 1: 1 is sent at the acknowledge timing

Bit	Bit Name	Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Setting condition] • When data is transferred from ICDRT to I and ICDRT becomes empty • When the TRS bits are set • When the start (re-transmit included) cond been issued • When switched from reception to transmis slave mode [Clearing conditions] • When 0 is written to this bit after reading T (When the CPU is used to clear this flag b 0 while the corresponding interrupt is enal sure to read the flag after writing 0 to it.) • When data is written to ICDRT
6	TEND	0	R/W	Transmit End [Setting condition] • When the ninth clock of SCL rises while th flag is 1 [Clearing conditions] • When 0 is written to this bit after reading T (When the CPU is used to clear this flag b 0 while the corresponding interrupt is enal sure to read the flag after writing 0 to it.) • When data is written to ICDRT

				sure to read the flag after writing 0 to it.) • When data is read from ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from the device in transmission while the ACKE bit i is set to 1 [Clearing condition] • When 0 is written to this bit after reading N 1 (When the CPU is used to clear this flag by 0 while the corresponding interrupt is enabl sure to read the flag after writing 0 to it.)
3	STOP	0	R/W	Stop Condition Detection Flag [Setting condition] • When a stop condition is detected after fra transfer [Clearing condition] • When 0 is written to this bit after reading S (When the CPU is used to clear this flag by 0 while the corresponding interrupt is enabl sure to read the flag after writing 0 to it.)

				• When the internal SDA and the SDA pin le disagree at the rising of SCL in master tra mode • When the SDA pin outputs a high level in mode while a start condition is detected [Clearing condition] • When 0 is written to this bit after reading A (When the CPU is used to clear this flag b 0 while the corresponding interrupt is enal sure to read the flag after writing 0 to it.)
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 whe frame following a start condition matches bits SVA0 in SAR. [Setting conditions] • When the slave address is detected in sla receive mode • When the general call address is detected receive mode [Clearing condition] • When 0 is written to this bit after reading A

20.3.6 Slave Address Register (SAR)

SAR is sets the slave address. In slave mode, if the upper 7 bits of SAR match the upper 7 the first frame received after a start condition, the LSI operates as the slave device.

Bit	7	6	5	4	3	2	1
Bit Name	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0
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 to 1	SVA6 to SVA0	0	R/W	Slave Address 6 to 0 These bits set a unique address differing from addresses of other slave devices connected to bus.
0	—	0	R/W	Reserved Although this bit is readable/writable, only 0 sh written to.

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

20.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit read-only register that stores the receive data. When one byte of data
received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data ca
received. ICDRR is a receive-only register; therefore, this register cannot be written to b
CPU.

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

20.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is an 8-bit write-only register that is used to transmit/receive data. In transmissio
transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception,
transferred from ICDRS to ICDRR after one by of data is received. This register cannot
from or 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	—	—	—	—	—	—	—

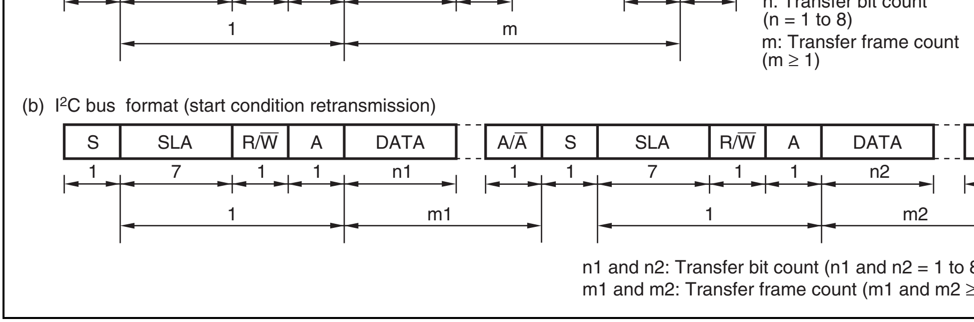


Figure 20.3 I²C Bus Formats

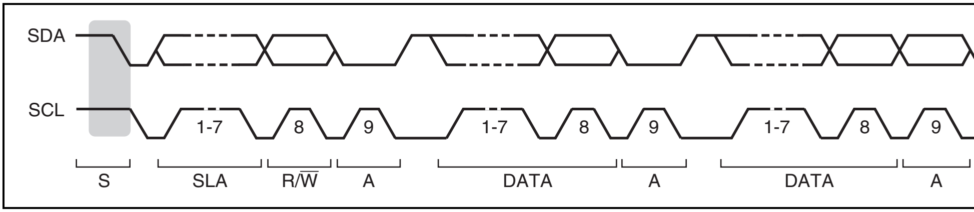


Figure 20.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is hi

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer; from the slave device to the master devic R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0.

A: Acknowledge. The receive device drives SDA low.

DATA: Transferred data

P: Stop condition. The master device drives SDA from low to high while SCL is hi

ICCRA to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using t
instruction. (The start condition is issued.) This generates the start condition.

3. After confirming that TDRE in ICSR has been set, write the transmit data (the first b the slave address and R/W) to ICDRT. After this, when TDRE is automatically clear data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR i at the rising of the ninth transmit clock pulse. Read the ACKBR bit in ICIER to conf the slave device has been selected. Then, write the second byte data to ICDRT. Whe is 1, the slave device has not been acknowledged, so issue a stop condition. To issue condition, write 0 to BBSY and SCP using the MOV instruction. SCL is fixed to a lo until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the e byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR is 1) receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TE NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mo

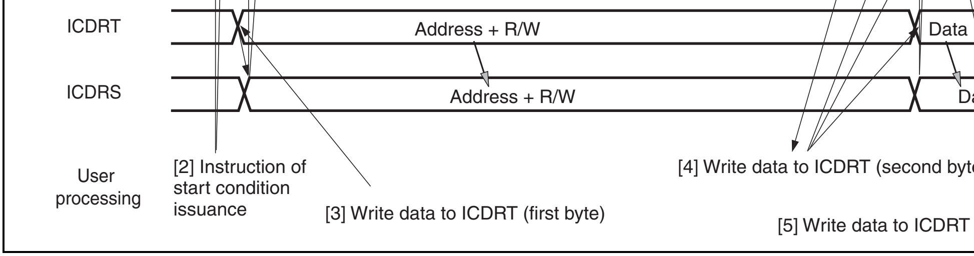


Figure 20.5 Master Transmit Mode Operation Timing 1

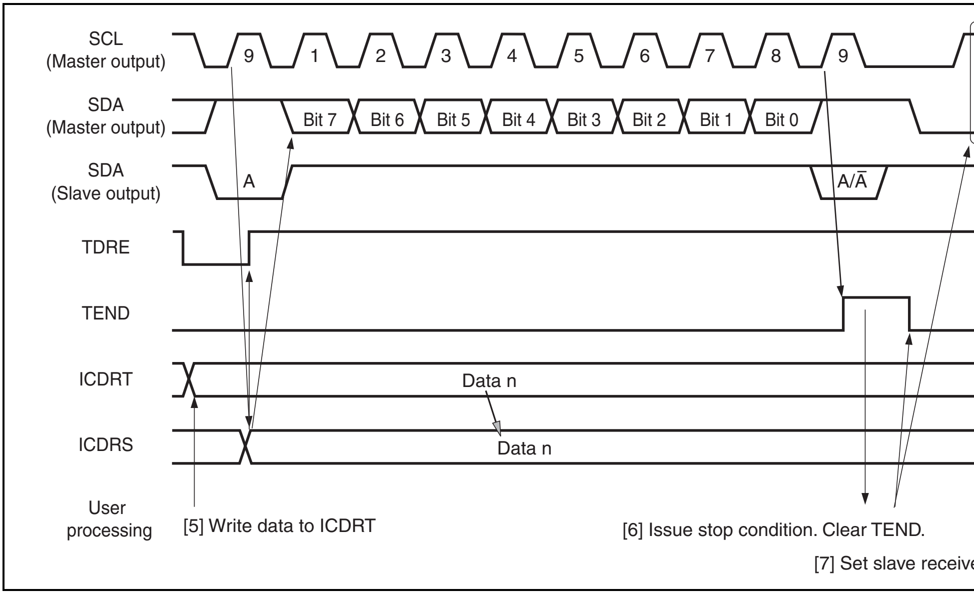


Figure 20.6 Master Transmit Mode Operation Timing 2

data is received, in synchronization with the internal clock. The master mode outpu
specified by the ACKBT in ICIER to SDA, at the ninth receive clock pulse.

3. After the reception of the first frame data is completed, the RDRF bit in ICSR is set
rising of the ninth receive clock pulse. At this time, the received data is read by readi
ICDRR. At the same time, RDRF is cleared.
4. The continuous reception is performed by reading ICDRR and clearing RDRF to 0 e
RDRF is set. If the eighth receive clock pulse falls after reading ICDRR by other pro
while RDRF is 1, SCL is fixed to a low level until ICDRR is read.
5. If the next frame is the last receive data, set the RCVD bit in ICCRA before reading
This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at the rising of the ninth receive clock pulse, the stop
is issued.
7. When the STOP bit in ICSR is set to 1, read ICDRR and clear RCVD to 0.
8. The operation returns to the slave receive mode.

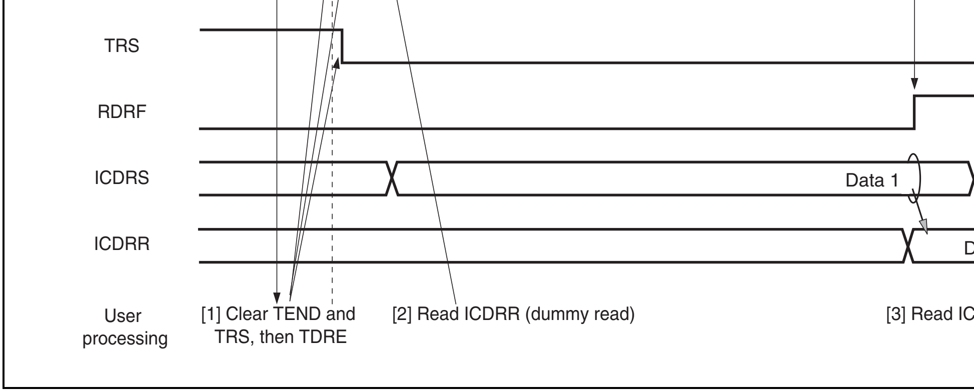


Figure 20.7 Master Receive Mode Operation Timing 1

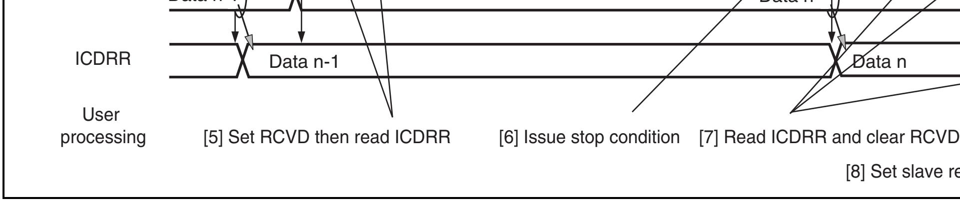


Figure 20.8 Master Receive Mode Operation Timing 2

20.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, and the master device
the receive clock pulse and returns an acknowledge signal. Figures 20.9 and 20.10 show
operation timings in slave transmit mode. The transmission procedure and operations in
transmit mode are described below.

1. Set the ICR bit in the corresponding register to 1, then set the ICE bit in ICCRA to 1
   WAIT in ICMR and CKS3 to CKS0 in ICCRA (initial setting). Set the MST and TR
   ICCRA to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following the detection of the start
   condition, the slave device outputs the level specified by ACKBT in ICIER to SDA,
   rising of the ninth clock pulse. At this time, if the eighth bit data (R/$\overline{W}$) is 1, TRS in
   and TDRE in ICSR are set to 1, and the mode changes to slave transmit mode autom
   The continuous transmission is performed by writing the transmit data to ICDRT eve
   TDRE is set.
3. If TDRE is set after writing the last transmit data to ICDRT, wait until TEND in ICS
   1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for end processing, and read ICDRR (dummy read) to release SCL.
5. Clear TDRE.

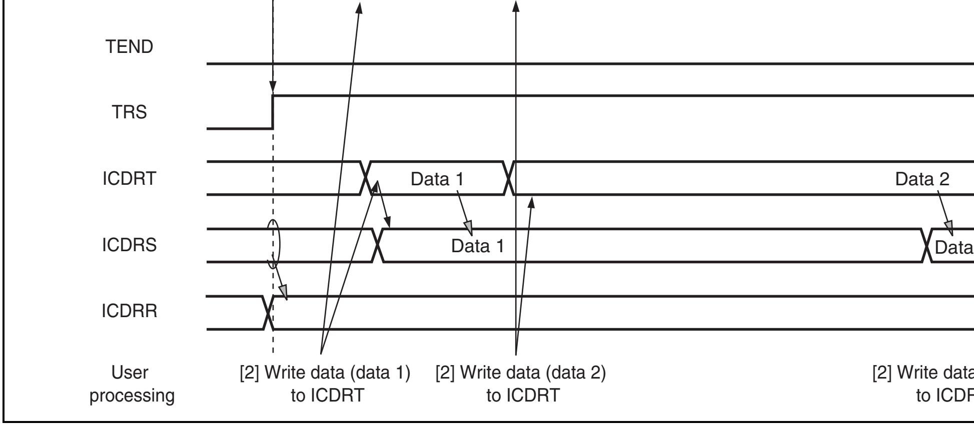


Figure 20.9 Slave Transmit Mode Operation Timing 1

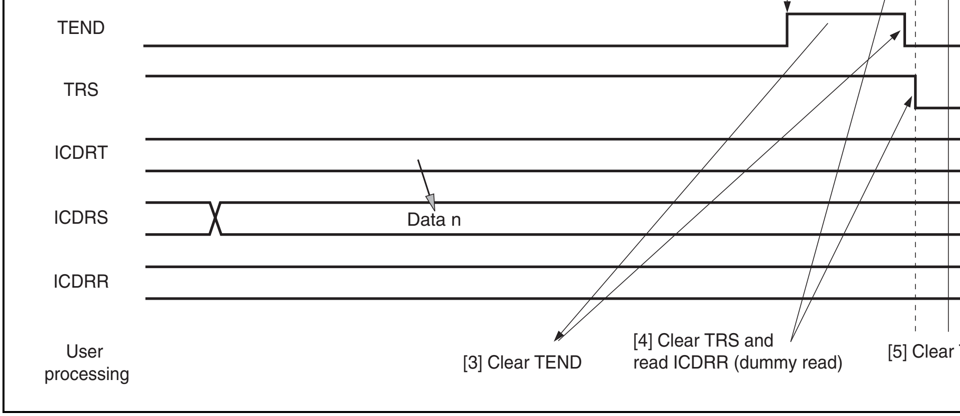


Figure 20.10 Slave Transmit Mode Operation Timing 2

2. When the slave address matches in the first frame following detection of the start con[dition] the slave address outputs the level specified by ACKBT in ICIER to SDA, at the risin[g edge of the] ninth clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy rea[d]). (Since the read data shows the slave address and R/$\overline{W}$, it is not used).
3. Read ICDRR every time RDRF is set. If the eighth clock pulse falls while RDRF is 1, fixed to a low level until ICDRR is read. The change of the acknowledge (ACKBT) s[etting] before reading ICDRR to be returned to the master device is reflected in the next trans[fer] frame.
4. The last byte data is read by reading ICDRR.

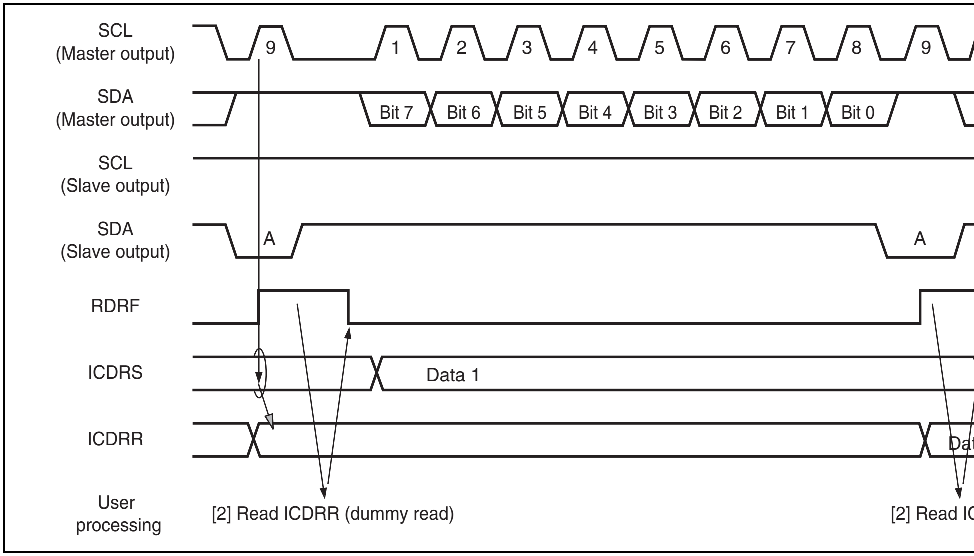


Figure 20.11 Slave Receive Mode Operation Timing 1

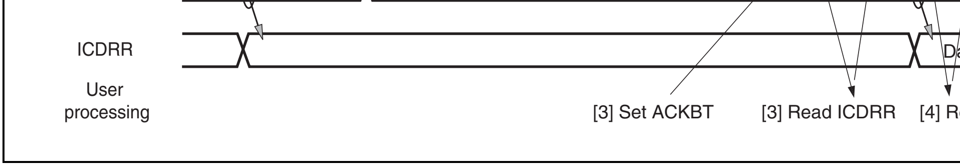


Figure 20.12 Slave Receive Mode Operation Timing 2

20.4.6 Noise Canceler

The logic levels at the SCL and SDA pins are routed through the noise cancelers before latched internally. Figure 20.13 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The signal inp (or SDA) is sampled on the system clock, but is not passed forward to the next circuit un outputs of both latches agree. If they do not agree, the previous value is held.

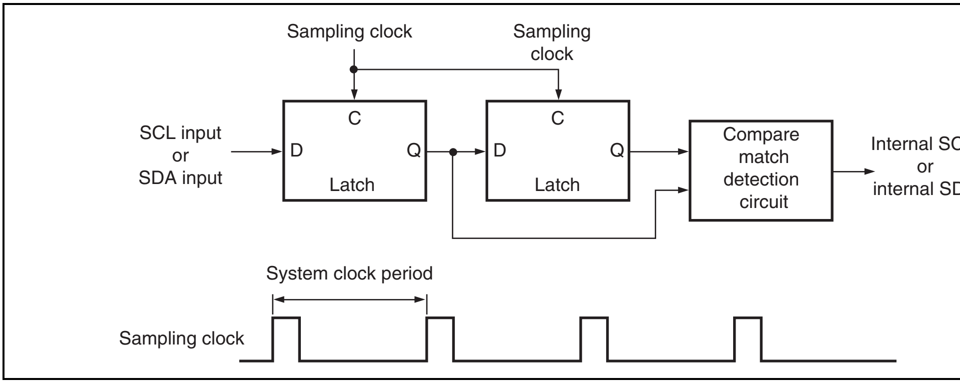
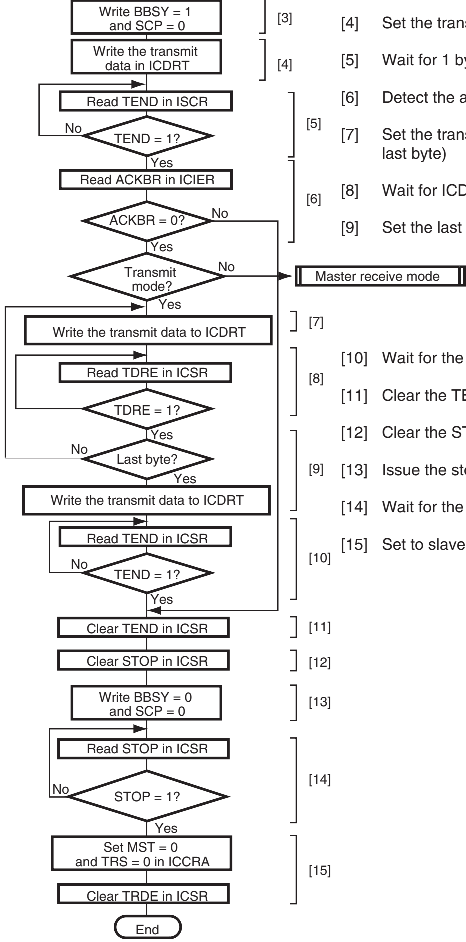


Figure 20.13 Block Diagram of Noise Canceler



[4] Set the transmit data for the first byte (slave address + R/W)

[5] Wait for 1 byte of data to be transmitted

[6] Detect the acknowledge bit, transferred from the specified slave devic

[7] Set the transmit data for the second and subsequent data (except for t last byte)

[8] Wait for ICDRT empty

[9] Set the last byte of transmit data

[10] Wait for the completion of transmission of the last byte

[11] Clear the TEND flag

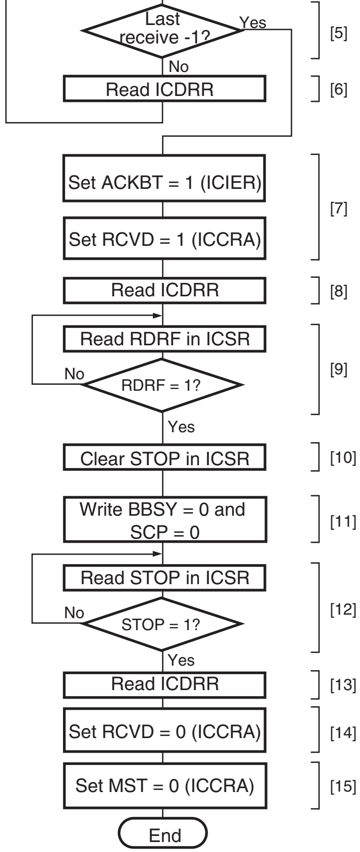
[12] Clear the STOP flag

[13] Issue the stop condition

[14] Wait for the creation of the stop condition

[15] Set to slave receive mode. Clear TDRE.

Figure 20.14 Sample Flowchart of Master Transmit Mode



(RCVD = 1).*2

[8] Read receive data of (last byte -1).*2

[9] Wait for the last byte to be received

[10] Clear the STOP flag

[11] Issue the stop condition

[12] Wait for the creation of stop condition

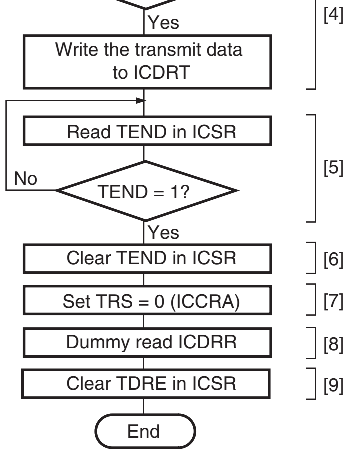
[13] Read the receive data of the last byte

[14] Clear RCVD to 0

[15] Set to slave receive mode

Note: 1. Do not generate an interrupt during steps [1] to [3].
2. For one-byte reception, steps [2] to [6] do not need to be executed. After step [1], execute step [7]
In step [8], read ICDRR (dummy read).

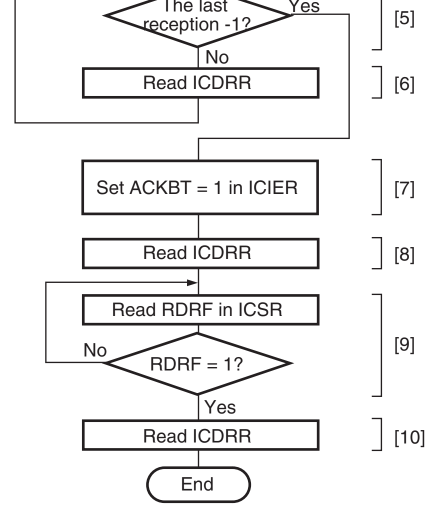
Figure 20.15 Sample Flowchart for Master Receive Mode



[8] Dummy read ICDRR to free the SCL line.

[9] Clear the TDRE flag.

Figure 20.16 Sample Flowchart for Slave Transmit Mode



[8] Read the receive data of (last byte -1).*

[9] Wait for the reception of the last byte to be cor

[10] Read the last byte of receive data.

Note: * For one-byte reception, steps [2] to [6] do not need to be executed. After step [1], execute step [7]. In step [8], read ICDRR (dummy read).

Figure 20.17 Sample Flowchart for Slave Receive Mode

Receive Data Full	RXI	(RDRF = 1) · (RIE = 1)
Stop Recognition	STPI	(STOP = 1) · (STIE = 1)
NACK Detection	NAKI	{(NACKF = 1) + (AL = 1)} · (NAKIE = 1)
Arbitration Lost		

When one of the interrupt conditions in table 20.3 is 1 and the I bit in CCR is 0, the CPU
interrupt exception handling. Clear the interrupt sources during interrupt exception handli
that the TDRE and TEND bits are automatically cleared to 0 by writing data to ICDRT, a
RDRF bit is cleared to 0 by reading ICDRR. In particular, the TDRE bit can be set again
same time as data are for transmission written to ICDRT, and 1 extra byte can be transmi
TDRE is again cleared to 0.

Figure 20.18 shows the timing of the bit synchronous circuit, and table 20.4 shows the ti
SCL output changes from low to Hi-Z and the period which SCL is monitored.

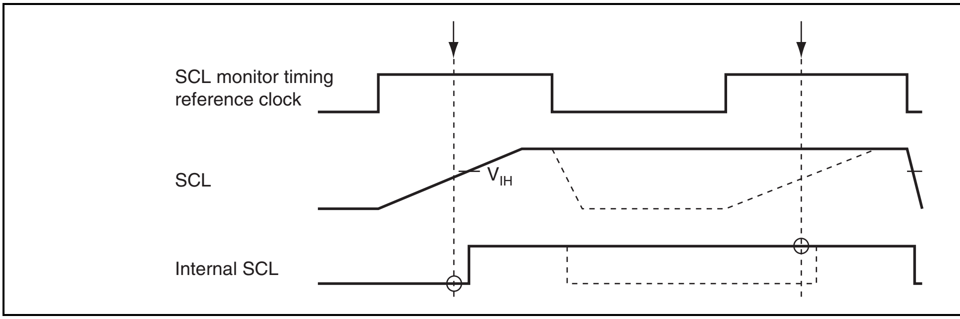


Figure 20.18 Timing of the Bit Synchronous Circuit

Table 20.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 tcyc
	1	19.5 tcyc
1	0	17.5 tcyc
	1	41.5 tcyc

during the eighth clock.

2. The WAIT bit in the I^2C bus mode register (ICMR) must be held 0.

   If the WAIT bit is set to 1, when a slave device holds the SCL signal low more than o
   transfer clock cycle during the eighth clock, the high level period of the ninth clock m
   shorter than a given period.

3. Restriction in transfer rate setting value in multi-master mode

   When the transfer rate of I^2C transfer of this LSI is slower than that of other master, th
   signal the width of which is unexpected may be output. To avoid this phenomenon, se
   transfer rate of 1/1.8 or more of the fastest rate of other master to the transfer rate of I
   transfer rate. For example, if the fastest rate of other masters is 400 kbps, the I^2C trans
   of this LSI should be 223 kbps (= 400/1.8) or more.

4. Restriction in bit manipulation when the MST and TRS bits are set in multi-master m

   When the MST and TRS bits are set to master slave mode by manipulating these bits
   sequentially, the conflict state occurs as follows according to the timing that arbitratio
   The AL bit in ICSR is set to 0, and set to master mode (MST = 1, TRS = 1). There are
   following methods to avoid this phenomenon.

   - In multi-master mode, set the MST and TRS bits by MOV instruction.
   - When arbitration is lost, confirm that the MST and TRS bits are set to 0. If these
     set to other than 0, set these bits to 0.

5. Notes on master receive mode

   In master receive mode, the RDRF bit is set to 0 at the eighth rising clock, the SCL si
   pulled to "Low" state. When ICDRR is read near at the eighth falling clock, the SCL
   level is released and the ninth clock is outputted by fixing the eighth clock of receive
   "Low" state. Reading ICDRR is not required. As a result, the failure to receive data o
   There are the following methods to avoid this phenomenon.

- 10-bit resolution
- Eight or four input channels (total eight input channels for the two units)

  Four channels x two units (for unit 0 and unit 1)

  Eight channels x one unit (for unit 0)
- Conversion time: 2.7 µs per channel (in peripheral clock mode)

  1.0 µs per channel (in system clock mode*[3])
- 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*[1]
- Eight data registers for the A/D converter unit 0 and four data registers for unit 1 (to
  data registers for the two units)

  Results of A/D conversion are held in a 16-bit data register for each channel.
- Sample and hold functionality
- Three types of conversion start

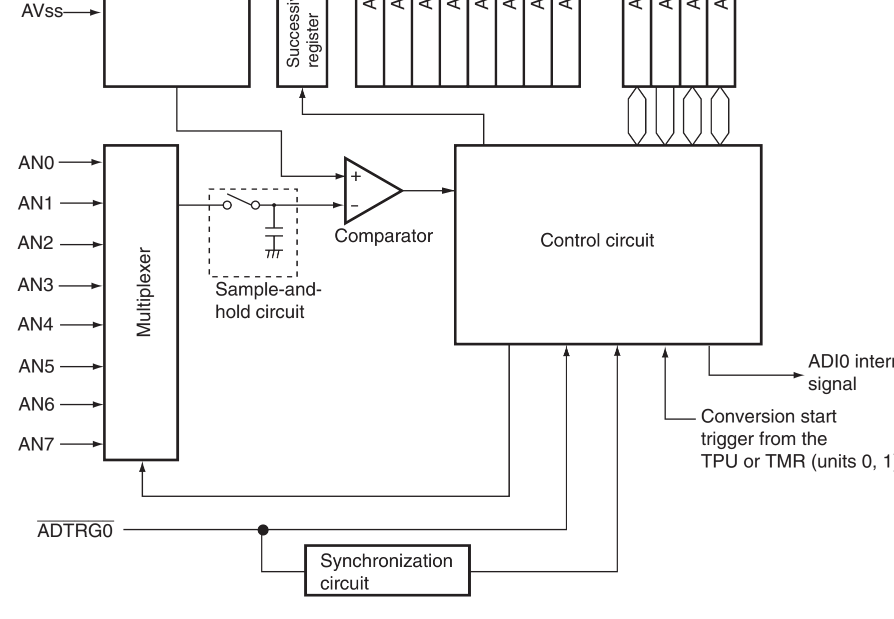
  Conversion can be started by software, a conversion start trigger by the 16-bit timer (TPU)*[1] or 8-bit timer (TMR)*[2], or an external trigger signal.
- Function of starting units simultaneously

  A/D conversion for multiple units can be started by external trigger ($\overline{\text{ADTRG0}}$).
- Interrupt source

  A/D conversion end interrupt (ADI) request can be generated.
- Module stop state specifiable

Notes: 1. Only supported in the A/D converter unit 0.

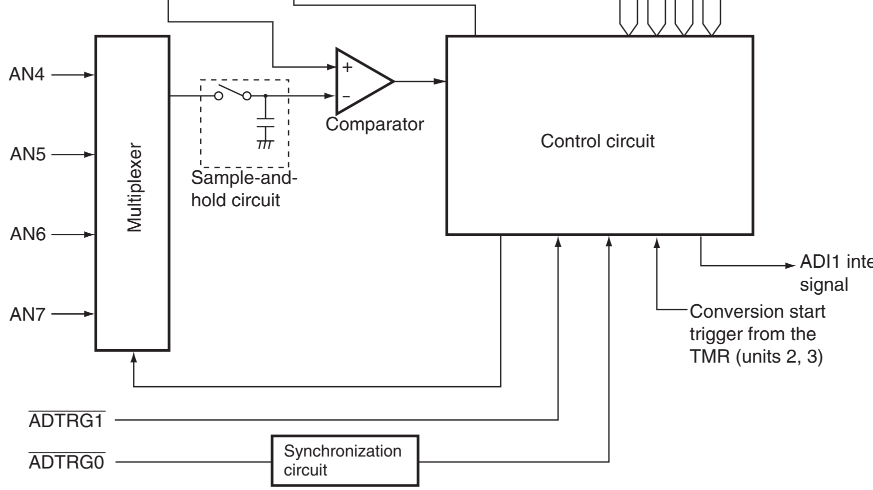
2. For unit 0, A/D conversion can be started by a conversion start trigger by th units 0 and 1 whereas for unit 1 A/D conversion can be started by a convers trigger by the TMR units 2 and 3.



[Legend]

ADCR_0:	A/D control register_0	ADDRC_0:	A/D data register C_0
ADCSR_0:	A/D control/status register_0	ADDRD_0:	A/D data register D_0
ADSSTR_0:	A/D sampling state register_0	ADDRE_0:	A/D data register E_0
ADMOSEL_0:	A/D mode selection register_0	ADDRF_0:	A/D data register F_0
ADDRA_0:	A/D data register A_0	ADDRG_0:	A/D data register G_0
ADDRB_0:	A/D data register B_0	ADDRH_0:	A/D data register H_0

Figure 21.1 Block Diagram of A/D Converter Unit 0 (AD_0)



[Legend]

ADCR_1:	A/D control register_1	ADDRC_1:	A/D data register C_1
ADCSR_1:	A/D control/status register_1	ADDRD_1:	A/D data register D_1
ADSSTR_1:	A/D sampling state register_1	ADDRE_1:	A/D data register E_1
ADMOSEL_1:	A/D mode selection register_1	ADDRF_1:	A/D data register F_1
ADDRA_1:	A/D data register A_1	ADDRG_1:	A/D data register G_1
ADDRB_1:	A/D data register B_1	ADDRH_1:	A/D data register H_1

Figure 21.2 Block Diagram of A/D Converter Unit 1 (AD_1)

		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 0	$\overline{ADTRG0}$	Input	External trigger inpu starting A/D conver
1	AD_1	Analog input pin 4	AN4	Input	Analog inputs
		Analog input pin 5	AN5	Input	
		Analog input pin 6	AN6	Input	
		Analog input pin 7	AN7	Input	
		A/D external trigger input pin 0	$\overline{ADTRG0}$	Input	External trigger inpu starting A/D conver
		A/D external trigger input pin 1	$\overline{ADTRG1}$	Input	External trigger inpu starting A/D conver
Common		Analog power supply pin	AV_{CC}	Input	Analog block power
		Analog ground pin	AV_{SS}	Input	Analog block groun
		Reference voltage pin	Vref	Input	A/D conversion refe voltage

Note: * Selectable by setting of the TRGS1, TRGS0, and EXTRGS bits in ADCR.

- A/D data register E_0 (ADDRE_0)
- A/D data register F_0 (ADDRF_0)
- A/D data register G_0 (ADDRG_0)
- A/D data register H_0 (ADDRH_0)
- A/D control/status register_0 (ADCSR_0)
- A/D control register_0 (ADCR_0)
- A/D mode selection register_0 (ADMOSEL_0)
- A/D sampling state register_0 (ADSSTR_0)

Unit 1 (A/D_1) registers:

- A/D data register A_1 (ADDRA_1)
- A/D data register B_1 (ADDRB_1)
- A/D data register C_1 (ADDRC_1)
- A/D data register D_1 (ADDRD_1)
- A/D data register E_1 (ADDRE_1)
- A/D data register F_1 (ADDRF_1)
- A/D data register G_1 (ADDRG_1)
- A/D data register H_1 (ADDRH_1)
- A/D control/status register_1 (ADCSR_1)
- A/D control register_1 (ADCR_1)
- A/D mode selection register_1 (ADMOSEL_1)
- A/D sampling state register_1 (ADSSTR_1)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit Name											—	—	—	—	—
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register Storing Conversion Result	
	Unit 0	Unit 1*[2]
AN0	ADDRA_0 (Unit 0)	—
AN1	ADDRB_0 (Unit 0)	—
AN2	ADDRC_0 (Unit 0)	—
AN3	ADDRD_0 (Unit 0)	—
AN4	ADDRE_0 (Unit 0)*[1]	ADDRE_1 (Unit 1)*[1]
AN5	ADDRF_0 (Unit 0)*[1]	ADDRF_1 (Unit 1)*[1]
AN6	ADDRG_0 (Unit 0)*[1]	ADDRG_1 (Unit 1)*[1]
AN7	ADDRH_0 (Unit 0)*[1]	ADDRH_1 (Unit 1)*[1]

Notes: 1. A/D conversion should not be performed on the same channel by multiple units

2. The ADDRA_1 to ADDRD_1 registers for unit 1 are not used.

Bit	Bit Name	Value	R/W	Description
7	ADF	0	R/(W)*[1]	A/D End Flag A status flag that indicates the end of A/D conv [Setting conditions] • Completion of A/D conversion in single mod • Completion of A/D conversion on all specifie channels in scan mode [Clearing conditions] • Writing of 0 after reading ADF = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enabled to read the flag after writing 0 to it.) • Reading from ADDR after activation of the D DTC by an ADI interrupt
6	ADIE	0	R/W	A/D Interrupt Enable Setting this bit to 1 enables ADI interrupts by A

				cleared to 0 upon completion of A/D conversion t the selected channels to stop A/D conversion. The ADST bit is automatically cleared at a differe from that of setting the ADF bit. The ADST bit is before setting the ADF bit.
4	EXCKS*[2]	0	R/W	Extended Clock Selection Sets the A/D conversion time in accord with bits and CKS0 in ADCR and the ICKSEL bit in ADMC For details, see section 21.3.4, A/D Control Regi (ADCR_0) for Unit 0. Write to the EXCKS bit at t time as bits CKS1 and CKS0 in ADCR.

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

Notes: 1. Only 0 can be written to this bit, to clear the flag.
2. The full-spec emulator (E6000H) should not be used, but the on-chip emulato
USB) is usable.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • Completion of A/D conversion in single mode • Completion of A/D conversion on all specified chann scan mode [Clearing conditions] • Writing of 0 after reading ADF = 1 (When the CPU is used to clear this flag by writing 0 the corresponding interrupt is enabled, be sure to re flag after writing 0 to it.) • Reading from ADDR after activation of the DMAC or an ADI interrupt
6	ADIE	0	R/W	A/D Interrupt Enable Setting this bit to 1 enables ADI interrupts by ADF.
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters wait state. Setting this bit to 1 starts A/D conversion. In single mod is cleared to 0 automatically when A/D conversion on th specified channel ends. In scan mode, A/D conversion c sequentially on the specified channels until this bit is cle by software, a reset, or hardware standby mode. In add when the ADSTCLR bit in ADCR is 1, the ADST bit is automatically cleared to 0 upon completion of A/D conv all of the selected channels to stop A/D conversion. The ADST bit is automatically cleared at a different time that of setting the ADF bit. The ADST bit is cleared befo setting the ADF bit.

Bit	Bit Name	Initial Value	R/W	Description
0	CH0	0	R/W	00xx: Setting prohibited 0100: AN4 0101: AN5 0110: AN6 0111: AN7 1xxx: Setting prohibited • When SCANE = 1 and SCANS = 0 00xx: Setting prohibited 0100: AN4 0101: AN4 and AN5 0110: AN4 to AN6 0111: AN4 to AN7 1xxx: Setting prohibited • When SCANE = 1 and SCANS = 1 xxxx: Setting prohibited

[Legend]

x: Don't care

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

Bit	Bit Name	Value	R/W	Description
7 6 0	TRGS1 TRGS0 EXTRGS	0 0 0	R/W R/W R/W	Timer Trigger Select 1 and 0, Extended Trigger Select These bits select enabling or disabling of the start of A/D conversion by a trigger signal. 000: Disables starting of A/D conversion by external trigg 010: A/D conversion is started by conversion trigger from (unit 0) 100: A/D conversion is started by conversion trigger from (units 0 and 1) 110: A/D conversion is started by the $\overline{\text{ADTRG0}}$ signal*[1] 001: External trigger is invalid 011: Setting prohibited 101: Setting prohibited 111: A/D conversion is started by the $\overline{\text{ADTRG0}}$ signal*[1] (s units simultaneously)
5 4	SCANE SCANS	0 0	R/W R/W	Scan Mode These bits select the A/D conversion operating mode. 0x: Single mode 10: Scan mode. A/D conversion is performed continuous channels 1 to 4. 11: Scan mode. A/D conversion is performed continuous channels 1 to 8.

Bit	Bit Name	Initial Value	R/W	Description
				0001: A/D conversion time = 268 states*[4] (max.) 0010: A/D conversion time = 138 states*[4] (max.) 0011: A/D conversion time = 73 states*[4] (max.) 01xx: Prohibited setting 1000: A/D conversion time = 336 states*[4] (max.) 1001: A/D conversion time = 172 states*[4] (max.) 1010: A/D conversion time = 90 states*[4] (max.) 1011: A/D conversion time = 49 states*[4] (max.) 11xx: A/D conversion time = 34 states*[4]*[5] (max.) This setting applies when Pφ = Iφ/2*[6].
1	ADSTCLR*[2]	0	R/W	A/D Start Clear Enables or disables automatic clearing of the ADST bit i mode. 0: The ADST bit is not automatically cleared to 0 in scan 1: The ADST bit is cleared to 0 upon completion of A/D conversion for all of the selected channels in scan m

[Legend]

x: Don't care

Notes: 1. To set A/D conversion to start by the $\overline{\text{ADTRG}}$ pin, the DDR bit and ICR bit for corresponding pin should be set to 0 and 1, respectively. For details, see sec Ports.

2. The full-spec emulator (E6000H) should not be used, but the on-chip emula USB) is usable.

3. ICKSEL = 1: The full-spec emulator (E6000H) should not be used, but the o emulator (E10A-USB) is usable.

4. Cycles of Pφ

5. Set the number of "states" (clock cycles) for sampling to 25 (ADSSTR_0 = D

6. When Pφ = Iφ, Iφ/4, or Iφ/8, settings of the form 11xx are prohibited.

Bit	Bit Name	Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0, Extended Trigger Selec
6	TRGS0	0	R/W	These bits select enabling or disabling of the start of conversion by a trigger signal.
0	EXTRGS	0	R/W	000: Disables starting of A/D conversion by external t 010: Setting prohibited 100: Setting prohibited 110: A/D conversion is started by the $\overline{\text{ADTRG1}}$ signa 001: Setting prohibited 011: External trigger is invalid 101: A/D conversion is started by conversion trigger f TMR (units 2 and 3) 111: A/D conversion is started by the $\overline{\text{ADTRG0}}$ signa units simultaneously)
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	These bits select the A/D conversion operating mode 0x: Single mode 10: Scan mode. A/D conversion is performed continu channels 1 to 4. 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
				0000: A/D conversion time = 528 states*[3] (max.) 0001: A/D conversion time = 268 states*[3] (max.) 0010: A/D conversion time = 138 states*[3] (max.) 0011: A/D conversion time = 73 states*[3] (max.) 01xx: Prohibited setting 1000: A/D conversion time = 336 states*[3] (max.) 1001: A/D conversion time = 172 states*[3] (max.) 1010: A/D conversion time = 90 states*[3] (max.) 1011: A/D conversion time = 49 states*[3] (max.) 11xx: A/D conversion time = 34 states*[3]*[4] (max.) This setting applies when Pϕ = Iϕ/2*[5].
1	ADSTCLR	0	R/W	A/D Start Clear Enables or disables automatic clearing of the ADST scan mode. 0: The ADST bit is not automatically cleared to 0 in mode. 1: The ADST bit is cleared to 0 upon completion of conversion for all of the selected channels in sca

[Legend]

x: Don't care

Notes: 1. To set A/D conversion to start by the $\overline{\text{ADTRG}}$ pin, the DDR bit and ICR bit for corresponding pin should be set to 0 and 1, respectively. For details, see sec Ports.

2. ICKSEL = 1: Access to the full-spec emulator (E6000H) is prohibited but the emulator (E10A-USB) is usable.

3. Cycles of Pϕ

4. Set the number of "states" (clock cycles) for sampling to 25 (ADSSTR_1 = D

5. When Pf = If, If/4, or If/8, settings of the form 11xx are prohibited.

Bit	Bit Name	Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write value sh always be 0.
5	—	0	R	
4	—	0	R	
3	—	0	R	
2	—	0	R	
1	ICKSEL*[1]	0	R/W	System Clock Mode Selection This bit is used to select the system clock mode. 0: Peripheral clock mode 1: System clock mode*[2] For details, see section 21.4.5, Setting the System C Mode.
0	—	0	R/W	Reserved This bit is always read as 0. The write value should be 0.

Notes: 1. The full-spec emulator (E6000H) should not be used, but the on-chip emulator USB) is usable.

2. In system clock mode, operate all units with the system clock.

Bit	Bit Name	Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write value sh always be 0.
5	—	0	R	
4	—	0	R	
3	—	0	R	
2	—	0	R	
1	ICKSEL*[1]	0	R/W	System Clock Mode Selection This bit is used to select the system clock mode. 0: Peripheral clock mode 1: System clock mode*[2] For details, see section 21.4.5, Setting the System Mode.
0	—	0	R/W	Reserved This bit is always read as 0. The write value should be 0.

Notes: 1. Access to the full-spec emulator (E6000H) is prohibited, but the on-chip emul (E10A-USB) is usable.

2. In system clock mode, operate all units with the system clock.

Bit	Bit Name	Initial Value	R/W	Description
7	SMP7	0	R/W	• When EXCKS = 0
6	SMP6	0	R/W	Set these bits to H'0F.
5	SMP5	0	R/W	• When EXCKS = 1
4	SMP4	0	R/W	If ICKSEL = 0, set these bits to H'0F.
3	SMP3	1	R/W	If ICKSEL = 1*, set these bits to H'19.
2	SMP2	1	R/W	
1	SMP1	1	R/W	
0	SMP0	1	R/W	

Note: * The full-spec emulator (E6000H) should not be used, but the on-chip emulator USB) is usable.

Bit	Bit Name	Initial Value	R/W	Description
7	SMP7	0	R/W	• When EXCKS = 0
6	SMP6	0	R/W	Set these bits to H'0F.
5	SMP5	0	R/W	• When EXCKS = 1
4	SMP4	0	R/W	If ICKSEL = 0, set these bits to H'0F.
3	SMP3	1	R/W	If ICKSEL = 1*, set these bits to H'19.
2	SMP2	1	R/W	
1	SMP1	1	R/W	
0	SMP0	1	R/W	

Note: * Access to the full-spec emulator (E6000H) is prohibited but the on-chip emula
USB) is usable.

single channel.

1. A/D conversion for the selected channel is started when the ADST bit in ADCSR is se software, TPU*[1], TMR*[2], 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 at 1 during A/D conversion, and is automatically cleared to 0 v 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 a wait sta

Notes: 1. Only possible in unit 0.
2. As conversion start trigger, units 0 and 1 of TMR, and units 2 and 3 of TMR a available in unit 0, and unit 1, respectively.

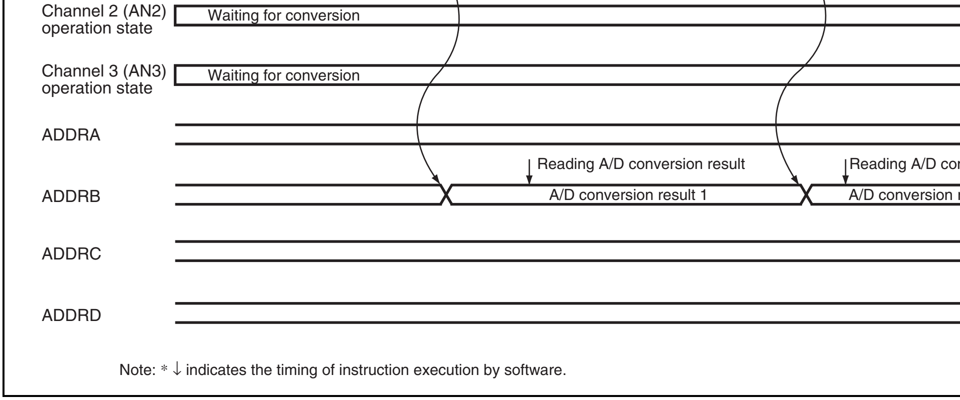


Figure 21.3 Example of A/D Converter Operation (Single Mode, Channel 1 Se

21.4.2 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*[1] channels. Two types of scan mode are provided, that is, co scan mode where A/D conversion is repeatedly performed and one-cycle scan mode*[2] w conversion is performed for the specified channels for one cycle.

transferred to the corresponding ADDR of each channel.

3. When A/D conversion of all selected channels is completed, the ADF bit in ADCSR i
If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. A/D conv
the first channel in the group starts again.

4. The ADST bit is not cleared automatically, and steps 2 to 3 are repeated as long as the
bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the
converter enters wait state. If the ADST bit is later set to 1, A/D conversion starts aga
the first channel in the group.

Notes: 1. Consecutive A/D conversion on eight channels is only possible in unit 0.

2. Unit 0: The full-spec emulator (E6000H) should not be used, but the on-chip
(E10A-USB) is usable.

3. As conversion start trigger, units 0 and 1 of TMR, and units 2 and 3 of TMR
available in unit 0, and unit 1, respectively.

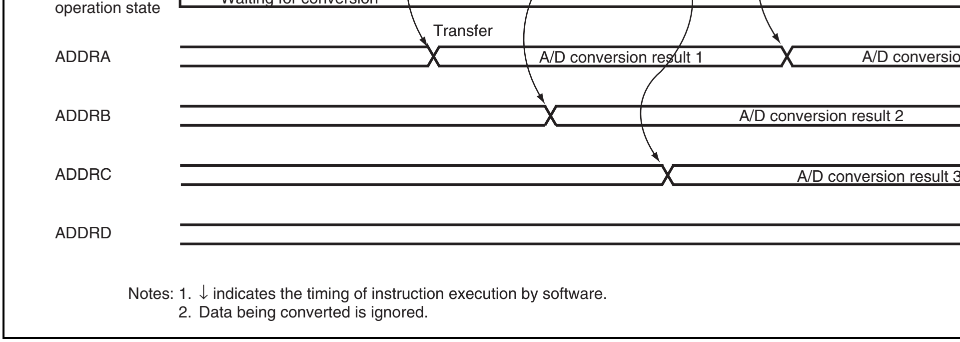


Figure 21.4 Example of A/D Conversion (Continuous Scan Mode, Three Channels (AN0 to AN2) Selected)

5. The ADST bit is automatically cleared when A/D conversion is completed for all of th channels that have been selected. A/D conversion stops and the A/D converter enters state.

Note: * For unit 0, the full-spec emulator (E6000H) should not be used, but the on-chi emulator (E10A-USB) is usable.

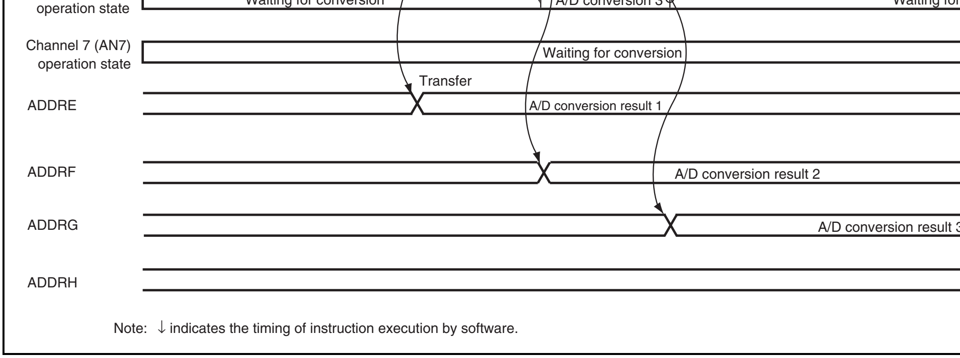


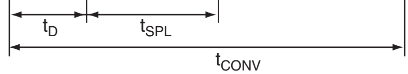
Figure 21.5 Example of A/D Conversion (One-Cycle Scan Mode, Three Channels (AN4 to AN6) Selected)

In scan mode, the values given in tables 21.3 to 21.5 apply to the first conversion time. Th given in table 21.6 apply to the second and subsequent rounds of conversion. In either cas CKS1 and CKS0 bits in ADCR, the ICKSEL*[1] bit in ADMOSEL, and the EXCKS*[2] bit i ADCSR should be set so that the conversion time is within the ranges indicated by the A/ conversion characteristics.

Notes: 1. Unit 0: The full-spec emulator (E6000H) should not be used, but the on-chip e (E10A-USB) is usable.

Unit 1: Access to the full-spec emulator (E6000H) is prohibited but the on-chi emulator (E10A-USB) is usable.

2. Unit 0: The full-spec emulator (E6000H) should not be used, but the on-chip (E10A-USB) is usable.



[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 21.6 Periods of A/D Conversion

Table 21.3 Characteristics of A/D Conversion (Unit 0: when EXCKS* = 0, ICKSI and ADSSTR* = H'0F) (1)

Item	Symbol	CKS1 = 0						CKS1 = 1				
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CK	
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	T
A/D conversion start delay time	t_D	3	—	14	3	—	10	3	—	8	3	
Input sampling time	t_{SPL}	—	312	—	—	156	—	—	78	—	—	
A/D conversion time	t_{CONV}	517	—	528	261	—	268	133	—	138	69	

Notes: Values in the table are numbers of states.
* The full-spec emulator (E6000H) should not be used, but the on-chip emulato USB) is usable.

Notes: Values in the table are numbers of states.

* The full-spec emulator (E6000H) should not be used, but the on-chip emulator
USB) is usable.

Table 21.4 Characteristics of A/D Conversion (Unit 1: when EXCKS = 0, ICKSEL ADSSTR* = H'0F) (1)

		CKS1 = 0						CKS1 = 1				
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS	
Item	Symbol	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Ty
A/D conversion start delay time	t_D	4	—	14	4	—	10	4	—	8	4	—
Input sampling time	t_{SPL}	—	312	—	—	156	—	—	78	—	—	3
A/D conversion time	t_{CONV}	518	—	528	262	—	268	134	—	138	70	—

Notes: Values in the table are numbers of states.

* Access to the full-spec emulator (E6000H) is prohibited but the on-chip emulat
USB) is usable.

Notes: Values in the table are numbers of states.

* Access to the full-spec emulator (E6000H) is prohibited but the on-chip emula
USB) is usable.

**Table 21.5 Characteristics of A/D Conversion (When EXCKS*1 = 1, ICKSEL*1 =
ADSSTR*2 = H'19)**

Item	Symbol	Iϕ:Pϕ = 1:1/2				
		Unit 0			Unit 1	
		Min.	Typ.	Max.	Min.	Typ.
A/D conversion start delay time	t_D	2.5	—	6.5	3.5	—
Input sampling time	t_{SPL}	—	12.5	—	—	12.5
A/D conversion time	t_{CONV}	30	—	34	31	—

Notes: Values in the table are numbers of states (cycles of Pϕ). Make the sampling setti
(ADSSRT = D'25).

1. Unit 0: The full-spec emulator (E6000H) should not be used, but the on-chip e
(E10A-USB) is usable.
2. Unit 0: The full-spec emulator (E6000H) should not be used, but the on-chip e
(E10A-USB) is usable.
Unit 1: Access to the full-spec emulator (E6000H) is prohibited but the on-chi
(E10A-USB) is usable.

	1	0	80*[1]
		1	40*[1]
1*[3]*[4]	—	—	25*[2]

Notes: 1. Make the sampling setting15 (ADSSRT = D'15).

2. When Pϕ = Iϕ/2, make the sampling setting 25 (ADSSRT = D'25).

3. Unit 0: The full-spec emulator (E6000H) should not be used, but the on-chip er (E10A-USB) is usable.

4. Unit 1: Access to the full-spec emulator (E6000H) is prohibited but the on-chip (E10A-USB) is usable.

TRGS1, TRGS0, and EXTRGS bits are set to B'111 in ADCR_0 and ADCR_1. A/D co
starts when the ADST bit in ADCSR is set to 1 on the falling edge of the $\overline{ADTRG0}$ pin.
timing is different from the one when multiple units do not start simultaneously. Figure
shows the timing.

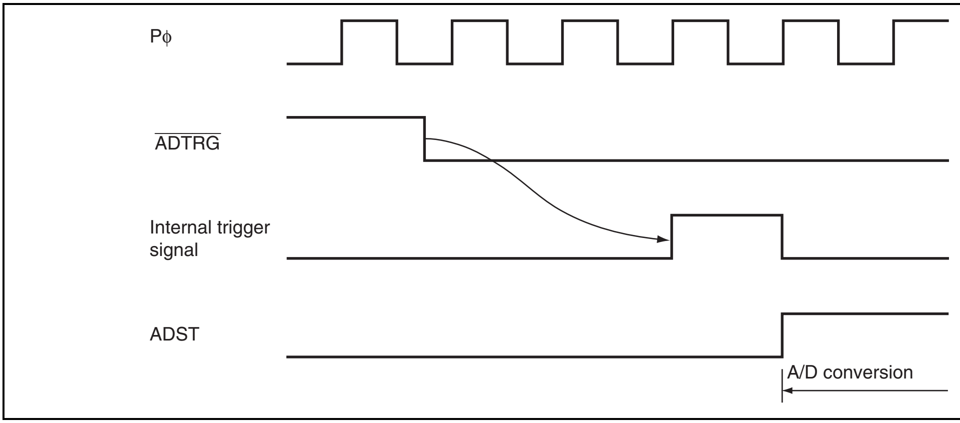


Figure 21.7 External Trigger Input Timing (TRGS1, TRGS0, and EXTRGS ≠

Figure 21.8 External Trigger Input Timing when Multiple Units Start Simultan (TRSG1, TRGS0, and EXTRGS = B'111)

21.4.5 Setting the System Clock Mode

In system clock mode, set Iϕ = 50 MHz, Pϕ = Iϕ/2, and make the sampling setting 25. A/ conversion*[1] with a conversion time of 1 µs per channel is possible. For information on c the frequency of the system clock relative to the input clock, see section 26, Clock Pulse Generator.

When the ADST bit is cleared to 0, start A/D conversion following the procedures shown

1. Set Pϕ = Iϕ/2.
2. Release the A/D converter from the module-stopped state.
3. Set the EXCKS*[2] bit in ADCSR to 1 (making setting of the number of states for samp ADSSTR*[2]*[3] effective).
4. Set*[2]*[3] the ICKSEL bit in ADMODSEL*[2]*[3] to 1 (selecting system clock mode).
5. Write H'19 to ADSSTR*[2]*[3] (setting the number of states for sampling to 25).
6. Start A/D conversion (set the ADST bit to 1 or have the trigger signal initiate convers

Notes: 1. The full-spec emulator (E6000H) should not be used, but the on-chip emulato USB) is usable.
2. For unit 0, the full-spec emulator (E6000H) should not be used, but the on-chi emulator (E10A-USB) is usable.
3. For unit 1, access to the full-spec emulator (E6000H) is prohibited, but the on- emulator (E10A-USB) is usable.

Table 21.7 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activ
ADI	A/D conversion end	ADF	Possible*	Possible

Note: * Only possible in unit 0.

when the digital output changes from the minimum voltage value B'0000000000 (H'0
B'0000000001 (H'001) (see figure 21.10).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion charac
  when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3F
  figure 21.10).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero vol
  the full-scale voltage. Does not include the offset error, full-scale error, or quantizatio
  (see figure 21.10).

- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offse
  full-scale error, quantization error, and nonlinearity error.

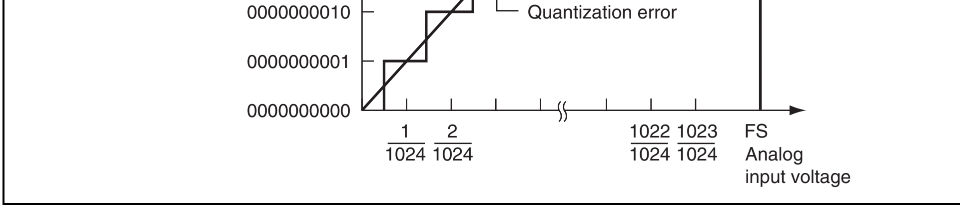


Figure 21.9 A/D Conversion Accuracy Definitions

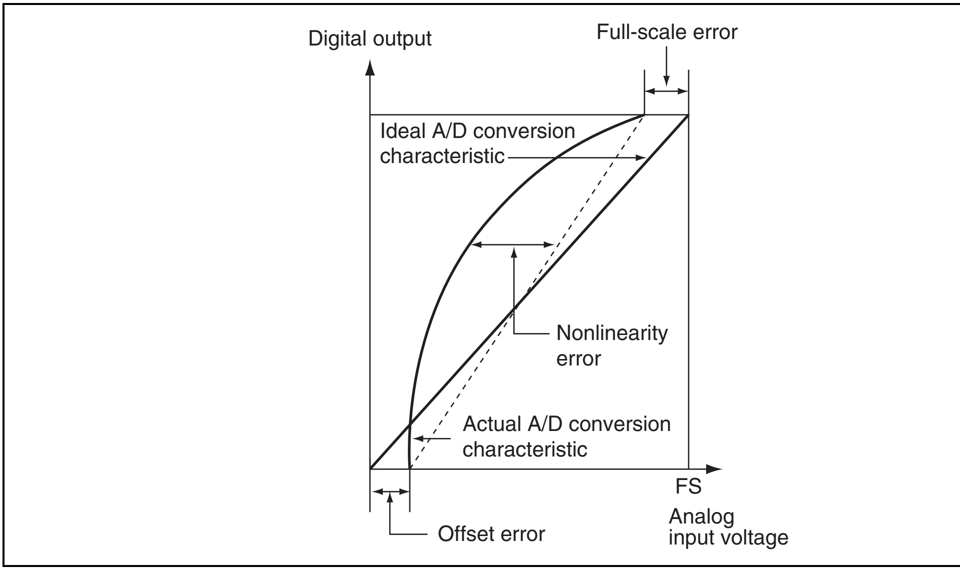


Figure 21.10 A/D Conversion Accuracy Definitions

21.7.2 A/D Input Hold Function in Software Standby Mode

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, set the CKS1 and C to 1 and clear the ADST, TRGS1, TRGS0, and EXTRGS bits all to 0 to disable A/D conv After that, enter software standby mode after executing a dummy read from ADCSR.

21.7.3 Notes on Stopping the A/D Converter

When the A/D start bit (ADST) is cleared during A/D conversion by software, A/D conve results may be stored incorrectly (ADDR), or when A/D conversion restarts, the interrupt be misset.

To avoid these events, follow the steps below.

(1) In Single Mode or Scan Mode (One-Cycle Scan Mode)

As the ADST bit is automatically cleared when A/D conversion is completed, do not clea during A/D conversion.

(2) In Scan Mode (Continuous Scan Mode)

- When the A/D Converter is Activated by Software

  Do not clear the ADST bit during A/D conversion. To stop A/D conversion, rewrite th SCANE bit to change modes from scan mode to single mode. By rewriting the SCAN A/D converter is stopped without clearing the ADST bit by software.

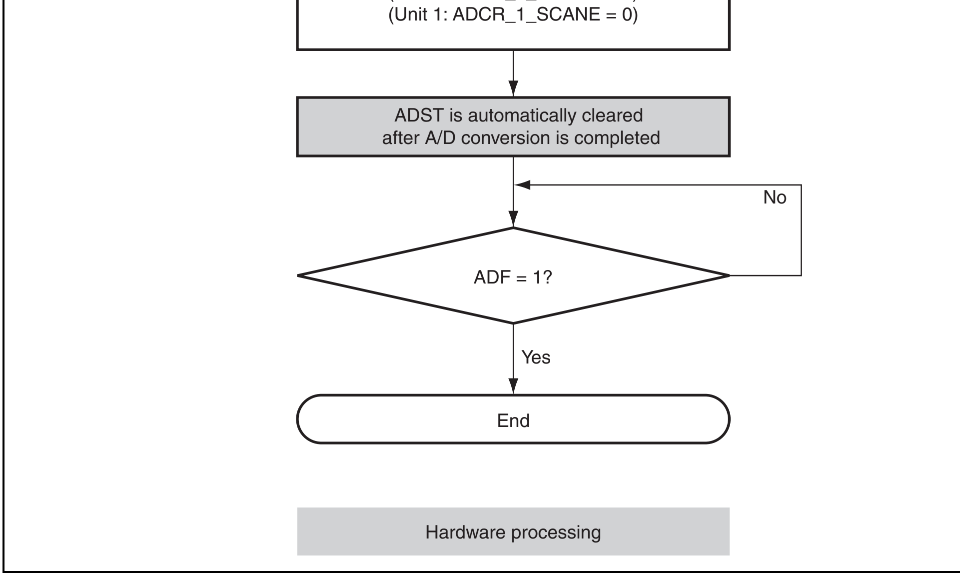


Figure 21.11 Stopping Continuous Scan Mode Activated by Software

- When the A/D Converter is Activated by an External Trigger

  Do not clear the ADST bit during A/D conversion. To stop A/D conversion, disable triggers and then rewrite the SCANE bit to change modes from scan mode to single This stops A/D conversion without clearing the ADST bit by software.

  However, after rewriting the SCANE bit, it may take up to 1.5-channel A/D convers stop A/D conversion and set the A/D end flag (ADF) to 1. Moreover, the ADDR val A/D conversion is completed should not be used.

  For details of settings, see figure 21.12.

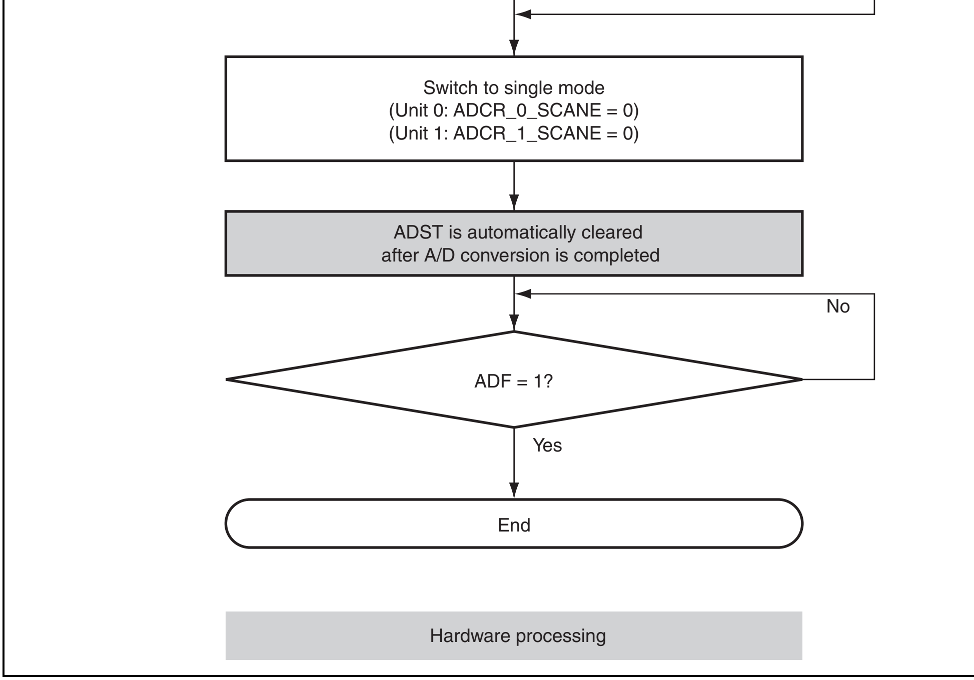


Figure 21.12 Stopping Continuous Scan Mode Activated by External Trigg

21.7.4 Notes in System Clock Mode

1. For board design, see section 21.7.8, Notes on Board Design.
2. In system clock mode, operate all units with the system clock (Iϕ). The system clock a peripheral module clock should not be used together.

and the signal source impedance is ignored. However, since a low-pass filter effect is ob
this case, it may not be possible to follow an analog signal with a large differential coeff
(e.g., 5 mV/μs or greater) (see figure 21.13). When converting a high-speed analog signa
conversion in scan mode, a low-impedance buffer should be inserted.

Notes: 1. Unit 0: The full-spec emulator (E6000H) should not be used, but the on-chip
(E10A-USB) is usable.
2. Unit 1: Access to the full-spec emulator (E6000H) is prohibited, but the on-c
emulator (E10A-USB) is usable.

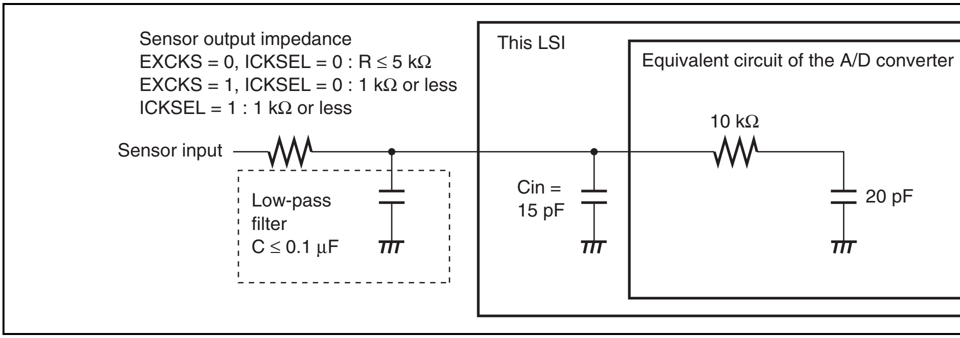


Figure 21.13 Example of Analog Input Circuit

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 $AV_{SS} \leq V_{AN} \leq V_{ref}$.
- Relation between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}

  As the relationship between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}, set $AV_{CC} = V_{CC} \pm 0.3$ V and AV
  If the A/D converter is not used, set $AV_{CC} = V_{CC}$ and $AV_{SS} = V_{SS}$.
- Vref setting range

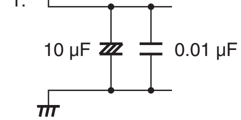
  The reference voltage at the Vref pin should be set in the range $V_{ref} \leq AV_{cc}$.

2. Lines should be connected with the analog reference power supply pin (AV_{CC}), analo supply pin (V_{ref}), and analog ground pin (AV_{SS}) with low impedance as possible.
3. The analog ground pin (AV_{SS}) should be connected at one point to a stable ground (V board.

21.7.9 Notes on Noise Countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an surge at the analog input pins (AN0 to AN7) should be connected to AV_{CC} and AV_{SS} as s figure 21.14. Also, the bypass capacitors connected to AV_{CC} and V_{ref} and the filter capaci connected to the AN0 to AN7 pins must be connected to AV_{SS}. The bypass capacitors be AV_{CC} and AV_{SS}, or V_{ref} and AV_{SS} should be placed as close to pins as possible.

If a filter capacitor is connected, the input currents at the AN0 to AN7 pins are averaged error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if current charged and discharged by the capacitance of the sample-and-hold circuit in the converter exceeds the current input via the input impedance (R_{in}), an error will arise in th input pin voltage. Careful consideration is therefore required when deciding the circuit c

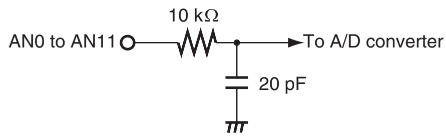


2. R_{in}: Input impedance

Figure 21.14 Example of Analog Input Protection Circuit

Table 21.8 Analog Pin Specifications

Item		Min.	Max.
Analog input capacitance		—	20
Permissible signal source impedance	EXCKS = 0, ICKSEL = 0	—	5
	EXCKS = 1, ICKSEL = 0	—	1
	ICKSEL = 1	—	1



Note: Values are reference values.

Figure 21.15 Analog Input Pin Equivalent Circuit

RENESAS

- Module stop state specifiable

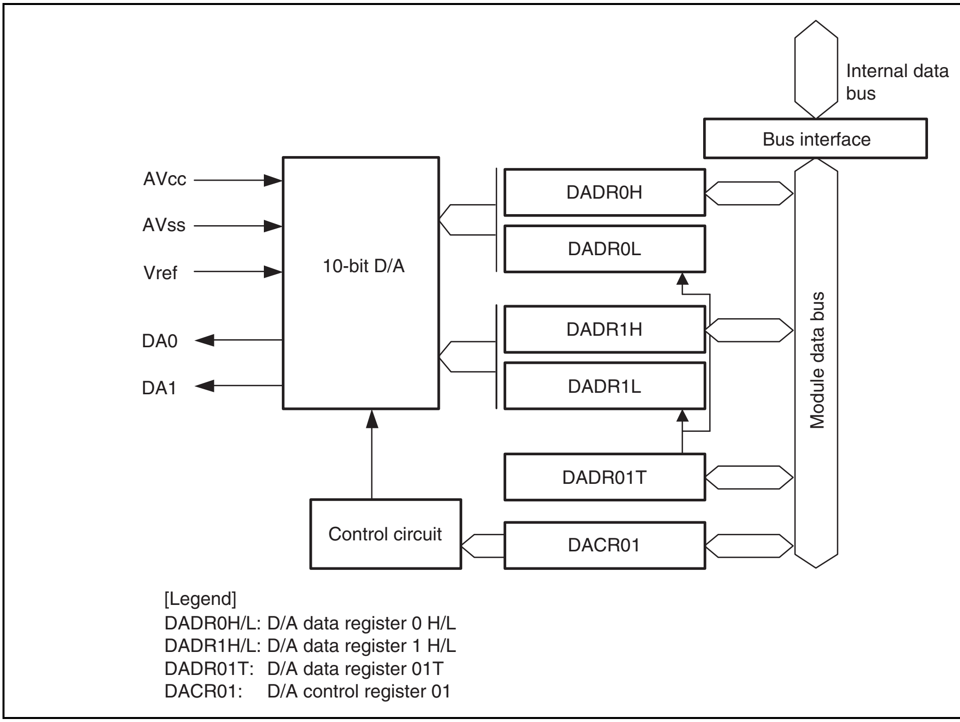


Figure 22.1 Block Diagram of the 10-Bit D/A Converter

Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

22.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 0H (DADR0H)
- D/A data register 0L (DADR0L)
- D/A data register 1H (DADR1H)
- D/A data register 1L (DADR1L)
- D/A data register 01T (DADR01T)
- D/A control register 01 (DACR01)

DADR1L are non-readable registers. Writing is accomplished by transferring values fro temporary register, DADR01T.

When the value in a DADR is to be updated, the new lower-order two bits of the value r previously have been written to DADR01T. The corresponding DADR is actually updat same time as a new value is written to DADR0H or DADR1H. The eight higher-order b reflected as written in DADR0H or DADR1H, while the two lower-order bits are update transfer from DADR01T to DADR0L or DADR1L.

22.3.2 D/A Data Registers 0H and 1H (DADR0H and DADR1H)

DADR0H and DADR1H are 8-bit readable/writable registers. When a value is written t DADR0H or DADR1H, the corresponding bits of DADR01T are simultaneously transfe DADR0L or DADR1L.

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

22.3.4 D/A Data Register 01T (DADR01T)

DADR01T is an 8-bit temporary register that is used to transfer data to DADR0L and DA
Writing is only valid for bits 7 and 6. Values written to bits 5 to 0 are ignored. In reading,
recent setting (of bits 7 and 6) and the values for the two lower-order bits held in DADR0
DADR1L can be read. The value for the two lower-order bits of DADR1 can be read in b
4. The value for the two lower-order bits of DADR0 can be read in bits 3 and 2.

Bits 1 and 0 are reserved. In reading, these bits are read as 1.

Bits 7 and 6 of DADR01T are used to store the two lower-order bits of a 10-bit value for
conversion.

Bit	7	6	5	4	3	2	1
Bit Name	DADT1	DADT0	DAD1L1	DAD1L0	DAD0L1	DAD0L0	—
Initial Value	0	0	0	0	0	0	1
R/W	R/W	R/W	R	R	R	R	R

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.

22.3.6 Usage as an 8-Bit D/A Converter

In advance of usage as an 8-bit D/A converter, fix the lower-order two bits to 0 by clearin DADT1 and DADT0 bits in DADR01T to 0. By clearing these bits in DADR01T to 0 in a the D/A converter is made to function by simply setting DADR0H or DADR1H. That is, conversion of the eight higher-order bits proceeds when the two lower-order bits remain f 0.

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
Output values are expressed by the following formulae.

- Formula for 8-bit conversion

$$\frac{\text{Contents of DADR}}{256} \times \text{Vref}$$

- Formula for 10-bit conversion

$$\frac{\text{Contents of DADR}}{1024} \times \text{Vref}$$

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.

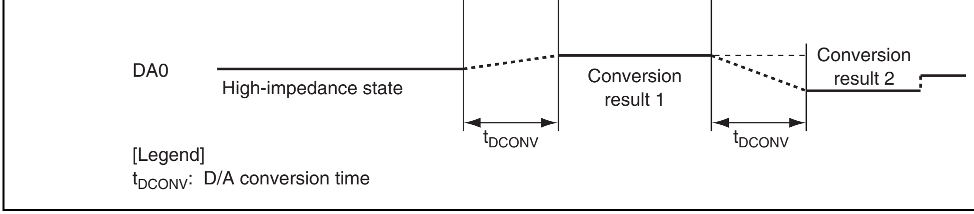


Figure 22.2 Example of D/A Converter Operation

When this LSI makes a transition to software standby mode with D/A conversion enable
D/A outputs are retained, and the flow of current from the analog power supply remains
as during D/A conversion. If the analog power-supply current has to be reduced in softw
standby mode, clear the DAOE0, DAOE1, and DAE bits to 0 to disable D/A conversion

22.5.3 Notes on Deep Software Standby Mode

When this LSI makes a transition to deep software standby mode with D/A conversion e
the D/A outputs enter high-impedance state.

22.5.4 Limitations on Emulators

The limitations described below apply to emulation of the 10-bit D/A converter.

In emulation with the full-spec emulator (E6000H), the resolution of the analog output b
eight bits, and the precision of D/A conversion is not guaranteed. Furthermore, when D/
read, the values in emulation differ from those for the actual product. Thus, as a precond
emulation with the full-spec emulator (E6000H), do not read DADR01T.

No particular limitations apply to emulation with the on-chip emulator (E10A-USB).

The above notes are summarized in table 22.3 below.

			bits are ignored)
Analog output: precision	D/A precision is guaranteed	As at left	D/A precision is n guaranteed

Flash memory version	H8SX/1652 H8SX/1655	40 Kbytes	H'FF2000 to H'
	H8SX/1652M H8SX/1655M		

	R5F61652M		(modes 1, 2, 3, 6, and 7
H8SX/1655	R5F61655 R5F61655M	512 Kbytes	H'000000 to H'07FFFF (modes 1, 2, 3, 6, and 7

- Two memory MATs

  The start addresses of two memory spaces (memory MATs) are allocated to the sam
  The mode setting in the initiation determines which memory MAT is initiated first. T
  memory MATs can be switched by using the bank-switching method after initiation.

  — User MAT initiated at a reset in user mode: 384 Kbytes/512 Kbytes

  — User boot MAT is initiated at reset in user boot mode: 16 Kbytes
- Programming/erasing interface by the download of on-chip program

  This LSI has a programming/erasing program. After downloading this program to th
  RAM, programming/erasure can be performed by setting the parameters.
- Programming/erasing time

  Programming time: 1 ms (typ.) for 128-byte simultaneous programming

  Erasing time: 600 ms (typ.) per 1 block (64 Kbytes)
- Number of programming

  The number of programming can be up to 100 times at the minimum. (1 to 100 time
  guaranteed.)
- Three on-board programming modes

  SCI boot mode: Using the on-chip SCI_4, the user MAT and user boot MAT can be
  programmed/erased. In SCI boot mode, the bit rate between the host and this LSI ca
  adjusted automatically.

  USB boot mode: Using the on-chip USB module, the user MAT can be programmed

  User programming mode: Using a desired interface, the user MAT can be programm of the flash memory (user MAT) area and the on-chip RAM.

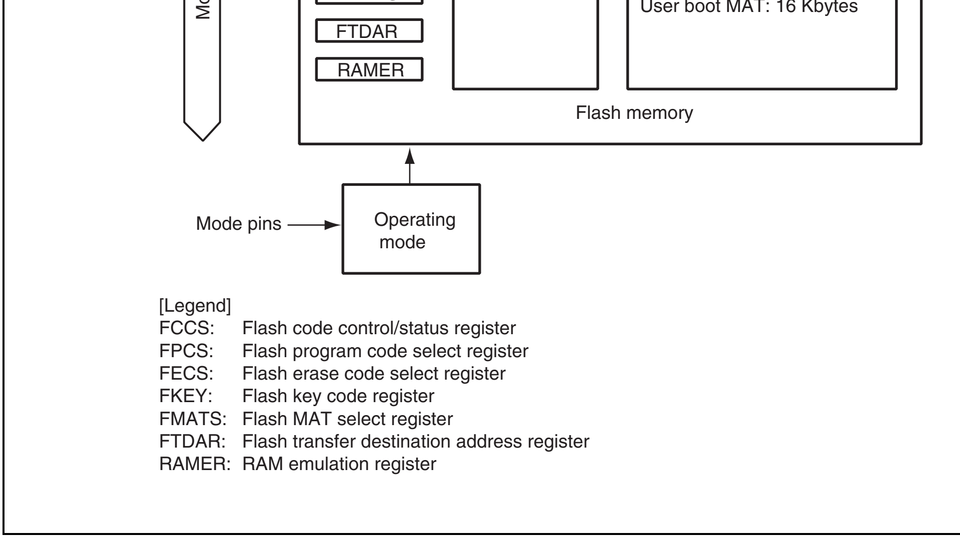


Figure 24.1 Block Diagram of Flash Memory

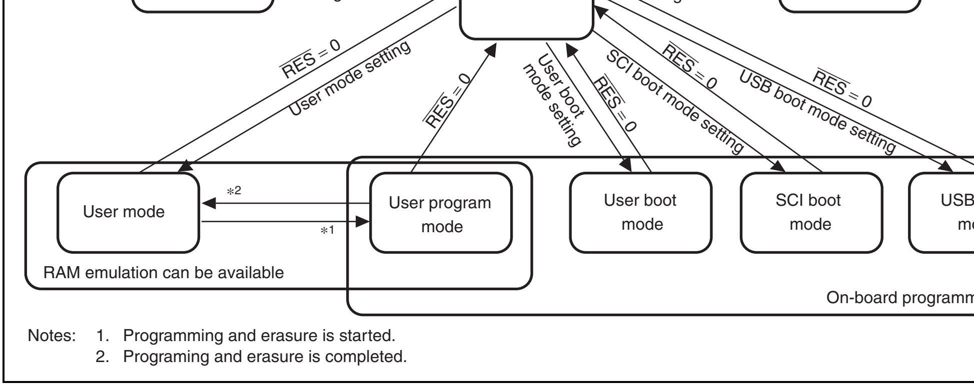


Figure 24.2 Mode Transition of Flash Memory

	MAT				MA
Programming/ erasing control	Command	Command	Programming/ erasing interface	Programming/ erasing interface	Comm
All erasure	O (Automatic)	O (Automatic)	O	O	O (Au
Block division erasure	O*[1]	O*[1]	O	O	×
Program data transfer	From host via SCI	From host via USB	From desired device via RAM	From desired device via RAM	Via pr
RAM emulation	×	×	O	O	×
Reset initiation MAT	Embedded program storage area	Embedded program storage area	User MAT	User boot MAT*[2]	—
Transition to user mode	Changing mode and reset	Changing mode and reset	Completing Programming/ erasure*[3]	Changing mode and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

2. First, the reset vector is fetched from the embedded program storage area. A flash memory related registers are checked, the reset vector is fetched from t boot MAT.

3. In this LSI, the user programming mode is defined as the period from the timi program concerning programming and erasure is started to the timing when t program is completed. For details on a program concerning programming anc see section 24.8.3, User Programming Mode.

The size of the user MAT is different from that of the user boot MAT. Addresses which e the size of the 16-Kbyte user boot MAT should not be accessed. If an attempt is made, da as an undefined value.

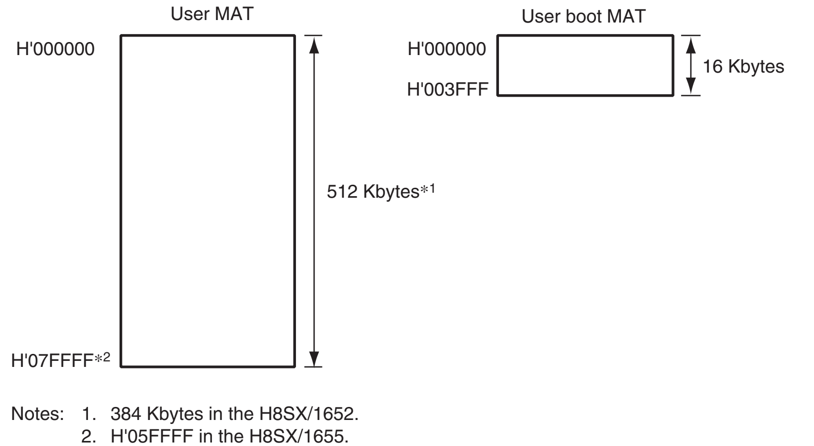


Figure 24.3 Memory MAT Configuration (H8SX/1655)

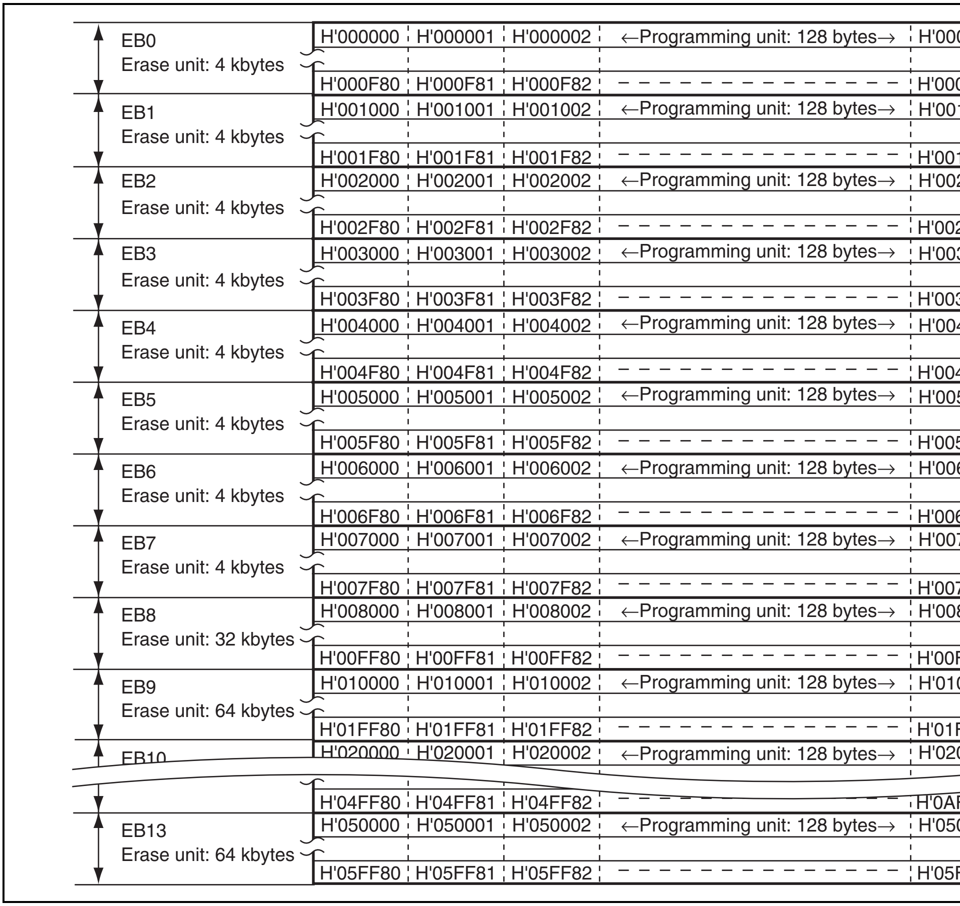


Figure 24.4 (1) User MAT Block Structure of H8SX/1652

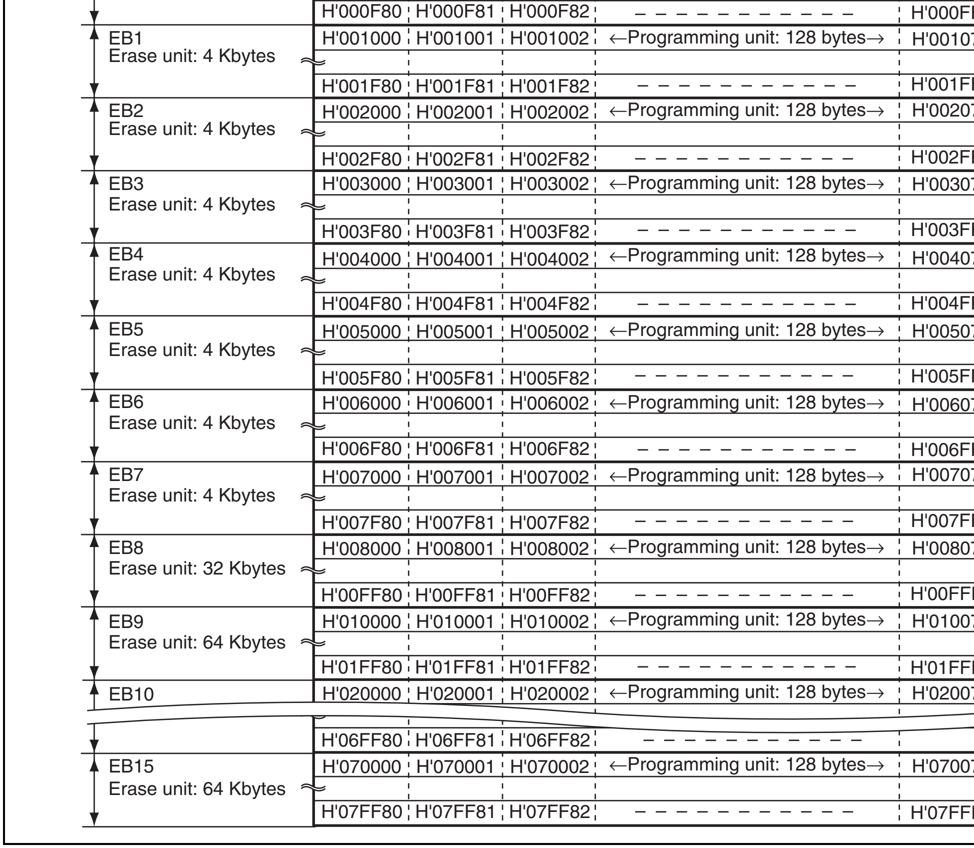


Figure 24.4 (2) User MAT Block Structure of H8SX/1655

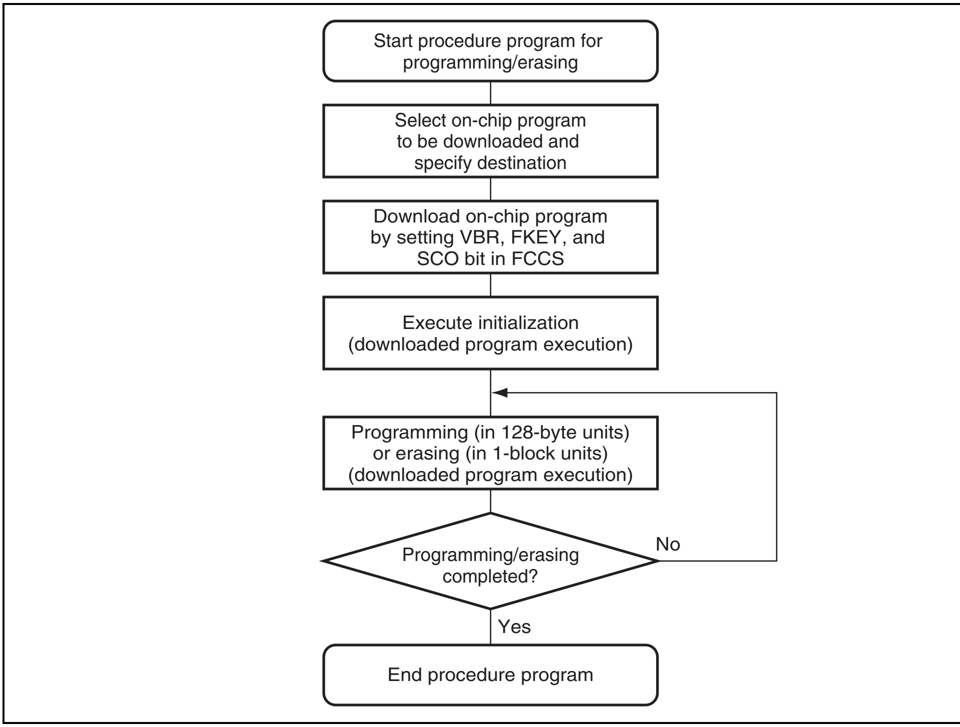


Figure 24.5 Procedure for Creating Procedure Program

(1) Selection of On-Chip Program to be Downloaded

This LSI has programming/erasing programs which can be downloaded to the on-chip R
on-chip program to be downloaded is selected by the programming/erasing interface reg
start address of the on-chip RAM where an on-chip program is downloaded is specified
flash transfer destination address register (FTDAR).

(3) Initialization of Programming/Erasure

A pulse with the specified period must be applied when programming or erasing. The spe
pulse width is made by the method in which wait loop is configured by the CPU instructi
Accordingly, the operating frequency of the CPU needs to be set before programming/era
operating frequency of the CPU is set by the programming/erasing interface parameter.

(4) Execution of Programming/Erasure

The start address of the programming destination and the program data are specified in 1
units when programming. The block to be erased is specified with the erase block numbe
erase-block units when erasing. Specifications of the start address of the programming de
program data, and erase block number are performed by the programming/erasing interfa
parameters, and the on-chip program is initiated. The on-chip program is executed by usi
JSR or BSR instruction and executing the subroutine call of the specified address in the o
RAM. The execution result is returned to the programming/erasing interface parameter.

The area to be programmed must be erased in advance when programming flash memory
interrupts are disabled during programming/erasure.

(5) When Programming/Erasure is Executed Consecutively

When processing does not end by 128-byte programming or 1-block erasure, consecutive
programming/erasure can be realized by updating the start address of the programming d
and program data, or the erase block number. Since the downloaded on-chip program is l
on-chip RAM even after programming/erasure completes, download and initialization ar
required when the same processing is executed consecutively.

PM2	Input	SCI boot/USB boot mode setting (valid when mode is selected by the MD2 to MD0 pins)
TxD4	Output	Serial transmit data output (used in SCI boot
RxD4	Input	Serial receive data input (used in SCI boot m
USD+, USD-	Input/output	USB data input/output (used in USB boot mo
VBUS	Input	USB cable connect/disconnect detection (used in USB boot mode)
PM3	Input	USB bus-power/self-power mode setting (used in USB boot mode)
PM4	Output	D+ pull-up control (used in USB boot mode)

- Flash transfer destination address register (FTDAR)

Programming/Erasing Interface Parameters:

- Download pass and fail result parameter (DPFR)
- Flash pass and fail result parameter (FPFR)
- Flash program/erase frequency parameter (FPEFEQ)
- Flash multipurpose address area parameter (FMPAR)
- Flash multipurpose data destination area parameter (FMPDR)
- Flash erase block select parameter (FEBS)

- RAM emulation register (RAMER)

There are several operating modes for accessing the flash memory. Respective operating registers, and parameters are assigned to the user MAT and user boot MAT. The correspo between operating modes and registers/parameters for use is shown in table 24.3.

Programming/ erasing interface parameters	DPFR	O	—	—	—	—	—
	FPFR	—	O	O	O	—	—
	FPEFEQ	—	O	—	—	—	—
	FMPAR	—	—	O	—	—	—
	FMPDR	—	—	O	—	—	—
	FEBS	—	—	—	O	—	—
RAM emulation	RAMER	—	—	—	—	—	O

Notes: 1. The setting is required when programming or erasing the user MAT in user b

2. The setting may be required according to the combination of initiation mode a target memory MAT.

24.7.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are 8-bit registers that can be accessed only These registers are initialized by a reset.

(1) Flash Code Control/Status Register (FCCS)

FCCS monitors errors during programming/erasing the flash memory and requests the o program to be downloaded to the on-chip RAM.

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

Note: * This is a write-only bit. This bit is always read as 0.

flash memory, the reset must be released after input period (period of $\overline{RES}$ = 0) of at least 100

0: Flash memory operates normally (Error prote invalid)

[Clearing condition]

- At a reset

1: An error occurs during programming/erasing memory (Error protection is valid)

[Setting conditions]

- When an interrupt, such as NMI, occurs du programming/erasure.
- When the flash memory is read during programming/erasure (including a vector re an instruction fetch).
- When the SLEEP instruction is executed du programming/erasure (including software st mode).
- When a bus master other than the CPU, su DMAC and DTC, obtains bus mastership du programming/erasure.

must be canceled, H'A5 must be written to FK
this operation must be executed in the on-chip
Dummy read of FCCS must be executed twice
immediately after setting this bit to 1. All interr
be disabled during download. This bit is cleare
when download is completed.

During program download initiated with this bit
particular processing which accompanies ban
switching of the program storage area is exec
Before a download request, initialize the VBR
to H'00000000. After download is completed,
contents can be changed.

0: Download of the programming/erasing prog
requested.

[Clearing condition]

- When download is completed

1: Download of the programming/erasing prog
requested.

[Setting conditions] (When all of the following
are satisfied)

- Not in RAM emulation mode (the RAMS bi
RAMER is cleared to 0)
- H'A5 is written to FKEY
- Setting of this bit is executed in the on-chip

Note: * This is a write-only bit. This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program to be downlo 0: Programming program is not selected. [Clearing condition] When transfer is completed 1: Programming program is selected.

(3) Flash Erase Code Select Register (FECS)

FECS selects the erasing program to be downloaded.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program to be downloaded 0: Erasing program is not selected. [Clearing condition] When transfer is completed 1: Erasing program is selected.

Bit	Bit Name	Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	When H'A5 is written to FKEY, writing to the S
5	K5	0	R/W	FCCS is enabled. When a value other than H'
4	K4	0	R/W	written, the SCO bit cannot be set to 1. Theref
3	K3	0	R/W	on-chip program cannot be downloaded to the RAM.
2	K2	0	R/W	Only when H'5A is written can programming/e
1	K1	0	R/W	the flash memory be executed. When a value H'5A is written, even if the programming/erasi
0	K0	0	R/W	program is executed, programming/erasure ca performed.
				H'A5: Writing to the SCO bit is enabled. (The S cannot be set to 1 when FKEY is a valu than H'A5.)
				H'5A: Programming/erasure of the flash memo enabled. (When FKEY is a value other t the software protection state is entered.
				H'00: Initial value

Bit	Bit Name	Initial Value	R/W	Description
7	MS7	0/1*	R/W	MAT Select
6	MS6	0	R/W	The memory MATs can be switched by writing to FMATS.
5	MS5	0/1*	R/W	When H'AA is written to FMATS, the user boot selected. When a value other than H'AA is writt user MAT is selected. Switch the MATs followin memory MAT switching procedure in section 24 Switching between User MAT and User Boot M user boot MAT cannot be selected by FMATS i programming mode. The user boot MAT can be selected in boot mode or programmer mode.
4	MS4	0	R/W	H'AA: The user boot MAT is selected. (The use selected when FMATS is a value other th H'AA.) (Initial value when initiated in user boot m
3	MS3	0/1*	R/W	H'00: The user MAT is selected. (Initial value when initiated in a mode exc user boot mode.)
2	MS2	0	R/W	
1	MS1	0/1*	R/W	
0	MS0	0	R/W	

Note: * This bit is set to 1 in user boot mode, otherwise cleared to 0.

Bit	Bit Name	Value	R/W	Description
7	TDER	0	R/W	Transfer Destination Address Setting Error This bit is set to 1 when an error has occurred the start address specified by bits TDA6 to TD A start address error is determined by whethe set in bits TDA6 to TDA0 is within the range o H'02 when download is executed by setting th in FCCS to 1. Make sure that this bit is cleare before setting the SCO bit to 1 and the value s by bits TDA6 to TDA0 should be within the ra H'00 to H'02. 0: The value specified by bits TDA6 to TDA0 i the range. 1: The value specified by bits TDA6 to TDA0 i H'03 and H'FF and download has stopped.
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the on-chip RAM start address of the download destination. A value between H'00 a and up to 4 Kbytes can be specified as the sta of the on-chip RAM.
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	H'00: H'FF9000 is specified as the start addre
1	TDA1	0	R/W	H'01: H'FFA000 is specified as the start addre
0	TDA0	0	R/W	H'02: H'FFB000 is specified as the start addre
				H'03 to H'7F: Setting prohibited. (Specifying a value from H'03 to the TDER bit to 1 and stops dow the on-chip program.)

processing result is written in R0. The programming/erasing interface parameters are usec download control, initialization before programming or erasing, programming, and erasin 24.4 shows the usable parameters and target modes. The meaning of the bits in the flash p fail result parameter (FPFR) varies in initialization, programming, and erasure.

Table 24.4 Parameters and Target Modes

Parameter	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allo
DPFR	O	—	—	—	R/W	Undefined	On-c
FPFR	O	O	O	O	R/W	Undefined	R0L
FPEFEQ	—	O	—	—	R/W	Undefined	ER0
FMPAR	—	—	O	—	R/W	Undefined	ER1
FMPDR	—	—	O	—	R/W	Undefined	ER0
FEBS	—	—	—	O	R/W	Undefined	ER0

Note: * A single byte of the start address of the on-chip RAM specified by FTDAR

Download Control: The on-chip program is automatically downloaded by setting the SC FCCS to 1. The on-chip RAM area to download the on-chip program is the 4-Kbyte area from the start address specified by FTDAR. Download is set by the programming/erasing registers, and the download pass and fail result parameter (DPFR) indicates the return val register ER1. This parameter is called the flash multipurpose address area parameter (FM

The program data is always in 128-byte units. When the program data does not satisfy 1 128-byte program data is prepared by filling the dummy code (H'FF). The boundary of t address of the programming destination on the user MAT is aligned at an address where eight bits (A7 to A0) are H'00 or H'80.

The program data for the user MAT must be prepared in consecutive areas. The program must be in a consecutive space which can be accessed using the MOV.B instruction of th and is not in the flash memory space.

The start address of the area that stores the data to be written in the user MAT must be s general register ER0. This parameter is called the flash multipurpose data destination are parameter (FMPDR).

For details on the programming procedure, see section 24.8.3, User Programming Mode

Erasure: When the flash memory is erased, the erase block number on the user MAT m passed to the erasing program which is downloaded.

The erase block number on the user MAT must be set in general register ER0. This para called the flash erase block select parameter (FEBS).

One block is selected from the block numbers of 0 to 19 as the erase block number.

For details on the erasing procedure, see section 24.8.3, User Programming Mode.

7 to 3	—	—	—	Unused These bits return 0.
2	SS	—	R/W	Source Select Error Detect Only one type can be specified for the on-chip p which can be downloaded. When the program t downloaded is not selected, more than two type programs are selected, or a program which is n mapped is selected, an error occurs. 0: Download program selection is normal 1: Download program selection is abnormal
1	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'A5) and returns the 0: FKEY setting is normal (H'A5) 1: FKEY setting is abnormal (value other than H
0	SF	—	R/W	Success/Fail Returns the download result. Reads back the p downloaded to the on-chip RAM and determine whether it has been transferred to the on-chip R 0: Download of the program has ended normall error) 1: Download of the program has ended abnorm (error occurs)

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused These bits return 0.
1	FQ	—	R/W	Frequency Error Detect Compares the specified CPU operating freque the operating frequencies supported by this LS returns the result. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Returns the initialization result. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error oc

6	MD	—	R/W	Programming Mode Related Setting Error Dete Detects the error protection state and returns th When the error protection state is entered, this to 1. Whether the error protection state is enter can be confirmed with the FLER bit in FCCS. F conditions to enter the error protection state, se 24.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming can performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect Writes 1 to this bit when the specified data coul written because the user MAT was not erased. is set to 1, there is a high possibility that the us has been written to partially. In this case, after r the error factor, erase the user MAT. If FMATS H'AA and the user boot MAT is selected, an err when programming is performed. In this case, b user MAT and user boot MAT have not been w Programming the user boot MAT should be per boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally (progra result is not guaranteed)

				When an address not in the flash memory are specified as the start address of the storage d for the program data, an error occurs. 0: Setting of the start address of the storage d for the program data is normal 1: Setting of the start address of the storage d for the program data is abnormal
1	WA	—	R/W	Write Address Error Detect When the following items are specified as the address of the programming destination, an e occurs. • An area other than flash memory • The specified address is not aligned with t byte boundary (lower eight bits of the addr other than H'00 and H'80) 0: Setting of the start address of the programm destination is normal 1: Setting of the start address of the programm destination is abnormal
0	SF	—	R/W	Success/Fail Returns the programming result. 0: Programming has ended normally (no error 1: Programming has ended abnormally (error

6	MD	—	R/W	Erasure Mode Related Setting Error Detect Detects the error protection state and returns th When the error protection state is entered, this to 1. Whether the error protection state is enter can be confirmed with the FLER bit in FCCS. F conditions to enter the error protection state, se 24.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming can performed (FLER = 1)
5	EE	—	R/W	Erasure Execution Error Detect Returns 1 when the user MAT could not be era when the flash memory related register settings partially changed. If this bit is set to 1, there is a possibility that the user MAT has been erased p In this case, after removing the error factor, era user MAT. If FMATS is set to H'AA and the use MAT is selected, an error occurs when erasure performed. In this case, both the user MAT and boot MAT have not been erased. Erasing of the boot MAT should be performed in boot mode o programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally

				0: Setting of erase block number is normal 1: Setting of erase block number is abnormal
2, 1	—	—	—	Unused These bits return 0.
0	SF	—	R/W	Success/Fail Indicates the erasure result. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occur

(3) Flash Program/Erase Frequency Parameter (FPEFEQ: General Register ER

FPEFEQ sets the operating frequency of the CPU. The operating frequency available in ranges from 8 MHz to 50 MHz.

Bit	31	30	29	28	27	26	25
Bit Name	—	—	—	—	—	—	—

Bit	23	22	21	20	19	18	17
Bit Name	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9
Bit Name	F15	F14	F13	F12	F11	F10	F9

Bit	7	6	5	4	3	2	1
Bit Name	F7	F6	F5	F4	F3	F2	F1

be rounded in a number of three decimal pl
be shown in a number of two decimal place

2. The value multiplied by 100 is converted to binary digit and is written to FPEFEQ (gene register ER0).

For example, when the operating frequency of is 35.000 MHz, the value is as follows:

1. The number of three decimal places of 35.0 rounded.
2. The formula of 35.00 × 100 = 3500 is conve the binary digit and B'0000 1101 1010 1100 (H'0DAC) is set to ER0.

Bit	23	22	21	20	19	18	17
Bit Name	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17

Bit	15	14	13	12	11	10	9
Bit Name	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9

Bit	7	6	5	4	3	2	1
Bit Name	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	These bits store the start address of the progr destination on the user MAT. Consecutive 128 programming is executed starting from the spe start address of the user MAT. Therefore, the start address of the programming destination 128-byte boundary, and MOA6 to MOA0 are a cleared to 0.

Bit	23	22	21	20	19	18	17	
Bit Name	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MO

Bit	15	14	13	12	11	10	9	
Bit Name	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	M

Bit	7	6	5	4	3	2	1	
Bit Name	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	M

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	These bits store the start address of the area w stores the program data for the user MAT. Con 128-byte data is programmed to the user MAT from the specified start address.

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

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

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	R	Reserved These are read-only bits and cannot be modifie
3	RAMS	0	R/W	RAM Select Selects the function which emulates the flash m using the on-chip RAM. 0: Disables RAM emulation function 1: Enables RAM emulation function (all blocks user MAT are protected against programmin erasing)
2 1 0	RAM2 RAM1 RAM0	0 0 0	R/W R/W R/W	Flash Memory Area Select These bits select the user MAT area overlaid w on-chip RAM when RAMS = 1. The following ar correspond to the 4-Kbyte erase blocks. 000: H'000000 to H'000FFF (EB0) 001: H'001000 to H'001FFF (EB1) 010: H'002000 to H'002FFF (EB2) 011: H'003000 to H'003FFF (EB3) 100: H'004000 to H'004FFF (EB4) 101: H'005000 to H'005FFF (EB5) 110: H'006000 to H'006FFF (EB6) 111: H'007000 to H'007FFF (EB7)

Table 24.5 On-Board Programming Mode Setting

Mode Setting	EMLE	MD2	MD1	MD0	PM
User boot mode	0	0	0	1	—
SCI boot mode	0	0	1	0	0
USB boot mode	0	0	1	0	1
User programming mode	0	1	1	0	—
	0	1	1	1	—

24.8.1 SCI Boot Mode

SCI boot mode executes programming/erasing of the user MAT by means of the control and program data transmitted from the externally connected host via the on-chip SCI_4.

In SCI boot mode, the tool for transmitting the control command and program data, and program data must be prepared in the host. The serial communication mode is set to asy mode. The system configuration in SCI boot mode is shown in figure 24.6. Interrupts ar in SCI boot mode. Configure the user system so that interrupts do not occur.

Figure 24.6 System Configuration in SCI Boot Mode

(1) Serial Interface Setting by Host

The SCI_4 is set to asynchronous mode, and the serial transmit/receive format is set to 8-
one stop bit, and no parity.

When a transition to SCI boot mode is made, the boot program embedded in this LSI is in

When the boot program is initiated, this LSI measures the low period of asynchronous se
communication data (H'00) transmitted consecutively by the host, calculates the bit rate,
adjusts the bit rate of the SCI_4 to match that of the host.

When bit rate adjustment is completed, this LSI transmits 1 byte of H'00 to the host as th
adjustment end sign. When the host receives this bit adjustment end sign normally, it tran
byte of H'55 to this LSI. When reception is not executed normally, initiate boot mode aga
bit rate may not be adjusted within the allowable range depending on the combination of
rate of the host and the system clock frequency of this LSI. Therefore, the transfer bit rat
host and the system clock frequency of this LSI must be as shown in table 24.6.

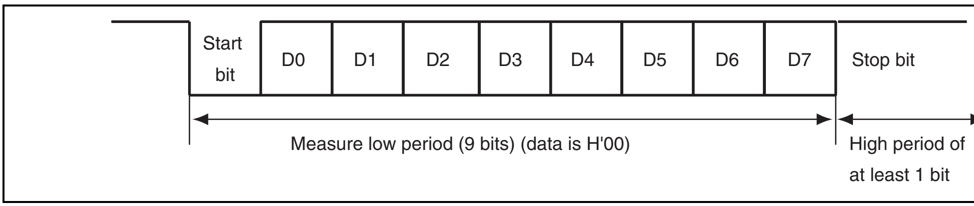


Figure 24.7 Automatic-Bit-Rate Adjustment Operation

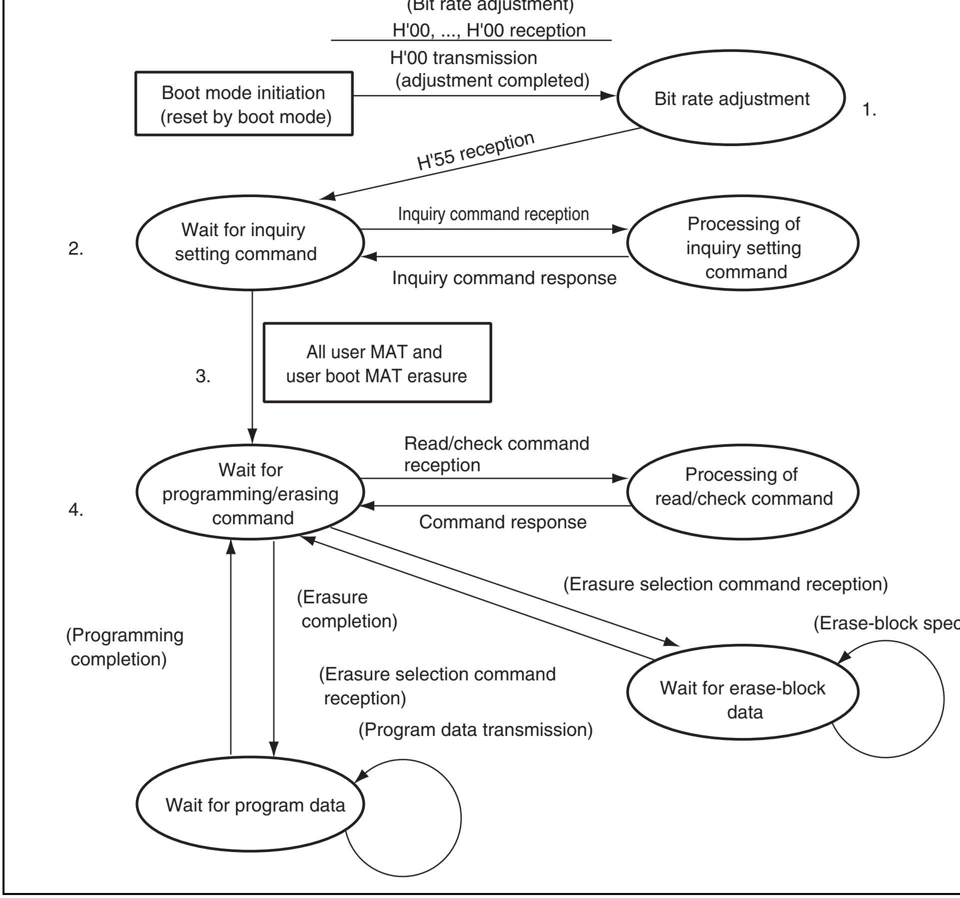


Figure 24.8 SCI Boot Mode State Transition Diagram

waiting for erase block data is entered. The erase block number must be transmitted a
erasing command is transmitted. When the erasure is finished, the erase block numbe
set to H'FF and transmitted. Then the state of waiting for erase block data is returned
state of waiting for programming/erasing command. Erasure must be executed when t
specified block is programmed without a reset start after programming is executed in
mode. When programming can be executed by only one operation, all blocks are erase
entering the state of waiting for programming/erasing command or another command
this case, the erasing operation is not required. The commands other than the
programming/erasing command perform sum check, blank check (erasure check), and
read of the user MAT/user boot MAT and acquisition of current status information.

Memory read of the user MAT/user boot MAT can only read the data programmed after a
MAT/user boot MAT has automatically been erased. No other data can be read.

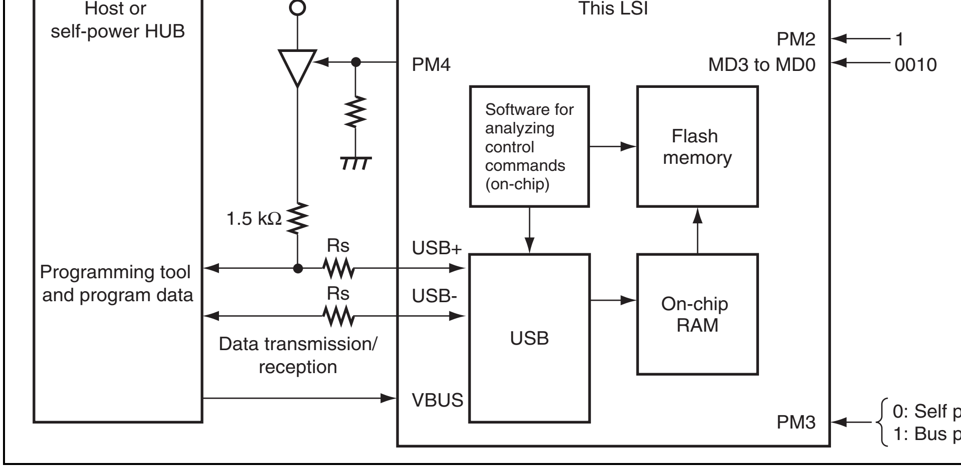


Figure 24.9 System Configuration in USB Boot Mode

	For bus power mode (PM3 = 1)	500

Endpoint configuration	EP0 Control (in out) 8 bytes
	Configuration 1
	└ InterfaceNumber0
	  └ AlternateSetting0
	    ├ EP1 Bulk (out) 64 bytes
	    └ EP2 Bulk (in) 64 bytes	

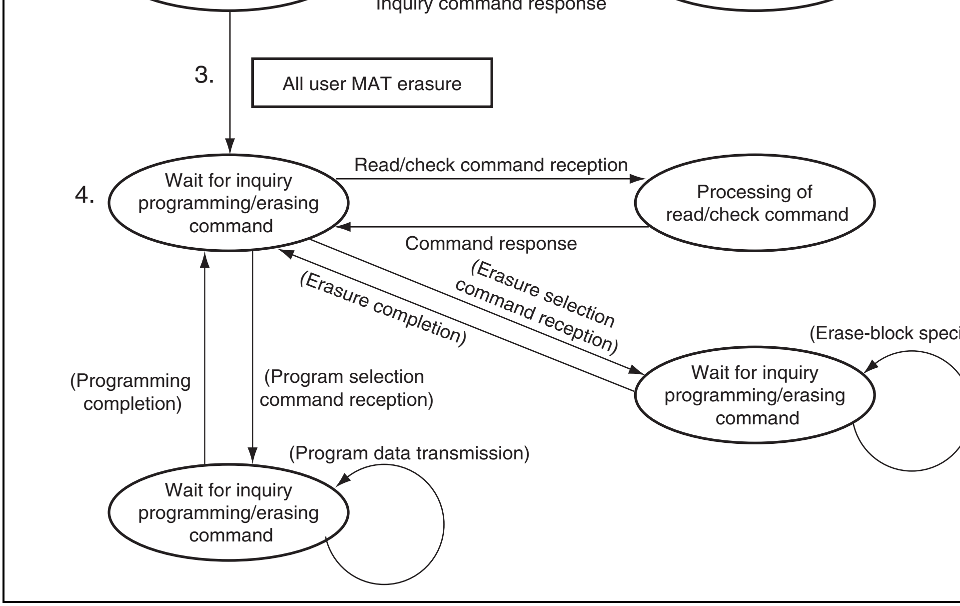
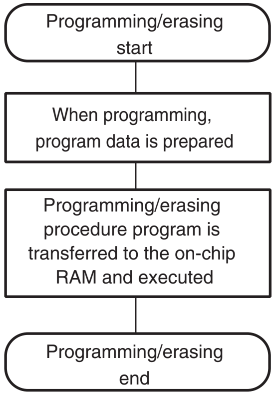


Figure 24.10 USB Boot Mode State Transition Diagram

check), and memory read of the user MAT, and acquiring the current status informatio

(3) Notes on USB Boot Mode Execution

- The clock of 48 MHz needs to be supplied to the USB module. Set the external clock frequency and clock pulse generator so as to supply 48 MHz as the clock for the USB For details, refer to section 26, Clock Pulse Generator.
- Use the PM4 pin for the D+ pull-up control connection.
- For the stable supply of the power during the flash memory programming and erasing cable should not be connected via the bus powered HUB.
- If the bus powered HUB is disconnected during the flash memory programming and e permanent damage to the LSI may result.
- If the USB bus in the bus power mode enters the suspend mode, this does not make th transition to the software standby mode in the power-down state.



1. Exit RAM emulation mode beforehand. Download is not allowe in emulation mode.

2. When the program data is adjusted in emulation mode, select the download destination specified by FTDAR carefully. Make sure that the download area does not overlap the emulation area.

3. Programming/erasing is executed only in the on-chip RAM.

4. After programming/erasing is finished, protect the flash memor by the hardware protection.

Figure 24.11 Programming/Erasing Flow

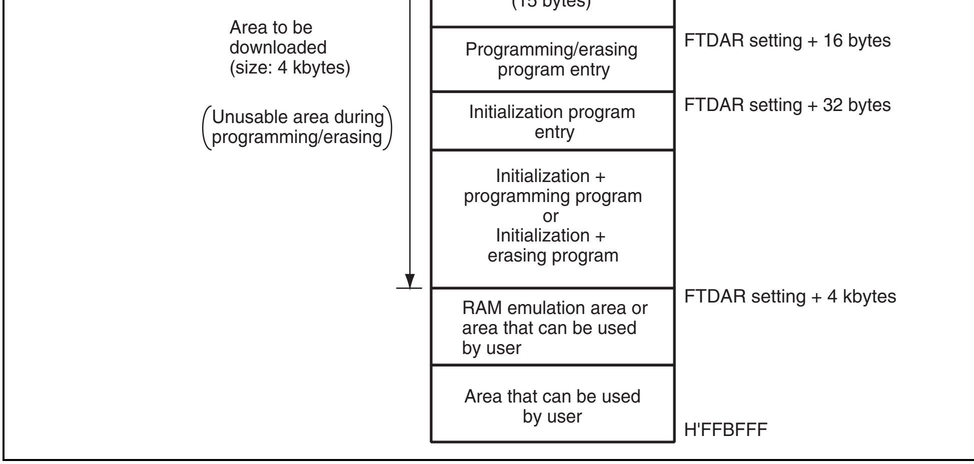


Figure 24.12 RAM Map when Programming/Erasure is Executed

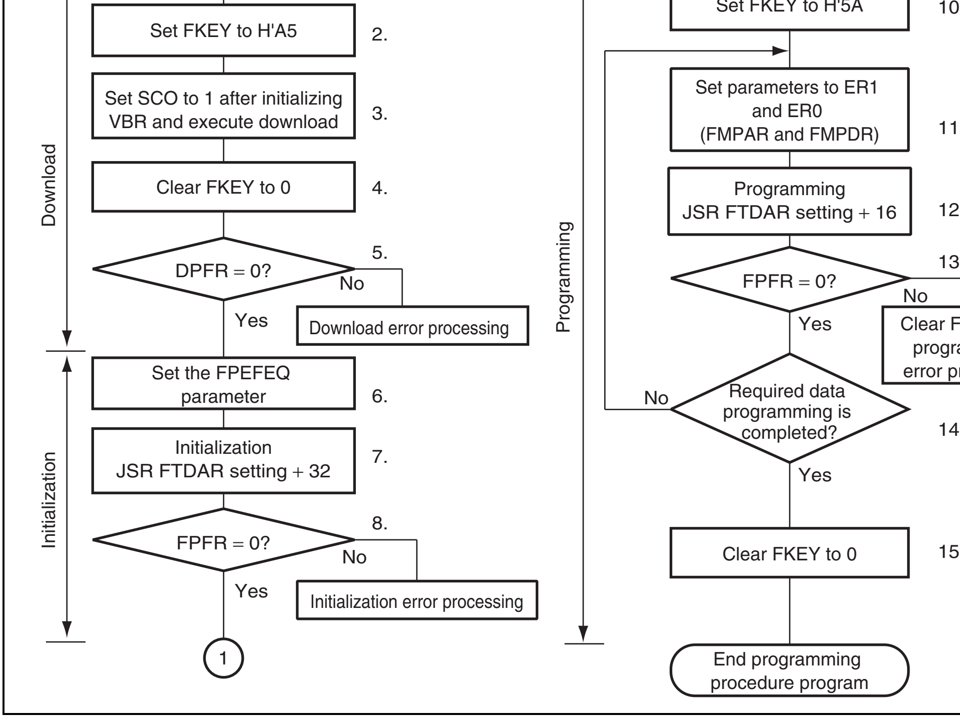


Figure 24.13 Programming Procedure in User Programming Mode

H'FF, the program processing time can be shortened.

1. Select the on-chip program to be downloaded and the download destination. When the
   bit in FPCS is set to 1, the programming program is selected. Several programming/er
   programs cannot be selected at one time. If several programs are selected, a download
   returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the d
   destination is specified by FTDAR.
2. Write H'A5 in FKEY. If H'A5 is not written to FKEY, the SCO bit in FCCS cannot be
   to request download of the on-chip program.
3. After initializing VBR to H'00000000, set the SCO bit to 1 to execute download. To s
   SCO bit to 1, all of the following conditions must be satisfied.
   - RAM emulation mode has been canceled.
   - H'A5 is written to FKEY.
   - Setting the SCO bit is executed in the on-chip RAM.

   When the SCO bit is set to 1, download is started automatically. Since the SCO bit is
   to 0 when the procedure program is resumed, the SCO bit cannot be confirmed to be 1
   procedure program. The download result can be confirmed by the return value of the
   parameter. To prevent incorrect decision, before setting the SCO bit to 1, set one byte
   on-chip RAM start address specified by FTDAR, which becomes the DPFR paramete
   value other than the return value (e.g. H'FF). Since particular processing that is accom
   by bank switching as described below is performed when download is executed, initia
   VBR contents to H'00000000. Dummy read of FCCS must be performed twice immed
   after the SCO bit is set to 1.
   - The user-MAT space is switched to the on-chip program storage area.
   - After the program to be downloaded and the on-chip RAM start address specified
     FTDAR are checked, they are transferred to the on-chip RAM.
   - FPCS, FECS, and the SCO bit in FCCS are cleared to 0.

— If access to the flash memory is requested by the DMAC or DTC during downlo
operation cannot be guaranteed. Make sure that an access request by the DMAC
not generated.

4. FKEY is cleared to H'00 for protection.
5. The download result must be confirmed by the value of the DPFR parameter. Check
of the DPFR parameter (one byte of start address of the download destination specifi
FTDAR). If the value of the DPFR parameter is H'00, download has been performed
If the value is not H'00, the source that caused download to fail can be investigated b
description below.
   — If the value of the DPFR parameter is the same as that before downloading, the s
   the start address of the download destination in FTDAR may be abnormal. In thi
   confirm the setting of the TDER bit in FTDAR.
   — If the value of the DPFR parameter is different from that before downloading, ch
   bit or FK bit in the DPFR parameter to confirm the download program selection
   setting, respectively.
6. The operating frequency of the CPU is set in the FPEFEQ parameter for initializatio
settable operating frequency of the FPEFEQ parameter ranges from 8 to 50 MHz. W
frequency is set otherwise, an error is returned to the FPFR parameter of the initializ
program and initialization is not performed. For details on setting the frequency, see
24.7.2 (3), Flash Program/Erase Frequency Parameter (FPEFEQ: General Register E
CPU).

— Since the stack area is used in the initialization program, a stack area of 128 bytes
maximum must be allocated in RAM.

— Interrupts can be accepted during execution of the initialization program. Make su
program storage area and stack area in the on-chip RAM and register values are no
overwritten.

8. The return value in the initialization program, the FPFR parameter is determined.

9. All interrupts and the use of a bus master other than the CPU are disabled during programming/erasure. The specified voltage is applied for the specified time when programming or erasing. If interrupts occur or the bus mastership is moved to other th CPU during programming/erasure, causing a voltage exceeding the specifications to b applied, the flash memory may be damaged. Therefore, interrupts are disabled by sett (I bit) in the condition code register (CCR) to B'1 in interrupt control mode 0 and by s bits 2 to 0 (I2 to I0 bits) in the extend register (EXR) to B'111 in interrupt control mod Accordingly, interrupts other than NMI are held and not executed. Configure the user so that NMI interrupts do not occur. The interrupts that are held must be executed afte programming completes. When the bus mastership is moved to other than the CPU, su the DMAC or DTC, the error protection state is entered. Therefore, make sure the DM not acquire the bus.

10. FKEY must be set to H'5A and the user MAT must be prepared for programming.

11. The parameters required for programming are set. The start address of the programmi destination on the user MAT (FMPAR parameter) is set in general register ER1. The s address of the program data storage area (FMPDR parameter) is set in general register

— Example of FMPAR parameter setting: When an address other than one in the use area is specified for the start address of the programming destination, even if the programming program is executed, programming is not executed and an error is re the FPFR parameter. Since the program data for one programming operation is 12 the lower eight bits of the address must be H'00 or H'80 to be aligned with the 128 boundary.

The general registers other than ER0 and ER1 are held in the programming progr

— R0L is a return value of the FPFR parameter.

— Since the stack area is used in the programming program, a stack area of 128 byt maximum must be allocated in RAM.

13. The return value in the programming program, the FPFR parameter is determined.

14. Determine whether programming of the necessary data has finished. If more than 12 data are to be programmed, update the FMPAR and FMPDR parameters in 128-byte repeat steps 11 to 14. Increment the programming destination address by 128 bytes a the programming data pointer correctly. If an address which has already been progra written to again, not only will a programming error occur, but also flash memory wil damaged.

15. After programming finishes, clear FKEY and specify software protection. If this LSI restarted by a reset immediately after programming has finished, secure the reset inp (period of $\overline{RES}$ = 0) of at least 100 µs.

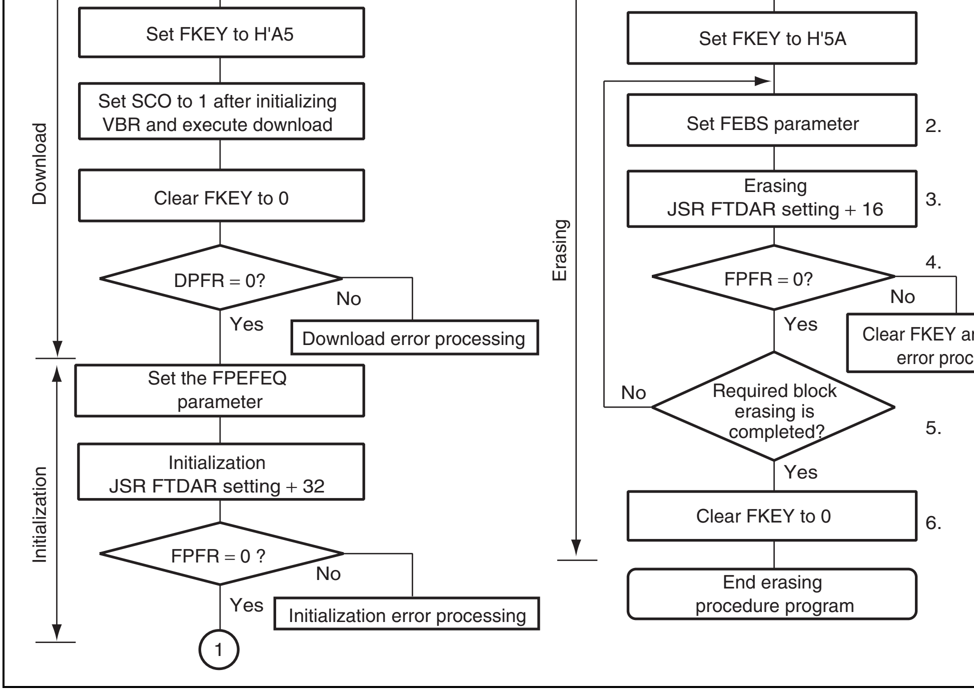


Figure 24.14 Erasing Procedure in User Programming Mode

bit in FECS is set to 1, the programming program is selected. Several programming/
programs cannot be selected at one time. If several programs are selected, a downloa
returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the
destination is specified by FTDAR.

For the procedures to be carried out after setting FKEY, see section 24.8.3 (2), Progr
Procedure in User Programming Mode.

2. Set the FEBS parameter necessary for erasure. Set the erase block number (FEBS pa
of the user MAT in general register ER0. If a value other than an erase block number
user MAT is set, no block is erased even though the erasing program is executed, and
is returned to the FPFR parameter.

3. Erasure is executed. Similar to as in programming, the entry point of the erasing prog
the address which is 16 bytes after #DLTOP (start address of the download destinatio
specified by FTDAR). Call the subroutine to execute erasure by using the following

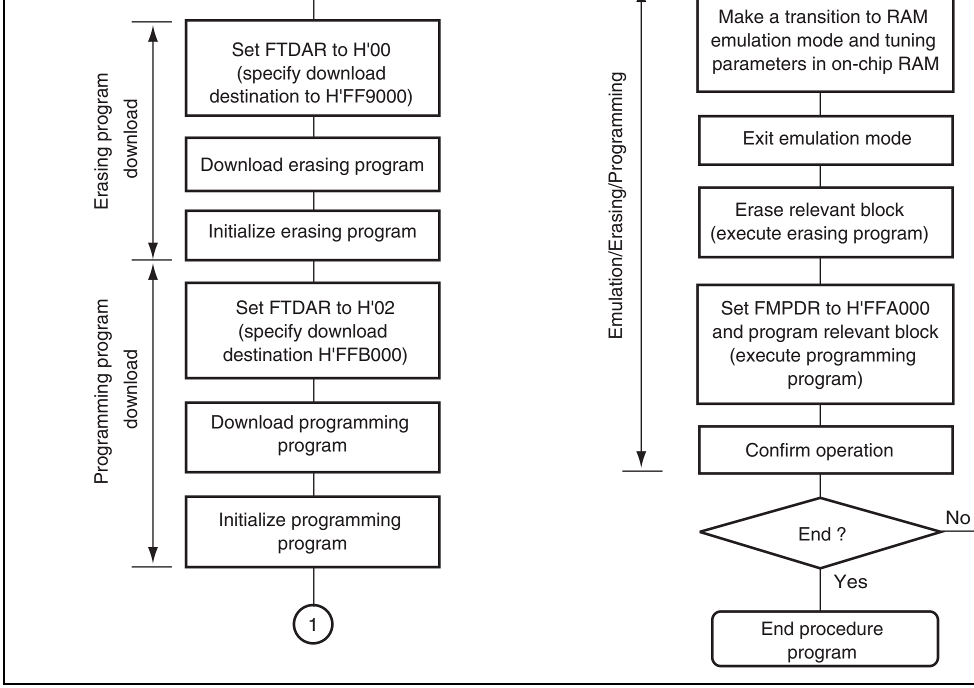
```
MOV.L #DLTOP+16, ER2      ; Set entry address to ER2
JSR   @ER2                ; Call erasing routine
NOP
```

- The general registers other than ER0 and ER1 are held in the erasing program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the erasing program, a stack area of 128 bytes at th
maximum must be allocated in RAM.

4. The return value in the erasing program, the FPFR parameter is determined.

5. Determine whether erasure of the necessary blocks has finished. If more than one blo
be erased, update the FEBS parameter and repeat steps 2 to 5.

6. After erasure completes, clear FKEY and specify software protection. If this LSI is r
a reset immediately after erasure has finished, secure the reset input period (period o
of at least 100 μs.



24.15 Repeating Procedure of Erasing, Programming, and RAM Emulation in User Programming Mode

Initialization must be executed for both entry addresses: #DLTOP (start address of d
destination for erasing program) + 32 bytes, and #DLTOP (start address of downloa
destination for programming program) + 32 bytes.

24.8.4 User Boot Mode

Branching to a programming/erasing program prepared by the user enables user boot mo
is a user-defined boot mode to be used.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasure
user boot MAT is only enabled in boot mode or programmer mode.

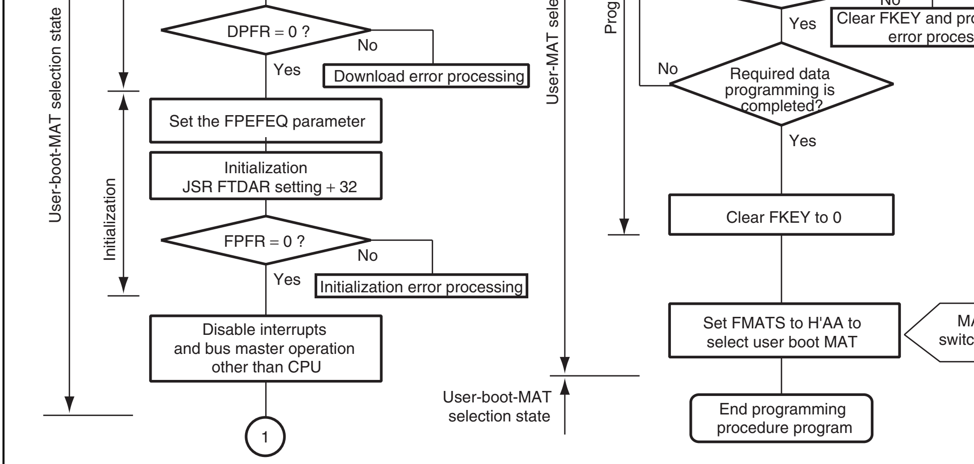
(1) Initiation in User Boot Mode

When the reset start is executed with the mode pins set to user boot mode, the built-in ch
routine runs and checks the user MAT and user boot MAT states. While the check routi
running, NMI and all other interrupts cannot be accepted. Next, processing starts from t
execution start address of the reset vector in the user boot MAT. At this point, the user b
is selected (FMATS = H'AA) as the execution memory MAT.

(2) User MAT Programming in User Boot Mode

Figure 24.16 shows the procedure for programming the user MAT in user boot mode.

The difference between the programming procedures in user programming mode and us
mode is the memory MAT switching as shown in figure 24.16. For programming the us
user boot mode, additional processing made by setting FMATS is required: switching fr
user boot MAT to the user MAT, and switching back to the user boot MAT after progra
completes.



Note: The MAT must be switched by FMATS to perform the programming error processing in the user boot MAT

Figure 24.16 Procedure for Programming User MAT in User Boot Mode

description in section 24.11, Switching between User MAT and User Boot MAT.

Except for memory MAT switching, the programming procedure is the same as that in u
programming mode.

The area that can be executed in the steps of the procedure program (on-chip RAM, user
and external space) is shown in section 24.8.5, On-Chip Program and Storable Area for
Data.

(3) User MAT Erasing in User Boot Mode

Figure 24.17 shows the procedure for erasing the user MAT in user boot mode.

The difference between the erasing procedures in user programming mode and user boo
the memory MAT switching as shown in figure 24.17. For erasing the user MAT in user
mode, additional processing made by setting FMATS is required: switching from the us
MAT to the user MAT, and switching back to the user boot MAT after erasing complete

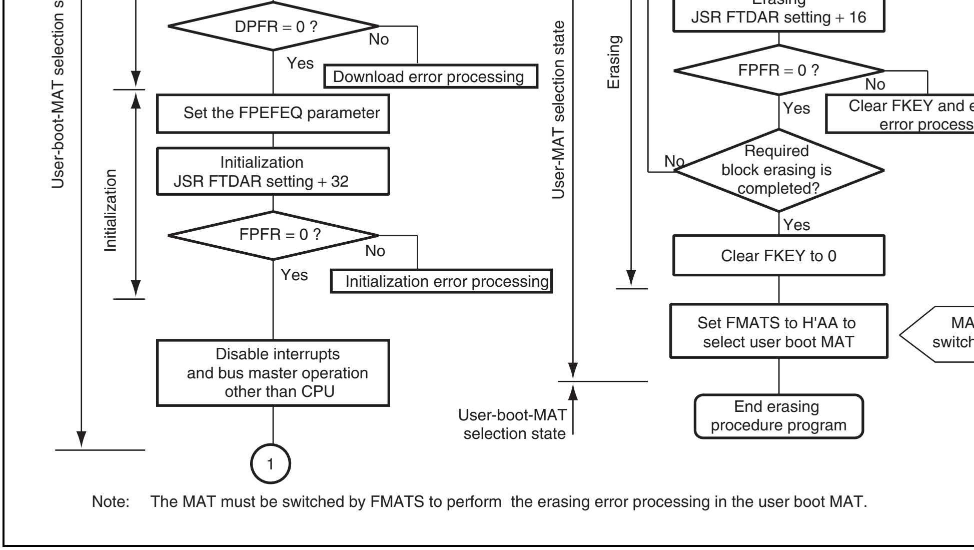


Figure 24.17 Procedure for Erasing User MAT in User Boot Mode

Data.

24.8.5 On-Chip Program and Storable Area for Program Data

In the descriptions in this manual, the on-chip programs and program data storage areas assumed to be in the on-chip RAM. However, they can be executed from part of the flas which is not to be programmed or erased as long as the following conditions are satisfie

- The on-chip program is downloaded to and executed in the on-chip RAM specified b FTDAR. Therefore, this on-chip RAM area is not available for use.
- Since the on-chip program uses a stack area, allocate 128 bytes at the maximum as a area.
- Download requested by setting the SCO bit in FCCS to 1 should be executed from th RAM because it will require switching of the memory MATs.
- In an operating mode in which the external address space is not accessible, such as s mode, the required procedure programs, NMI handling vector table, and NMI handli should be transferred to the on-chip RAM before programming/erasure starts (downl is determined).
- The flash memory is not accessible during programming/erasure. Programming/eras executed by the program downloaded to the on-chip RAM. Therefore, the procedure that initiates operation, the NMI handling vector table, and the NMI handling routin stored in the on-chip RAM other than the flash memory.
- After programming/erasure starts, access to the flash memory should be inhibited un is cleared. The reset input state (period of $\overline{\text{RES}}$ = 0) must be set to at least 100 µs wh operating mode is changed and the reset start executed on completion of programmi Transitions to the reset state are inhibited during programming/erasure. When the res is input, a reset input state (period of $\overline{\text{RES}}$ = 0) of at least 100 µs is needed before th signal is released.

executed are determined by the combination of the processing contents, operating mode, structure of the memory MATs, as shown in tables 24.8 to 24.12.

Table 24.8 Executable Memory MAT

	Operating Mode	
Processing Contents	**User Programming Mode**	**User Boot Mode***
Programming	See table 24.9	See table 24.11
Erasing	See table 24.10	See table 24.12

Note: * Programming/Erasure is possible to the user MAT.

FCCS (download)			
Operation for clearing FKEY	O	O	O
Decision of download result	O	O	O
Operation for download error	O	O	O
Operation for setting initialization parameter	O	O	O
Execution of initialization	O	×	O
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
NMI handling routine	O	×	O
Operation for disabling interrupts	O	O	O
Operation for writing H'5A to FKEY	O	O	O
Operation for setting programming parameter	O	×	O
Execution of programming	O	×	O
Decision of programming result	O	×	O
Operation for programming error	O	×	O
Operation for clearing FKEY	O	×	O

Note: * Transferring the program data to the on-chip RAM beforehand enables this a used.

Operation for clearing FKEY	O	O	O
Decision of download result	O	O	O
Operation for download error	O	O	O
Operation for setting initialization parameter	O	O	O
Execution of initialization	O	×	O
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
NMI handling routine	O	×	O
Operation for disabling interrupts	O	O	O
Operation for writing H'5A to FKEY	O	O	O
Operation for setting erasure parameter	O	×	O
Execution of erasure	O	×	O
Decision of erasure result	O	×	O
Operation for erasure error	O	×	O
Operation for clearing FKEY	O	×	O

FCCS (download)				
Operation for clearing FKEY	O	O		O
Decision of download result	O	O		O
Operation for download error	O	O		O
Operation for setting initialization parameter	O	O		O
Execution of initialization	O	×		O
Decision of initialization result	O	O		O
Operation for initialization error	O	O		O
NMI handling routine	O	×		O
Operation for disabling interrupts	O	O		O
Switching memory MATs by FMATS	O	×	O	
Operation for writing H'5A to FKEY	O	×	O	
Operation for setting programming parameter	O	×	O	
Execution of programming	O	×	O	
Decision of programming result	O	×	O	
Operation for programming error	O	×*2	O	
Operation for clearing FKEY	O	×	O	
Switching memory MATs by FMATS	O	×		O

Notes: 1. Transferring the program data to the on-chip RAM beforehand enables this a used.

2. Switching memory MATs by FMATS by a program in the on-chip RAM enable area to be used.

Operation for clearing FKEY	O	O		O
Decision of download result	O	O		O
Operation for download error	O	O		O
Operation for setting initialization parameter	O	O		O
Execution of initialization	O	×		O
Decision of initialization result	O	O		O
Operation for initialization error	O	O		O
NMI handling routine	O	×		O
Operation for disabling interrupts	O	O		O
Switching memory MATs by FMATS	O	×	O	
Operation for writing H'5A to FKEY	O	×	O	
Operation for setting erasure parameter	O	×	O	
Execution of erasure	O	×	O	
Decision of erasure result	O	×	O	
Operation for erasure error	O	×*	O	
Operation for clearing FKEY	O	×	O	
Switching memory MATs by FMATS	O	×	O	

Note: * Switching memory MATs by FMATS by a program in the on-chip RAM enables area to be used.

program is initiated, and the error in programming/erasure is indicated by the FPFR para

Table 24.13 Hardware Protection

		Function to be Pr	
Item	**Description**	**Download**	**Progr Erasi**
Reset protection	• The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and the programming/erasing protection state is entered. • The reset state will not be entered by a reset using the $\overline{RES}$ pin unless the $\overline{RES}$ pin is held low until oscillation has settled after a power is initially supplied. In the case of a reset during operation, hold the $\overline{RES}$ pin low for the $\overline{RES}$ pulse width given in the AC characteristics. If a reset is input during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	O	O

by SCO bit	entered when the SCO bit in FCCS is cleared to 0 to disable download of the programming/erasing programs.		
Protection by FKEY	The programming/erasing protection state is entered because download and programming/erasure are disabled unless the required key code is written in FKEY.	O	O
Emulation protection	The programming/erasing protection state is entered when the RAMS bit in the RAM emulation register (RAMER) is set to 1.	O	O

24.9.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when a CPU runaw occurs or operations not according to the programming/erasing procedures are detected d programming/erasure of the flash memory. Aborting programming or erasure in such cas prevents damage to the flash memory due to excessive programming or erasing.

If an error occurs during programming/erasure of the flash memory, the FLER bit in FCC to 1 and the error protection state is entered.

- When an interrupt request, such as NMI, occurs during programming/erasure.
- When the flash memory is read from during programming/erasure (including a vector an instruction fetch).
- When a SLEEP instruction is executed (including software-standby mode) during programming/erasure.
- When a bus master other than the CPU, such as the DMAC and DTC, obtains bus ma during programming/erasure.

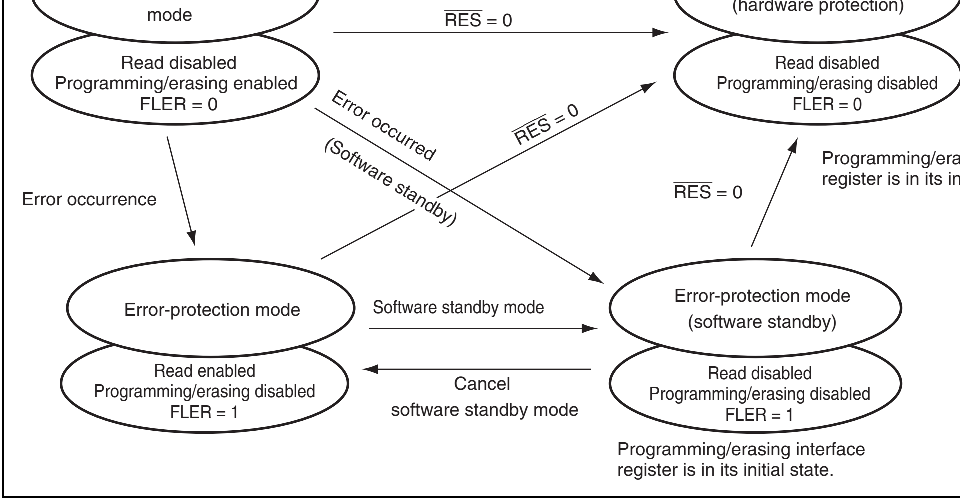


Figure 24.18 Transitions to Error Protection State

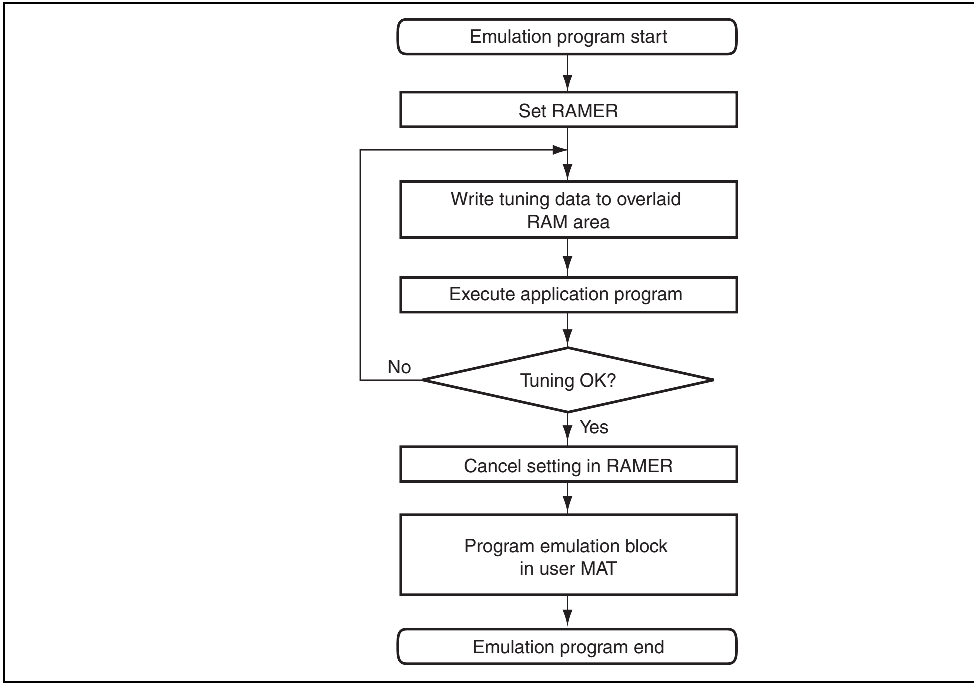


Figure 24.19 RAM Emulation Flow

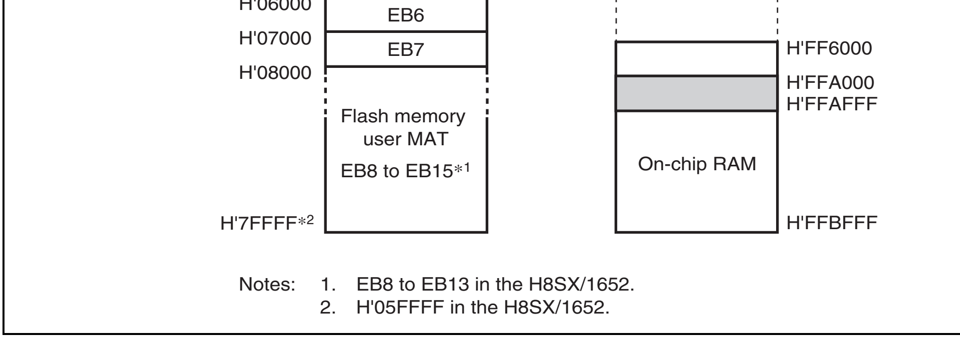


Figure 24.20 Address Map of Overlaid RAM Area (H8SX/1655)

The flash memory area that can be emulated is the one area selected by bits RAM2 to R
RAMER from among the eight blocks, EB0 to EB7, of the user MAT.

To overlay a part of the on-chip RAM with block EB0 for realtime emulation, set the R
RAMER to 1 and bits RAM2 to RAM0 to B'000.

For programming/erasing the user MAT, the procedure programs including a download
of the on-chip program must be executed. At this time, the download area should be spe
that the overlaid RAM area is not overwritten by downloading the on-chip program. Sin
in which the tuned data is stored is overlaid with the download area when FTDAR = H'
tuned data must be saved in an unused area beforehand.

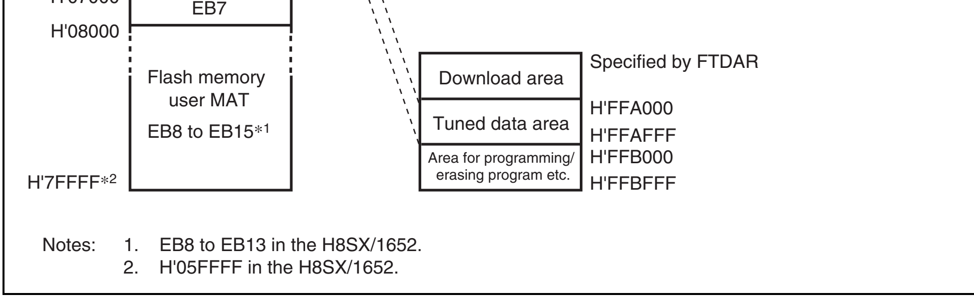


Figure 24.21 Programming Tuned Data (H8SX/1655)

1. After tuning program data is completed, clear the RAMS bit in RAMER to 0 to cance overlaid RAM.
2. Transfer the user-created procedure program to the on-chip RAM.
3. Start the procedure program and download the on-chip program to the on-chip RAM. address of the download destination should be specified by FTDAR so that the tuned does not overlay the download area.
4. When block EB0 of the user MAT has not been erased, the programming program mu downloaded after block EB0 is erased. Specify the tuned data saved in the FMPAR ar FMPDR parameters and then execute programming.

Note: Setting the RAMS bit to 1 makes all the blocks of the user MAT enter the programming/erasing protection state (emulation protection state) regardless of th of the RAM2 to RAM0 bits. Under this condition, the on-chip program cannot be downloaded. When data is to be actually programmed and erased, clear the RAM to 0.

for eight times (this prevents access to the flash memory during memory MAT switc

3. If an interrupt request has occurred during memory MAT switching, there is no guar which memory MAT is accessed. Always mask the maskable interrupts before switc memory MATs. In addition, configure the system so that NMI interrupts do not occu memory MAT switching.
4. After the memory MATs have been switched, take care because the interrupt vector also have been switched. If interrupt processing is to be the same before and after me MAT switching, transfer the interrupt processing routines to the on-chip RAM and s VBR to place the interrupt vector table in the on-chip RAM.
5. The size of the user MAT is different from that of the user boot MAT. Addresses wh the size of the 16-Kbyte user boot MAT should not be accessed. If an attempt is mad read as an undefined value.

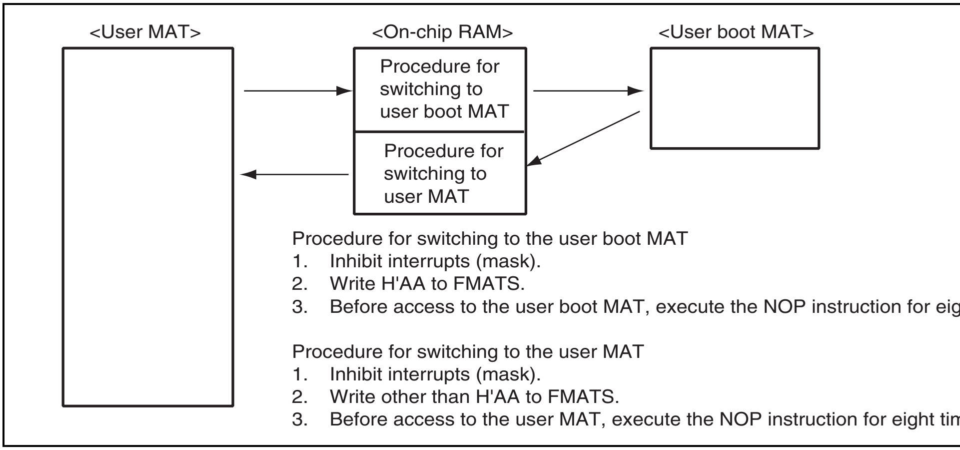


Figure 24.22 Switching between User MAT and User Boot MAT

	H8SX/1655	512 Kbytes	
User boot MAT	H8SX/1652	16 Kbytes	FZTATUSBT1
	H8SX/1655		

24.13 Standard Serial Communications Interface Specifications for Mode

The boot program initiated in boot mode performs serial communications using the host a chip SCI_4. The serial communications interface specifications are shown below.

The boot program has three states.

1. Bit-rate-adjustment state

   In this state, the boot program adjusts the bit rate to achieve serial communications wi host. Initiating boot mode enables starting of the boot program and entry to the bit-rat adjustment state. The program receives the command from the host to adjust the bit ra adjusting the bit rate, the program enters the inquiry/selection state.

2. Inquiry/selection state

   In this state, the boot program responds to inquiry commands from the host. The devi clock mode, and bit rate are selected. After selection of these settings, the program is enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure to chip RAM and erases the user MATs and user boot MATs before the transition.

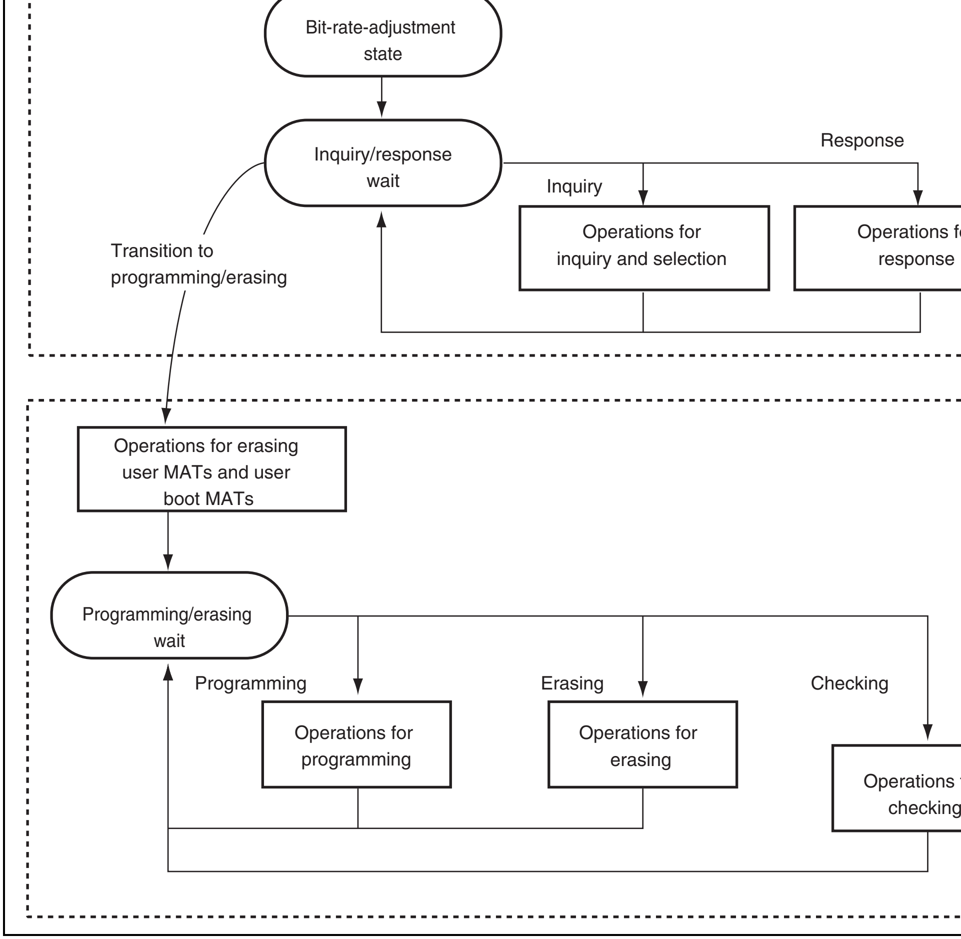


Figure 24.23 Boot Program States

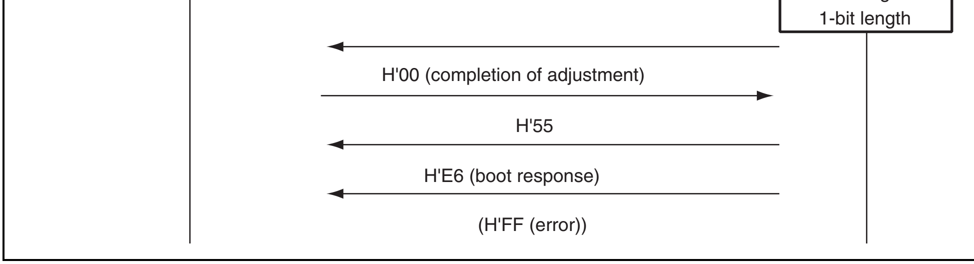


Figure 24.24 Bit-Rate-Adjustment Sequence

(2) Communications Protocol

After adjustment of the bit rate, the protocol for serial communications between the host a
boot program is as shown below.

1. One-byte commands and one-byte responses

   These one-byte commands and one-byte responses consist of the inquiries and the AC
   successful completion.

2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selections
   responses to inquiries.

   The program data size is not included under this heading because it is determined in a
   command.

3. Error response

   The error response is a response to inquiries. It consists of an error response and an er
   and comes two bytes.

4. Programming of 128 bytes

   The size is not specified in commands. The size of n is indicated in response to the programming unit inquiry.

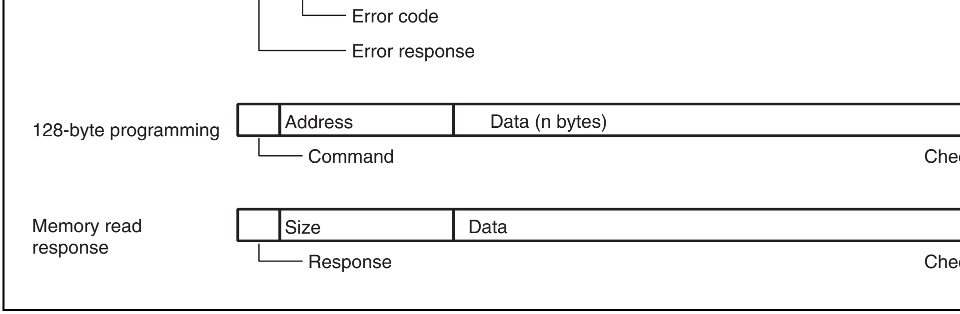


Figure 24.25 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasi checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, amou and checksum
- Checksum (one byte): The checksum is calculated so that the total of all values from command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error response (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

H'10	Device selection	Selection of device code
H'21	Clock mode inquiry	Inquiry regarding numbers of clock m and values of each mode
H'11	Clock mode selection	Indication of the selected clock mode
H'22	Multiplication ratio inquiry	Inquiry regarding the number of frequ multiplied clock types, the number of multiplication ratios, and the values o multiple
H'23	Operating clock frequency inquiry	Inquiry regarding the maximum and r values of the main clock and periphe
H'24	User boot MAT information inquiry	Inquiry regarding the number of user MATs and the start and last address each MAT
H'25	User MAT information inquiry	Inquiry regarding the a number of us and the start and last addresses of e
H'26	Block for erasing information Inquiry	Inquiry regarding the number of block the start and last addresses of each
H'27	Programming unit inquiry	Inquiry regarding the unit of program
H'3F	New bit rate selection	Selection of new bit rate
H'40	Transition to programming/erasing state	Erasing of user MAT and user boot M entry to programming/erasing state
H'4F	Boot program status inquiry	Inquiry into the operated status of the program

response to the supported device inquiry.

Command: H'20

- Command, H'20, (one byte): Inquiry regarding supported devices

Response:

H'30	Size	Number of devices	
Number of characters	Device code		Product name
...			
SUM			

- Response, H'30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transmitted, excluding the command, size, an checksum, that is, the amount of data contributes by the number of devices, characte codes and product names
- Number of devices (one byte): The number of device types supported by the boot pr
- Number of characters (one byte): The number of characters in the device codes and program's name
- Device code (four bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one byte): Checksum

  The checksum is calculated so that the total number of all values from the command the SUM byte becomes H'00.

- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response	H'90	ERROR

- Error response, H'90, (one byte): Error response to the device selection command
  ERROR : (one byte): Error code
  H'11: Sum check error
  H'21: Device code error, that is, the device code does not match

(c) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode in

Command	H'21

- Command, H'21, (one byte): Inquiry regarding clock mode

Response	H'31	Size	Mode	···	SUM

- Response, H'31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the modes
- Mode (two bytes): Values of the supported clock modes
  H'00: MD_CLK = 0 (8 to 18 MHz input)
  H'01: MD_CLK = 1 (16 MHz input)
- SUM (one byte): Checksum

- SUM (one byte): Checksum

Response | H'06

- Response, H'06, (one byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response | H'91 | ERROR

- Error response, H'91, (one byte): Error response to the clock mode selection comma
- ERROR : (one byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Number of multiplication ratios	Multiplica-tion ratio	...				
...						
SUM						

- Response, H'32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of types of multiplicat number of multiplication ratios, and the multiplication ratios
- Number of types of multiplication (one byte): The number of types of multiplication t the device can be set

  (e.g. when there are two multiplied clock types, which are the main and peripheral clo number of types will be H'02.)
- Number of multiplication ratios (one byte): The number of types of multiplication rati each type

  (e.g. the number of multiplication ratios to which the main clock can be set and the pe clock can be set.)
- Multiplication ratio (one byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-frequen multiplier is four, the value of multiplication ratio will be H'04.)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the divided by two, the value of division ratio will be H'FE. H'FE = -2)

  The number of multiplication ratios returned is the same as the number of multiplicati and as many groups of data are returned as there are types of multiplication.
- SUM (one byte): Checksum

operating clock frequency	frequency
...	
SUM	

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operati frequency types
  (e.g. when there are two operating clock frequency types, which are the main and pe clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of th multiplied or divided clock frequency.

  The minimum and maximum values of the operating clock frequency represent the v MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the v 17.00 MHz, it will be 2000, which is H'07D0.)
- Maximum value (two bytes): Maximum value among the multiplied or divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating clo frequencies.
- SUM (one byte): Checksum

- Response, H'34, (one byte): Response to user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start ac
  and area-last address
- Number of Areas (one byte): The number of consecutive user boot MAT areas
  When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Area-start address (four byte): Start address of the area
- Area-last address (four byte): Last address of the area
  There are as many groups of data representing the start and last addresses as there are
- SUM (one byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command

H'25

- Command, H'25, (one byte): Inquiry regarding user MAT information

Response

H'35	Size	Number of areas	
Start address area			Last address area
...			
SUM			

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start ac
  and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area

Response

H'36	Size	Number of blocks	
Block start address			Block last address
...			
SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-s addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block
- Block last Address (four bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there ar
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command

H'27

- Command, H'27, (one byte): Inquiry regarding programming unit

Response

H'37	Size	Programming unit	SUM

- Response, H'37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicate the programming unit, which is fi
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (one byte): Checksum

- Size (one byte): The number of bytes that represents the bit rate, input frequency, nur types of multiplication, and multiplication ratio
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is
- Input frequency (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 1(when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Number of types of multiplication (one byte): The number of multiplication to which device can be set.
- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for the operating frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clo frequency is multiplied by four, the multiplication ratio will be H'04.)
  Division ratio: The inverse of the division ratio, as a negative number (e.g. when the c frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for the peripheral frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clo frequency is multiplied by four, the multiplication ratio will be H'04.)
  (Division ratio: The inverse of the division ratio, as a negative number (E.g. when the divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- SUM (one byte): Checksum

Response | H'06 |

- Response, H'06, (one byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

H'27: Operating frequency error

The frequency is not within the specified range.

(4) Receive Data Check

The methods for checking of receive data are listed below.

1. Input frequency

   The received value of the input frequency is checked to ensure that it is within the ra
   minimum to maximum frequencies which matches the clock modes of the specified
   When the value is out of this range, an input-frequency error is generated.

2. Multiplication ratio

   The received value of the multiplication ratio or division ratio is checked to ensure th
   matches the clock modes of the specified device. When the value is out of this range
   frequency error is generated.

3. Operating frequency error

   Operating frequency is calculated from the received value of the input frequency and
   multiplication or division ratio. The input frequency is input to the LSI and the LSI i
   at the operating frequency. The expression is given below.

   Operating frequency = Input frequency × Multiplication ratio, or
   Operating frequency = Input frequency ÷ Division ratio

   The calculated operating frequency should be checked to ensure that it is within the r
   minimum to maximum frequencies which are available with the clock modes of the
   device. When it is out of this range, an operating frequency error is generated.

When the new bit rate is selectable, the rate will be set in the register after sending ACK i
response. The host will send an ACK with the new bit rate for confirmation and the boot
will response with that rate.

Confirmation | H'06 |

- Confirmation, H'06, (one byte): Confirmation of a new bit rate

Response | H'06 |

- Response, H'06, (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 24.26.

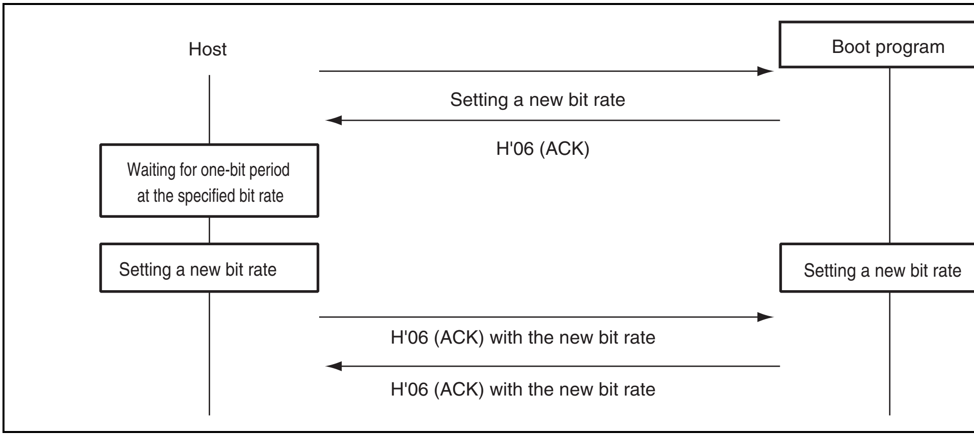


Figure 24.26 New Bit-Rate Selection Sequence

Command	H'40

- Command, H'40, (one byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (one byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MAT and user boot MAT have bee
  by the transferred erasing program.

Error Response	H'C0	H'51

- Error response, H'C0, (one byte): Error response for user boot MAT blank check
- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

(6) Command Error

A command error will occur when a command is undefined, the order of commands is i
or a command is unacceptable. Issuing a clock-mode selection command before a devic
or an inquiry command after the transition to programming/erasing state command, are

Error Response	H'80	H'xx

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

be made, such as the multiplication-ratio inquiry (H'22) or operating frequency inquir which are needed for a new bit-rate selection.

6. A new bit rate should be selected with the new bit-rate selection (H'3F) command, acc to the returned information on multiplication ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user boot MAT a MAT should be made to inquire about the user boot MATs information inquiry (H'24 MATs information inquiry (H'25), erased block information inquiry (H'26), and progr unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

H'43	User MAT programming selection	Transfers the user MAT programm program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user
H'4B	User MAT sum check	Checks the checksum of the user
H'4C	User boot MAT blank check	Checks the blank data of the user
H'4D	User MAT blank check	Checks the blank data of the user
H'4F	Boot program status inquiry	Inquires into the boot program's st

command represents the data programmed according to the method specified by the se
command. When more than 128-byte data is programmed, 128-byte commands should
repeatedly be executed. Sending a 128-byte programming command with H'FFFFFFF
address will stop the programming. On completion of programming, the boot program
wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programmin
another method or of another MAT, the procedure must be repeated from the program
selection command.

The sequence for the programming selection and 128-byte programming commands is
in figure 24.27.

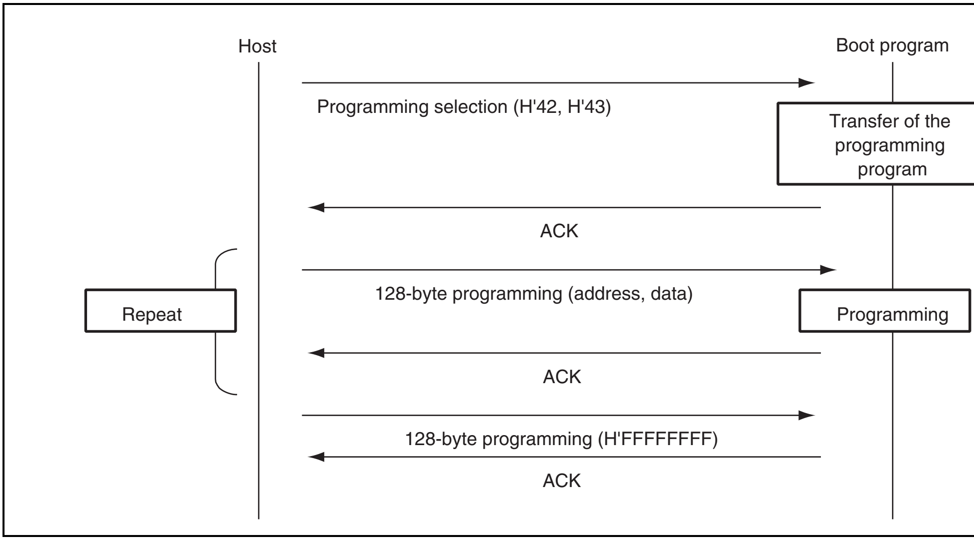


Figure 24.27 Programming Sequence

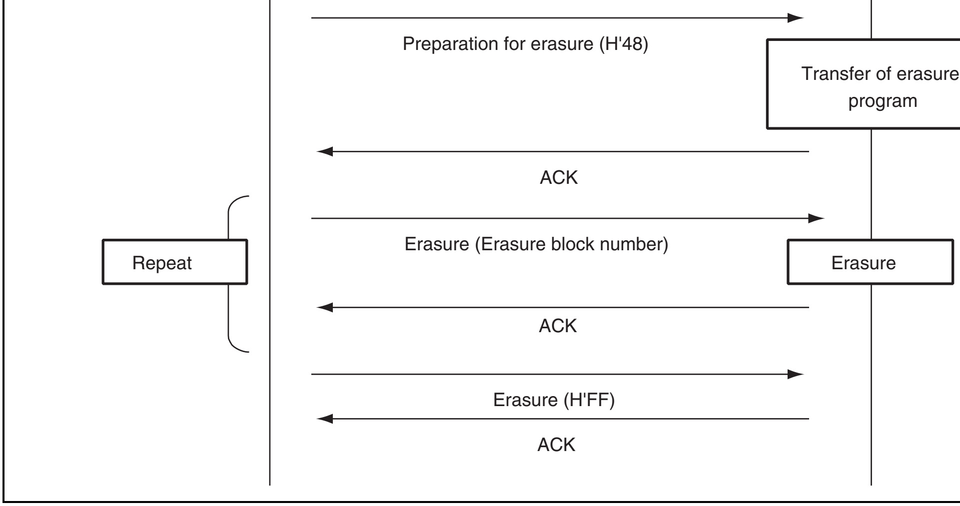


Figure 24.28 Erasure Sequence

Error Response	H'C2	ERROR

- Error response: H'C2 (1 byte): Error response to user boot MAT programming selecti
- ERROR: (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complete

(b) User MAT Programming Selection

The boot program will transfer a program for user MAT programming selection. The data
programmed to the user MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (one byte): User MAT programming selection

Response	H'06

- Response, H'06, (one byte): Response to user MAT programming selection

  When the programming program has been transferred, the boot program will return A

Error Response	H'C3	ERROR

- Error response: H'C3 (1 byte): Error response to user MAT programming selection
- ERROR: (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complete

- Programming Address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00: H'01000000)
- Program data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response | H'06 |

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response | H'D0 | ERROR |

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum Error
  H'2A: Address error
  The address is not in the specified MAT.
  H'53: Programming error
  A programming error has occurred and programming cannot be contin

The specified address should match the unit for programming of data. For example, whe
programming is in 128-byte units, the lower eight bits of the address should be H'00 or I
When there are less than 128 bytes of data to be programmed, the host should fill the res
H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will sto
programming operation. The boot program will interpret this as the end of the programm
wait for selection of programming or erasing.

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum error

  H'53: Programming error

  An error has occurred in programming and programming cannot be con

(d) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the tran erasure program.

Command	H'48

- Command, H'48, (one byte): Erasure selection

Response	H'06

- Response, H'06, (one byte): Response for erasure selection
  After the erasure program has been transferred, the boot program will return ACK.

Error Response	H'C8	ERROR

- Error Response, H'C8, (one byte): Error response to erasure selection
- ERROR: (one byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complete

Response	H'06

- Response, H'06, (one byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error Response, H'D8, (one byte): Response to Erasure
- ERROR (one byte): Error code

  H'11: Sum check error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasure error
  An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a selec
command.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H'FF
  Stop code for erasure
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the proc
  should be executed from the erasure selection command.

H'01: User MAT

An address error occurs when the area setting is incorrect.

- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response

H'52	Read size						
Data	···						
SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Error Response

H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

H'11: Sum check error

H'2A: Address error

The read address is not in the MAT.

H'2B: Size error

The read size exceeds the MAT.

This is fixed to 4.

- Checksum of user boot program (four bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(h) User MAT Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the us
program.

Command	H'4B

- Command, H'4B, (one byte): Sum check for user program

Response	H'5B	Size	Checksum of user program	SUM

- Response, H'5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

Error Response | H'CC | H'52

- Error Response, H'CC, (one byte): Response to blank check for user boot MAT
- Error Code, H'52, (one byte): Erasure has not been completed.

(j) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the resul

Command | H'4D

- Command, H'4D, (one byte): Blank check for user MATs

Response | H'06

- Response, H'06, (one byte): Response to the blank check for user MATs
  If the contents of all user MATs are blank (H'FF), the boot program will return ACK.

Error Response | H'CD | H'52

- Error Response, H'CD, (one byte): Error response to the blank check of user MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

- Status (one byte): State of the boot program
- ERROR (one byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (one byte): Sum check

Table 24.18 Status Code

Code	Description
H'11	Device selection wait
H'12	Clock mode selection wait
H'13	Bit rate selection wait
H'1F	Programming/erasing state transition wait (bit rate selection is completed)
H'31	Programming state for erasure
H'3F	Programming/erasing selection wait (erasure is completed)
H'4F	Program data receive wait
H'5F	Erase block specification wait (erasure is completed)

H'26	Multiplication ratio error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data length error
H'51	Erasure error
H'52	Erasure incomplete error
H'53	Programming error
H'54	Selection processing error
H'80	Command error
H'FF	Bit-rate-adjustment confirmation error

3.3-V programming voltage. Use only the specified socket adapter.

5. Do not turn off the Vcc power supply nor remove the chip from the PROM program programming/erasure in which a high voltage is applied to the flash memory. Doing damage the flash memory permanently. If a reset is input, the reset must be released reset input period of at least 100ms.
6. The flash memory is not accessible until FKEY is cleared after programming/erasure the operating mode is changed and this LSI is restarted by a reset immediately after programming/erasure has finished, secure the reset input period (period of $\overline{RES}$ = 0) 100µs. Transition to the reset state during programming/erasure is inhibited. If a rese the reset must be released after the reset input period of at least 100µs.
7. At powering on the Vcc power supply, fix the $\overline{RES}$ pin to low and set the flash mem hardware protection state. This power on procedure must also be satisfied at a power power-on caused by a power failure and other factors.
8. In on-board programming mode or programmer mode, programming of the 128-byte programming-unit block must be performed only once. Perform programming in the where the programming-unit block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of progra erasure in on-board programming mode, it is recommended that automatic programm performed after execution of automatic erasure.
10. To program the flash memory, the program data and program must be allocated to a which are higher than those of the external interrupt vector table and H'FF must be w all the system reserved areas in the exception handling vector table.
11. The programming program that includes the initialization routine and the erasing pro includes the initialization routine are each 4 Kbytes or less. Accordingly, when the C frequency is 35 MHz, the download for each program takes approximately 60 µs at t maximum.

immediately after setting it to 1. Otherwise, download cannot be performed normally.
Immediately after executing the instruction to set the SCO bit to 1, dummy read of the
must be executed twice.

15. The contents of general registers ER0 and ER1 are not saved during download of an o
program, initialization, programming, or erasure. When needed, save the general regis
before a download request or before execution of initialization, programming, or erasu
the procedure program.

valid

- Six test modes:

  BYPASS mode

  EXTEST mode

  SAMPLE/PRELOAD mode

  CLAMP mode

  HIGHZ mode

  IDCODE mode

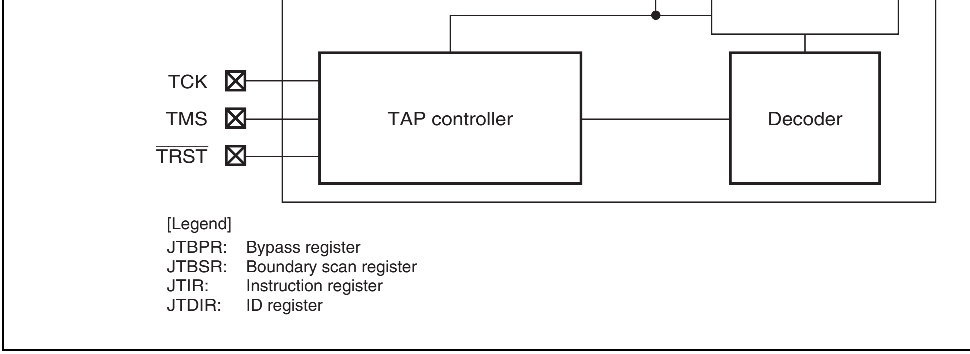


Figure 25.1 Block Diagram of Boundary Scan Function

25.3 Input/Output Pins

Table 25.1 shows the I/O pins used in the boundary scan function.

Table 25.1 Pin Configuration

Pin Name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. Input the clock the duty cycle of wh percent when boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin
$\overline{\text{TRST}}$	Input	Test reset input pin

TDI and TDO pins in BYPASS mode. The boundary scan register (JTBSR), which is a
register (see table 25.4), is connected between the TDI and TDO pins when test data are
shifted in. None of the registers is accessible from the CPU.

Table 25.2 shows the availability of serial transfer for the registers.

Table 25.2 Serial Transfers for Registers

Register Abbreviation	Serial Input	Serial Output
JTIR	Available	Not available
JTBPR	Available	Available
JTBSR	Available	Available
JTID	Not available	Available

Bit Name	TS3	TS2	TS1	TS0			
Initial Value	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 12	TS[3:0]	All 0	R/W	Test Bit Set Specify an instruction as shown in table 25.3.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu always 0.

0	1	1	1	Reserved
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	BYPASS

25.4.2 Bypass Register (JTBPR)

JTBPR is a 1-bit register and is connected between the TDI and TDO pins when JTIR is BYPASS mode. JTBPR cannot be read from or written to by the CPU.

25.4.3 Boundary Scan Register (JTBSR)

JTBSR is a shift register to control the external input and output pins of this LSI and is across the pads. The initial values are undefined. JTBSR cannot be accessed by the CPU EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions are issued to apply boundary-scan testing conformant to the JTAG standard.

Table 25.5 shows the correspondence between the JTBSR bits and the pins of this LSI.

			Output	[illegible]
6	D2	PM1	Input	284
			Output enable	283
			Output	282
7	E1	PM2	Input	280
			Output enable	279
			Output	278
8	F1	PF4	Input	265
			Output enable	264
			Output	263
9	F4	PF3	Input	262
			Output enable	261
			Output	260
11	F2	PF2	Input	259
			Output enable	258
			Output	257
12	G1	PF1	Input	256
			Output enable	255
			Output	254
13	H4	PF0	Input	253
			Output enable	252
			Output	251
14	G3	PE7	Input	250
			Output enable	249
			Output	248
15	G2	PE6	Input	247
			Output enable	246
			Output	245

20	H3	PE3	Input
			Output ena
			Output
21	J4	PE2	Input
			Output ena
			Output
22	J2	PE1	Input
			Output ena
			Output
23	K1	PE0	Input
			Output ena
			Output
24	J3	PD7	Input
			Output ena
			Output
25	K4	PD6	Input
			Output ena
			Output
27	K2	PD5	Input
			Output ena
			Output
28	K3	PD4	Input
			Output ena
			Output
29	L1	PD3	Input
			Output ena
			Output

34	N2	PM3	Input	201
			Output enable	200
			Output	199
35	N3	PM4	Input	197
			Output enable	196
			Output	195
40	L4	VBUS	Input	194
41	L5	MD_C LK	Input	193
43	M6	P20	Input	183
			Output enable	182
			Output	181
45	K6	P21	Input	180
			Output enable	179
			Output	178
46	N6	P22	Input	177
			Output enable	176
			Output	175
47	M7	P23	Input	174
			Output enable	173
			Output	172
48	L6	P24	Input	171
			Output enable	170
			Output	169

52	L9	NMI	Input
53	M9	PH0	Input
			Output e
			Output
54	L10	PH1	Input
			Output e
			Output
55	K10	PH2	Input
			Output e
			Output
56	N10	PH3	Input
			Output e
			Output
61	M12	PH7	Input
			Output e
			Output
58	M11	PH4	Input
			Output e
			Output
59	N11	PH5	Input
			Output e
			Output
60	N12	PH6	Input
			Output e
			Output

63	M13	PI0	Input	110	86	E11	P16	Input
			Output enable	109				Output enabl
			Output	108				Output
68	K11	PI4	Input	107	87	E10	P17	Input
			Output enable	106				Output enabl
			Output	105				Output
69	K12	PI5	Input	104	89	B13	P60	Input
			Output enable	103				Output enabl
			Output	102				Output
71	J10	PI7	Input	101	90	A13	P61	Input
			Output enable	100				Output enabl
			Output	99				Output
70	J13	PI6	Input	98	97	D10	MD0	Input
			Output enable	97	109	B6	MD1	Input
			Output	96				
72	J11	P10	Input	95	110	D6	PA0	Input
			Output enable	94				Output enabl
			Output	93				Output
73	H11	P11	Input	92	111	A5	PA1	Input
			Output enable	91				Output enabl
			Output	90				Output
74	J12	P12	Input	89	112	B4	PA2	Input
			Output enable	88				Output enabl
			Output	87				Output

116	C4	PA6	Input	14
			Output enable	13
			Output	12
118	A2	PA7	Input	11
			Output enable	10
			Output	9
120	B2	PB0	Input	8
			Output enable	7
			Output	6
1	A1	PB1	Input	5
			Output enable	4
			Output	3
2	B1	PB2	Input	2
			Output enable	1
			Output	0
to TDO				

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

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 0	DID31 to DID0	H'0807F447	R/W	JTID is a register the value showing the decide IDCODE is fixed.

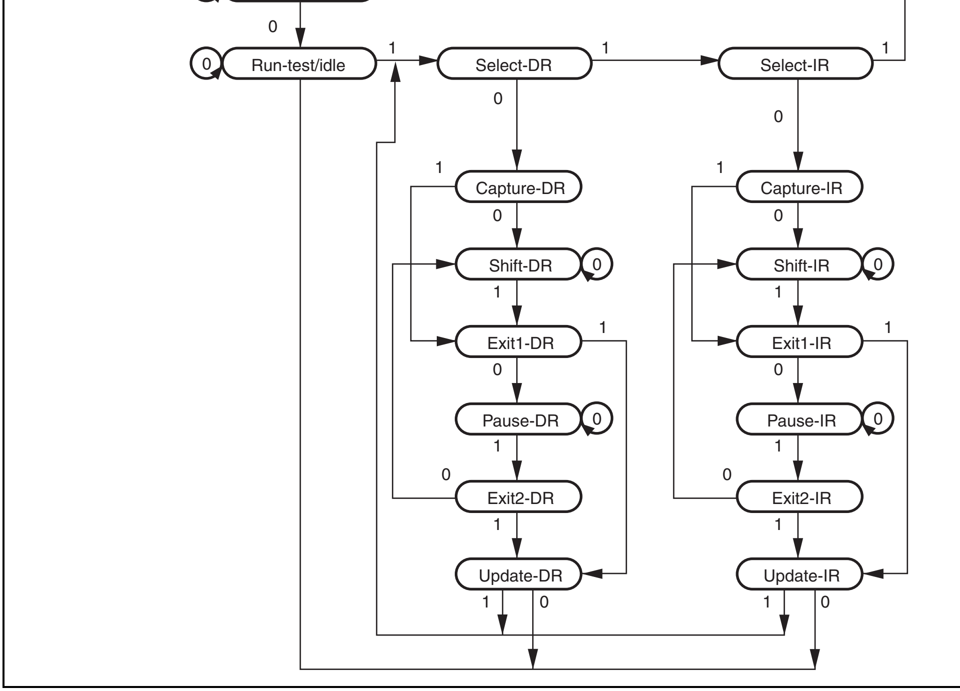


Figure 25.2 State Transitions of the TAP Controller

EXTEST (Instruction Code: B'0000): The EXTEST instruction is used to test external when this LSI is installed on the printed circuit board. If this instruction is executed, outp are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the b scan register to the print circuit board, and input pins are used to input test result.

SAMPLE/PRELOAD (Instruction Code: B'0100): The SAMPLE/PRELOAD instructi used to input data from the LSI internal circuits to the boundary scan register, output data scan path, and reload the data to the scan path. While this instruction is executed, input si directly input to the LSI and output signals are also directly output to the external circuits system circuit is not affected by this function.

In SAMPLE operation, the boundary scan register latches the snap shot of data transferre input pins to internal circuit or data transferred from internal circuit to output pins. The la data is read from the scan path. The scan register latches the snap data at the rising edge o TCK in Capture-DR state. The scan register latches snap shot without affecting the LSI n operation.

In PRELOAD operation, initial value is written from the scan path to the parallel output l the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is without executing this PRELOAD operation, undefined values are output from the beginn the end (transfer to the output latch) of the EXTEST sequence. (In EXTEST instruction, o parallel latches are always output to the output pins.)

IDCODE (Instruction Code: B'0001): When the IDCODE instruction is selected, IDCO register value is output to the TDO in Shift-DR state of the TAP controller. In this case, I register value is output from the LSB. During this instruction execution, test circuit does n the system circuit. INSTR is initialized by the IDCODE instruction in Test-Logic-Reset s the TAP controller.

register is maintained regardless of the state of the TAP controller.

BYPASS is connected between TDI and TDO pins, leading to the same operation as wh
BYPASS instruction has been selected.

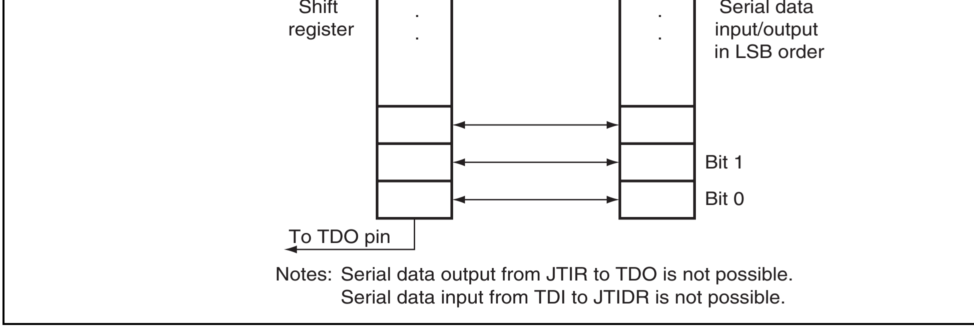


Figure 25.3 Serial Data Input/Output

2. If a pin with open-drain function is SAMPLEed while its open-drain function is enabl while the corresponding OUT register is set to 1, the corresponding Control register is to 0 (the pin status is Hi-Z). If the pin is SAMPLEed while the corresponding OUT re cleared to 0, the corresponding Control register is 1 (the pin status is 0)
3. Pins of the boundary scan (TCK, TDI, TMS, and $\overline{TRST}$) have to be pulled up by pull- resistors.
4. Power supply pins (V_{CC}, V_{CL}, V_{SS}, AV_{CC}, AV_{SS}, Vref, $PLLV_{CC}$, $PLLV_{SS}$, DrV_{CC}, and DrV cannot be boundary-scanned.
5. Clock pins (EXTAL and XTAL) cannot be boundary-scanned.
6. Reset and standby signals ($\overline{RES}$ and $\overline{STBY}$) cannot be boundary-scanned.
7. Boundary scan pins (TCK, TMS, $\overline{TRST}$, TDI, and TDO) cannot be boundary-scanned
8. The boundary scan function is not available when this LSI are in the following states.

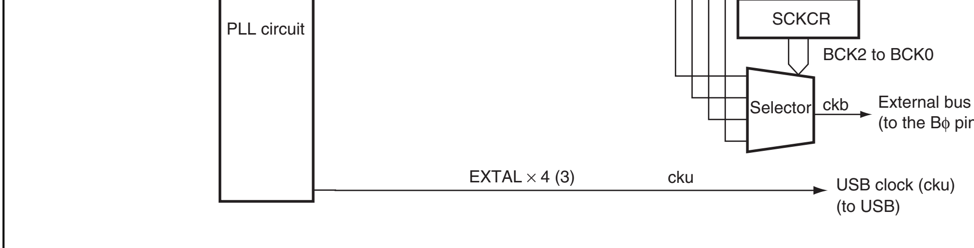
(1) Reset state

(2) Hardware standby mode, software standby mode, and deep software standby mode

This LSI supports four clocks: a system clock provided to the CPU and bus masters, a p
module clock provided to the peripheral modules, an external bus clock provided to the
bus and a USB clock provided to the USB module. Frequencies of the peripheral modul
the external bus clock, and the system clock can be set independently, although the peri
module clock and the external bus clock operate with the frequency lower than the syste
frequency.

The USB module requires the 48-MHz clock. Set the external clock frequency and the M
pin so that the USB clock (cku) frequency becomes 48 MHz.

Note that the MD_CLK pin setting also changes the frequencies of the peripheral modul
the external bus clock, and the system clock.



Note: * Values in parentheses are setting values when MD_CLK = 1.

Figure 26.1 Block Diagram of Clock Pulse Generator

Table 26.1 Selection of Clock Pulse Generator

MD_CLK	EXTAL Input Clock Frequencies	Iϕ/Pϕ/Bϕ	USB Clock (cku)
0	8 MHz to 18 MHz	EXTAL ×4, ×2, ×1, ×1/2	EXTAL ×4
1	16 MHz	EXTAL ×2, ×1, ×1/2	EXTAL ×3

Bit	15	14	13	12	11	10	9
Bit Name	PSTOP1	—	—	—	—	ICK2	ICK1
Initial Value	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	—	PCK2	PCK1	PCK0	—	BCK2	BCK1
Initial Value	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PSTOP1	0	R/W	Bϕ Clock Output Enable Controls ϕ output on PA7. • Normal operation 0: ϕ output 1: Fixed high
14 to 11	—	All 0	R/W	Reserved Although these bits are readable/writable, onl be written to.

				011: × 1/2 Setting prohibitec 1XX: Setting prohibited The frequencies of the peripheral module clock external bus clock change to the same frequer system clock if the frequency of the system clo lower than that of the two clocks.
7	—	0	R/W	Reserved Although this bit is readable/writable, only 0 sh written to.
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 peripher module clock. The ratio to the input clock is as PCK (2:0) MD_CLK = 0 MD_CLK = 1 000: × 4 × 2 001: × 2 × 1 010: × 1 × 1/2 011: × 1/2 Setting prohibitec 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 peripheral module clock higher than that of the clock, the clocks will have the same frequency

001:	× 2	× 1
010:	× 1	× 1/2
011:	× 1/2	Setting prohibite
1XX:	Setting prohibited	

The frequency of the external bus clock shoul than that of the system clock. Though these b set so as to make the frequency of the extern clock higher than that of the system clock, the will have the same frequency in reality.

Note: X: Don't care

16 MHz.

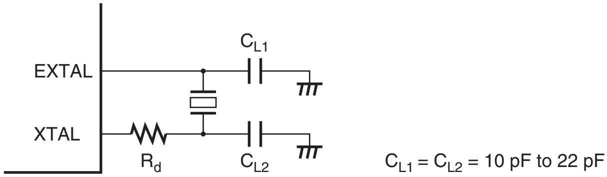


Figure 26.2 Connection of Crystal Resonator (Example)

Table 26.2 Damping Resistance Value

Frequency (MHz)	8	12	16	18
R_d (Ω)	200	0	0	0

Figure 26.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator th
the characteristics shown in table 26.3.

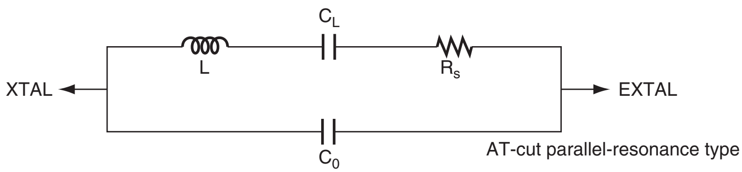


Figure 26.3 Crystal Resonator Equivalent Circuit

pin, put the external clock in high level during standby mode.

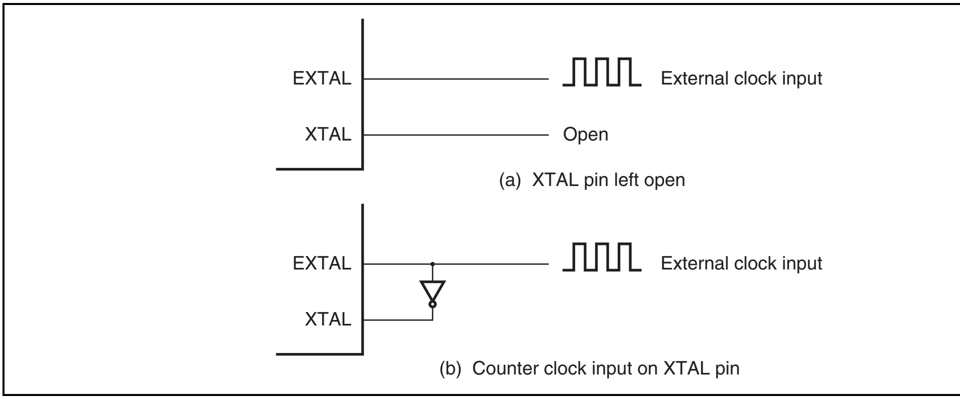


Figure 26.4 External Clock Input (Examples)

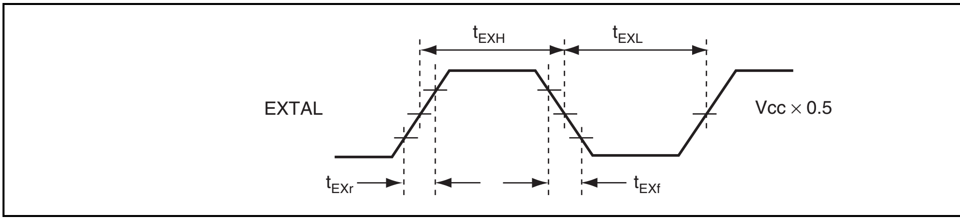


Figure 26.5 External Clock Input Timing

updated frequency.

I ϕ max = 50 MHz, P ϕ max = 35 MHz, and Bϕ max = 50 MHz,

the frequencies should satisfy the conditions 8 MHz ≤ Iϕ ≤ 50 MHz, 8 MHz ≤ Pϕ ≤ 3
and 8 MHz ≤ Bϕ ≤ 50 MHz.

2. All the on-chip peripheral modules (except for the EXDMAC, DMAC, and DTC) op
the Pϕ. Note therefore that the time processing of modules such as a timer and SCI d
before and after changing the clock division ratio.

   In addition, wait time for clearing software standby mode differs by changing the clo
division ratio. For details, see section 27.7.3, Setting Oscillation Settling Time after
Software Standby Mode.

3. The relationship among the system clock, peripheral module clock, and external bus
≥ Pϕ and Iϕ ≥ Bϕ. In addition, the system clock setting has the highest priority. Acco
Pϕ or Bϕ may have the frequency set by bits ICK2 to ICK0 regardless of the settings
PCK2 to PCK0 or BCK2 to BCK0.

4. Note that the frequency of ϕ will be changed in the middle of a bus cycle when settin
while executing the external bus cycle with the write-data-buffer function and EXDM

5. Figure 26.6 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 26.6 Clock Modification Timing

26.5.2 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.

26.5.3 Notes on Board Design

When using the crystal resonator, place the crystal resonator and its load capacitors as clo
XTAL and EXTAL pins as possible. Other signal lines should be routed away from the o
circuit as shown in figure 26.7 to prevent induction from interfering with correct oscillati

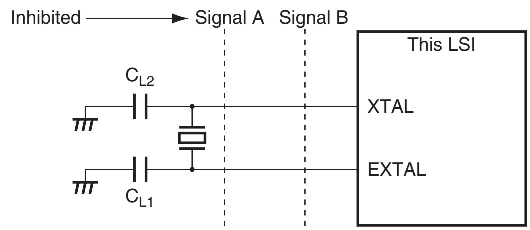


Figure 26.7 Note on Board Design for Oscillation Circuit

Note: * CB and CPB are laminated ceramic capacitors.

Figure 26.8 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.
- Five power-down modes

  Sleep mode

  All-module-clock-stop mode

  Software standby mode

  Deep software standby mode

  Hardware standby mode

Table 27.1 shows conditions to shift to a power-down mode, states of the CPU and perip
modules, and clearing method for each mode. After the reset state, since this LSI operat
normal program execution state, the modules, other than the DMAC, DTC, and EXDM
stopped.

Oscillator	Operating	Operating	Stopped	Stopped	Stopp
CPU	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (undefined)	Stopp (unde
On-chip RAM 4 (H'FF2000 to H'FF3FFF)	Operating (retained)	Stopped (retained)	Stopped (retained)	Stopped (undefined)	Stopp (unde
On-chip RAMs 3 to 0 (H'FF4000 to H'FFBFFF)	Operating (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained/ undefined)*5	Stopp (unde
Universal Serial Bus interface	Operating	Stopped (retained)	Stopped (retained)	Stopped (retained/ undefined)*5	Stopp (unde
Watchdog timer	Operating	Operating	Stopped (retained)	Stopped (undefined)	Stopp (unde
8-bit timer (unit 0/1)	Operating	Operating*4	Stopped (retained)	Stopped (undefined)	Stopp (unde
Voltage detection circuit*9	Operating	Operating	Operating	Operating	Stopp
Power-on reset circuit*9	Operating	Operating	Operating	Operating	Stopp
Other peripheral modules	Operating	Stopped*1	Stopped*1	Stopped*7 (undefined)	Stopp (unde
I/O ports	Operating	Retained	Retained*6	Stopped*6 (undefined)	Hi-Z

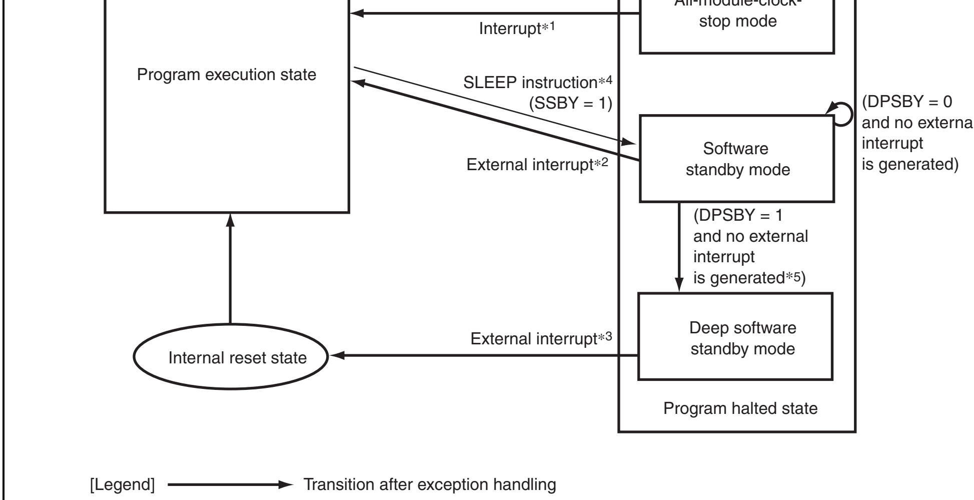
Notes: "Stopped (retained)" in the table means that the internal values are retained and in operations are suspended.

7. Some peripheral modules enter a state where the register values are retained

8. An external interrupt or USB suspend/resume interrupt.

9. External interrupt and voltage monitoring interrupt*[10].

10. Supported only by the H8SX/1655M Group.



Notes: 1. NMI, IRQ0 to IRQ11, 8-bit timer interrupt, watchdog timer interrupt, and voltage monitoring interrupt*6.
Note that the 8-bit timer interrupt is valid when the MSTPCRA9 or MSTPCRA8 bit is cleared to 0.
2. NMI, IRQ0 to IRQ11, and voltage monitoring interrupt*6.
Note that IRQ is valid only when the corresponding bit in SSIER is set to 1.
3. NMI, $\overline{IRQ0}$-A to $\overline{IRQ3}$-A, and voltage monitoring interrupt*6.
Note that IRQ and voltage monitoring*6 interrupts are valid only when the corresponding bit in DPSIER is set
4. The SLPIE bit in SBYCR is cleared to 0.
5. If a conflict between a transition to deep software standby mode and generation of software standby mode clearing source occurs, a mode transition may be made from software standby mode to program execution st through execution of interrupt exception handling. In this case, a transition to deep software standby mode is made. For details, refer to section 27.12, Usage Notes.
6. Supported only by the H8SX/1655M Group.

From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ is driven low.
From any state except hardware standby mode, a transition to the reset state occurs when $\overline{RES}$ is driven low.

Figure 27.1 Mode Transitions

- Deep standby wait control register (DPSWCR)
- Deep standby interrupt enable register (DPSIER)
- Deep standby interrupt flag register (DPSIFR)
- Deep standby interrupt edge register (DPSIEGR)
- Reset status register (RSTSR)
- Deep standby backup register n (DPSBKRn) (n: 15 to 0)

27.2.1 Standby Control Register (SBYCR)

SBYCR controls software standby mode.

Bit	15	14	13	12	11	10	9
Bit name	SSBY	OPE	—	STS4	STS3	STS2	STS1
Initial value:	0	1	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	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

				is used in watchdog timer mode, the setting of th disabled. In this case, a transition is always mad 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 ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and retained or these lines are set to the high-Z state software standby mode or deep software standb 0: In software standby mode or deep software st mode, address bus and bus control signal line high-impedance. 1: In software standby mode or deep software st mode, output states of address bus and bus c signals are retained.
13	—	0	R/W	Reserved This bit is always read as 0. The write value shou always be 0.

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

				executed, this bit remains set to 1. For clearing, this bit.
6 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.

[Legend]

x: Don't care

Note: With the F-ZTAT version, the flash memory settling time must be reserved.

27.2.2 Module Stop Control Registers A and B (MSTPCRA and MSTPCRB)

MSTPCRA and MSTPCRB control module stop state. Setting a bit to 1 makes the corres module enter module stop state, while clearing the bit to 0 clears module stop state.

- MSTPCRA

Bit	15	14	13	12	11	10	9	
Bit name	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9	M
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	M
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 state fo current consumption by stopping the bus contro I/O ports operations when the CPU executes th instruction after module stop state has been se on-chip peripheral modules controlled by MSTF 0: All-module-clock-stop mode disabled 1: All-module-clock-stop mode enabled
14	MSTPA14	0	R/W	EXDMA controller (EXDMAC)
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.
5	MSTPA5	1	R/W	D/A converter (channels 1 and 0)

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_0: PO7 to
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 communications interface_4 (SCI_4)
11	MSTPB11	1	R/W	Reserved This bit is always read as 1. The write value shou always be 1.
10	MSTPB10	1	R/W	Serial communications interface_2 (SCI_2)
9	MSTPB9	1	R/W	Serial communications interface_1 (SCI_1)
8	MSTPB8	1	R/W	Serial communications interface_0 (SCI_0)
7	MSTPB7	1	R/W	I^{2}C bus interface 2_1 (IIC2_1)
6	MSTPB6	1	R/W	I^{2}C bus interface 2_0 (IIC2_0)
5	MSTPB5	1	R/W	User break controller (UBC)
4	MSTPB4	1	R/W	Reserved
3	MSTPB3	1	R/W	These bits are always read as 1. The write value always be 1.
2	MSTPB2	1	R/W	
1	MSTPB1	1	R/W	
0	MSTPB0	1	R/W	

Bit	15	14	13	12	11	10	9
Bit name	MSTPC15	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	MSTPC9
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	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	Serial communications interface_5 (SCI_5), (IrDA)
14	MSTPC14	1	R/W	Serial communications interface_6 (SCI_6)
13	MSTPC13	1	R/W	8-bit timer (TMR_4, TMR_5)
12	MSTPC12	1	R/W	8-bit timer (TMR_6, TMR_7)
11	MSTPC11	1	R/W	Universal Serial Bus interface (USB)
10	MSTPC10	1	R/W	Cyclic redundancy check calculator
9	MSTPC9	1	R/W	A/D converter (unit 1)
8	MSTPC8	1	R/W	Programmable pulse generator (PPG_1: PO31 to P

value.

27.2.4 Deep Standby Control Register (DPSBYCR)

DPSBYCR controls deep software standby mode.

DPSBYCR is not initialized by the internal reset signal upon exit from deep software star
mode.

Bit	7	6	5	4	3	2	1	
Bit name	DPSBY	IOKEEP	RAMCUT2	RAMCUT1	—	—	—	RA
Initial value:	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1	0	Enters software standby after execution of a SLE instruction.
1	1	Enters deep software st mode after execution of instruction.

When deep software standby mode is canceled interrupt, this bit remains at 1. Write a 0 here to Setting of this bit has no effect when the WDT i watchdog timer mode. In this case, executing th instruction always initiates entry to sleep mode module-clock-stop mode. Be sure to clear this b when setting the SLPIE bit to 1.

				simultaneously with exit from deep software standby mode.
				1 The retained port states are releas when a 0 is written to this bit follow from deep software standby mode.
				In operation in external extended mode, howeve address bus, bus control signals ($\overline{CS0}$, $\overline{AS}$, $\overline{RD}$, and $\overline{LWR}$), and data bus are set to the initial stat exit from deep software standby mode.
5	RAMCUT2	0	R/W	On-chip RAM Power Off 2 RAMCUT 2, 1, and 0 control the internal power s the on-chip RAM and USB in deep software star mode. For details, see descriptions of the RAMC
4	RAMCUT1	0	R/W	On-chip RAM Power Off 1 RAMCUT 2, 1, and 0 control the internal power s the on-chip RAM and USB in deep software star mode. For details, see descriptions of the RAMC
3 to 1	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
0	RAMCUT0	1	R/W	On-chip RAM Power Off 0 RAMCUT 2, 1, and 0 control the internal power s the on-chip RAM and USB in deep software star mode. RAMCUT 2 to 0 000: Power is supplied to the on-chip RAM and U 111: Power is not supplied to the on-chip RAM a Settings other than above are prohibited.

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Module
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.

During the oscillation settling period, counting is
performed with the clock frequency input to the E

000000: Reserved

000001: Reserved

000010: Reserved

000011: Reserved

000100: Reserved

000101: Wait time = 64 states

000110: Wait time = 512 states

000111: Wait time = 1024 states

001000: Wait time = 2048 states

001001: Wait time = 4096 states

001010: Wait time = 16384 states

001011: Wait time = 32768 states

001100: Wait time = 65536 states

001101: Wait time = 131072 states

001110: Wait time = 262144 states

001111: Wait time = 524288 states

01xxxx: Reserved

[Legend]

x: Don't care

Note: * Supported only by the H8SX/1655M Group.

Bit	Bit Name	Initial Value	R/W	Module
7	—	0	R/W	Reserved This bit is always read as 0. The write value should be 0.
6	DUSBIE	0	R/W	USB Suspend/Resume Interrupt Enable Enables/disables exit from deep software standby the USB suspend/resume interrupt signal. 0: Disables exit from deep software standby mode USB suspend/resume interrupt signal. 1: Enables exit from deep software standby mode USB suspend/resume interrupt signal.
5	—	0	R/W	Reserved This bit is always read as 0. The write value should be 0.
4	DLVDIE*	0	R/W	LVD Interrupt Enable Enables/disables exit from deep software standby the voltage monitoring interrupt signal. 0: Disables exit from deep software standby mode voltage monitoring interrupt signal. 1: Enables exit from deep software standby mode voltage monitoring interrupt signal.

				Enables or disables exit from deep software standb by $\overline{\text{IRQ2}}$-A. 0: Disables exit from deep software standby mode b $\overline{\text{IRQ2}}$-A 1: Enables exit from deep software standby mode b $\overline{\text{IRQ2}}$-A
1	DIRQ1E	0	R/W	IRQ1 Interrupt Enable Enables or disables exit from deep software standby by $\overline{\text{IRQ1}}$-A. 0: Disables exit from deep software standby mode b $\overline{\text{IRQ1}}$-A. 1: Enables exit from deep software standby mode b $\overline{\text{IRQ1}}$-A.
0	DIRQ0E	0	R/W	IRQ0 Interrupt Enable Enables or disables exit from deep software standby by $\overline{\text{IRQ0}}$-A. 0: Disables exit from deep software standby mode b $\overline{\text{IRQ0}}$-A. 1: Enables exit from deep software standby mode b $\overline{\text{IRQ0}}$-A.

Note: * Supported only by the H8SX/1655M Group.

Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

Notes: 1. Only 0 can be written to clear the flag.
2. Supported only by the H8SX/1655M Group.

Bit	Bit Name	Initial Value	R/W	Module
7	DNMIF	0	R/(W)*1	NMI Flag [Setting condition] NMI input specified in DPSIEGR is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.
6	DUSBIF	0	R/(W)*1	USB Suspend/Resume Interrupt Flag [Setting condition] When the USB suspend/resume interrupt occurs [Clearing condition] Writing a 0 to this bit after reading it as 1.
5	—	0	R	Reserved This bit is always read as 0. The write value sho be 0.
4	DLVDIF*2	0	R/(W)*1	LVD Interrupt Flag [Setting condition] Voltage monitoring interrupt is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.

				[Clearing condition] Writing a 0 to this bit after reading it as 1.
1	DIRQ1F	0	R/(W)*[1]	IRQ1 Interrupt Flag [Setting condition] $\overline{\text{IRQ1}}$-A input specified in DPSIEGR is generated [Clearing condition] Writing a 0 to this bit after reading it as 1.
0	DIRQ0F	0	R/(W)*[1]	IRQ0 Interrupt Flag [Setting condition] $\overline{\text{IRQ0}}$-A input specified in DPSIEGR is generated [Clearing condition] Writing a 0 to this bit after reading it as 1.

Notes: 1. Only 0 can be written to clear the flag.
2. Supported only by the H8SX/1655M Group.

R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W

Bit	Bit Name	Initial Value	R/W	Module
7	DNMIEG	0	R/W	NMI Edge Select Selects the active edge for NMI pin input. 0: The interrupt request is generated by a fallin 1: The interrupt request is generated by a risin
6 to 4	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
3	DIRQ3EG	0	R/W	IRQ3 Interrupt Edge Select Selects the active edge for $\overline{IRQ3}$-A pin input. 0: The interrupt request is generated by a fallin 1: The interrupt request is generated by a risin
2	DIRQ2EG	0	R/W	IRQ2 Interrupt Edge Select Selects the active edge for $\overline{IRQ2}$-A pin input. 0: The interrupt request is generated by a fallin 1: The interrupt request is generated by a risin
1	DIRQ1EG	0	R/W	IRQ1 Interrupt Edge Select Selects the active edge for $\overline{IRQ1}$-A pin input. 0: The interrupt request is generated by a fallin 1: The interrupt request is generated by a risin

The DPSRSTF bit in RSTSR indicates that deep software standby mode has been cancele
interrupt.

RSTSR is not initialized by the internal reset signal upon exit from deep software standby

Bit	7	6	5	4	3	2	1	
Bit name	DPSRSTF	—	—	—	—	LVDF*2	—	P
Initial value:	0	0	0	0	0	0*3	0*3	
R/W:	R/(W)*1	R/W	R/W	R/W	R/W	R/W*4	R/W	

Notes: 1. Only 0 can be written to clear the flag.
2. Supported only by the H8SX/1655M Group.
3. Initial value is undefined in the H8SX/1655M Group.
4. Only 0 can be written to clear the flag in the H8SX/1655M Group.
5. Readable only in the H8SX/1655M Group.

Bit	Bit Name	Initial Value	R/W	Module
7	DPSRSTF	0	R/(W)*	Deep Software Standby Reset Flag Indicates that deep software standby mode has canceled by an interrupt source specified in DP DPSIEGR and an internal reset is generated. [Setting condition] Deep software standby mode is canceled by ar source. [Clearing condition] Writing a 0 to this bit after reading it as 1.
6 to 3	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.

				This bit indicates that the voltage-detection circui detected a low voltage (Vcc at or below Vdet). For details, see section 5, Voltage Detection Circ
1	—	Undefined	R/W	Reserved These bits are always read as 0. The write value always be 0.
0	PORF	Undefined	R	Power-on Reset Flag This bit indicates that a power-on reset has been generated. For details, see section 4, Reset.

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

27.2.10 Deep Standby Backup Register (DPSBKRn)

DPSBKRn (n = 15 to 0) is a 16-bit readable/writable register to store data during deep s standby mode.

Although data in on-chip RAM is not retained in deep software standby mode, data in th is retained.

DPSBKRn (n = 15 to 0) is not initialized by the internal reset signal upon exit from deep standby mode.

Bit	7	6	5	4	3	2	1
Bit name	BKUPn7	BKUPn6	BKUPn5	BKUPn4	BKUPn3	BKUPn2	BKUPn1
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

n: 15 to 0

the operating clock specified by bits ICK2 to ICK0.

27.4 Module Stop State

Module stop functionality can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCRA, MSTPCRB, or MSTPCRC is set to 1, operation stops at the end of the bus cycle and a transition is made to a module stop state. CPU continues operating independently.

When the corresponding MSTP bit is cleared to 0, a module stop state is cleared and the starts operating at the end of the bus cycle. In a module stop state, the internal states of m other than the SCI are retained.

After the reset state is cleared, all modules other than the EXDMAC, DMAC, and DTC a chip RAM are placed in a module stop state.

The registers of the module for which the module stop state is selected cannot be read fro written to.

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, a voltage monitoring reset*, or a power-on reset*.

- Exit from sleep mode 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 maske
  CPU.

- Exit from sleep mode 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.

- Exit from sleep mode by $\overline{\text{STBY}}$ pin

  When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mo

- Exit from sleep mode by reset caused by watchdog timer overflow

  Sleep mode is exited by an internal reset caused by a watchdog timer overflow.

- Exit from voltage monitoring reset*

  Sleep mode is exited by a voltage monitoring reset of the voltage detection circuit.

- Exit from power-on reset*

  Sleep mode is exited by a power-on reset.

Note: * Supported only by the H8SX/1655M Group.

modules controlled by MSTPCRC (MSTPCRC[15:8] = H'FFFF).

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*[1], watchdog timer, and voltage detecti
circuit*[2]), and the CPU returns to the normal program execution state via the exception ha
state. All-module-clock-stop mode is not cleared if interrupts are disabled, if interrupts ot
NMI are masked on the CPU side, or if the relevant interrupt is designated as a DTC activ
source.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Notes: 1. Operation or stopping of the 8-bit timer can be selected by bits MSTPA9 and
in MSTPCRA.

2. Supported only by the H8SX/1655M Group.

mode the oscillator stops, allowing power consumption to be significantly reduced.

If the WDT is used in watchdog timer mode, it is impossible to make a transition to soft standby mode. The WDT should be stopped before the SLEEP instruction execution.

27.7.2 Exit from Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI or IRQ0 to IRQ11*[1]) or internal interrupt (voltage monitoring interrupt *[2] or USB suspend/resume), a voltage mo reset*[2], a power-on reset*[2] or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

1. Exit from software standby mode by interrupt

   When an NMI, IRQ0 to IRQ11*[1], or USB suspend/resume interrupt request signal is clock oscillation starts, and after the elapse of the time set in bits STS4 to STS0 in S stable clocks are supplied to the entire LSI, software standby mode is cleared, and in exception handling is started.

   When clearing software standby mode with an IRQ0 to IRQ11*[1] interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than IRQ0 to IRQ11*[1] is generated. Software standby mode cannot be cleared if the inter been masked on the CPU side or has been designated as a DTC activation source.

2. Exit from voltage monitoring reset*[2]

   When a voltage monitoring reset is generated by the fall of power-voltage, software mode is cleared and a clock oscillation starts. At the same time, a clock signal is sup throughout the LSI. After that, if power voltage rises, the voltage detection reset is re while the clock oscillation stabilization time is well kept. Thereafter, CPU starts the exception handling.

5. Exit from software standby mode by STBY pin

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Notes: 1. By setting the SSIn bit in SSIER to 1, IRQ0 to IRQ11 can be used as a softwa standby mode clearing source.

2. Supported only by the H8SX/1655M Group.

27.7.3 Setting Oscillation Settling Time after Exit from Software Standby Mode

Bits STS4 to STS0 in SBYCR should be set as described below.

1. Using a crystal resonator

   Set bits STS4 to STS0 so that the standby time is at least equal to the oscillation settli Table 27.2 shows the standby times for operating frequencies and settings of bits STS STS0.

2. Using an external clock

   A PLL circuit settling time is necessary. Refer to table 27.2 to set the standby time.

			1	0	512	14.6	20.5	25.6	39.4	51.2	64.0
				1	1024	29.3	41.0	51.2	78.8	102.4	128.
	1	0	0	0	2048	58.5	81.9	102.4	157.5	204.8	256.
				1	4096	0.12	0.16	0.20	0.32	0.41	0.51
			1	0	16384	0.47	0.66	0.82	1.26	1.64	2.05
				1	32768	0.94	1.31	1.64	2.52	3.28	4.10
		1	0	0	65536	1.87	2.62	3.28	5.04	6.55	8.19
				1	131072	3.74	5.24	6.55	10.08	13.11	16.3
			1	0	262144	7.49	10.49	13.11	20.16	26.21	32.7
				1	524288	14.98	20.97	26.21	40.33	52.43	65.5
1	0	0	0	0	Reserved	—	—	—	—	—	—

[Legend]

: Recommended setting when external clock is in use

: Recommended setting when crystal oscillator is in use

Note: * Pϕ is the output from the peripheral module frequency divider. The oscillation time, which includes a period where the oscillation by an oscillator is not stab depends on the resonator characteristics. The above figures are for referenc

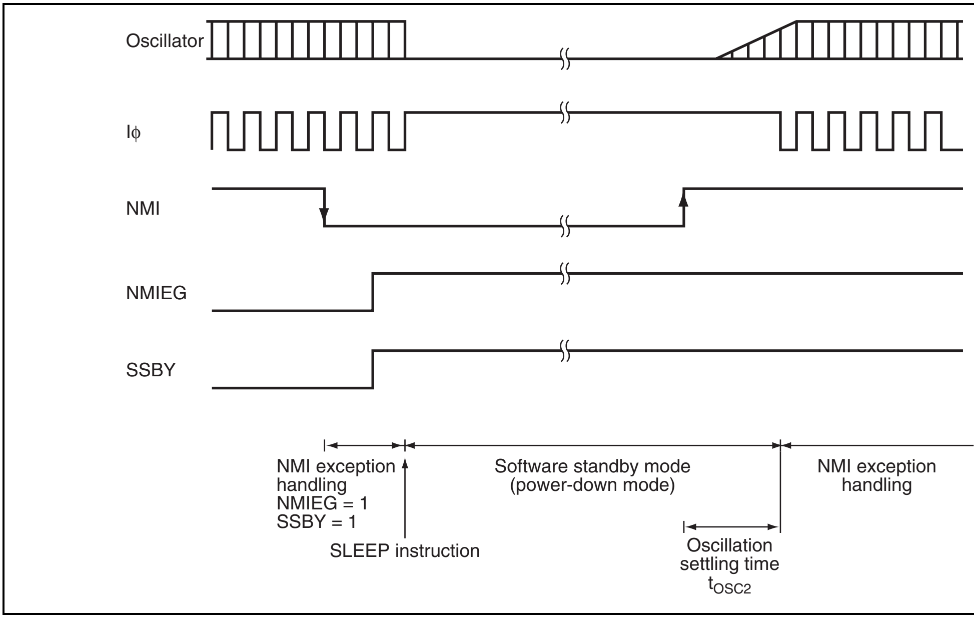


Figure 27.2 Software Standby Mode Application Example

software standby mode will be cleared regardless of the DPSBY bit setting, and the inte
exception handling starts after the oscillation settling time for software standby mode sp
the bits STS4 to STS0 in SBYCR has elapsed.

When both of the SSBY bit in SBYCR and the CPSBY bit in DPSBYCR are set to 1 an
software standby mode clearing source occurs, a transition to deep software standby mo
made immediately after software standby mode is entered.

In deep software standby mode, the CPU, on-chip peripheral functions (except for the U
chip RAM 4, and oscillator functionality are all halted. In addition, the internal power su
these modules stops, resulting in a significant reduction in power consumption. At this t
contents of all the registers of the CPU, on-chip peripheral functions (except for the US
chip RAM 4 become undefined.

Contents of the on-chip RAMs 3 to 0 and USB registers can be retained when all the bit
RAMCUT2 to RAMCUT0 in DPSBYCR have been cleared to 0. If these bits are set to
internal power supply to the on-chip RAMs 3 to 0 and USB stops and the power consum
further reduced. At this time, the contents of the on-chip RAMs 3 to 0 and USB register
undefined.

The voltage detection circuit*, and power-on reset circuit* can operate in deep software
mode.

The I/O ports can be retained in the same state as in software standby mode.

Note: * Supported only by the H8SX/1655M Group.

by the DIRQnE (n = 3 to 0) bit in DPSIER. The rising or falling edge of the signals c
specified with DPSIEGR. The DLVDIF bit is set to 1 when a voltage-monitoring inte
occurs. The DUSBIF bit is set to 1 when a USB suspend/resume interrupt occurs.

When deep software standby mode clearing source is generated, internal power suppl
simultaneously with the start of clock oscillation, and internal reset signal is generate
entire LSI. Once the time specified by the WTSTS5 to WTSTS0 bits in DPSWCR ha
a stable clock signal is being supplied throughout the LSI and the internal reset is clea
Deep software standby mode is canceled on clearing of the internal reset, and then the
exception handling starts.

When deep software standby mode is canceled by an external interrupt pin or internal
signal, the DPSRSTF bit in RSTSR is set to 1.

2. Exit from deep software standby mode by a voltage-monitoring reset*

   When a voltage monitoring reset is generated by the power-supply voltage falling, the
   released from deep software standby mode and internal power supply starts simultane
   with the start of clock oscillation. At the same time, a clock signal is supplied through
   LSI. When the power-supply voltage has risen sufficiently, the LSI is released from th
   voltage-detection reset state after the clock oscillation stabilization time has been secu
   CPU then starts reset-exception handling.

3. Exit from power-on reset*

   When a power-on reset is generated by the power-supply voltage falling, the LSI is re
   from deep software standby mode. If the power-supply voltage then rises sufficiently,
   oscillation starts and the LSI is released from the power-on reset state after the clock
   oscillation stabilization time has been secured. As soon as the clock oscillation starts,
   signal is provided to the LSI. The internal power supply restarts during the power-on
   time. After release from the power-on reset state, the CPU starts reset-exception hand

27.8.3 Pin State on Exit from Deep Software Standby Mode

In deep software standby mode, the ports retain the states that were held during software
mode. The internal of the LSI is initialized by an internal reset caused by deep software
mode, and the reset exception handling starts as soon as deep software standby mode is
The following shows the port states at this time.

(1) Pins for address bus, bus control and data bus

Pins for the address bus, bus control signals ($\overline{\text{CS0}}$, $\overline{\text{AS}}$, $\overline{\text{HWR}}$, and $\overline{\text{LWR}}$), and data bus
depending on the CPU.

(2) Pins other than address bus, bus control and data bus pins

Whether the ports are initialized or retain the states that were held during software stand
can be selected by the IOKEEP bit.

- When IOKEEP = 0

  Ports are initialized by an internal reset caused by deep software standby mode.
- When IOKEEP = 1

  The port states that were held in deep software standby mode are retained regardless
  internal state though the internal of the LSI is initialized by an internal reset caused
  software standby mode. At this time, the port states that were held in software stand
  are retained even if settings of I/O ports or peripheral modules are set. Subsequently
  retained port states are released when the IOKEEP bit is cleared to 0 and operation i
  performed according to the internal settings.

The IOKEEP bit is not initialized by an internal reset caused by canceling deep standby

1. Change the value of the PSTOP1 bit from 0 to 1 to fix the Bϕ output at the high level
   that the Bϕ output was already fixed high).
2. Clear the IOKEEP bit to 0 to end retention of the Bϕ state.
3. Clear the PSTOP1 bit to 0 to enable Bϕ output.

For the port state when the IOKEEP bit is set to 1, see section 27.8.3, Pin State on Exit fr
Software Standby Mode.

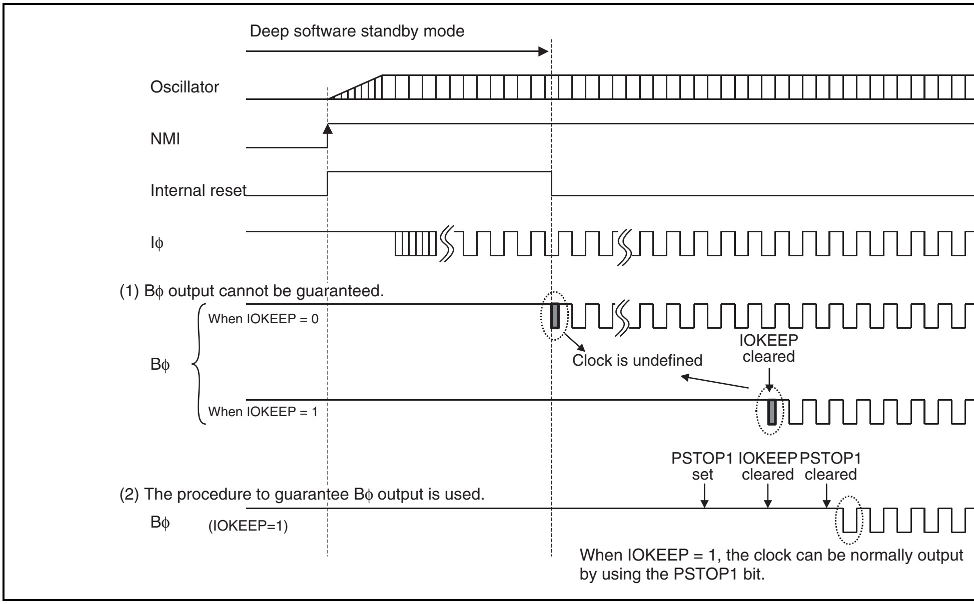


Figure 27.3 Bϕ Operation after Exit from Deep Software Standby Mode

				1	64	3.6	4.0	4.6	5.3	6.4	8.0
			1	0	512	28.4	32.0	36.6	42.7	51.2	64
				1	1024	56.9	64.0	73.1	85.3	102.4	12
	1	0	0	0	2048	113.8	128.0	146.3	170.7	204.8	25
				1	4096	0.23	0.26	0.29	0.34	0.41	0.5
			1	0	16384	0.91	1.02	1.17	1.37	1.64	2.0
				1	32768	1.82	2.05	2.34	2.73	3.28	4.1
		1	0	0	65536	3.64	4.10	4.68	5.46	6.55	8.1
				1	131072	7.28	8.19	9.36	10.92	13.11	16
			1	0	262144	14.56	16.38	18.72	21.85	26.21	32
				1	524288	29.13	32.77	37.45	43.69	52.43	65
1	0	0	0	0	Reserved	—	—	—	—	—	—

[Legend]

(light shading): Recommended setting when external clock is in use

(dark shading): Recommended setting when crystal oscillator is in use

Note: * The oscillation settling time, which includes a period where the oscillation by a oscillator is not stable, depends on the resonator characteristics. The above figures are for reference.

transition to deep software standby mode is triggered by execution of a SLEEP instructi

After that, deep software standby mode is canceled at the rising edge on the NMI pin.

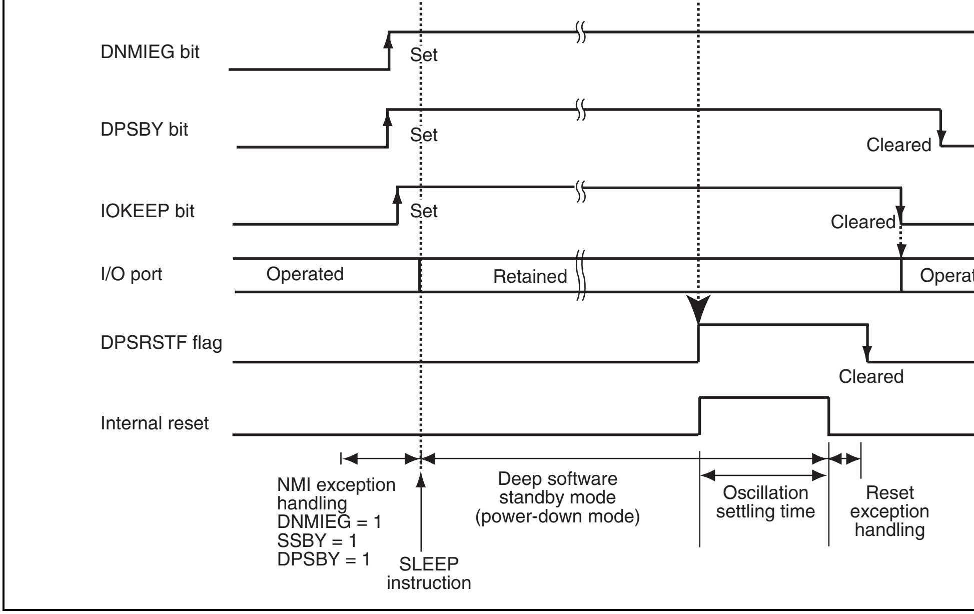


Figure 27.4 Deep Software Standby Mode Application Example (IOKEEP =

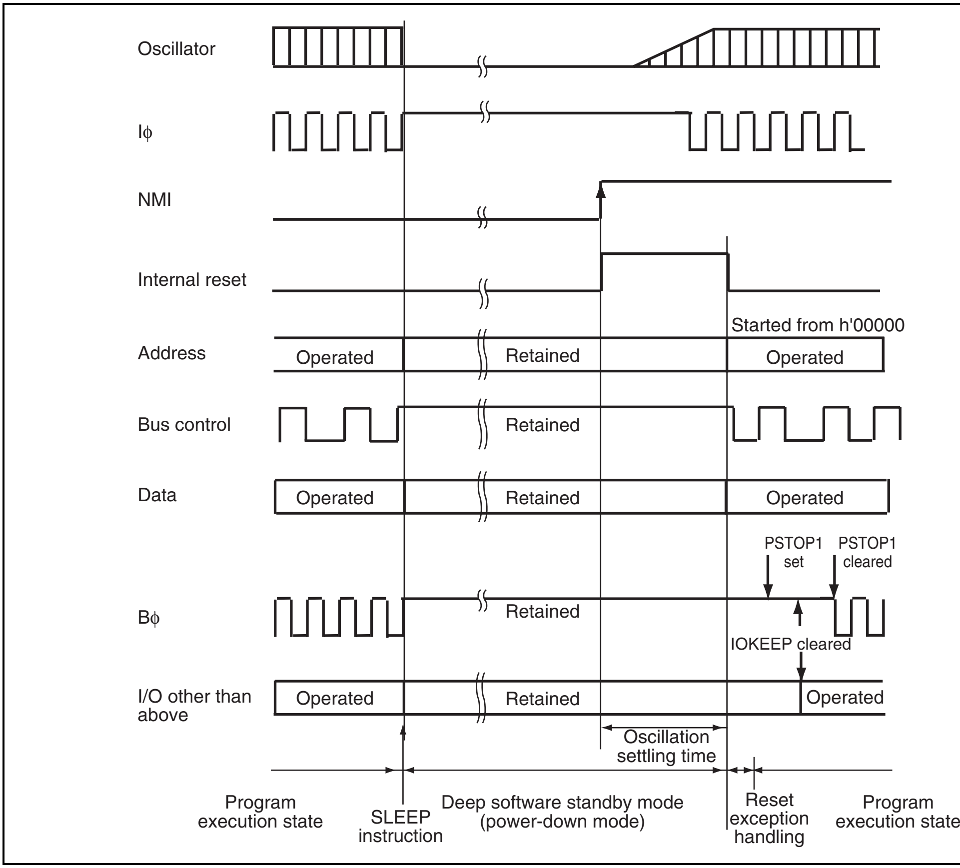


Figure 27.5 Example of Deep Software Standby Mode Operation in External Extended Mode (IOKEEP = OPE = 1)

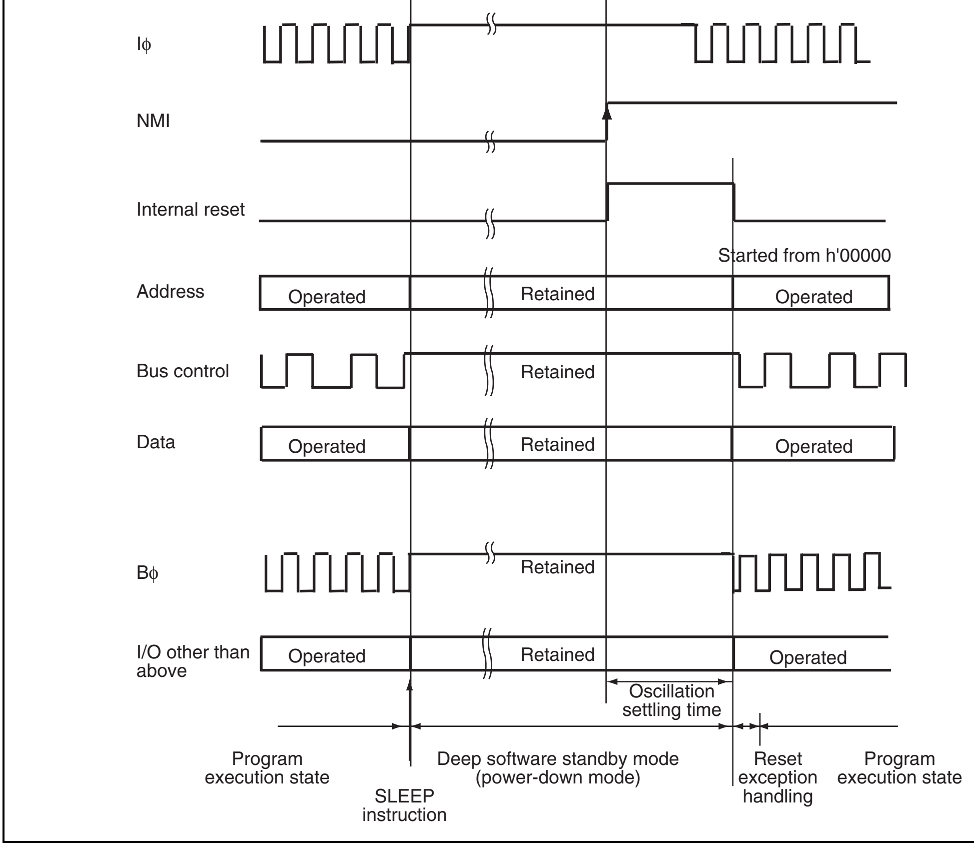


Figure 27.6 Example of Deep Software Standby Mode Operation in External Extended Mode (IOKEEP = OPE = 0)

output is also set.

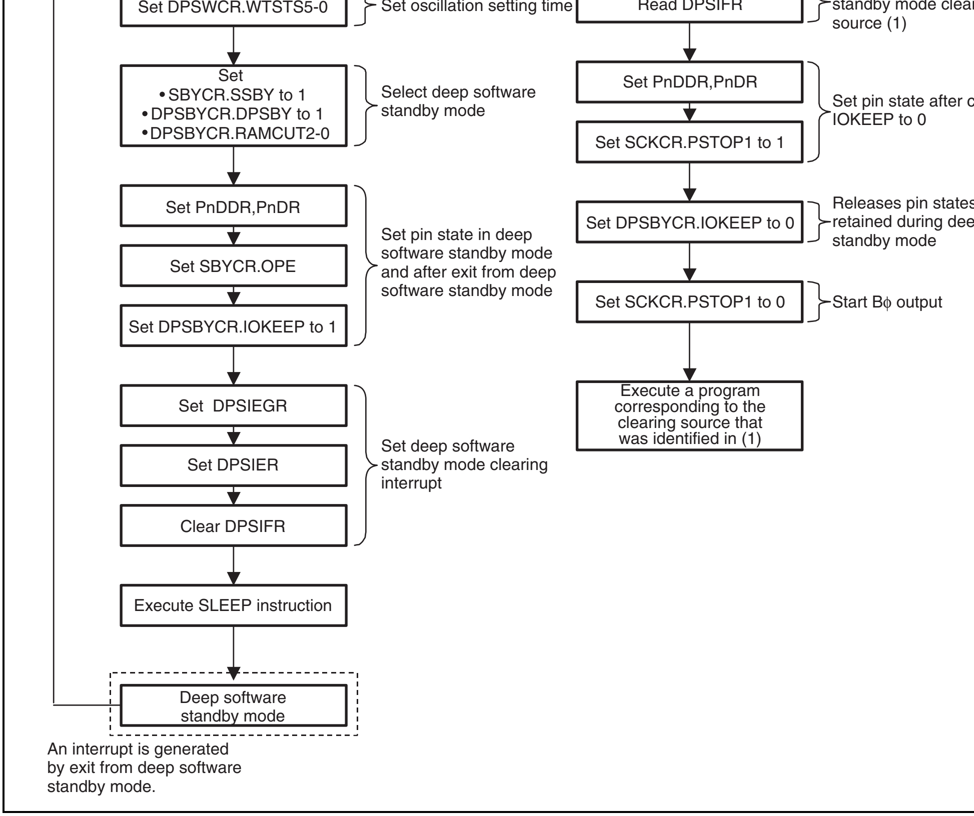


Figure 27.7 Flowchart of Deep Software Standby Mode Operation

Do not change the states of mode pins (MD2 to MD0) while this LSI is in hardware stan

27.9.2 Clearing Hardware Standby Mode

Hardware standby mode is cleared by means of the $\overline{STBY}$ pin and the $\overline{RES}$ pin. When th
pin is driven high while the $\overline{RES}$ pin is low, the reset state is entered and clock oscillatio
started. Ensure that the $\overline{RES}$ pin is held low until clock oscillation settles (for details on
oscillation settling time, refer to table 27.2). When the $\overline{RES}$ pin is subsequently driven h
transition is made to the program execution state via the reset exception handling state.

27.9.3 Hardware Standby Mode Timing

Figure 27.8 shows an example of hardware standby mode timing.

When the $\overline{STBY}$ pin is driven low after the $\overline{RES}$ pin has been driven low, a transition is
hardware standby mode. Hardware standby mode is cleared by driving the $\overline{STBY}$ pin hi
waiting for the oscillation settling time, then changing the $\overline{RES}$ pin from low to high.

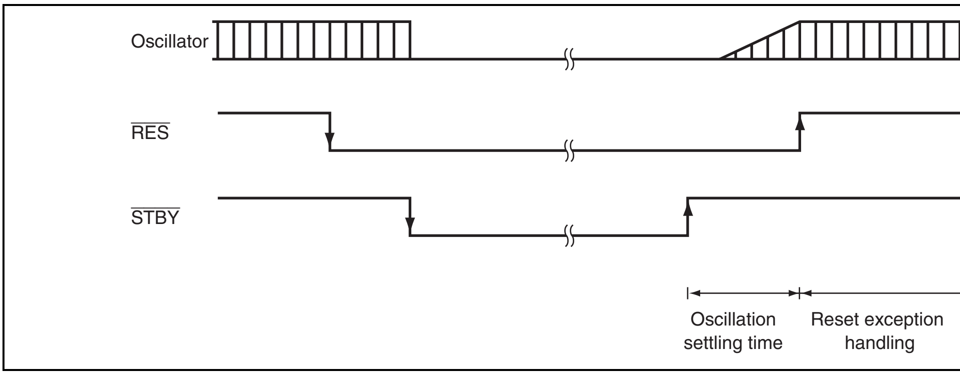


Figure 27.8 Hardware Standby Mode Timing

Timing.

In a power-on reset*, power on while driving the $\overline{STBY}$ or $\overline{RES}$ pin to a high-level.

Note: * Supported only by the H8SX/1655M Group.

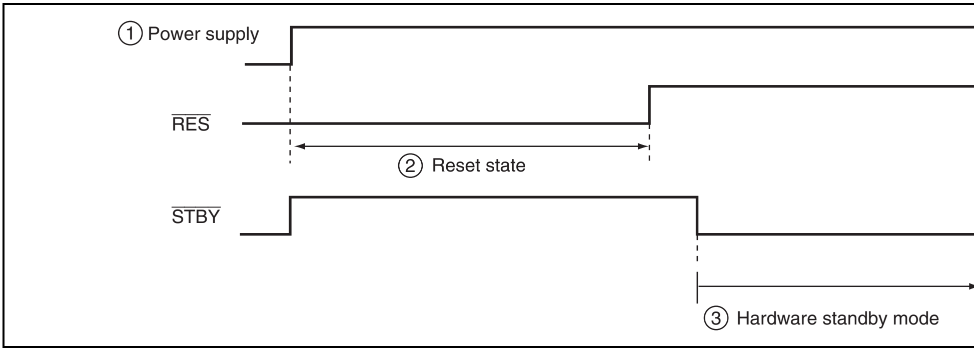


Figure 27.9 Timing Sequence at Power-On

When a SLEEP instruction is executed while the SLPIE bit is cleared to 0, a transition is
the power-down state. Exit from the power-down state is initiated by an exit-initiating in
source (see figure 27.10).

When an interrupt that causes exit from the power-down state is generated immediately
execution of a SLEEP instruction, exception handling for the interrupt starts. On return f
exception service routine, the SLEEP instruction is executed to enter the power-down st
case, exit from the power-down state will not take place until the next time an exit-initia
interrupt is generated (see figure 27.11).

As stated above, setting the SLPIE bit to 1 causes sleep instruction exception handling to
the execution of the SLEEP instruction. If this setting is made in the exception service ro
an interrupt that initiates exit from the power-down state, handling of the sleep instructio
exception due to the execution of a SLEEP instruction will proceed even if the interrupt
generated immediately beforehand (see figure 27.12). Consequently, the CPU will execu
instruction that follows the SLEEP instruction, after handling of the sleep instruction exc
and exception service routine, and will not enter the power-down state.

Thus, when the SLPIE bit is set to 1 to enable the sleep exception handling, clear the SS
SBYCR to 0.

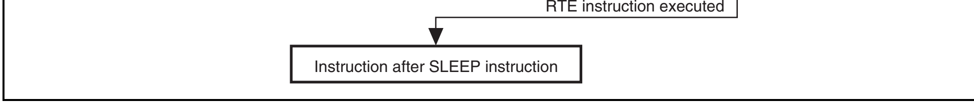


**Figure 27.10 When an Interrupt that Initiates Exit from the Power-Down Sta
is Generated after SLEEP Instruction Execution**

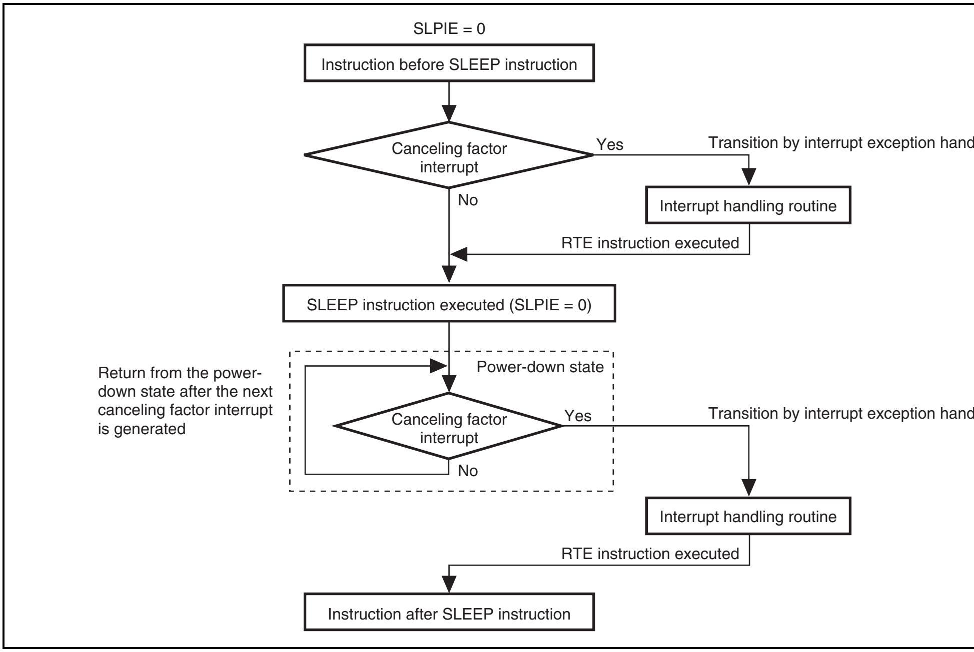


**Figure 27.11 When an Interrupt that Initiates Exit from the Power-Down St
is Generated before SLEEP Instruction Execution
(Sleep-Instruction Exception Handling does not Proceed)**

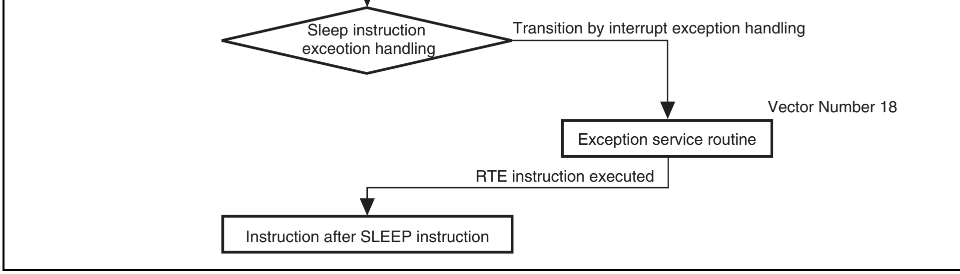


Figure 27.12 When an Interrupt that Initiates Exit from the Power-Down S is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Proceeds)

Register Setting Value		Normal Operating Mode	Sleep Mode	All-Module-Clock-Stop Mode	Software Standby Mode		Deep Software Standby Mode	
DDR	PSTOP1				OPE = 0	OPE = 1	IOKEEP = 0	IOKEEP = 1
0	x	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
1	0	Bϕ output	Bϕ output	Bϕ output	High	High	High	High
1	1	High	High	High	High	High	High	High

[Legend]

x = Don't care

Current consumption increases during the oscillation settling standby period.

27.12.3 Module Stop State of EXDMAC, DMAC, or DTC

Depending on the operating state of the EXDMAC, DMAC, and DTC, bits MSTPA14, MSTPA13, and MSTPA12 may not be set to 1, respectively. The module stop state setti EXDMAC, DMAC, or DTC should be carried out only when the EXDMAC, DMAC, or not activated.

For details, refer to section 10, DMA Controller (DMAC), section 11, EXDMA Control (EXDMAC), and section 12, Data Transfer Controller (DTC).

27.12.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in a module stop state. Consequently module stop state is entered when an interrupt has been requested, it will not be possible the CPU interrupt source or the DMAC or DTC activation source. Interrupts should ther disabled before entering a module stop state.

27.12.5 Writing to MSTPCRA, MSTPCRB, and MSTPCRC

MSTPCRA, MSTPCRB, and MSTPCRC should only be written to by the CPU.

If a conflict between a transition to deep software standby mode and generation of softwa
standby mode clearing source occurs, a transition to deep software standby mode is not m
the software standby mode clearing sequence is executed. In this case, an interrupt except
handling for the input interrupt starts after the oscillation settling time for software standb
(set by the STS4 to STS0 bits in SBYCR) has elapsed.

Note that if a conflict between a deep software standby mode transition and NMI interrup
the NMI interrupt exception handling routine is required.

If a conflict between transitions to deep software standby mode, the IRQ0 to IRQ11 inter
voltage-monitoring interrupt*occurs, a transition to deep software standby mode can be n
without executing the interrupt handling by clearing the SSIn bits in SSIER to 0 beforeha

Note: * Supported only by the H8SX/1655M Group.

27.12.8 Bϕ Output State

Bϕ output is undefined for a maximum of one cycle immediately after deep software stan
mode is canceled with the IOKEEP bit cleared to 0 or immediately after the IOKEEP bit
after cancellation of deep software standby mode with the IOKEEP bit set to 1.

However, Bϕ can be normally output by setting the IOKEEP and PSTOP1 bits. For detail
section 27.8.4, Bϕ Operation after Exit from Deep Software Standby Mode.

clock. For details, refer to section 9.5.4, 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 o
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 specifi
an on-chip peripheral module, refer to the section on that on-chip peripheral module.

Time constant registerA_5	TCORA_5	8	H'FEA45	TMR_5	16	3P
Time constant registerB_4	TCORB_4	8	H'FEA46	TMR_4	16	3P
Time constant registerB_5	TCORB_5	8	H'FEA47	TMR_5	16	3P
Timer counter_4	TCNT_4	8	H'FEA48	TMR_4	16	3P
Timer counter_5	TCNT_5	8	H'FEA49	TMR_5	16	3P
Timer counter control register_4	TCCR_4	8	H'FEA4A	TMR_4	16	3P
Timer counter control register_5	TCCR_5	8	H'FEA4B	TMR_5	16	3P
CRC control register	CRCCR	8	H'FEA4C	CRC	16	3P
CRC data input register	CRCDIR	8	H'FEA4D	CRC	16	3P
CRC data output register	CRCDOR	16	H'FEA4E	CRC	16	3P
Timer control register_6	TCR_6	8	H'FEA50	TMR_6	16	3P
Timer control register_7	TCR_7	8	H'FEA51	TMR_7	16	3P
Timer control/status register_6	TCSR_6	8	H'FEA52	TMR_6	16	3P
Timer control/status register_7	TCSR_7	8	H'FEA53	TMR_7	16	3P
Time constant registerA_6	TCORA_6	8	H'FEA54	TMR_6	16	3P
Time constant registerA_7	TCORA_7	8	H'FEA55	TMR_7	16	3P
Time constant registerB_6	TCORB_6	8	H'FEA56	TMR_6	16	3P
Time constant registerB_7	TCORB_7	8	H'FEA57	TMR_7	16	3P
Timer counter_6	TCNT_6	8	H'FEA58	TMR_6	16	3P
Timer counter_7	TCNT_7	8	H'FEA59	TMR_7	16	3P
Timer counter control register_6	TCCR_6	8	H'FEA5A	TMR_6	16	3P
Timer counter control register_7	TCCR_7	8	H'FEA5B	TMR_7	16	3P
A/D data register A_1	ADDRA_1	16	H'FEA80	A/D_1	16	3P
A/D data register B_1	ADDRB_1	16	H'FEA82	A/D_1	16	3P

A/D control register_1	ADCR_1	8	H'FEAA1	A/D_1	16	3F
A/D mode selection register_1	ADMOSEL_1	8	H'FEAA2	A/D_1	16	3F
A/D sampling state register_1	ADSSTR_1	8	H'FEAAB	A/D_1	16	3F
Interrupt flag register 0	IFR0	8	H'FEE00	USB	8	3F
Interrupt flag register 1	IFR1	8	H'FEE01	USB	8	3F
Interrupt flag register 2	IFR2	8	H'FEE02	USB	8	3F
Interrupt enable register 0	IER0	8	H'FEE04	USB	8	3F
Interrupt enable register 1	IER1	8	H'FEE05	USB	8	3F
Interrupt enable register 2	IER2	8	H'FEE06	USB	8	3F
Interrupt select register 0	ISR0	8	H'FEE08	USB	8	3F
Interrupt select register 1	ISR1	8	H'FEE09	USB	8	3F
Interrupt select register 2	ISR2	8	H'FEE0A	USB	8	3F
EP0i data register	EPDR0i	8	H'FEE0C	USB	8	3F
EP0o data register	EPDR0o	8	H'FEE0D	USB	8	3F
EP0s data register	EPDR0s	8	H'FEE0E	USB	8	3F
EP1 data register	EPDR1	8	H'FEE10	USB	8	3F
EP2 data register	EPDR2	8	H'FEE14	USB	8	3F
EP3 data register	EPDR3	8	H'FEE18	USB	8	3F
EP0o receive data size register	EPSZ0o	8	H'FEE24	USB	8	3F
EP1 receive data size register	EPSZ1	8	H'FEE25	USB	8	3F
Data status register	DASTS	8	H'FEE27	USB	8	3F
FIFO clear register	FCLR	8	H'FEE28	USB	8	3F
End point store register	EPSTL	8	H'FEE2A	USB	8	3F
Trigger register	TRG	8	H'FEE2C	USB	8	3F

Port M data register	PMDR	8	H'FEE51	I/O port	8	3P
Port M register	PORTM	8	H'FEE52	I/O port	8	3P
Port M input buffer control register	PMICR	8	H'FEE53	I/O port	8	3P
Serial mode register_5	SMR_5	8	H'FF600	SCI_5	8	3P
Bit rate register_5	BRR_5	8	H'FF601	SCI_5	8	3P
Serial control register_5	SCR_5	8	H'FF602	SCI_5	8	3P
Transmit data register_5	TDR_5	8	H'FF603	SCI_5	8	3P
Serial status register_5	SSR_5	8	H'FF604	SCI_5	8	3P
Receive data register_5	RDR_5	8	H'FF605	SCI_5	8	3P
Smart card mode register_5	SCMR_5	8	H'FF606	SCI_5	8	3P
Serial extended mode register_5	SEMR_5	8	H'FF608	SCI_5	8	3P
IrDA control register	IrCR	8	H'FF60C	SCI_5	8	3P
Serial mode register_6	SMR_6	8	H'FF610	SCI_6	8	3P
Bit rate register_6	BRR_6	8	H'FF611	SCI_6	8	3P
Serial control register_6	SCR_6	8	H'FF612	SCI_6	8	3P
Transmit data register_6	TDR_6	8	H'FF613	SCI_6	8	3P
Serial status register_6	SSR_6	8	H'FF614	SCI_6	8	3P
Receive data register_6	RDR_6	8	H'FF615	SCI_6	8	3P
Smart card mode register_6	SCMR_6	8	H'FF616	SCI_6	8	3P
Serial extended mode register_6	SEMR_6	8	H'FF618	SCI_6	8	3P
PPG output control register_1	PCR_1	8	H'FF636	PPG_1	8	3P
PPG output mode register_1	PMR_1	8	H'FF637	PPG_1	8	3P
Next data enable register H_1	NDERH_1	8	H'FF638	PPG_1	8	3P
Next data enable register L_1	NDERL_1	8	H'FF639	PPG_1	8	3P

Break address register AL	BARAL	16	H'FFA02	UBC	16	2I
Break address mask register AH	BAMRAH	16	H'FFA04	UBC	16	2I
Break address mask register AL	BAMRAL	16	H'FFA06	UBC	16	2I
Break address register BH	BARBH	16	H'FFA08	UBC	16	2I
Break address register BL	BARBL	16	H'FFA0A	UBC	16	2I
Break address mask register BH	BAMRBH	16	H'FFA0C	UBC	16	2I
Break address mask register BL	BAMRBL	16	H'FFA0E	UBC	16	2I
Break address register CH	BARCH	16	H'FFA10	UBC	16	2I
Break address register CL	BARCL	16	H'FFA12	UBC	16	2I
Break address mask register CH	BAMRCH	16	H'FFA14	UBC	16	2I
Break address mask register CL	BAMRCL	16	H'FFA16	UBC	16	2I
Break address register DH	BARDH	16	H'FFA18	UBC	16	2I
Break address register DL	BARDL	16	H'FFA1A	UBC	16	2I
Break address mask register DH	BAMRDH	16	H'FFA1C	UBC	16	2I
Break address mask register DL	BAMRDL	16	H'FFA1E	UBC	16	2I
Break control register A	BRCRA	16	H'FFA28	UBC	16	2I
Break control register B	BRCRB	16	H'FFA2C	UBC	16	2I
Break control register C	BRCRC	16	H'FFA30	UBC	16	2I
Break control register D	BRCRD	16	H'FFA34	UBC	16	2I
A/D sampling state register_0	ADSSTR_0	8	H'FEADB	A/D_0	16	2F
Timer start register	TSTRB	8	H'FFB00	TPU (unit 1)	16	2F
Timer synchronous register	TSYRB	8	H'FFB01	TPU (unit 1)	16	2F
Timer control register_6	TCR_6	8	H'FFB10	TPU_6	16	2F

Timer general register B_6	TGRB_6	16	H'FFB1A	TPU_6	16	2P
Timer general register C_6	TGRC_6	16	H'FFB1C	TPU_6	16	2P
Timer general register D_6	TGRD_6	16	H'FFB1E	TPU_6	16	2P
Timer control register_7	TCR_7	8	H'FFB20	TPU_7	16	2P
Timer mode register_7	TMDR_7	8	H'FFB21	TPU_7	16	2P
Timer I/O control register_7	TIOR_7	8	H'FFB22	TPU_7	16	2P
Timer interrupt enable register_7	TIER_7	8	H'FFB24	TPU_7	16	2P
Timer status register_7	TSR_7	8	H'FFB25	TPU_7	16	2P
Timer counter_7	TCNT_7	16	H'FFB26	TPU_7	16	2P
Timer general register A_7	TGRA_7	16	H'FFB28	TPU_7	16	2P
Timer general register B_7	TGRB_7	16	H'FFB2A	TPU_7	16	2P
Timer control register_8	TCR_8	8	H'FFB30	TPU_8	16	2P
Timer mode register_8	TMDR_8	8	H'FFB31	TPU_8	16	2P
Timer I/O control register_8	TIOR_8	8	H'FFB32	TPU_8	16	2P
Timer interrupt enable register_8	TIER_8	8	H'FFB34	TPU_8	16	2P
Timer status register_8	TSR_8	8	H'FFB35	TPU_8	16	2P
Timer counter_8	TCNT_8	16	H'FFB36	TPU_8	16	2P
Timer general register A_8	TGRA_8	16	H'FFB38	TPU_8	16	2P
Timer general register B_8	TGRB_8	16	H'FFB3A	TPU_8	16	2P
Timer control register_9	TCR_9	8	H'FFB40	TPU_9	16	2P
Timer mode register_9	TMDR_9	8	H'FFB41	TPU_9	16	2P
Timer I/O control register H_9	TIORH_9	8	H'FFB42	TPU_9	16	2P
Timer I/O control register L_9	TIORL_9	8	H'FFB43	TPU_9	16	2P
Timer interrupt enable register_9	TIER_9	8	H'FFB44	TPU_9	16	2P

Timer mode register_10	TMDR_10	8	H'FFB51	TPU_10	16	2P
Timer I/O control register_10	TIOR_10	8	H'FFB52	TPU_10	16	2P
Timer interrupt enable register_10	TIER_10	8	H'FFB54	TPU_10	16	2P
Timer status register_10	TSR_10	8	H'FFB55	TPU_10	16	2P
Timer counter_10	TCNT_10	16	H'FFB56	TPU_10	16	2P
Timer general register A_10	TGRA_10	16	H'FFB58	TPU_10	16	2P
Timer general register B_10	TGRB_10	16	H'FFB5A	TPU_10	16	2P
Timer control register_11	TCR_11	8	H'FFB60	TPU_11	16	2P
Timer mode register_11	TMDR_11	8	H'FFB61	TPU_11	16	2P
Timer I/O control register_11	TIOR_11	8	H'FFB62	TPU_11	16	2P
Timer interrupt enable register_11	TIER_11	8	H'FFB64	TPU_11	16	2P
Timer status register_11	TSR_11	8	H'FFB65	TPU_11	16	2P
Timer counter_11	TCNT_11	16	H'FFB66	TPU_11	16	2P
Timer general register A_11	TGRA_11	16	H'FFB68	TPU_11	16	2P
Timer general register B_11	TGRB_11	16	H'FFB6A	TPU_11	16	2P
Port 1 data direction register	P1DDR	8	H'FFB80	I/O port	8	2P
Port 2 data direction register	P2DDR	8	H'FFB81	I/O port	8	2P
Port 6 data direction register	P6DDR	8	H'FFB85	I/O port	8	2P
Port A data direction register	PADDR	8	H'FFB89	I/O port	8	2P
Port B data direction register	PBDDR	8	H'FFB8A	I/O port	8	2P
Port D data direction register	PDDDR	8	H'FFB8C	I/O port	8	2P
Port E data direction register	PEDDR	8	H'FFB8D	I/O port	8	2P
Port F data direction register	PFDDR	8	H'FFB8E	I/O port	8	2P
Port 1 input buffer control register	P1ICR	8	H'FFB90	I/O port	8	2P

Port F input buffer control register	PFICR	8	H'FFB9E	I/O port	8	2P
Port H register	PORTH	8	H'FFBA0	I/O port	8	2P
Port I register	PORTI	8	H'FFBA1	I/O port	8	2P
Port J register	PORTJ	8	H'FFBA2	I/O port	8	2P
Port K register	PORTK	8	H'FFBA3	I/O port	8	2P
Port H data register	PHDR	8	H'FFBA4	I/O port	8	2P
Port I data register	PIDR	8	H'FFBA5	I/O port	8	2P
Port J data register	PJDR	8	H'FFBA6	I/O port	8	2P
Port K data register	PKDR	8	H'FFBA7	I/O port	8	2P
Port H data direction register	PHDDR	8	H'FFBA8	I/O port	8	2P
Port I data direction register	PIDDR	8	H'FFBA9	I/O port	8	2P
Port J data direction register	PJDDR	8	H'FFBAA	I/O port	8	2P
Port K data direction register	PKDDR	8	H'FFBAB	I/O port	8	2P
Port H input buffer control register	PHICR	8	H'FFBAC	I/O port	8	2P
Port I input buffer control register	PIICR	8	H'FFBAD	I/O port	8	2P
Port J input buffer control register	PJICR	8	H'FFBAE	I/O port	8	2P
Port K input buffer control register	PKICR	8	H'FFBAF	I/O port	8	2P
Port D pull-up MOS control register	PDPCR	8	H'FFBB4	I/O port	8	2P
Port E pull-up MOS control register	PEPCR	8	H'FFBB5	I/O port	8	2P
Port F pull-up MOS control register	PFPCR	8	H'FFBB6	I/O port	8	2P
Port H pull-up MOS control register	PHPCR	8	H'FFBB8	I/O port	8	2P
Port I pull-up MOS control register	PIPCR	8	H'FFBB9	I/O port	8	2P
Port J pull-up MOS control register	PJPCR	8	H'FFBBA	I/O port	8	2P
Port K pull-up MOS control register	PKPCR	8	H'FFBBB	I/O port	8	2P

Port function control register 7	PFCR7	8	H'FFBC7	I/O port	8	2F
Port function control register 8	PFCR8	8	H'FFBC8	I/O port	8	2F
Port function control register 9	PFCR9	8	H'FFBC9	I/O port	8	2F
Port function control register A	PFCRA	8	H'FFBCA	I/O port	8	2F
Port function control register B	PFCRB	8	H'FFBCB	I/O port	8	2F
Port function control register C	PFCRC	8	H'FFBCC	I/O port	8	2F
Port function control register D	PFCRD	8	H'FFBCD	I/O port	8	2F
Software standby release IRQ enable register	SSIER	16	H'FFBCE	INTC	8	2F
Deep standby backup register 0	DPSBKR0	8	H'FFBF0	SYSTEM	8	2I
Deep standby backup register 1	DPSBKR1	8	H'FFBF1	SYSTEM	8	2I
Deep standby backup register 2	DPSBKR2	8	H'FFBF2	SYSTEM	8	2I
Deep standby backup register 3	DPSBKR3	8	H'FFBF3	SYSTEM	8	2I
Deep standby backup register 4	DPSBKR4	8	H'FFBF4	SYSTEM	8	2I
Deep standby backup register 5	DPSBKR5	8	H'FFBF5	SYSTEM	8	2I
Deep standby backup register 6	DPSBKR6	8	H'FFBF6	SYSTEM	8	2I
Deep standby backup register 7	DPSBKR7	8	H'FFBF7	SYSTEM	8	2I
Deep standby backup register 8	DPSBKR8	8	H'FFBF8	SYSTEM	8	2I
Deep standby backup register 9	DPSBKR9	8	H'FFBF9	SYSTEM	8	2I
Deep standby backup register 10	DPSBKR10	8	H'FFBFA	SYSTEM	8	2I
Deep standby backup register 11	DPSBKR11	8	H'FFBFB	SYSTEM	8	2I
Deep standby backup register 12	DPSBKR12	8	H'FFBFC	SYSTEM	8	2I
Deep standby backup register 13	DPSBKR13	8	H'FFBFD	SYSTEM	8	2I
Deep standby backup register 14	DPSBKR14	8	H'FFBFE	SYSTEM	8	2I

DMA address control register_0	DACR_0	32	H'FFC18	DMAC_0	16	2Iφ
DMA source address register_1	DSAR_1	32	H'FFC20	DMAC_1	16	2Iφ
DMA destination address register_1	DDAR_1	32	H'FFC24	DMAC_1	16	2Iφ
DMA offset register_1	DOFR_1	32	H'FFC28	DMAC_1	16	2Iφ
DMA transfer count register_1	DTCR_1	32	H'FFC2C	DMAC_1	16	2Iφ
DMA block size register_1	DBSR_1	32	H'FFC30	DMAC_1	16	2Iφ
DMA mode control register_1	DMDR_1	32	H'FFC34	DMAC_1	16	2Iφ
DMA address control register_1	DACR_1	32	H'FFC38	DMAC_1	16	2Iφ
DMA source address register_2	DSAR_2	32	H'FFC40	DMAC_2	16	2Iφ
DMA destination address register_2	DDAR_2	32	H'FFC44	DMAC_2	16	2Iφ
DMA offset register_2	DOFR_2	32	H'FFC48	DMAC_2	16	2Iφ
DMA transfer count register_2	DTCR_2	32	H'FFC4C	DMAC_2	16	2Iφ
DMA block size register_2	DBSR_2	32	H'FFC50	DMAC_2	16	2Iφ
DMA mode control register_2	DMDR_2	32	H'FFC54	DMAC_2	16	2Iφ
DMA address control register_2	DACR_2	32	H'FFC58	DMAC_2	16	2Iφ
DMA source address register_3	DSAR_3	32	H'FFC60	DMAC_3	16	2Iφ
DMA destination address register_3	DDAR_3	32	H'FFC64	DMAC_3	16	2Iφ
DMA offset register_3	DOFR_3	32	H'FFC68	DMAC_3	16	2Iφ
DMA transfer count register_3	DTCR_3	32	H'FFC6C	DMAC_3	16	2Iφ
DMA block size register_3	DBSR_3	32	H'FFC70	DMAC_3	16	2Iφ
DMA mode control register_3	DMDR_3	32	H'FFC74	DMAC_3	16	2Iφ
DMA address control register_3	DACR_3	32	H'FFC78	DMAC_3	16	2Iφ
EXDMA source address register_0	EDSAR_0	32	H'FFC80	EXDMAC_0	16	2Iφ

EXDMA source address register_1	EDSAR_1	32	H'FFCA0	EXDMAC_1	16	2I
EXDMA destination address register_1	EDDAR_1	32	H'FFCA4	EXDMAC_1	16	2I
EXDMA offset register_1	EDOFR_1	32	H'FFCA8	EXDMAC_1	16	2I
EXDMA transfer count register_1	EDTCR_1	32	H'FFCAC	EXDMAC_1	16	2I
EXDMA block size register_1	EDBSR_1	32	H'FFCB0	EXDMAC_1	16	2I
EXDMA mode control register_1	EDMDR_1	32	H'FFCB4	EXDMAC_1	16	2I
EXDMA address control register_1	EDACR_1	32	H'FFCB8	EXDMAC_1	16	2I
EXDMA source address register_2	EDSAR_2	32	H'FFCC0	EXDMAC_2	16	2I
EXDMA destination address register_2	EDDAR_2	32	H'FFCC4	EXDMAC_2	16	2I
EXDMA offset register_2	EDOFR_2	32	H'FFCC8	EXDMAC_2	16	2I
EXDMA transfer count register_2	EDTCR_2	32	H'FFCCC	EXDMAC_2	16	2I
EXDMA block size register_2	EDBSR_2	32	H'FFCD0	EXDMAC_2	16	2I
EXDMA mode control register_2	EDMDR_2	32	H'FFCD4	EXDMAC_2	16	2I
EXDMA address control register_2	EDACR_2	32	H'FFCD8	EXDMAC_2	16	2I
EXDMA source address register_3	EDSAR_3	32	H'FFCE0	EXDMAC_3	16	2I
EXDMA destination address register_3	EDDAR_3	32	H'FFCE4	EXDMAC_3	16	2I
EXDMA offset register_3	EDOFR_3	32	H'FFCE8	EXDMAC_3	16	2I
EXDMA transfer count register_3	EDTCR_3	32	H'FFCEC	EXDMAC_3	16	2I
EXDMA block size register_3	EDBSR_3	32	H'FFCF0	EXDMAC_3	16	2I
EXDMA mode control register_3	EDMDR_3	32	H'FFCF4	EXDMAC_3	16	2I
EXDMA address control register_3	EDACR_3	32	H'FFCF8	EXDMAC_3	16	2I
Cluster buffer register 0	CLSBR0	32	H'FFD00	EXDMAC	16	2I

DMA module request select register_0	DMRSR_0	8	H'FFD20	DMAC_0	16	2Iφ
DMA module request select register_1	DMRSR_1	8	H'FFD21	DMAC_1	16	2Iφ
DMA module request select register_2	DMRSR_2	8	H'FFD22	DMAC_2	16	2Iφ
DMA module request select register_3	DMRSR_3	8	H'FFD23	DMAC_3	16	2Iφ
Interrupt priority register A	IPRA	16	H'FFD40	INTC	16	2Iφ
Interrupt priority register B	IPRB	16	H'FFD42	INTC	16	2Iφ
Interrupt priority register C	IPRC	16	H'FFD44	INTC	16	2Iφ
Interrupt priority register E	IPRE	16	H'FFD48	INTC	16	2Iφ
Interrupt priority register F	IPRF	16	H'FFD4A	INTC	16	2Iφ
Interrupt priority register G	IPRG	16	H'FFD4C	INTC	16	2Iφ
Interrupt priority register H	IPRH	16	H'FFD4E	INTC	16	2Iφ
Interrupt priority register I	IPRI	16	H'FFD50	INTC	16	2Iφ
Interrupt priority register J	IPRJ	16	H'FFD52	INTC	16	2Iφ
Interrupt priority register K	IPRK	16	H'FFD54	INTC	16	2Iφ
Interrupt priority register L	IPRL	16	H'FFD56	INTC	16	2Iφ
Interrupt priority register M	IPRM	16	H'FFD58	INTC	16	2Iφ
Interrupt priority register N	IPRN	16	H'FFD5A	INTC	16	2Iφ
Interrupt priority register O	IPRO	16	H'FFD5C	INTC	16	2Iφ
Interrupt priority register Q	IPRQ	16	H'FFD60	INTC	16	2Iφ
Interrupt priority register R	IPRR	16	H'FFD62	INTC	16	2Iφ
IRQ sense control register H	ISCRH	16	H'FFD68	INTC	16	2Iφ
IRQ sense control register L	ISCRL	16	H'FFD6A	INTC	16	2Iφ
DTC vector base register	DTCVBR	32	H'FFD80	BSC	16	2Iφ
Bus width control register	ABWCR	16	H'FFD84	BSC	16	2Iφ

Bus control register 2	BCR2	8	H'FFD94	BSC	16	2I
Endian control register	ENDIANCR	8	H'FFD95	BSC	16	2I
SRAM mode control register	SRAMCR	16	H'FFD98	BSC	16	2I
Burst ROM interface control register	BROMCR	16	H'FFD9A	BSC	16	2I
Address/data multiplexed I/O control register	MPXCR	16	H'FFD9C	BSC	16	2I
RAM emulation register	RAMER	8	H'FFD9E	BSC	16	2I
Mode control register	MDCR	16	H'FFDC0	SYSTEM	16	2I
System control register	SYSCR	16	H'FFDC2	SYSTEM	16	2I
System clock control register	SCKCR	16	H'FFDC4	SYSTEM	16	2I
Standby control register	SBYCR	16	H'FFDC6	SYSTEM	16	2I
Module stop control register A	MSTPCRA	16	H'FFDC8	SYSTEM	16	2I
Module stop control register B	MSTPCRB	16	H'FFDCA	SYSTEM	16	2I
Module stop control register C	MSTPCRC	16	H'FFDCC	SYSTEM	16	2I
Flash code control/status register	FCCS	8	H'FFDE8	FLASH	16	2I
Flash program code select register	FPCS	8	H'FFDE9	FLASH	16	2I
Flash erase code select register	FECS	8	H'FFDEA	FLASH	16	2I
Flash key code register	FKEY	8	H'FFDEC	FLASH	16	2I
Flash MAT select register	FMATS	8	H'FFDED	FLASH	16	2I
Flash transfer destination address register	FTDAR	8	H'FFDEE	FLASH	16	2I
Deep standby control register	DPSBYCR	8	H'FFE70	SYSTEM	8	2I
Deep standby wait control register	DPSWCR	8	H'FFE71	SYSTEM	8	2I
Deep standby interrupt enable register	DPSIER	8	H'FFE72	SYSTEM	8	2I
Deep standby interrupt flag register	DPSIFR	8	H'FFE73	SYSTEM	8	2I

Serial control register_4	SCR_4	8	H'FFE92	SCI_4	8	2P
Transmit data register_4	TDR_4	8	H'FFE93	SCI_4	8	2P
Serial status register_4	SSR_4	8	H'FFE94	SCI_4	8	2P
Receive data register_4	RDR_4	8	H'FFE95	SCI_4	8	2P
Smart card mode register_4	SCMR_4	8	H'FFE96	SCI_4	8	2P
I^2C bus control register A_0	ICCRA_0	8	H'FFEB0	IIC2_0	8	2P
I^2C bus control register B_0	ICCRB_0	8	H'FFEB1	IIC2_0	8	2P
I^2C bus mode register_0	ICMR_0	8	H'FFEB2	IIC2_0	8	2P
I^2C bus interrupt enable register_0	ICIER_0	8	H'FFEB3	IIC2_0	8	2P
I^2C bus status register_0	ICSR_0	8	H'FFEB4	IIC2_0	8	2P
Slave address register_0	SAR_0	8	H'FFEB5	IIC2_0	8	2P
I^2C bus transmit data register_0	ICDRT_0	8	H'FFEB6	IIC2_0	8	2P
I^2C bus receive data register_0	ICDRR_0	8	H'FFEB7	IIC2_0	8	2P
I^2C bus control register A_1	ICCRA_1	8	H'FFEB8	IIC2_1	8	2P
I^2C bus control register B_1	ICCRB_1	8	H'FFEB9	IIC2_1	8	2P
I^2C bus mode register_1	ICMR_1	8	H'FFEBA	IIC2_1	8	2P
I^2C bus interrupt enable register_1	ICIER_1	8	H'FFEBB	IIC2_1	8	2P
I^2C bus status register_1	ICSR_1	8	H'FFEBC	IIC2_1	8	2P
Slave address register_1	SAR_1	8	H'FFEBD	IIC2_1	8	2P
I^2C bus transmit data register_1	ICDRT_1	8	H'FFEBE	IIC2_1	8	2P
I^2C bus receive data register_1	ICDRR_1	8	H'FFEBF	IIC2_1	8	2P
Timer control register_2	TCR_2	8	H'FFEC0	TMR_2	16	2P
Timer control register_3	TCR_3	8	H'FFEC1	TMR_3	16	2P
Timer control/status register_2	TCSR_2	8	H'FFEC2	TMR_2	16	2P

Timer counter control register_2	TCCR_2	8	H'FFECA	TMR_2	16	2P
Timer counter control register_3	TCCR_3	8	H'FFECB	TMR_3	16	2P
Timer control register_4	TCR_4	8	H'FFEE0	TPU_4	16	2P
Timer mode register_4	TMDR_4	8	H'FFEE1	TPU_4	16	2P
Timer I/O control register_4	TIOR_4	8	H'FFEE2	TPU_4	16	2P
Timer interrupt enable register_4	TIER_4	8	H'FFEE4	TPU_4	16	2P
Timer status register_4	TSR_4	8	H'FFEE5	TPU_4	16	2P
Timer counter_4	TCNT_4	16	H'FFEE6	TPU_4	16	2P
Timer general register A_4	TGRA_4	16	H'FFEE8	TPU_4	16	2P
Timer general register B_4	TGRB_4	16	H'FFEEA	TPU_4	16	2P
Timer control register_5	TCR_5	8	H'FFEF0	TPU_5	16	2P
Timer mode register_5	TMDR_5	8	H'FFEF1	TPU_5	16	2P
Timer I/O control register_5	TIOR_5	8	H'FFEF2	TPU_5	16	2P
Timer interrupt enable register_5	TIER_5	8	H'FFEF4	TPU_5	16	2P
Timer status register_5	TSR_5	8	H'FFEF5	TPU_5	16	2P
Timer counter_5	TCNT_5	16	H'FFEF6	TPU_5	16	2P
Timer general register A_5	TGRA_5	16	H'FFEF8	TPU_5	16	2P
Timer general register B_5	TGRB_5	16	H'FFEFA	TPU_5	16	2P
DTC enable register A	DTCERA	16	H'FFF20	INTC	16	2I
DTC enable register B	DTCERB	16	H'FFF22	INTC	16	2I
DTC enable register C	DTCERC	16	H'FFF24	INTC	16	2I
DTC enable register D	DTCERD	16	H'FFF26	INTC	16	2I
DTC enable register E	DTCERE	16	H'FFF28	INTC	16	2I
DTC enable register F	DTCERF	16	H'FFF2A	INTC	16	2I

Port 5 register	PORT5	8	H'FFF44	I/O port	8	2Pφ
Port 6 register	PORT6	8	H'FFF45	I/O port	8	2Pφ
Port A register	PORTA	8	H'FFF49	I/O port	8	2Pφ
Port B register	PORTB	8	H'FFF4A	I/O port	8	2Pφ
Port D register	PORTD	8	H'FFF4C	I/O port	8	2Pφ
Port E register	PORTE	8	H'FFF4D	I/O port	8	2Pφ
Port F register	PORTF	8	H'FFF4E	I/O port	8	2Pφ
Port 1 data register	P1DR	8	H'FFF50	I/O port	8	2Pφ
Port 2 data register	P2DR	8	H'FFF51	I/O port	8	2Pφ
Port 6 data register	P6DR	8	H'FFF55	I/O port	8	2Pφ
Port A data register	PADR	8	H'FFF59	I/O port	8	2Pφ
Port B data register	PBDR	8	H'FFF5A	I/O port	8	2Pφ
Port D data register	PDDR	8	H'FFF5C	I/O port	8	2Pφ
Port E data register	PEDR	8	H'FFF5D	I/O port	8	2Pφ
Port F data register	PFDR	8	H'FFF5E	I/O port	8	2Pφ
Serial mode register_2	SMR_2	8	H'FFF60	SCI_2	8	2Pφ
Bit rate register_2	BRR_2	8	H'FFF61	SCI_2	8	2Pφ
Serial control register_2	SCR_2	8	H'FFF62	SCI_2	8	2Pφ
Transmit data register_2	TDR_2	8	H'FFF63	SCI_2	8	2Pφ
Serial status register_2	SSR_2	8	H'FFF64	SCI_2	8	2Pφ
Receive data register_2	RDR_2	8	H'FFF65	SCI_2	8	2Pφ
Smart card mode register_2	SCMR_2	8	H'FFF66	SCI_2	8	2Pφ
D/A data register 0	DADR0H	8	H'FFF68	D/A	8	2Pφ
D/A data register 1	DADR1H	8	H'FFF69	D/A	8	2Pφ

Output data register L	PODRL	8	H'FFF7B	PPG_0	8	2F
Next data register H*[1]	NDRH	8	H'FFF7C	PPG_0	8	2F
Next data register L*[1]	NDRL	8	H'FFF7D	PPG_0	8	2F
Next data register H*[1]	NDRH	8	H'FFF7E	PPG_0	8	2F
Next data register L*[1]	NDRL	8	H'FFF7F	PPG_0	8	2F
Serial mode register_0	SMR_0	8	H'FFF80	SCI_0	8	2F
Bit rate register_0	BRR_0	8	H'FFF81	SCI_0	8	2F
Serial control register_0	SCR_0	8	H'FFF82	SCI_0	8	2F
Transmit data register_0	TDR_0	8	H'FFF83	SCI_0	8	2F
Serial status register_0	SSR_0	8	H'FFF84	SCI_0	8	2F
Receive data register_0	RDR_0	8	H'FFF85	SCI_0	8	2F
Smart card mode register_0	SCMR_0	8	H'FFF86	SCI_0	8	2F
Serial mode register_1	SMR_1	8	H'FFF88	SCI_1	8	2F
Bit rate register_1	BRR_1	8	H'FFF89	SCI_1	8	2F
Serial control register_1	SCR_1	8	H'FFF8A	SCI_1	8	2F
Transmit data register_1	TDR_1	8	H'FFF8B	SCI_1	8	2F
Serial status register_1	SSR_1	8	H'FFF8C	SCI_1	8	2F
Receive data register_1	RDR_1	8	H'FFF8D	SCI_1	8	2F
Smart card mode register_1	SCMR_1	8	H'FFF8E	SCI_1	8	2F
A/D data register A_0	ADDRA_0	16	H'FFF90	A/D_0	16	2F
A/D data register B_0	ADDRB_0	16	H'FFF92	A/D_0	16	2F
A/D data register C_0	ADDRC_0	16	H'FFF94	A/D_0	16	2F
A/D data register D_0	ADDRD_0	16	H'FFF96	A/D_0	16	2F
A/D data register E_0	ADDRE_0	16	H'FFF98	A/D_0	16	2F

Timer counter	TCNT	8	H'FFFA5	WDT	16	2P
Reset control/status register	RSTCSR	8	H'FFFA7	WDT	16	2P
Timer control register_0	TCR_0	8	H'FFFB0	TMR_0	16	2P
Timer control register_1	TCR_1	8	H'FFFB1	TMR_1	16	2P
Timer control/status register_0	TCSR_0	8	H'FFFB2	TMR_0	16	2P
Timer control/status register_1	TCSR_1	8	H'FFFB3	TMR_1	16	2P
Time constant register A_0	TCORA_0	8	H'FFFB4	TMR_0	16	2P
Time constant register A_1	TCORA_1	8	H'FFFB5	TMR_1	16	2P
Time constant register B_0	TCORB_0	8	H'FFFB6	TMR_0	16	2P
Time constant register B_1	TCORB_1	8	H'FFFB7	TMR_1	16	2P
Timer counter_0	TCNT_0	8	H'FFFB8	TMR_0	16	2P
Timer counter_1	TCNT_1	8	H'FFFB9	TMR_1	16	2P
Timer counter control register_0	TCCR_0	8	H'FFFBA	TMR_0	16	2P
Timer counter control register_1	TCCR_1	8	H'FFFBB	TMR_1	16	2P
Timer start register	TSTR	8	H'FFFBC	TPU	16	2P
Timer synchronous register	TSYR	8	H'FFFBD	TPU	16	2P
Timer control register_0	TCR_0	8	H'FFFC0	TPU_0	16	2P
Timer mode register_0	TMDR_0	8	H'FFFC1	TPU_0	16	2P
Timer I/O control register H_0	TIORH_0	8	H'FFFC2	TPU_0	16	2P
Timer I/O control register L_0	TIORL_0	8	H'FFFC3	TPU_0	16	2P
Timer interrupt enable register_0	TIER_0	8	H'FFFC4	TPU_0	16	2P
Timer status register_0	TSR_0	8	H'FFFC5	TPU_0	16	2P
Timer counter_0	TCNT_0	16	H'FFFC6	TPU_0	16	2P
Timer general register A_0	TGRA_0	16	H'FFFC8	TPU_0	16	2P

Timer status register_1	TSR_1	8	H'FFFD5	TPU_1	16	2F
Timer counter_1	TCNT_1	16	H'FFFD6	TPU_1	16	2F
Timer general register A_1	TGRA_1	16	H'FFFD8	TPU_1	16	2F
Timer general register B_1	TGRB_1	16	H'FFFDA	TPU_1	16	2F
Timer control register_2	TCR_2	8	H'FFFE0	TPU_2	16	2F
Timer mode register_2	TMDR_2	8	H'FFFE1	TPU_2	16	2F
Timer I/O control register_2	TIOR_2	8	H'FFFE2	TPU_2	16	2F
Timer interrupt enable register_2	TIER_2	8	H'FFFE4	TPU_2	16	2F
Timer status register_2	TSR_2	8	H'FFFE5	TPU_2	16	2F
Timer counter_2	TCNT_2	16	H'FFFE6	TPU_2	16	2F
Timer general register A_2	TGRA_2	16	H'FFFE8	TPU_2	16	2F
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

NDRL addresses for pulse output groups 4 and 5 are H'FF63F and H'FF63D, respectively.

2. Supported only by the H8SX/1655M Group.

TCSR_4	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_5	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_4								
TCORA_5								
TCORB_4								
TCORB_5								
TCNT_4								
TCNT_5								
TCCR_4	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_5	—	—	—	—	TMRIS	—	ICKS1	ICKS0
CRCCR	DORCLR	—	—	—	—	LMS	G1	G0
CRCDIR								
CRCDOR								
TCR_6	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_7	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_6	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_7	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_6								
TCORA_7								
TCORB_6								
TCORB_7								
TCNT_6								
TCNT_7								

ADDRD_1								
ADDRE_1								
ADDRF_1								
ADDRG_1								
ADDRH_1								
ADCSR_1	ADF	ADIE	ADST	EXCKS	CH3	CH2	CH1	CH0
ADCR_1	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS
ADMOSEL_1	—	—	—	—	—	—	ICKSEL	—
ADSSTR_1	SMP7	SMP6	SMP5	SMP4	SMP3	SMP2	SMP1	SMP0
IFR0	BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS
IFR1	—	—	—	—	VBUS MN	EP3 TR	EP3 TS	VBUSF
IFR2	—	—	SURSS	SURSF	CFDN	—	SETC	SETI
IER0	BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS
IER1	—	—	—	—	—	EP3 TR	EP3 TS	VBUSF
IER2	SSRSME	—	—	SURSE	CFDN	—	SETCE	SETIE
ISR0	BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS

EPDR3	D7	D6	D5	D4	D3	D2	D1	D0
EPSZ0o	—	—	—	—	—	—	—	—
EPSZ1	—	—	—	—	—	—	—	—
DASTS	—	—	EP3 DE	EP2 DE	—	—	—	EP0i DE
FCLR	—	EP3 CLR	EP1 CLR	EP2 CLR	—	—	EP0o CLR	EP0i CLR
EPSTL	—	—	—	—	EP3STL	EP2STL	EP1STL	EP0STL
TRG	—	EP3 PKTE	EP1 RDFN	EP2 PKTE	—	EP0s RDFN	EP0o RDFN	EP0i PKTE
DMA	—	—	—	—	—	PULLUP_E	EP2DMAE	EP1DMAE
CVR	CNFV1	CNFV0	INTV1	INTV0	—	ALTV2	ALTV1	ALTV0
CTLR	—	—	—	RWUPS	RSME	RWMD	ASCE	—
EPIR	D7	D6	D5	D4	D3	D2	D1	D0
TRNTREG0	PTSTE	—	—	—	SUSPEND	txenl	txse0	txdata
TRNTREG1	—	—	—	—	—	xver_data	dpls	dmns
PMDDR	—	—	—	PM4DDR	PM3DDR	PM2DDR	PM1DDR	PM0DDR
PMDR	—	—	—	PM4DR	PM3DR	PM2DR	PM1DR	PM0DR
PORTM	—	—	—	PM4	PM3	PM2	PM1	PM0
PMICR	—	—	—	PM4ICR	PM3ICR	PM2ICR	PM1ICR	PM0ICR
SMR_5*[1]	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
	(GM)	(BLK)	(PE)	(O/$\overline{E}$)	(BCP1)	(BCP0)		
BRR_5								
SCR_5*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_5								

	(GM)	(BLK)	(PE)	(O/$\overline{E}$)	(BCP1)	(BCP0)		
BRR_6								
SCR_6*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_6								
SSR_6*[1]	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
				(ERS)				
RDR_6								
SCMR_6	—	—	—	—	SDIR	SINV	—	SMIF
SEMR_6	—	—	—	ABCS	ACS3	ACS2	ACS1	ACS0
PCR_1	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
PMR_1	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
NDERH_1	NDER31	NDER30	NDER29	NDER28	NDER27	NDER26	NDER25	NDER24
NDERL_1	NDER23	NDER22	NDER21	NDER20	NDER19	NDER18	NDER17	NDER16
PODRH_1	POD31	POD30	POD29	POD28	POD27	POD26	POD25	POD24
PODRL_1	POD23	POD22	POD21	POD20	POD19	POD18	POD17	POD16
NDRH_1*[2]	NDR31	NDR30	NDR29	NDR28	NDR27	NDR26	NDR25	NDR24
NDRL_1*[2]	NDR23	NDR22	NDR21	NDR20	NDR19	NDR18	NDR17	NDR16
NDRH_1*[2]	—	—	—	—	NDR27	NDR26	NDR25	NDR24
NDRL_1*[2]	—	—	—	—	NDR19	NDR18	NDR17	NDR16
BARAH	BARA31	BARA30	BARA29	BARA28	BARA27	BARA26	BARA25	BARA24
	BARA23	BARA22	BARA21	BARA20	BARA19	BARA18	BARA17	BARA16

	BARB23	BARB22	BARB21	BARB20	BARB19	BARB18	BARB17	BARB16
BARBL	BARB15	BARB14	BARB13	BARB12	BARB11	BARB10	BARB9	BARB8
	BARB7	BARB6	BARB5	BARB4	BARB3	BARB2	BARB1	BARB0
BAMRBH	BAMRB31	BAMRB30	BAMRB29	BAMRB28	BAMRB27	BAMRB26	BAMRB25	BAMRB2
	BAMRB23	BAMRB22	BAMRB21	BAMRB20	BAMRB19	BAMRB18	BAMRB17	BAMRB1
BAMRBL	BAMRB15	BAMRB14	BAMRB13	BAMRB12	BAMRB11	BAMRB10	BAMRB9	BAMRB8
	BAMRB7	BAMRB6	BAMRB5	BAMRB4	BAMRB3	BAMRB2	BAMRB1	BAMRB0
BARCH	BARC31	BARC30	BARC29	BARC28	BARC27	BARC26	BARC25	BARC24
	BARC23	BARC22	BARC21	BARC20	BARC19	BARC18	BARC17	BARC16
BARCL	BARC15	BARC14	BARC13	BARC12	BARC11	BARC10	BARC9	BARC8
	BARC7	BARC6	BARC5	BARC4	BARC3	BARC2	BARC1	BARC0
BAMRCH	BAMRC31	BAMRC30	BAMRC29	BAMRC28	BAMRC27	BAMRC26	BAMRC25	BAMRC2
	BAMRC23	BAMRC22	BAMRC21	BAMRC20	BAMRC19	BAMRC18	BAMRC17	BAMRC1
BAMRCL	BAMRC15	BAMRC14	BAMRC13	BAMRC12	BAMRC11	BAMRC10	BAMRC9	BAMRC8
	BAMRC7	BAMRC6	BAMRC5	BAMRC4	BAMRC3	BAMRC2	BAMRC1	BAMRC0
BARDH	BARD31	BARD30	BARD29	BARD28	BARD27	BARD26	BARD25	BARD24
	BARD23	BARD22	BARD21	BARD20	BARD19	BARD18	BARD17	BARD16
BARDL	BARD15	BARD14	BARD13	BARD12	BARD11	BARD10	BARD9	BARD8
	BARD7	BARD6	BARD5	BARD4	BARD3	BARD2	BARD1	BARD0
BRCRA	—	—	CMFCPA	—	CPA2	CPA1	CPA0	—
	—	—	IDA1	IDA0	RWA1	RWA0	—	—
BRCRB	—	—	CMFCPB	—	CPB2	CPB1	CPB0	—
	—	—	IDB1	IDB0	RWB1	RWB0	—	—
BRCRC	—	—	CMFCPC	—	CPC2	CPC1	CPC0	—
	—	—	IDC1	IDC0	RWC1	RWC0	—	—

TIORH_6	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIORL_6	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
TIER_6	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_6	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_6								
TGRA_6								
TGRB_6								
TGRC_6								
TGRD_6								
TCR_7	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_7	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_7	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_7	—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_7	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_7								
TGRA_7								

TCNT_8								
TGRA_8								
TGRB_8								
TCR_9	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_9	—	—	BFB	BFA	MD3	MD2	MD1	MD0
TIORH_9	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIORL_9	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
TIER_9	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_9	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_9								
TGRA_9								
TGRB_9								
TGRC_9								
TGRD_9								

TGRA_10								
TGRB_10								
TCR_11	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_11	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_11	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_11	—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_11	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_11								
TGRA_11								
TGRB_11								
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
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	—	—	—	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR

PEICR	PE7ICR	PE6ICR	PE5ICR	PE4ICR	PE3ICR	PE2ICR	PE1ICR	PE0ICR
PFICR	—	—	—	PF4ICR	PF3ICR	PF2ICR	PF1ICR	PF0ICR
PORTH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
PORTI	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0
PORTJ	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
PORTK	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
PHDR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
PIDR	PI7DR	PI6DR	PI5DR	PI4DR	PI3DR	PI2DR	PI1DR	PI0DR
PJDR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
PKDR	PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR
PHDDR	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR
PIDDR	PI7DDR	PI6DDR	PI5DDR	PI4DDR	PI3DDR	PI2DDR	PI1DDR	PI0DDR
PJDDR	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR
PKDDR	PK7DDR	PK6DDR	PK5DDR	PK4DDR	PK3DDR	PK2DDR	PK1DDR	PK0DDR
PHICR	PH7ICR	PH6ICR	PH5ICR	PH4ICR	PH3ICR	PH2ICR	PH1ICR	PH0ICR
PIICR	PI7ICR	PI6ICR	PI5ICR	PI4ICR	PI3ICR	PI2ICR	PI1ICR	PI0ICR
PJICR	PJ7ICR	PJ6ICR	PJ5ICR	PJ4ICR	PJ3ICR	PJ2ICR	PJ1ICR	PJ0ICR
PKICR	PK7ICR	PK6ICR	PK5ICR	PK4ICR	PK3ICR	PK2ICR	PK1ICR	PK0ICR
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
PFPCR	—	—	—	PF4PCR	PF3PCR	PF2PCR	PF1PCR	PF0PCR
PHPCR	PH7PCR	PH6PCR	PH5PCR	PH4PCR	PH3PCR	PH2PCR	PH1PCR	PH0PCR

PFCR2	—	CS2S	BSS	BSE	—	RDWRE	ASOE	—
PFCR4	—	—	—	A20E	A19E	A18E	A17E	A16E
PFCR6	—	LHWROE	—	—	TCLKS	—	—	—
PFCR7	DMAS3A	DMAS3B	DMAS2A	DMAS2B	DMAS1A	DMAS1B	DMAS0A	DMAS0B
PFCR8	—	—	—	—	EDMAS1A	EDMAS1B	EDMAS0A	EDMAS0B
PFCR9	TPUMS5	TPUMS4	TPUMS3A	TPUMS3B	—	—	—	—
PFCRA	TPUMS11	TPUMS10	TPUMS9A	TPUMS9B	TPUMS8	TPUMS7	TPUMS6A	TPUMS6B
PFCRB	—	ITS14	—	—	ITS11	ITS10	ITS9	ITS8
PFCRC	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0
PFCRD	PCJKE	—	—	—	—	—	—	—
SSIER	—	—	—	—	SSI11	SSI10	SSI9	SSI8
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0
DPSBKR0	DKUP07	DKUP06	DKUP05	DKUP04	DKUP03	DKUP02	DKUP01	DKUP00
DPSBKR1	DKUP17	DKUP16	DKUP15	DKUP14	DKUP13	DKUP12	DKUP11	DKUP10
DPSBKR2	DKUP27	DKUP26	DKUP25	DKUP24	DKUP23	DKUP22	DKUP21	DKUP20
DPSBKR3	DKUP37	DKUP36	DKUP35	DKUP34	DKUP33	DKUP32	DKUP31	DKUP30
DPSBKR4	DKUP47	DKUP46	DKUP45	DKUP44	DKUP43	DKUP42	DKUP41	DKUP40
DPSBKR5	DKUP57	DKUP56	DKUP55	DKUP54	DKUP53	DKUP52	DKUP51	DKUP50
DPSBKR6	DKUP67	DKUP66	DKUP65	DKUP64	DKUP63	DKUP62	DKUP61	DKUP60
DPSBKR7	DKUP77	DKUP76	DKUP75	DKUP74	DKUP73	DKUP72	DKUP71	DKUP70
DPSBKR8	DKUP87	DKUP86	DKUP85	DKUP84	DKUP83	DKUP82	DKUP81	DKUP80
DPSBKR9	DKUP97	DKUP96	DKUP95	DKUP94	DKUP93	DKUP92	DKUP91	DKUP90
DPSBKR10	DKUP107	DKUP106	DKUP105	DKUP104	DKUP103	DKUP102	DKUP101	DKUP100

DDAR_0								
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

DDAR_1								
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

DDAR_2								
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

DDAR_3								
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

EDDAR_0								
EDOFR_0								
EDTCR_0								
EDBSR_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
EDMDR_0	DTE	EDACKE	ETENDE	EDRAKE	EDREQS	NRD	—	—
	ACT	—	—	—	ERRF	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	—	—	—	EDMAP2	DEMAP1	EDMAP0

EDDAR_1								
EDOFR_1								
EDTCR_1								
EDBSR_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
EDMDR_1	DTE	EDACKE	ETENDE	EDRAKE	EDREQS	NRD	—	—
	ACT	—	—	—	—	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	—	—	—	EDMAP2	DEMAP1	EDMAP0

EDDAR_2								
EDOFR_2								
EDTCR_2								
EDBSR_2	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH2
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH1
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
EDMDR_2	DTE	EDACKE	ETENDE	EDRAKE	EDREQS	NRD	—	—
	ACT	—	—	—	—	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	—	—	—	EDMAP2	DEMAP1	EDMAP0

EDDAR_3								
EDOFR_3								
EDTCR_3								
EDBSR_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
EDMDR_3	DTE	EDACKE	ETENDE	EDRAKE	EDREQS	NRD	—	—
	ACT	—	—	—	—	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	—	—	—	EDMAP2	DEMAP1	EDMAP0

CLSBR1	
CLSBR2	
CLSBR3	
CLSBR4	
CLSBR5	

DMRSR_0								
DMRSR_1								
DMRSR_2								
DMRSR_3								
IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0
IPRE	—	—	—	—	—	IPRE10	IPRE9	IPRE8
	—	—	—	—	—	—	—	—
IPRF	—	—	—	—	—	IPRF10	IPRF9	IPRF8
	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0
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
IPRJ	—	IPRIJ4	IPRJ13	IPRJ12	—	IPRJ10	IPRJ9	IPRJ8
	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0

	—	IPRN6	IPRN5	IPRN4	—	IPRN2	IPRN1	IPRN0
IPRO	—	IPRO14	IPRO13	IPRO12	—	IPRO10	IPRO9	IPRO8
	—	IPRO6	IPRO5	IPRO4	—	—	—	—
IPRQ	—	—	—	—	—	—	—	—
	—	IPRQ6	IPRQ5	IPRQ4	—	IPRQ2	IPRQ1	IPRQ0
IPRR	—	IPRR14	IPRR13	IPRR12	—	IPRR10	IPRR9	IPRR8
	—	IPRR6	IPRR5	IPRR4	—	IPRR2	IPRR1	IPRR0
ISCRH	—	—	—	—	—	—	—	—
	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR	IRQ8SF
ISCRL	IRQ7SR	IRQ7SF	IRQ6SR	IRQ6SF	IRQ5SR	IRQ5SF	IRQ4SR	IRQ4SF
	IRQ3SR	IRQ3SF	IRQ2SR	IRQ2SF	IRQ1SR	IRQ1SF	IRQ0SR	IRQ0SF
DTCVBR								
ABWCR	ABWH7	ABWH6	ABWH5	ABWH4	ABWH3	ABWH2	ABWH1	ABWH0
	ABWL7	ABWL6	ABWL5	ABWL4	ABWL3	ABWL2	ABWL1	ABWL0
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
	—	—	—	—	—	—	—	—
WTCRA	—	W72	W71	W70	—	W62	W61	W60
	—	W52	W51	W50	—	W42	W41	W40
WTCRB	—	W32	W31	W30	—	W22	W21	W20
	—	W12	W11	W10	—	W02	W01	W00

	DKC	—	—	—	—	—	—	—
BCR2	—	—	EBCCS	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
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0
MDCR	—	—	—	—	MDS3	MDS2	MDS1	MDS0
	—	—	—	—	—	—	—	—
SYSCR	—	—	MACS	—	FETCHMD	—	EXPE	RAME
	—	—	—	—	—	—	DTCMD	—
SCKCR	PSTOP1	—	—	—	—	ICK2	ICK1	ICK0
	—	PCK2	PCK1	PCK0	—	BCK2	BCK1	BCK0
SBYCR	SSBY	OPE	—	STS4	STS3	STS2	STS1	STS0
	SLPIE	—	—	—	—	—	—	—
MSTPCRA	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9	MSTPA8
	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
MSTPCRB	MSTPB15	MSTPB14	MSTPB13	MSTPB12	MSTPB11	MSTPB10	MSTPB9	MSTPB8
	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0

FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0
DPSBYCR	DPSBY	IOKEEP	RAMCUT2	RAMCUT1	—	—	—	RAMCUT
DPSWCR	—	—	WTSTS5	WTSTS4	WTSTS3	WTSTS2	WTSTS1	WTSTS0
DPSIER	—	DUSBIE	—	DLVDIE	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
DPSIFR	DNMIF	DUSBIF	—	DLVDIF	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
DPSIEGR	DNMIEG	—	—	—	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
RSTSR	DPSRSTF	—	—	—	—	LVDF	—	PORF
LVDCR*[3]	LVDE	LVDRI	—	LVDMON	—	—	—	—
SEMR_2	—	—	—	—	ABCS	ACS2	ACS1	ACS0
SMR_4*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	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
ICCRA_0	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0
ICCRB_0	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—
ICMR_0	—	WAIT	—	—	BCWP	BC2	BC1	BC0
ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	—

SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	—
ICDRT_1								
ICDRR_1								
TCR_2	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_3	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_2	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_3	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_2								
TCORA_3								
TCORB_2								
TCORB_3								
TCNT_2								
TCNT_3								
TCCR_2	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_3	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_4								

TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_5								
TGRA_5								
TGRB_5								
DTCERA	DTCEA15	DTCEA14	DTCEA13	DTCEA12	DTCEA11	DTCEA10	DTCEA9	DTCEA8
	DTCEA7	DTCEA6	DTCEA5	DTCEA4	—	—	—	—
DTCERB	DTCEB15	—	DTCEB13	DTCEB12	DTCEB11	DTCEB10	DTCEB9	DTCEB8
	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0
DTCERC	DTCEC15	DTCEC14	DTCEC13	DTCEC12	DTCEC11	DTCEC10	DTCEC9	DTCEC8
	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0
DTCERD	DTCED15	DTCED14	DTCED13	DTCED12	DTCED11	DTCED10	DTCED9	DTCED8
	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0
DTCERE	—	—	DTCEE13	DTCEE12	DTCEE11	DTCEE10	DTCEE9	DTCEE8
	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0
DTCERF	DTCEF15	DTCEF14	—	—	DTCEF11	DTCEF10	DTCEF9	—
	—	—	—	—	—	—	—	—
DTCCR	—	—	—	RRS	RCHNE	—	—	ERR
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—
CPUPCR	CPUPCE	DTCP2	DTCP1	DTCP0	IPSETE	CPUP2	CPUP1	CPUP0

PORT6	—	—	P65	P64	P63	P62	P61	P60
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PORTB	—	—	—	—	PB3	PB2	PB1	PB0
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PORTF	—	—	—	PF4	PF3	PF2	PF1	PF0
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
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	—	—	—	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

NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
NDRH*[2]	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
NDRL*[2]	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
NDRH*[2]	—	—	—	—	NDR11	NDR10	NDR9	NDR8
NDRL*[2]	—	—	—	—	NDR3	NDR2	NDR1	NDR0
SMR_0*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_0								
SCR_0*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_0								
SSR_0*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_0								
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF
SMR_1*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_1								
SCR_1*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1								
SSR_1*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT

ADDRD_0								
ADDRE_0								
ADDRF_0								
ADDRG_0								
ADDRH_0								
ADCSR_0	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0
ADCR_0	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	—	EXTRGS
ADMOSEL_0	—	—	—	—	—	—	ICKSEL	—
TCSR	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
TCNT								
RSTCSR	WOVF	RSTE	—	—	—	—	—	—
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_0								
TCORA_1								

TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_0								
TGRA_0								
TGRB_0								
TGRC_0								
TGRD_0								
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA

TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_2								
TGRA_2								
TGRB_2								
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_3	—	—	BFB	BFA	MD3	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	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_3								
TGRA_3								
TGRB_3								

respectively. Similarly, when the same output trigger is specified for pulse out
0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When different o
triggers are specified, the NDRL addresses for pulse output groups 0 and 1 a
H'FFF7F and H'FFF7D, respectively.

When the same output trigger is specified for pulse output groups 6 and 7 by
setting, the NDRH address is H'FF63C. When different output triggers are sp
NDRH addresses for pulse output groups 6 and 7 are H'FF63E and H'FF63C
respectively.

When the same output trigger is specified for pulse output groups 4 and 5 by
setting, the NDRL address is H'FF63D. When different output triggers are spe
NDRL addresses for pulse output groups 4 and 5 are H'FF63F and H'FF63D,
respectively.

3. Supported only by the H8SX/1655M Group.

TCORA_4	Initialized					Initialized	Initialized
TCORA_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORB_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORB_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCCR_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCCR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
CRCCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
CRCDIR	Initialized	—	—	—	—	Initialized*[1]	Initialized
CRCDOR	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_7	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCSR_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCSR_7	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORA_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORA_7	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORB_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORB_7	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_7	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCCR_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCCR_7	Initialized	—	—	—	—	Initialized*[1]	Initialized

ADDRG_1	Initialized	—	—	—	—	Initialized*	Initialized
ADDRH_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ADCSR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ADCR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ADMOSEL_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ADSSTR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
IFR0	Initialized	—	—	—	—	Initialized*[2]	Initialized
IFR1	Initialized	—	—	—	—	Initialized*[2]	Initialized
IFR2	Initialized	—	—	—	—	Initialized*[2]	Initialized
IER0	Initialized	—	—	—	—	Initialized*[2]	Initialized
IER1	Initialized	—	—	—	—	Initialized*[2]	Initialized
IER2	Initialized	—	—	—	—	Initialized*[2]	Initialized
ISR0	Initialized	—	—	—	—	Initialized*[2]	Initialized
ISR1	Initialized	—	—	—	—	Initialized*[2]	Initialized
ISR2	Initialized	—	—	—	—	Initialized*[2]	Initialized
EPDR0i	Initialized	—	—	—	—	Initialized*[2]	Initialized
EPDR0o	Initialized	—	—	—	—	Initialized*[2]	Initialized
EPDR0s	Initialized	—	—	—	—	Initialized*[2]	Initialized
EPDR1	Initialized	—	—	—	—	Initialized*[2]	Initialized
EPDR2	Initialized	—	—	—	—	Initialized*[2]	Initialized
EPDR3	Initialized	—	—	—	—	Initialized*[2]	Initialized
EPSZ0o	Initialized	—	—	—	—	Initialized*[2]	Initialized
EPSZ1	Initialized	—	—	—	—	Initialized*[2]	Initialized

CTLR	Initialized	—	—	—	—	Initialized*	Initialized
EPIR	Initialized	—	—	—	—	Initialized*[2]	Initialized
TRNTREG0	Initialized	—	—	—	—	Initialized*[2]	Initialized
TRNTREG1	Initialized	—	—	—	—	Initialized*[2]	Initialized
PMDDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PMDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PORTM	—	—	—	—	—	—	—
PMICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
SMR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
SCR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TDR_5	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SSR_5	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
RDR_5	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SCMR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
SEMR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
IrCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
SMR_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRR_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
SCR_6	Initialized	—	—	—	—	Initialized*[1]	Initialized
TDR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SSR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
RDR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized

PODRH_1	Initialized	—	—	—	—	Initialized*	Initialized
PODRL_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
NDRH_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
NDRL_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
BARAH	Initialized	—	—	—	—	Initialized*[1]	Initialized
BARAL	Initialized	—	—	—	—	Initialized*[1]	Initialized
BAMRAH	Initialized	—	—	—	—	Initialized*[1]	Initialized
BAMRAL	Initialized	—	—	—	—	Initialized*[1]	Initialized
BARBH	Initialized	—	—	—	—	Initialized*[1]	Initialized
BARBL	Initialized	—	—	—	—	Initialized*[1]	Initialized
BAMRBH	Initialized	—	—	—	—	Initialized*[1]	Initialized
BAMRBL	Initialized	—	—	—	—	Initialized*[1]	Initialized
BARCH	Initialized	—	—	—	—	Initialized*[1]	Initialized
BARCL	Initialized	—	—	—	—	Initialized*[1]	Initialized
BAMRCH	Initialized	—	—	—	—	Initialized*[1]	Initialized
BAMRCL	Initialized	—	—	—	—	Initialized*[1]	Initialized
BARDH	Initialized	—	—	—	—	Initialized*[1]	Initialized
BARDL	Initialized	—	—	—	—	Initialized*[1]	Initialized
BAMRDH	Initialized	—	—	—	—	Initialized*[1]	Initialized
BAMRDL	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRCRA	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRCRB	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRCRC	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRCRD	Initialized	—	—	—	—	Initialized*[1]	Initialized

TIORL_6	Initialized	—	—	—	—	Initialized*	Initialized
TIER_6	Initialized	—	—	—	—	Initialized*1	Initialized
TSR_6	Initialized	—	—	—	—	Initialized*1	Initialized
TCNT_6	Initialized	—	—	—	—	Initialized*1	Initialized
TGRA_6	Initialized	—	—	—	—	Initialized*1	Initialized
TGRB_6	Initialized	—	—	—	—	Initialized*1	Initialized
TGRC_6	Initialized	—	—	—	—	Initialized*1	Initialized
TGRD_6	Initialized	—	—	—	—	Initialized*1	Initialized
TCR_7	Initialized	—	—	—	—	Initialized*1	Initialized
TMDR_7	Initialized	—	—	—	—	Initialized*1	Initialized
TIOR_7	Initialized	—	—	—	—	Initialized*1	Initialized
TIER_7	Initialized	—	—	—	—	Initialized*1	Initialized
TSR_7	Initialized	—	—	—	—	Initialized*1	Initialized
TCNT_7	Initialized	—	—	—	—	Initialized*1	Initialized
TGRA_7	Initialized	—	—	—	—	Initialized*1	Initialized
TGRB_7	Initialized	—	—	—	—	Initialized*1	Initialized
TCR_8	Initialized	—	—	—	—	Initialized*1	Initialized
TMDR_8	Initialized	—	—	—	—	Initialized*1	Initialized
TIOR_8	Initialized	—	—	—	—	Initialized*1	Initialized

TMDR_9	Initialized	—	—	—	—	Initialized*	Initialized
TIORH_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TIORL_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TIER_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TSR_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRA_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRB_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRC_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRD_9	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_10	Initialized	—	—	—	—	Initialized*[1]	Initialized
TMDR_10	Initialized	—	—	—	—	Initialized*[1]	Initialized
TIOR_10	Initialized	—	—	—	—	Initialized*[1]	Initialized
TIER_10	Initialized	—	—	—	—	Initialized*[1]	Initialized
TSR_10	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_10	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRA_10	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRB_10	Initialized	—	—	—	—	Initialized*[1]	Initialized

TGRA_11	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRB_11	Initialized	—	—	—	—	Initialized*[1]	Initialized
P1DDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P2DDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P6DDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PADDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PBDDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PDDDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PEDDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFDDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P1ICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P2ICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P5ICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P6ICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PAICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PBICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PDICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PEICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PORTH	—	—	—	—	—	—	—
PORTI	—	—	—	—	—	—	—
PORTJ	—	—	—	—	—	—	—
PORTK	—	—	—	—	—	—	—

PJDDR	Initialized	—	—	—	—	Initialized*	Initialized
PKDDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PHICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PIICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PJICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PKICR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PDPCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PEPCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFPCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PHPCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PIPCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PJPCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PKPCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P2ODR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFODR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFCR0	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFCR1	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFCR2	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFCR4	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFCR6	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFCR7	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFCR8	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFCR9	Initialized	—	—	—	—	Initialized*[1]	Initialized

DPSBKR1	Initialized	—	—	—	—	—	Initialized
DPSBKR2	Initialized	—	—	—	—	—	Initialized
DPSBKR3	Initialized	—	—	—	—	—	Initialized
DPSBKR4	Initialized	—	—	—	—	—	Initialized
DPSBKR5	Initialized	—	—	—	—	—	Initialized
DPSBKR6	Initialized	—	—	—	—	—	Initialized
DPSBKR7	Initialized	—	—	—	—	—	Initialized
DPSBKR8	Initialized	—	—	—	—	—	Initialized
DPSBKR9	Initialized	—	—	—	—	—	Initialized
DPSBKR10	Initialized	—	—	—	—	—	Initialized
DPSBKR11	Initialized	—	—	—	—	—	Initialized
DPSBKR12	Initialized	—	—	—	—	—	Initialized
DPSBKR13	Initialized	—	—	—	—	—	Initialized
DPSBKR14	Initialized	—	—	—	—	—	Initialized
DPSBKR15	Initialized	—	—	—	—	—	Initialized
DSAR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
DDAR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
DOFR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
DBSR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
DMDR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
DACR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized

DACR_1	Initialized	—	—	—	—	Initialized*	Initialized
DSAR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
DDAR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
DOFR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
DBSR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
DMDR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
DACR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
DSAR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
DDAR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
DOFR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
DBSR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
DMDR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
DACR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDSAR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDDAR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDOFR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDTCR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDBSR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDMDR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDACR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized

EDACR_1	Initialized	—	—	—	—	Initialized*	Initialized
EDSAR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDDAR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDOFR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDTCR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDBSR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDMDR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDACR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDSAR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDDAR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDOFR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDTCR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDBSR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDMDR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
EDACR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
CLSBR0	—	—	—	—	—	—	—
CLSBR1	—	—	—	—	—	—	—
CLSBR2	—	—	—	—	—	—	—
CLSBR3	—	—	—	—	—	—	—
CLSBR4	—	—	—	—	—	—	—
CLSBR5	—	—	—	—	—	—	—
CLSBR6	—	—	—	—	—	—	—
CLSBR7	—	—	—	—	—	—	—

IPRC	Initialized	—	—	—	—	Initialized*	Initialized
IPRD	Initialized	—	—	—	—	Initialized*1	Initialized
IPRE	Initialized	—	—	—	—	Initialized*1	Initialized
IPRF	Initialized	—	—	—	—	Initialized*1	Initialized
IPRG	Initialized	—	—	—	—	Initialized*1	Initialized
IPRH	Initialized	—	—	—	—	Initialized*1	Initialized
IPRI	Initialized	—	—	—	—	Initialized*1	Initialized
IPRJ	Initialized	—	—	—	—	Initialized*1	Initialized
IPRK	Initialized	—	—	—	—	Initialized*1	Initialized
IPRL	Initialized	—	—	—	—	Initialized*1	Initialized
IPRM	Initialized	—	—	—	—	Initialized*1	Initialized
IPRN	Initialized	—	—	—	—	Initialized*1	Initialized
IPRO	Initialized	—	—	—	—	Initialized*1	Initialized
IPRQ	Initialized	—	—	—	—	Initialized*1	Initialized
IPRR	Initialized	—	—	—	—	Initialized*1	Initialized
ISCRH	Initialized	—	—	—	—	Initialized*1	Initialized
ISCRL	Initialized	—	—	—	—	Initialized*1	Initialized
DTCVBR	Initialized	—	—	—	—	Initialized*1	Initialized
ABWCR	Initialized	—	—	—	—	Initialized*1	Initialized
ASTCR	Initialized	—	—	—	—	Initialized*1	Initialized
WTCRA	Initialized	—	—	—	—	Initialized*1	Initialized
WTCRB	Initialized	—	—	—	—	Initialized*1	Initialized
RDNCR	Initialized	—	—	—	—	Initialized*1	Initialized

BROMCR	Initialized	—	—	—	—	Initialized*	Initialized
MPXCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
RAMER	Initialized	—	—	—	—	Initialized*[1]	Initialized
MDCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
SYSCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
SCKCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
SBYCR	Initialized	—	—	—	—	Initialized*[1]	Initialized
MSTPCRA	Initialized	—	—	—	—	Initialized*[1]	Initialized
MSTPCRB	Initialized	—	—	—	—	Initialized*[1]	Initialized
MSTPCRC	Initialized	—	—	—	—	Initialized*[1]	Initialized
FCCS	Initialized	—	—	—	—	Initialized*[1]	Initialized
FPCS	Initialized	—	—	—	—	Initialized*[1]	Initialized
FEC	Initialized	—	—	—	—	Initialized*[1]	Initialized
FKEY	Initialized	—	—	—	—	Initialized*[1]	Initialized
FMATS	Initialized	—	—	—	—	Initialized*[1]	Initialized
FTDAR	Initialized	—	—	—	—	Initialized*[1]	Initialized
DPSBYCR	Initialized	—	—	—	—	—	Initialized
DPSWCR	Initialized	—	—	—	—	—	Initialized
DPSIER	Initialized	—	—	—	—	—	Initialized
DPSIFR	Initialized	—	—	—	—	—	Initialized
DPSIEGR	Initialized	—	—	—	—	—	Initialized
RSTSR	Initialized	—	—	—	—	—	Initialized
LVDCR*[3]	Initialized*[4]	—	—	—	—	—	Initialized

RDR_4	Initialized	Initialized	—	Initialized	Initialized	Initialized*	Initialized
SCMR_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICCRA_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICCRB_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICMR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICIER_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICSR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
SAR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICDRT_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICDRR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICCRA_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICCRB_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICMR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICIER_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICSR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
SAR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICDRT_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
ICDRR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCSR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCSR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORA_2	Initialized	—	—	—	—	Initialized*[1]	Initialized

TCCR_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TMDR_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TIOR_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TIER_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TSR_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRA_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRB_4	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TMDR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TIOR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TIER_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TSR_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRA_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRB_5	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCERA	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCERB	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCERC	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCERD	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCERE	Initialized	—	—	—	—	Initialized*[1]	Initialized
DTCERF	Initialized	—	—	—	—	Initialized*[1]	Initialized

PORT2	—	—	—	—	—	—	—
PORT5	—	—	—	—	—	—	—
PORT6	—	—	—	—	—	—	—
PORTA	—	—	—	—	—	—	—
PORTB	—	—	—	—	—	—	—
PORTD	—	—	—	—	—	—	—
PORTE	—	—	—	—	—	—	—
PORTF	—	—	—	—	—	—	—
P1DR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P2DR	Initialized	—	—	—	—	Initialized*[1]	Initialized
P6DR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PADR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PBDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PDDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PEDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
PFDR	Initialized	—	—	—	—	Initialized*[1]	Initialized
SMR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
SCR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized
TDR_2	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SSR_2	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
RDR_2	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SCMR_2	Initialized	—	—	—	—	Initialized*[1]	Initialized

NDERH	Initialized	—	—	—	—	Initialized*[1]	Initialized
NDERL	Initialized	—	—	—	—	Initialized*[1]	Initialized
PODRH	Initialized	—	—	—	—	Initialized*[1]	Initialized
PODRL	Initialized	—	—	—	—	Initialized*[1]	Initialized
NDRH	Initialized	—	—	—	—	Initialized*[1]	Initialized
NDRL	Initialized	—	—	—	—	Initialized*[1]	Initialized
SMR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
SCR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
TDR_0	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SSR_0	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
RDR_0	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SCMR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
SMR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
BRR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
SCR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
TDR_1	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SSR_1	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
RDR_1	Initialized	Initialized	—	Initialized	Initialized	Initialized*[1]	Initialized
SCMR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized

ADDRG_0	Initialized	—	—	—	—	Initialized*	Initialized
ADDRH_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ADCSR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
ADCR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCSR	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT	Initialized	—	—	—	—	Initialized*[1]	Initialized
RSTCSR	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCSR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCSR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORA_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORA_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORB_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCORB_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCNT_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCCR_0	Initialized	—	—	—	—	Initialized*[1]	Initialized
TCCR_1	Initialized	—	—	—	—	Initialized*[1]	Initialized
TSTR	Initialized	—	—	—	—	Initialized*[1]	Initialized
TSYR	Initialized	—	—	—	—	Initialized*[1]	Initialized

TCNT_0	Initialized	—	—	—	—	Initialized*1	Initialized
TGRA_0	Initialized	—	—	—	—	Initialized*1	Initialized
TGRB_0	Initialized	—	—	—	—	Initialized*1	Initialized
TGRC_0	Initialized	—	—	—	—	Initialized*1	Initialized
TGRD_0	Initialized	—	—	—	—	Initialized*1	Initialized
TCR_1	Initialized	—	—	—	—	Initialized*1	Initialized
TMDR_1	Initialized	—	—	—	—	Initialized*1	Initialized
TIOR_1	Initialized	—	—	—	—	Initialized*1	Initialized
TIER_1	Initialized	—	—	—	—	Initialized*1	Initialized
TSR_1	Initialized	—	—	—	—	Initialized*1	Initialized
TCNT_1	Initialized	—	—	—	—	Initialized*1	Initialized
TGRA_1	Initialized	—	—	—	—	Initialized*1	Initialized
TGRB_1	Initialized	—	—	—	—	Initialized*1	Initialized
TCR_2	Initialized	—	—	—	—	Initialized*1	Initialized
TMDR_2	Initialized	—	—	—	—	Initialized*1	Initialized
TIOR_2	Initialized	—	—	—	—	Initialized*1	Initialized
TIER_2	Initialized	—	—	—	—	Initialized*1	Initialized
TSR_2	Initialized	—	—	—	—	Initialized*1	Initialized
TCNT_2	Initialized	—	—	—	—	Initialized*1	Initialized
TGRA_2	Initialized	—	—	—	—	Initialized*1	Initialized
TGRB_2	Initialized	—	—	—	—	Initialized*1	Initialized

TCNT_3	Initialized	—	—	—	—	Initialized*	Initialized
TGRA_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRB_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRC_3	Initialized	—	—	—	—	Initialized*[1]	Initialized
TGRD_3	Initialized	—	—	—	—	Initialized*[1]	Initialized

Notes: 1. Not initialized in deep software standby mode but initialized when deep softw
standby mode is released by the internal reset.

2. These registers are initialized when all the RAMCUT2 to RAMCUT0 bits in D
are set to 1, and not initialized when these bits are set to 0.

3. Supported only by the H8SX/1655M Group.

4. LVDCR is initialized by a pin reset or power-on reset not by a voltage-monito
deep software standby reset, or watchdog timer reset.

Input voltage (except for 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

Note: * The operating temperature range during programming/erasing of the flash me
0°C to +75°C for regular specifications and 0°C to +85°C for wide-range spec

trigger input voltage	TPU input pin, TMR input pin, port 2 port J, port K	VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.06$	—	—	V	
	$\overline{IRQ0}$-B to $\overline{IRQ7}$-B input pin	VT^-	$AV_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$AV_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$AV_{CC} \times 0.06$	—	—	V	
Input high voltage (except Schmitt trigger input pin)	MD, $\overline{RES}$, $\overline{STBY}$, EMLE, NMI	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL 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, $\overline{RES}$, $\overline{STBY}$, EMLE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, NMI		–0.3	—	$V_{CC} \times 0.2$		
	Other input 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} =
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	μA	V_{in} = V_{CC} –
	MD, $\overline{STBY}$, EMLE, NMI		—	—	1.0		
	Port 5		—	—	1.0		V_{in} = AV_{CC}

Supply current*[2]	Normal operation			I_{CC}*[3]	—	50	85	mA	f = 5
	Sleep mode				—	48	60		
	Standby mode	Software standby mode			—	0.15	1.1	mA	T_a ≤
					—	—	3.5		50°
		Deep software standby mode	RAM, USB retained		—	20	60	μA	T_a ≤
					—	—	200		50°
			RAM, USB power supply stopped		—	3	8		T_a ≤
					—	—	26		50°
		Hardware standby mode			—	2	7	μA	T_a ≤
					—	—	25		50°
	All-module-clock-stop mode*[4]				—	23	30	mA	
Analog power supply current	During A/D and D/A conversion			AI_{CC}	—	2.0	3.5	mA	
	Standby for A/D and D/A conversion				—	0.5	1.5	μA	
Reference power supply current	During A/D and D/A conversion			AI_{CC}	—	0.8	1.5	mA	
	Standby for A/D and D/A conversion				—	0.5	1.5	μA	
RAM standby voltage				V_{RAM}	2.5	—	—	V	

4. The values are for reference.
5. This applies when the $\overline{RES}$ pin is held low at power-on.

Table 29.3 Permissible Output Currents

Conditions: V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V
V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS} = 0 V*, T_a = –20°C to +75°C (regular specifi
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.
Permissible output low current (per pin)	Output pins	I_{OL}	—	—	2.0
Permissible output low current (total)	Total of 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 sh
be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

Schmitt trigger input voltage	IRQ input pin, TPU input pin, TMR input pin, port 2 port J, port K	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.06$	—	—	V	
	$\overline{IRQ0\text{-}B}$ to $\overline{IRQ7\text{-}B}$ input pin	VT^-	$AV_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$AV_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$AV_{CC} \times 0.06$	—	—	V	
Input high voltage (except Schmitt trigger input pin)	MD, $\overline{RES}$, $\overline{STBY}$, EMLE, NMI	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL 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, $\overline{RES}$, $\overline{STBY}$, EMLE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, NMI		–0.3	—	$V_{CC} \times 0.2$		
	Other input 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} =
Input leakage current	$\overline{RES}$	$\|I_{in}\|$	—	—	10.0	μA	V_{in} = V_{CC}
	MD, $\overline{STBY}$, EMLE, NMI		—	—	1.0		
	Port 5		—	—	1.0		V_{in} = AV_C

Supply current*[2]	Normal operation			I_{CC}*[3]	—	50	85	mA	f = 5
	Sleep mode				—	48	60		
	Standby mode	Software standby mode			—	0.15	1.1	mA	T_a ≤ 5
					—	—	3.5		50°C
		Deep software standby mode	RAM, USB retained		—	24	67	μA	T_a ≤ 5
					—	—	200		50°C
			RAM, USB power supply stopped		—	23	35		T_a ≤ 5
					—	—	60		50°C
		Hardware standby mode			—	2	7	μA	T_a ≤ 5
					—	—	25		50°C
	All-module-clock-stop mode*[4]				—	23	30	mA	
Analog power supply current	During A/D and D/A conversion			AI_{CC}	—	2.0	3.5	mA	
	Standby for A/D and D/A conversion				—	0.5	1.5	μA	
Reference power supply current	During A/D and D/A conversion			AI_{CC}	—	0.8	1.5	mA	
	Standby for A/D and D/A conversion				—	0.5	1.5	μA	
RAM standby voltage				V_{RAM}	2.5	—	—	V	

4. The values are for reference.
5. This applies when the $\overline{RES}$ pin is held low at power-on.

Table 29.5 Permissible Output Currents

Conditions: V_{CC} = PLLV_{CC} = DrV_{CC} = 2.95 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V,
V_{ref} = 3.0 V to AV_{CC}, V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS} = 0 V*,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.
Permissible output low current (per pin)	Output pins	I_{OL}	—	—	2.0
Permissible output low current (total)	Total of 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 tabl

Note: * 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}.

1.5 V (V_{CC} = 3.0 V to 3.6 V*)

Note: * Vcc=2.95 to 3.60V in the H8SX/1655M Group.

Figure 29.1 Output Load Circuit

Clock cycle time	t_{cyc}	20	125	ns	Figure 2
Clock high pulse width	t_{CH}	5	—	ns	
Clock low pulse width	t_{CL}	5	—	ns	
Clock rising time	t_{Cr}	—	5	ns	
Clock falling time	t_{Cf}	—	5	ns	
Oscillation settling time after reset (crystal)	t_{OSC1}	10	—	ms	Figure 2
Oscillation settling time after leaving software standby mode (crystal)	t_{OSC2}	10	—	ms	Figure 2
External clock output delay settling time	t_{DEXT}	1	—	ms	Figure 2
External clock input low pulse width	T_{EXL}	27.7	—	ns	Figure 2
External clock input high pulse width	T_{EXH}	27.7	—	ns	
External clock rising time	T_{EXr}	—	5	ns	
External clock falling time	T_{EXf}	—	5	ns	

Note: * V_{CC} = PLLV_{CC} = DrV_{CC} = 2.95V to 3.60V in the H8SX/1655M Group.

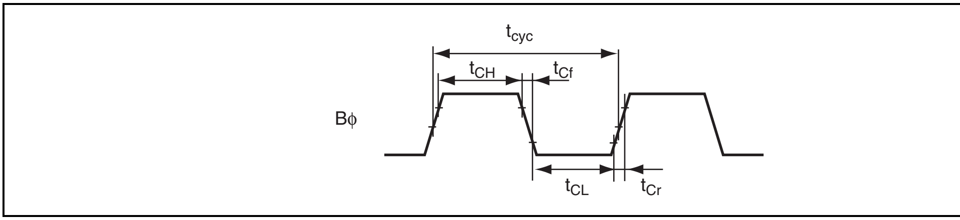


Figure 29.2 External Bus Clock Timing

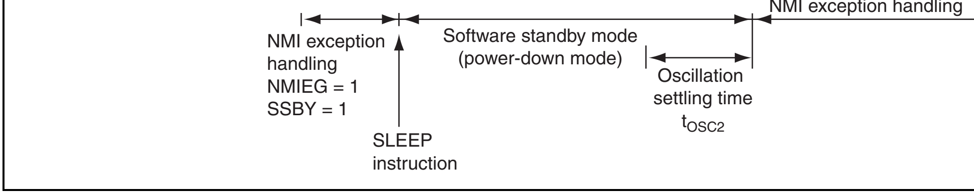


Figure 29.3 Oscillation Settling Timing after Software Standby Mode

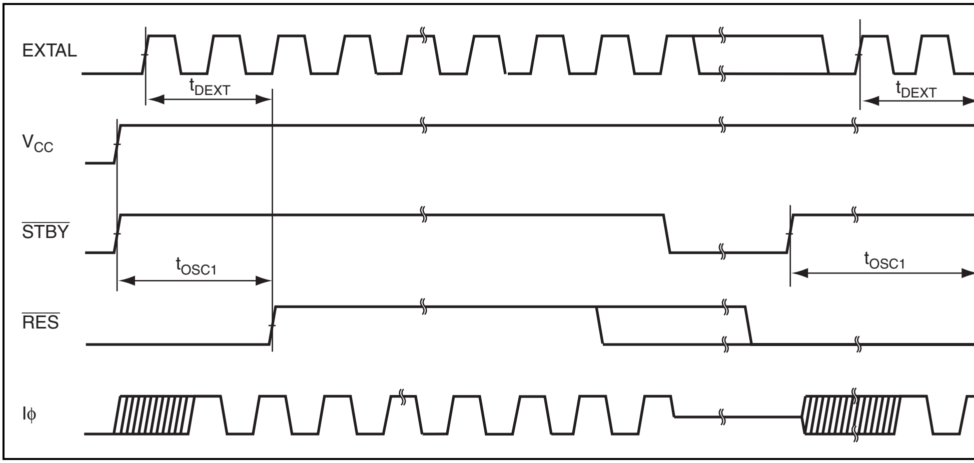


Figure 29.4 Oscillation Settling Timing

Table 29.7 Control Signal Timing

Conditions: V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 V to 3.6 V*, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0
AV_{CC}, V_{SS} = PLLV_{SS} = DrV_{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 Co
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 2
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 2
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (after leaving software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (after leaving software standby mode)	t_{IRQW}	200	—	ns	

Note: * V_{CC} = PLLV_{CC} = DrV_{CC} = 2.95V to 3.60V in the H8SX/1655M Group.

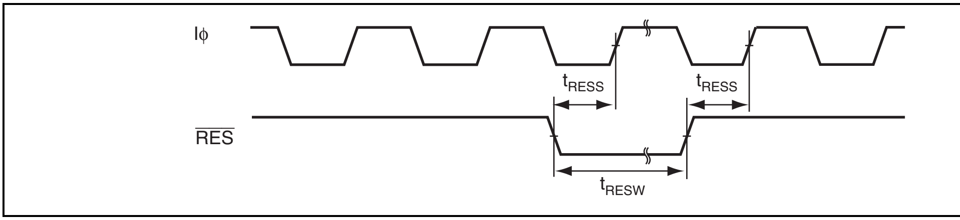


Figure 29.6 Reset Input Timing

Note: * SSIER must be set to cancel software standby mode.

Figure 29.7 Interrupt Input Timing

29.4.3 Bus Timing

Table 29.8 Bus Timing (1)

Conditions: V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 V to 3.6 V*, AV_{CC} = 3.0 V to 3.6 V,
V_{ref} = 3.0 V to AV_{CC}, V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS} = 0 V, Bϕ = 8 MHz to 5
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 Condit
Address delay time	t_{AD}	—	15	ns	Figures 29.20
Address setup time 1	t_{AS1}	$0.5 \times t_{CYC} - 8$	—	ns	
Address setup time 2	t_{AS2}	$1.0 \times t_{CYC} - 8$	—	ns	
Address setup time 3	t_{AS3}	$1.5 \times t_{CYC} - 8$	—	ns	
Address setup time 4	t_{AS4}	$2.0 \times t_{CYC} - 8$	—	ns	
Address hold time 1	t_{AH1}	$0.5 \times t_{CYC} - 8$	—	ns	
Address hold time 2	t_{AH2}	$1.0 \times t_{CYC} - 8$	—	ns	
Address hold time 3	t_{AH3}	$1.5 \times t_{CYC} - 8$	—	ns	

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) 1	t_{AA1}	—	$1.0 \times t_{CYC} - 20$	ns
Read data access time (from address) 2	t_{AA2}	—	$1.5 \times t_{CYC} - 20$	ns
Read data access time (from address) 3	t_{AA3}	—	$2.0 \times t_{CYC} - 20$	ns
Read data access time (from address) 4	t_{AA4}		$2.5 \times t_{CYC} - 20$	ns
Read data access time (from address) 5	t_{AA5}	—	$3.0 \times t_{CYC} - 20$	ns

Note: * $V_{CC} = PLLV_{CC} = DrV_{CC}$ = 2.95V to 3.60V in the H8SX/1655M Group.

WR pulse width 2	t_{WSW2}	$1.5 \times t_{CYC} - 13$	—	ns	
Write data delay time	t_{WDD}	—	20	ns	
Write data setup time 1	t_{WDS1}	$0.5 \times t_{CYC} - 13$	—	ns	
Write data setup time 2	t_{WDS2}	$1.0 \times t_{CYC} - 13$	—	ns	
Write data setup time 3	t_{WDS3}	$1.5 \times t_{CYC} - 13$	—	ns	
Write data hold time 1	t_{WDH1}	$0.5 \times t_{CYC} - 8$	—	ns	
Write data hold time 3	t_{WDH3}	$1.5 \times t_{CYC} - 8$	—	ns	
Byte control delay time	t_{UBD}	—	15	ns	Figures 29 29.14
Byte control pulse width 1	t_{UBW1}	—	$1.0 \times t_{CYC} - 15$	ns	Figure 29.1
Byte control pulse width 2	t_{UBW2}	—	$2.0 \times t_{CYC} - 15$	ns	Figure 29.1
Multiplexed address delay time 1	t_{MAD1}	—	15	ns	Figures 29 29.18
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	Figures 29 29.18
$\overline{\text{WAIT}}$ hold time	t_{WTH}	5.0	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	20	—	ns	Figure 29.1
$\overline{\text{BACK}}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	30	ns	
$\overline{\text{BREQO}}$ delay time	t_{BRQOD}	—	15	ns	Figure 29.2

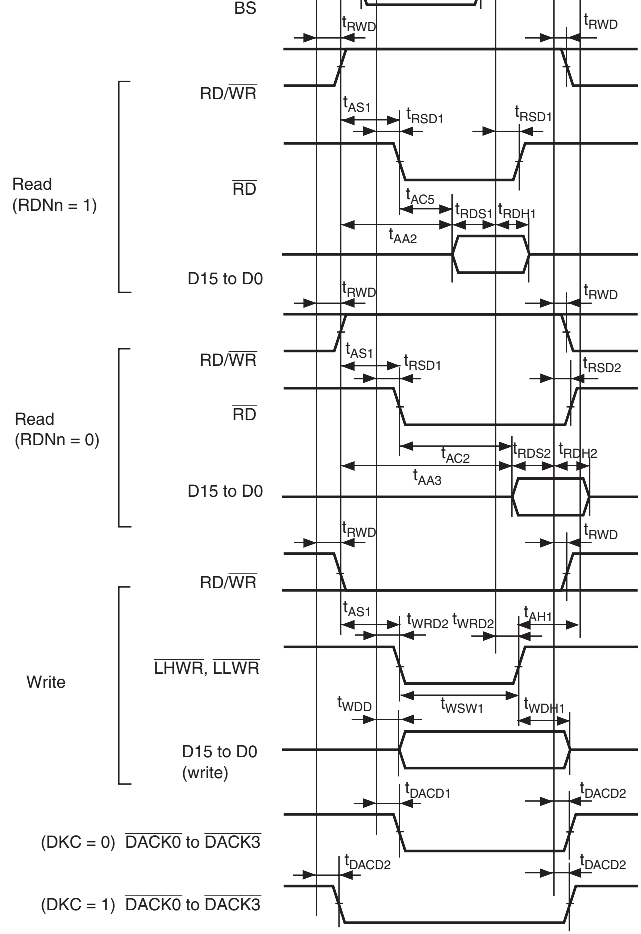


Figure 29.8 Basic Bus Timing: Two-State Access

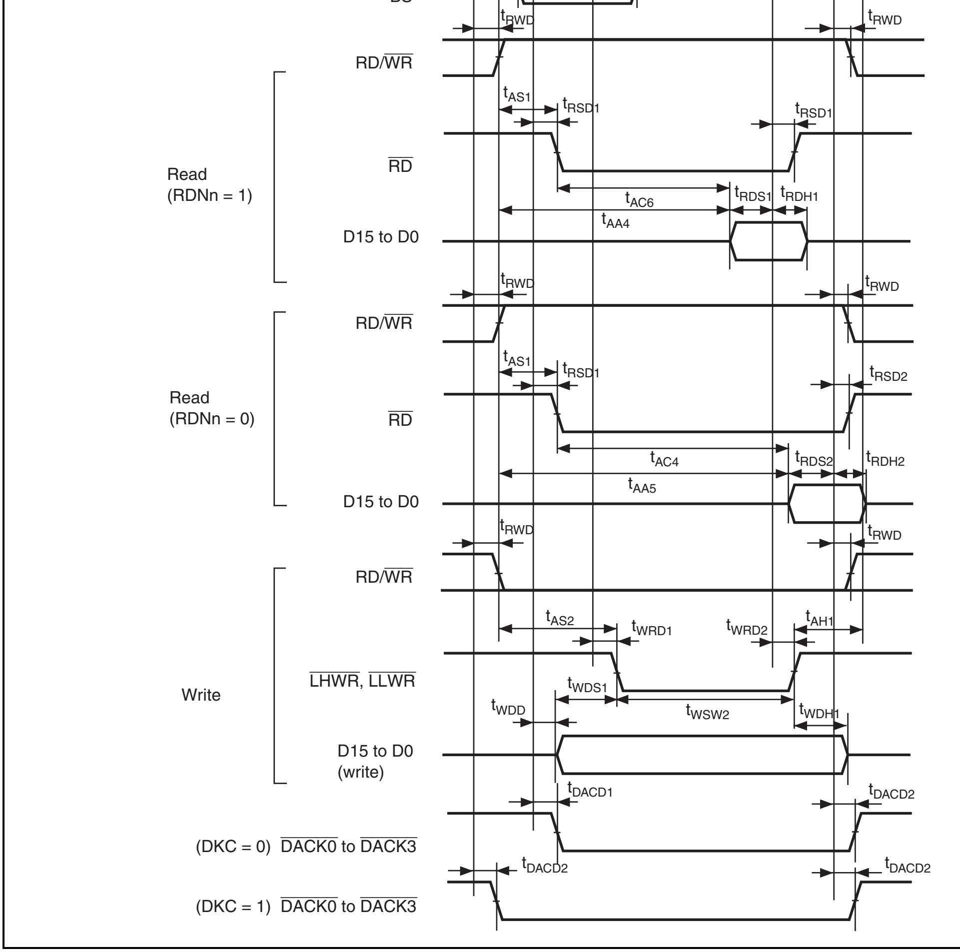


Figure 29.9 Basic Bus Timing: Three-State Access

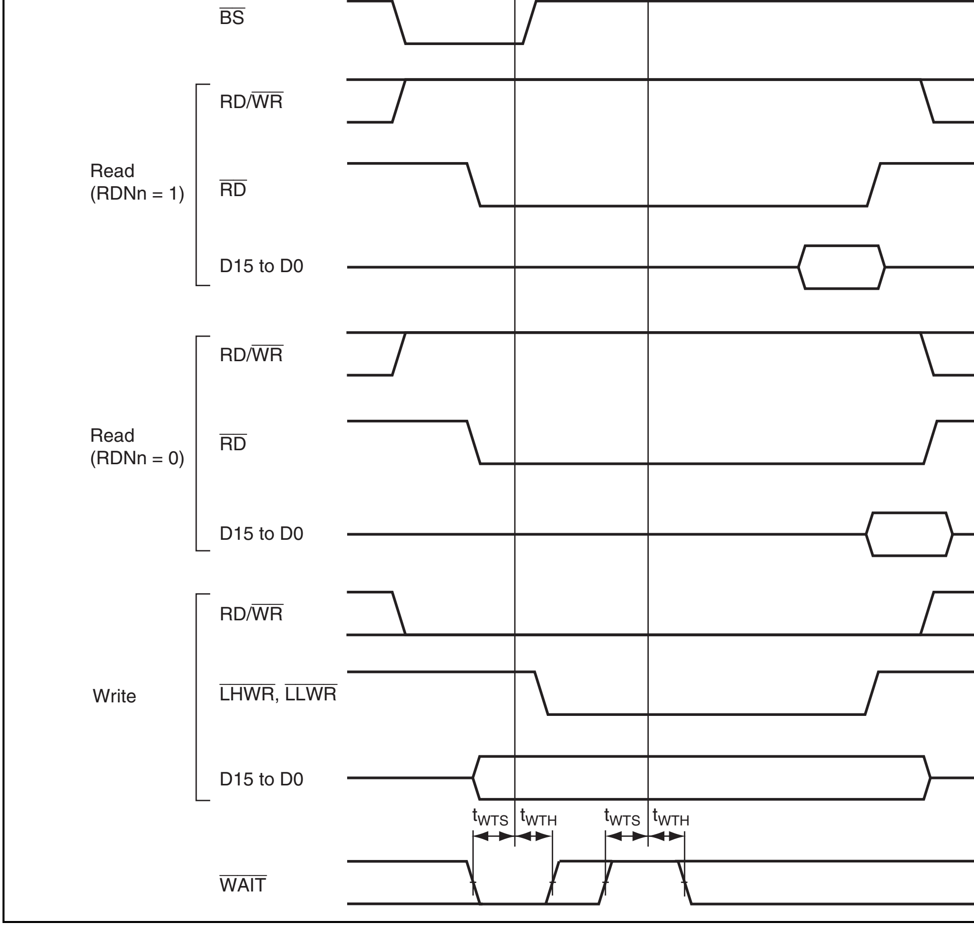


Figure 29.10 Basic Bus Timing: Three-State Access, One Wait

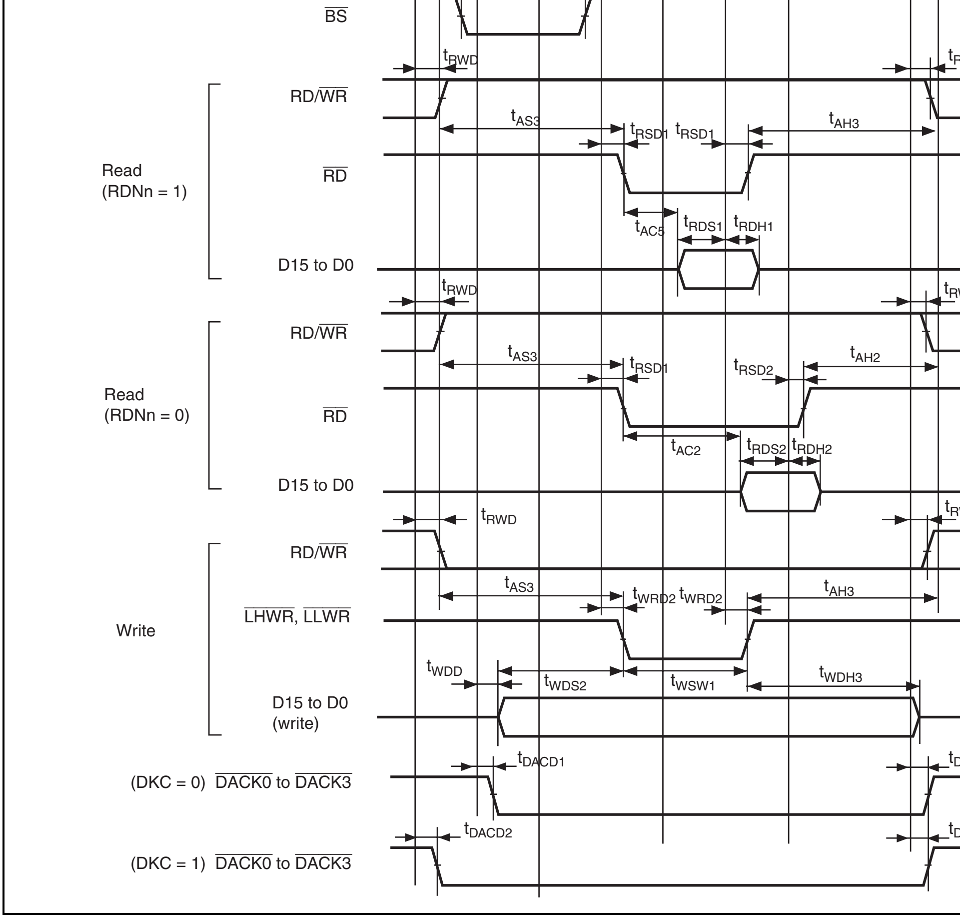


Figure 29.11 Basic Bus Timing: Two-State Access (CS Assertion Period Exte

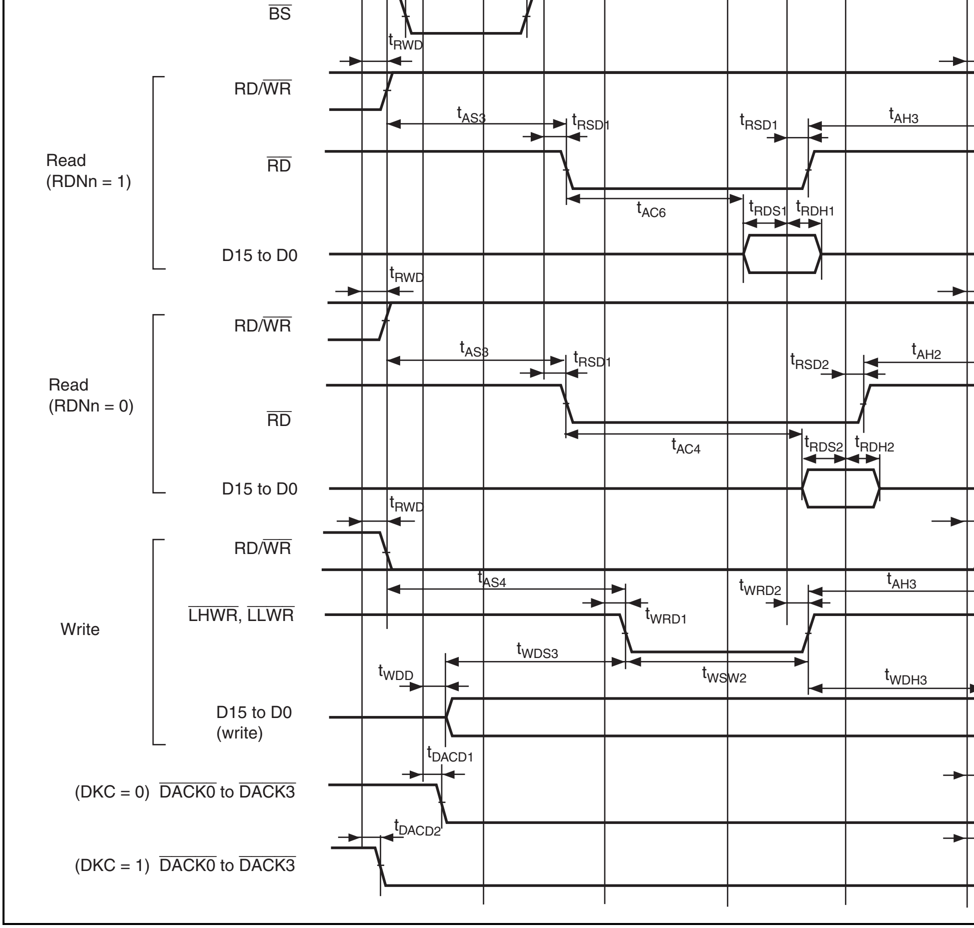


Figure 29.12 Basic Bus Timing: Three-State Access (CS Assertion Period Exter

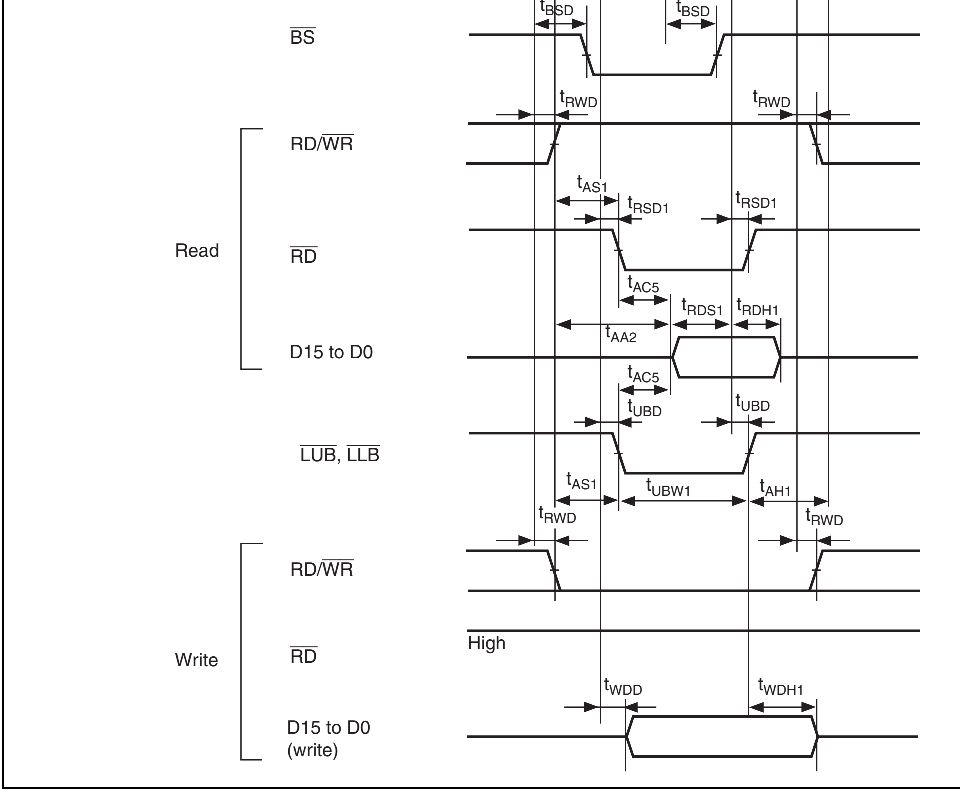


Figure 29.13 Byte Control SRAM: Two-State Read/Write Access

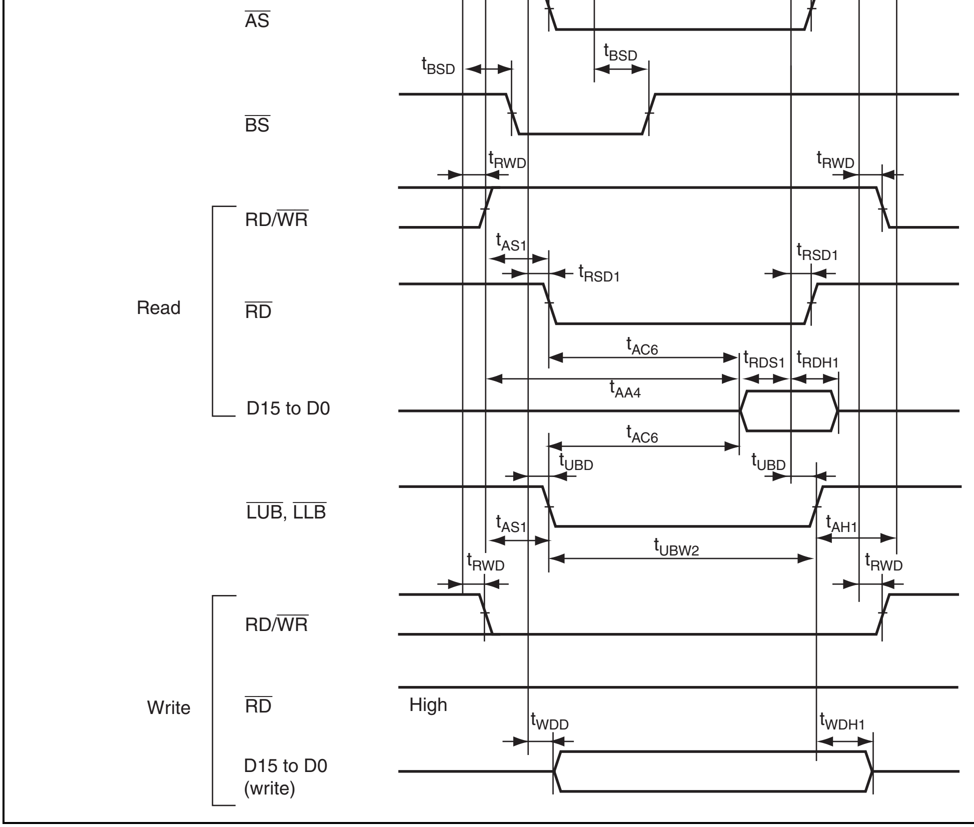


Figure 29.14 Byte Control SRAM: Three-State Read/Write Access

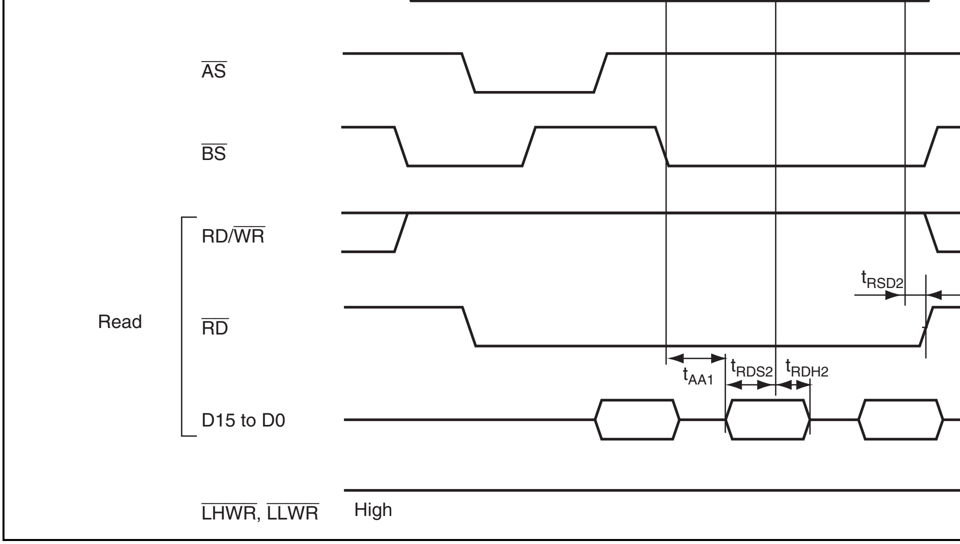


Figure 29.15 Burst ROM Access Timing: One-State Burst Access

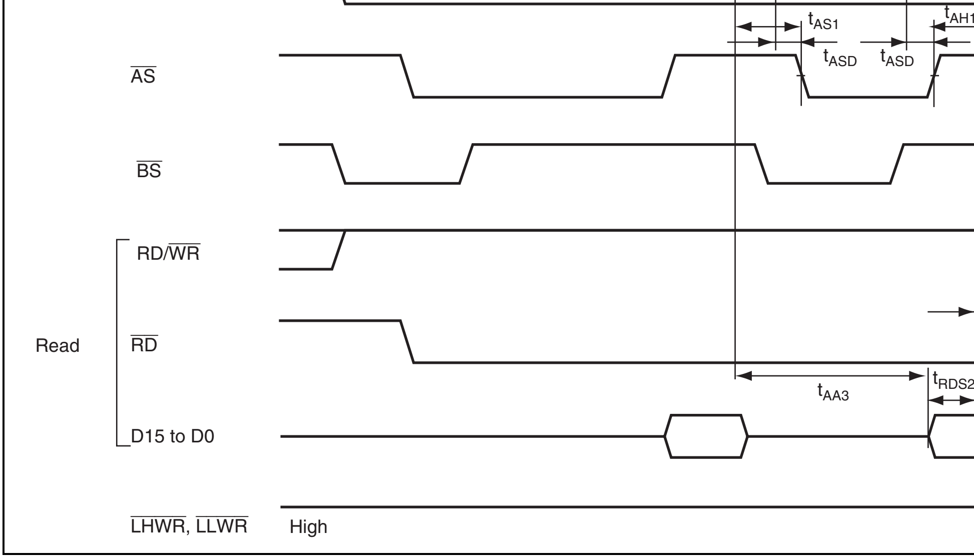


Figure 29.16 Burst ROM Access Timing: Two-State Burst Access

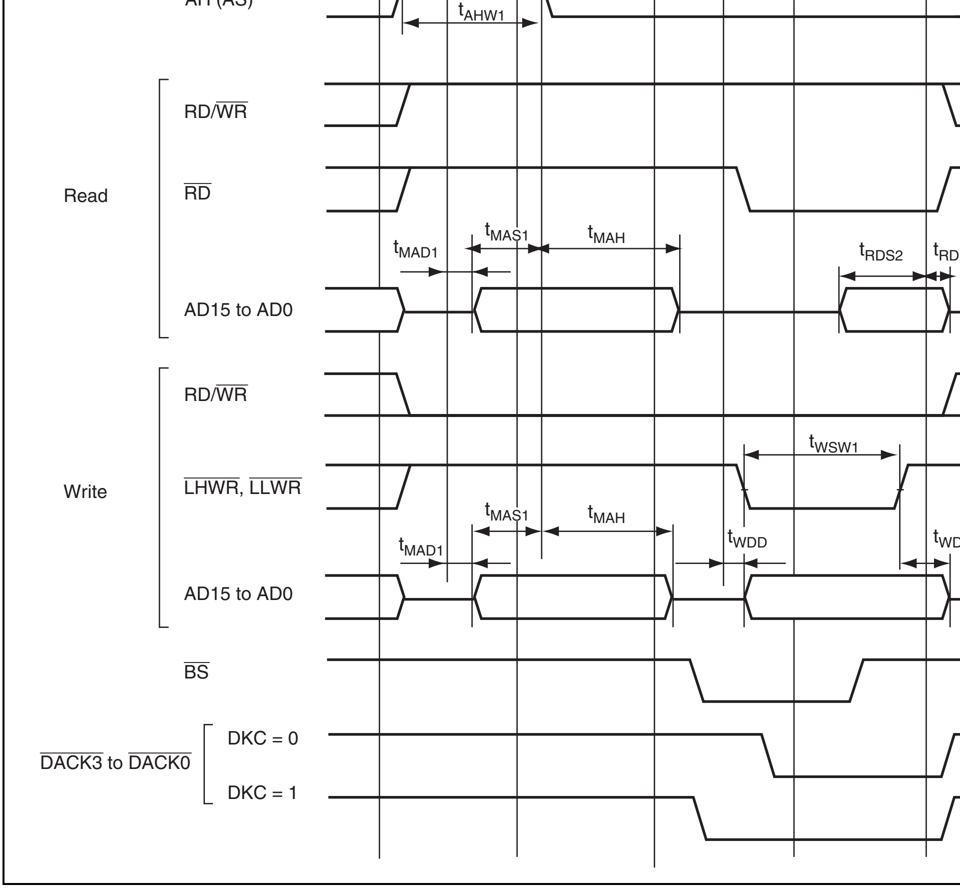


Figure 29.17 Address/Data Multiplexed Access Timing (No Wait) (Basic, Four-State Access)

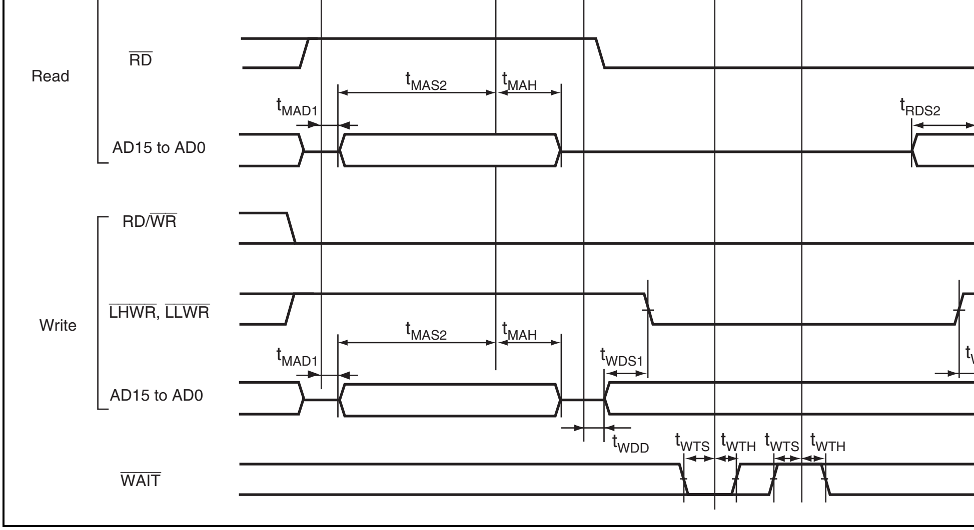


Figure 29.18 Address/Data Multiplexed Access Timing (Wait Control) (Address Cycle Program Wait × 1 + Data Cycle Program Wait × 1 + Data Cycle Pin Wait × 1)

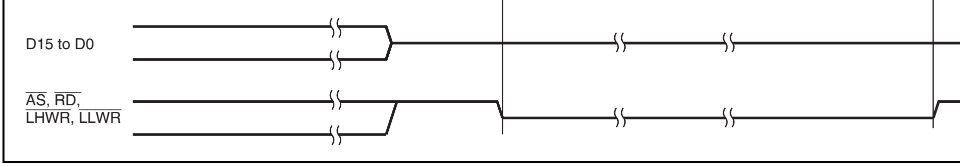


Figure 29.19 External Bus Release Timing

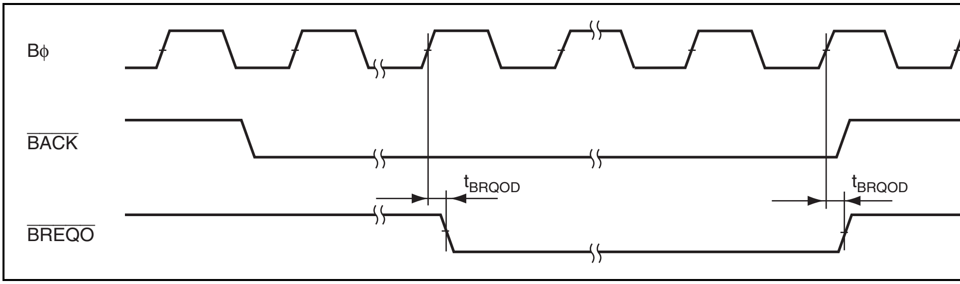


Figure 29.20 External Bus Request Output Timing

$\overline{\text{DREQ}}$ hold time	t_{DRQH}	5	—	ns	
$\overline{\text{TEND}}$ delay time	t_{TED}	—	15	ns	Figure 29.
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	15	ns	Figure 29.
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	15	ns	Figure 29.
$\overline{\text{EDREQ}}$ setup time	t_{EDRQS}	20	—	ns	Figure 29.
$\overline{\text{EDREQ}}$ hold time	t_{EDRQH}	5	—	ns	
$\overline{\text{ETEND}}$ delay time	t_{ETED}	—	15	ns	Figure 29.
$\overline{\text{EDACK}}$ delay time 1	t_{EDACD1}	—	15	ns	Figure 29.
$\overline{\text{EDACK}}$ delay time 2	t_{EDACD2}	—	15	ns	Figure 29.
$\overline{\text{EDRAK}}$ delay time	t_{EDRKD}	—	15	ns	Figure 29.

Note: * V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 2.95V to 3.60V in the H8SX/1655M Group.

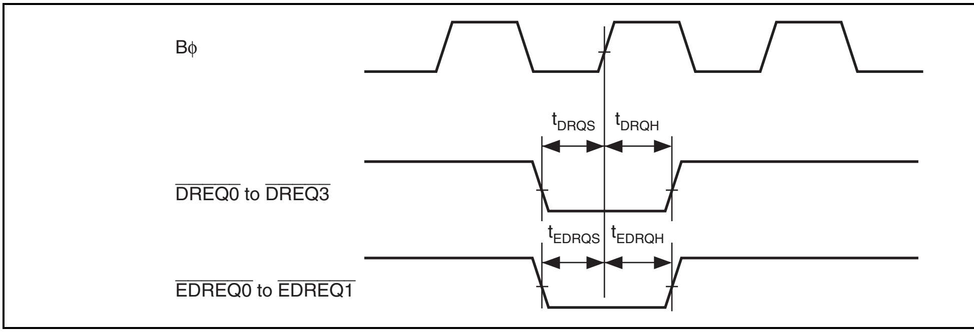


Figure 29.21 DMAC/EXDMAC ($\overline{\text{DREQ}}$ and $\overline{\text{EDREQ}}$) Input Timing

Figure 29.22 DMAC/EXDMAC ($\overline{\text{TEND}}$ and $\overline{\text{ETEND}}$) Output Timing

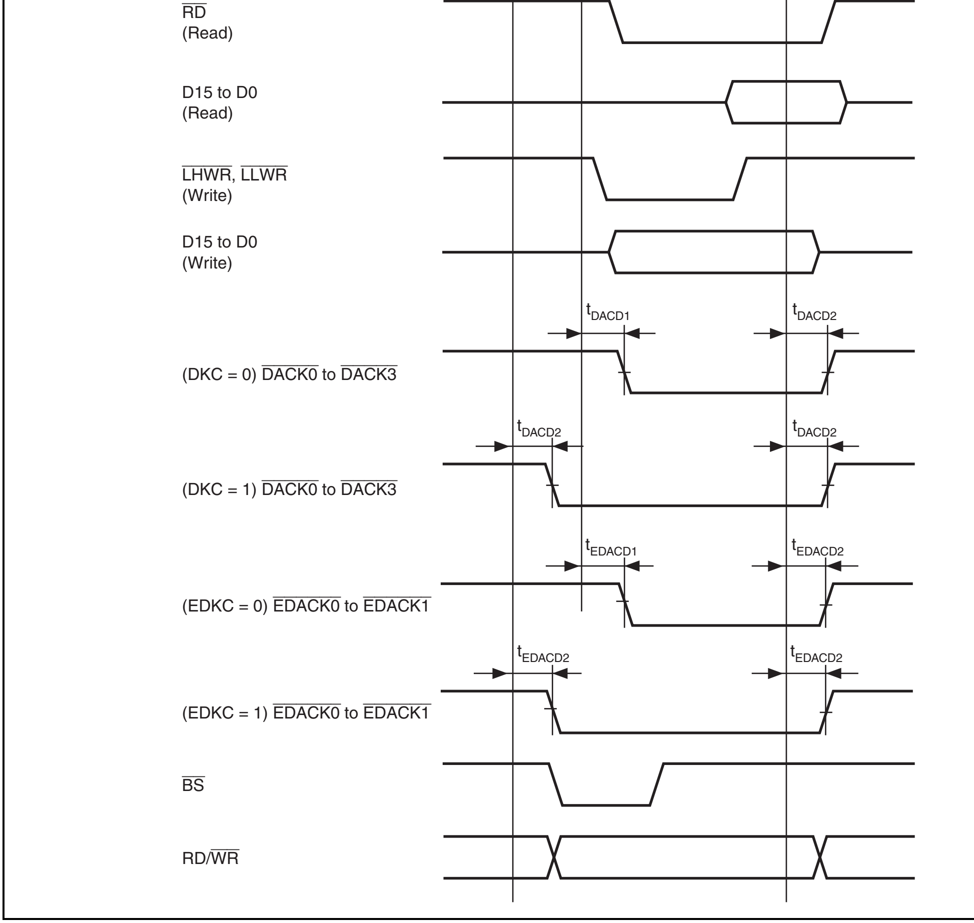


Figure 29.23 DMAC/EXDMAC Single-Address Transfer Timing: Two-State A

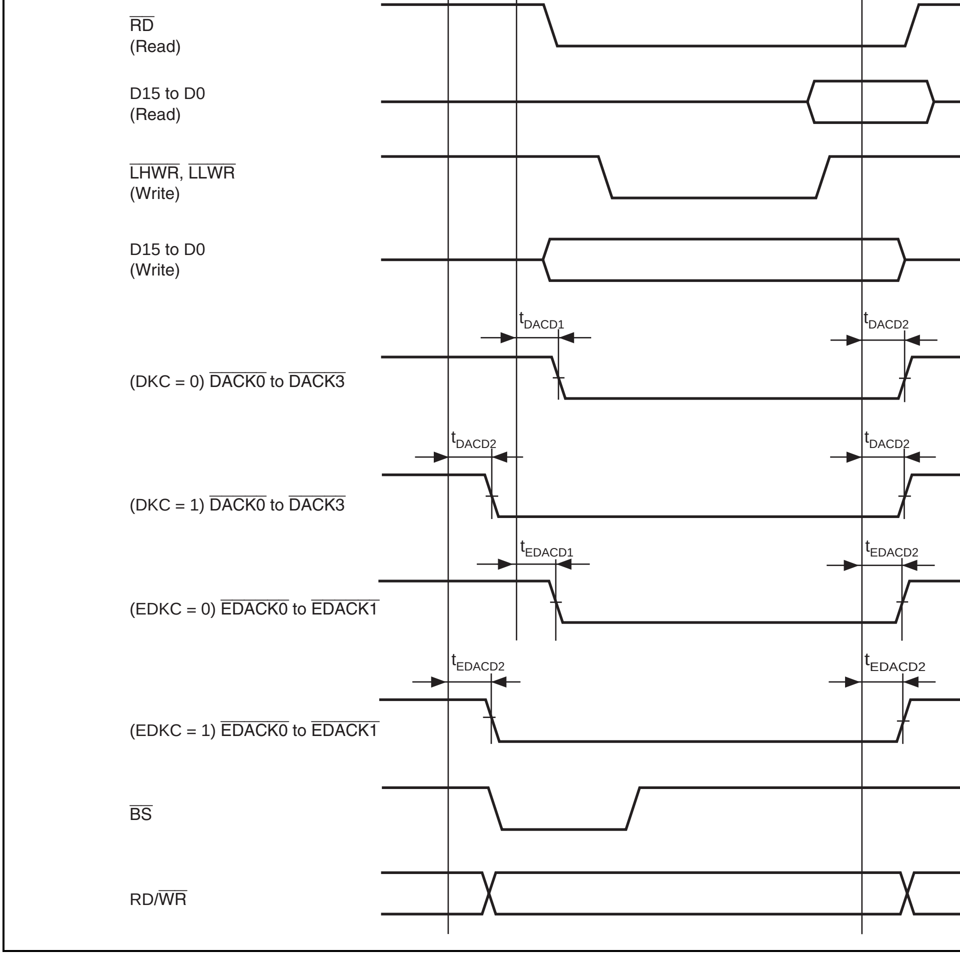


Figure 29.24 DMAC/EXDMAC Single-Address Transfer Timing: Three-State

29.4.5 Timing of On-Chip Peripheral Modules

Table 29.10 Timing of On-Chip Peripheral Modules

Conditions: $V_{CC} = PLLV_{CC} = DrV_{CC}$ = 3.0 V to 3.6 V*[1], AV_{CC} = 3.0 V to 3.6 V,
V_{ref} = 3.0 V to AV_{CC}, $V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS}$ = 0 V, Pφ = 8 MHz to 3
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
8-bit timer	Timer output delay time		t_{TMOD}	—	40	ns	Figure 2
	Timer reset input setup time		t_{TMRS}	25	—	ns	Figure 2
	Timer clock input setup time		t_{TMCS}	25	—	ns	Figure 2
	Timer clock pulse width	Single-edge setting	t_{TMCWH}	1.5	—	t_{cyc}	
		Both-edge setting	t_{TMCWL}	2.5	—	t_{cyc}	

	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
IIC2	SCL input cycle time	t_{SCL}	$12\ t_{CYC}$ + 600	—	ns	Figure
	SCL input high pulse width	t_{SCLH}	$3\ t_{CYC}$ + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	$5\ t_{CYC}$ + 300	—	ns	
	SCL, SDA input falling time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	—	$1\ t_{CYC}$	ns	
	SDA input bus free time	t_{BUF}	$5\ t_{CYC}$	—	ns	
	Start condition input hold time	t_{STAH}	$3\ t_{CYC}$	—	ns	
	Repeated start condition input setup time	t_{STAS}	$3\ t_{CYC}$	—	ns	
	Stop condition input setup time	t_{STOS}	$1\ t_{CYC}$ + 20	—	ns	
	Data input setup time	t_{SDAS}	0	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	Cb	—	400	pF	
	SCL, SDA falling time	t_{Sf}	—	300	ns	

	TDI setup time	t_{TDIS}	20	—	ns
	TDI hold time	t_{TDIH}	20	—	ns
	TDO data delay time	t_{TDOD}	—	23	ns

Notes: 1. V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 2.95V to 3.60V in the H8SX/1655M Group
2. $t_{TCKcyc} \geq t_{TCKcyc}$ must be satisfied.

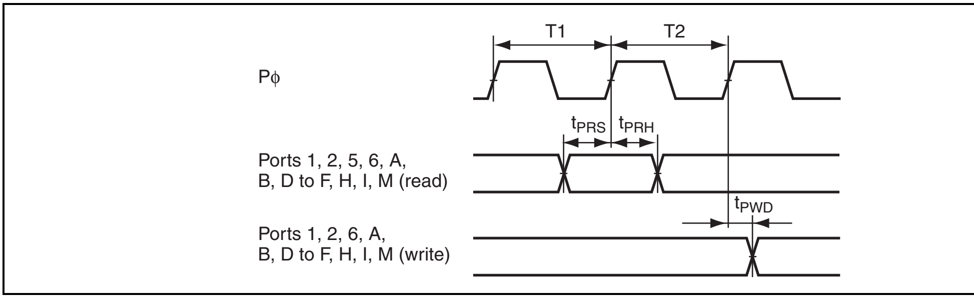


Figure 29.26 I/O Port Input/Output Timing

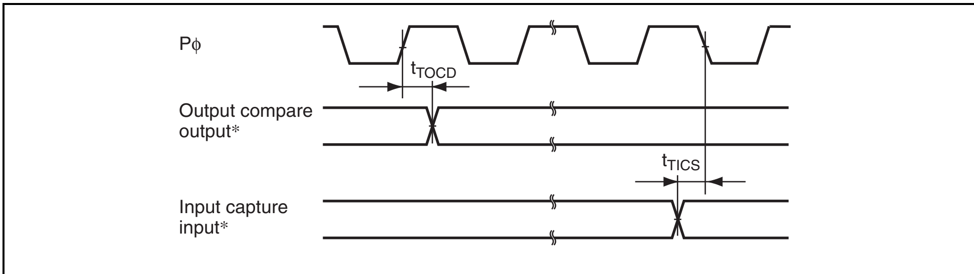


Figure 29.27 TPU Input/Output Timing

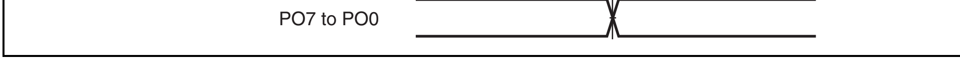


Figure 29.29 PPG Output Timing

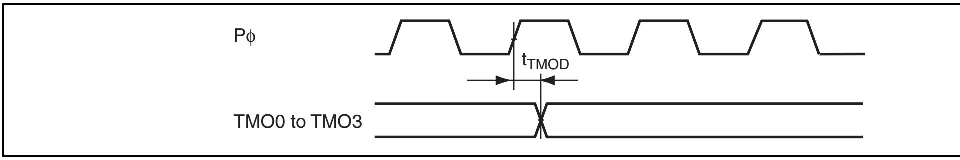


Figure 29.30 8-Bit Timer Output Timing

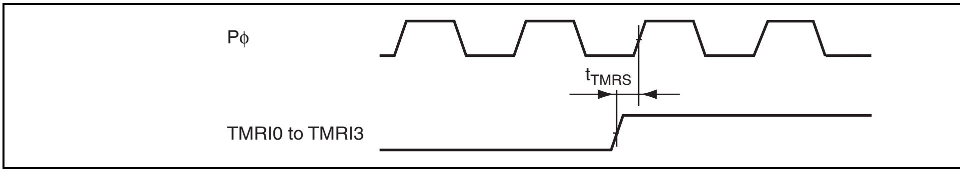


Figure 29.31 8-Bit Timer Reset Input Timing

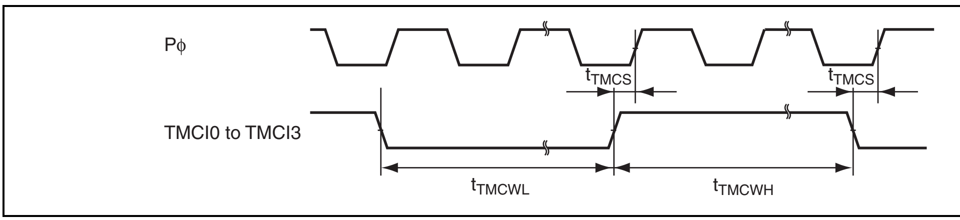


Figure 29.32 8-Bit Timer Clock Input Timing

Figure 29.34 SCK Clock Input Timing

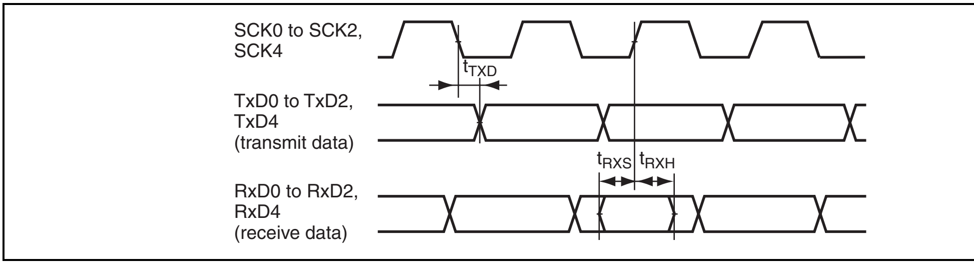


Figure 29.35 SCI Input/Output Timing: Clocked Synchronous Mode

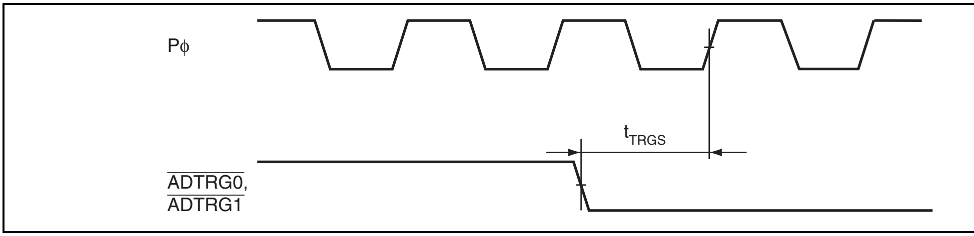
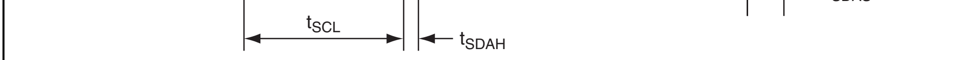


Figure 29.36 A/D Converter External Trigger Input Timing



Note: S, P, and Sr represent the following conditions:
S: Start condition
P: Stop condition
Sr: Repeated start condition

Figure 29.37 I²C Bus Interface2 Input/Output Timing (Option)

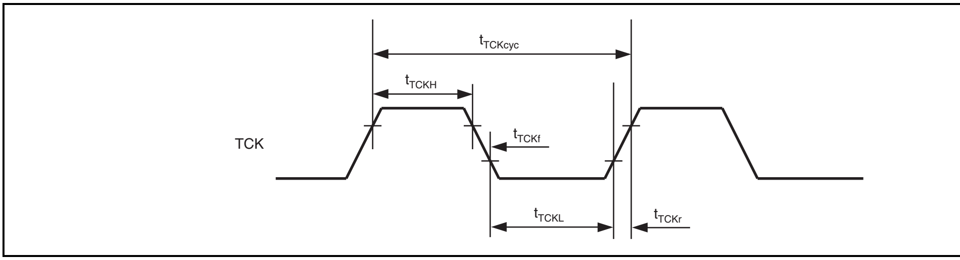


Figure 29.38 Boundary Scan TCK Timing

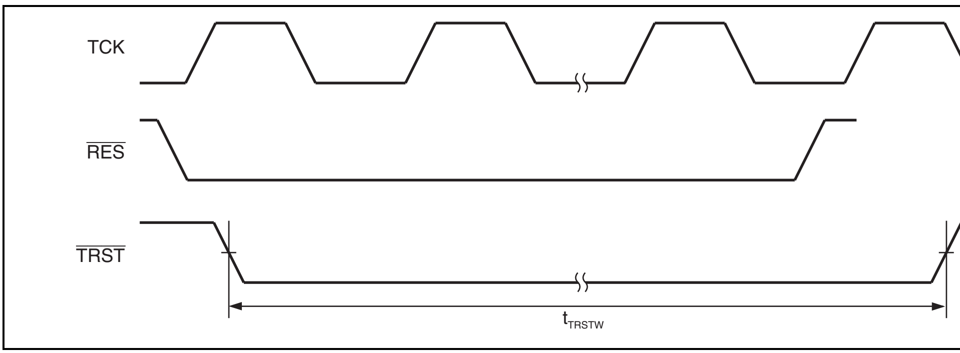


Figure 29.39 Boundary Scan $\overline{TRST}$ Timing

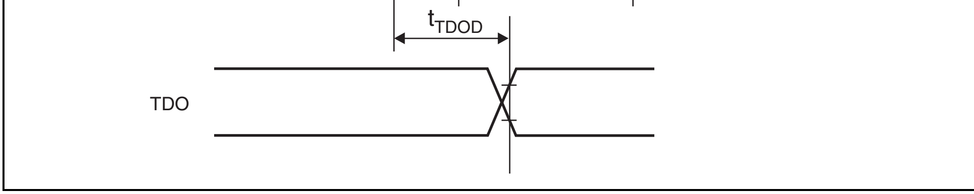


Figure 29.40 Boundary Scan Input/Output Timing

	Input low voltage	V_{IL}	—	0.8	V	
	Differential input sensitivity	V_{DI}	0.2	—	V	$\lvert (D+) - (D-) \rvert$
	Differential common mode range	V_{CM}	0.8	2.5	V	
Output	Output high voltage	V_{OH}	2.8	—	V	$I_{OH} = -200\ \mu A$
	Output low voltage	V_{OL}	—	0.3	V	$I_{OL} = 2$ mA
	Crossover voltage	V_{CRS}	1.3	2.0	V	
	Rising time	t_R	4	20	ns	
	Falling time	t_F	4	20	ns	
	Ratio of rising time to falling time	t_{RFM}	90	111.11	%	(T_R/T_F)
	Output resistance	Z_{DRV}	28	44	Ω	Including $R_S = 22\Omega$

Note: * Vcc = PLLVcc = DrVcc = 2.95V to 3.60V in the H8SX/1655M Group.

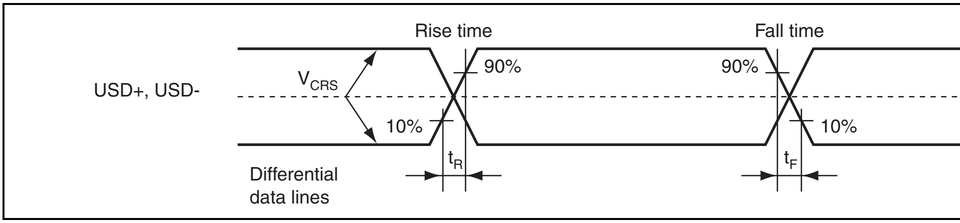


Figure 29.41 Data Signal Timing

Figure 29.42 Load Condition

29.6 A/D Conversion Characteristics

Table 29.12 A/D Conversion Characteristics in Peripheral Clock Mode (ICKSEL =

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V*, AV_{CC} = 3.0 V to 3.6 V,
Vref = 3.0 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, Pϕ = 8 MHz to
When all units operate in ICKSEL = 0,
Ta = –20°C to +75°C (regular specifications),
Ta = –40°C to +85°C (wide-range specifications)

Item		Min.	Typ.	Max.	Uni
Resolution		10	10	10	Bit
Conversion time		2.7	—	—	µs
Analog input capacitance		—	—	20	pF
Permissible signal source impedance	EXCKS = 0	—	—	5	kΩ
	EXCKS = 1	—	—	1	kΩ
Nonlinearity error		—	—	±3.5	LSE
Offset error		—	—	±3.5	LSE
Full-scale error		—	—	±3.5	LSE
Quantization error		—	±0.5	—	LSE
Absolute accuracy		—	—	±4.0	LSE

Note: * V_{CC} = $PLLV_{CC}$ = DrV_{CC}= 2.95V to 3.60V in the H8SX/1655M Group.

Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	1	kΩ
Nonlinearity error	—	—	±7.5	LS
Offset error	—	—	±7.5	LS
Full-scale error	—	—	±7.5	LS
Quantization error	—	±0.5	—	LS
Absolute accuracy	—	—	±8.0	LS

Note: * V_{cc} = $PLLV_{cc}$ = DrV_{cc}= 2.95V to 3.60V in the H8SX/1655M Group.

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

Note: * V_{CC} = $PLLV_{CC}$ = DrV_{CC}= 2.95V to 3.60V in the H8SX/1655M Group.

Table 29.15 10-Bit D/A Conversion Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V*, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V
AV_{CC}, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, Pϕ = 8 MHz to 35 MHz,
DADT[1:0] = Don't care (used as 10-bit D/A converter),
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	10	10	10	Bit	
Conversion time	—	—	10	μs	20-pF capaci
Absolute accuracy	—	±2.0	±3.0	LSB	16-MΩ resist

Note: * V_{CC} = $PLLV_{CC}$ = DrV_{CC}= 2.95V to 3.60V in the H8SX/1655M Group.

Item	Symbol	Min.	Typ.	Max.	Unit	T C
Programming time [*1, *2, *4]	t_P	—	1	10	ms/128 bytes	
Erasure time [*1, *2, *4]	t_E	—	40	130	ms/4-Kbyte block	
		—	300	800	ms/32-Kbyte block	
		—	600	1500	ms/64-Kbyte block	
Programming time (total) [*1, *2, *4]	Σ_{tP}	—	3.4	9	H8SX/1652, H8SX/1652M s/384 Kbytes	T fo
		—	4.5	12	H8SX/1655, H8SX/1655M s/512 Kbytes	
Erasure time (total) [*1, *2, *4]	Σ_{tE}	—	3.4	9	H8SX/1652, H8SX/1652M s/384 Kbytes	T
		—	4.5	12	H8SX/1655, H8SX/1655M s/512 Kbytes	
Programming and Erasure time (total) [*1, *2, *4]	Σ_{tPE}	—	6.8	18	H8SX/1652, H8SX/1652M s/384 Kbytes	T
		—	9.0	24	H8SX/1655, H8SX/1655M s/512 Kbytes	
Reprogramming count	N_{WEC}	100[*3]	—	—	times	
Data retention time [*4]	T_{DRP}	10	—	—	years	

Notes: 1. Programming time and erasure time depend on data in the flash memory.
2. Programming time and erasure time do not include time for data transfer.
3. All the characteristics after programming are guaranteed within this value (gu value is from 1 to Min. value).
4. Characteristics when programming is performed within the Min. value.
5. V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 2.95V to 3.60V in the H8SX/1655M Group.

Voltage detection level	Voltage detection circuit (LVD)	V_{det}	3.00	3.10	3.20	V	Figu
	Power-on reset (POR)	V_{POR}	2.48	2.58	2.68		Figu
Internal reset time		t_{POR}	20	35	50	ms	Figu
Power-off time*		t_{VOFF}	200	—	—	µs	Figu

Note: * Power-off time (t_{VOFF}) is the time over which V_{cc} is lower than minimum value of voltage-detection level of the POR and LVD.

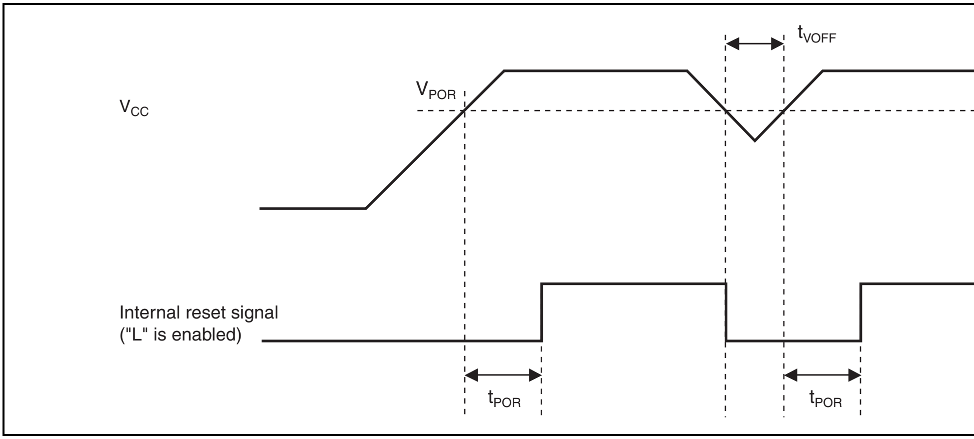


Figure 29.43 Power-On Reset Timing

Figure 29.44 Voltage Detection Circuit Timing

Port 2	All	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
P55 to P50	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
P56/ AN6/ DA0/ IRQ6-B	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	[DAOE0 = 1] Keep [DAOE0 = 0] Hi-Z	[DAOE0 = 1] Keep [DAOE0 = 0] Hi-Z
P57/ AN7/ DA1/ IRQ7-B	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	[DAOE1 = 1] Keep [DAOE1 = 0] Hi-Z	[DAOE1 = 1] Keep [DAOE1 = 0] Hi-Z
P65 to P60	All	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
PA0/ BREQO/ BS-A	All	Hi-Z	Hi-Z	[BREQO output] Hi-Z	[BREQO output] Hi-Z	[BREQO output] Hi-Z	[BREQO output] Hi-Z
				[BS output] Keep	[BS output] Hi-Z	[BS output] Keep	[BS output] Hi-Z
				[Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep
PA1/ BACK/ (RD/WR)	All	Hi-Z	Hi-Z	[BACK output] Hi-Z	[BACK output] Hi-Z	[BACK output] Hi-Z	[BACK output] Hi-Z
				[RD/WR output] Keep	[RD/WR output] Hi-Z	[RD/WR output] Keep	[RD/WR output] Hi-Z
				[Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep

$\overline{LLB}$	(EXPE = 0)							
	External extended mode (EXPE = 1)	H	Hi-Z	H	Hi-Z	H	Hi-Z	H
PA4/ $\overline{LHWR}$/ $\overline{LUB}$	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	K
	External extended mode (EXPE = 1)	H	Hi-Z	[$\overline{LHWR}$, $\overline{LUB}$ output] H [Other than above] Keep	[$\overline{LHWR}$, $\overline{LUB}$ output] Hi-Z [Other than above] Keep	[$\overline{LHWR}$, $\overline{LUB}$ output] H [Other than above] Keep	[$\overline{LHWR}$, $\overline{LUB}$ output] Hi-Z [Other than above] Keep	[L o H [O a K
PA5/$\overline{RD}$	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	K
	External extended mode (EXPE = 1)	H	Hi-Z	H	Hi-Z	H	Hi-Z	H
PA6/ $\overline{AS}$/ $\overline{AH}$/ $\overline{BS}$-B	Single-chip mode (EXPE = 0)	Hi-Z	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	[$\overline{AS}$, $\overline{BS}$ output] H [$\overline{AH}$ output] L [Other than above] Keep	[$\overline{AS}$, $\overline{AH}$, $\overline{BS}$ output] Hi-Z [Other than above] Keep	[A o H [O a K
	External extended mode (EXPE = 1)	H	Hi-Z					

$\overline{\text{CS5}}$-B	External extended mode (EXPE = 1)	H	Hi-Z	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep
PB1/ $\overline{\text{CS1}}$/ $\overline{\text{CS2}}$-B/ $\overline{\text{CS5}}$-A/ $\overline{\text{CS6}}$-B/ $\overline{\text{CS7}}$-B	All	Hi-Z	Hi-Z	[$\overline{\text{CS}}$ output] H [Other than above] Keep	[$\overline{\text{CS}}$ output] Hi-Z [Other than above] Keep	[$\overline{\text{CS}}$ output] H [Other than above] Keep	[$\overline{\text{CS}}$ output] Hi-Z [Other than above] Keep
PB2/ $\overline{\text{CS2}}$-A/ $\overline{\text{CS6}}$-A	All	Hi-Z	Hi-Z	[$\overline{\text{CS}}$ output] H [Other than above] Keep	[$\overline{\text{CS}}$ output] Hi-Z [Other than above] Keep	[$\overline{\text{CS}}$ output] H [Other than above] Keep	[$\overline{\text{CS}}$ output] Hi-Z [Other than above] Keep
PB3/ $\overline{\text{CS3}}$/ $\overline{\text{CS7}}$-A	All	Hi-Z	Hi-Z	[$\overline{\text{CS}}$ output] H [Other than above] Keep	[$\overline{\text{CS}}$ output] Hi-Z [Other than above] Keep	[$\overline{\text{CS}}$ output] H [Other than above] Keep	[$\overline{\text{CS}}$ output] Hi-Z [Other than above] Keep
Port D	External extended mode (EXPE = 1)	L	Hi-Z	Keep	Hi-Z	Keep	Hi-Z
	ROM enabled extended mode	Hi-Z	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	Keep	[Address output] Hi-Z [Other than above] Keep
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep

					Keep		Keep	Ke
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	Ke
PF3 to PF0	External extended mode (EXPE = 1)	L	Hi-Z	Keep	Hi-Z	Keep	Hi-Z	Hi-
	ROM enabled extended mode	Hi-Z	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	Keep	[Address output] Hi-Z [Other than above] Keep	[Ad ou Hi- [O ab Ke
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	Ke
PF4	External extended mode (EXPE = 1)	Hi-Z	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	Keep	[Address output] Hi-Z [Other than above] Keep	[Ad ou Hi- [O ab Ke
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	Ke

	mode (EXPE = 1)	mode						
		16-bit bus mode	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
Port J	Hi-Z		Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
Port K	Hi-Z		Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
Port M	Hi-Z		Hi-Z	Hi-Z	Keep	Keep	Keep	Keep

[Legend]

H: High-level output

L: Low-level output

Keep: Input pins become high-impedance, output pins retain their state.

Hi-Z: High impedance

H8SX/1655M	R5F61655M	R5F61655MFPV	PLQP0120LA-A (FP-
		R5F61655MLGV	PTLG0145JB-A (TLP

Note: * Pb-free version

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP120-14x14-0.40	PLQP0120LA-A	120P6R-A / FP-120B / FP-120BV	0.7g

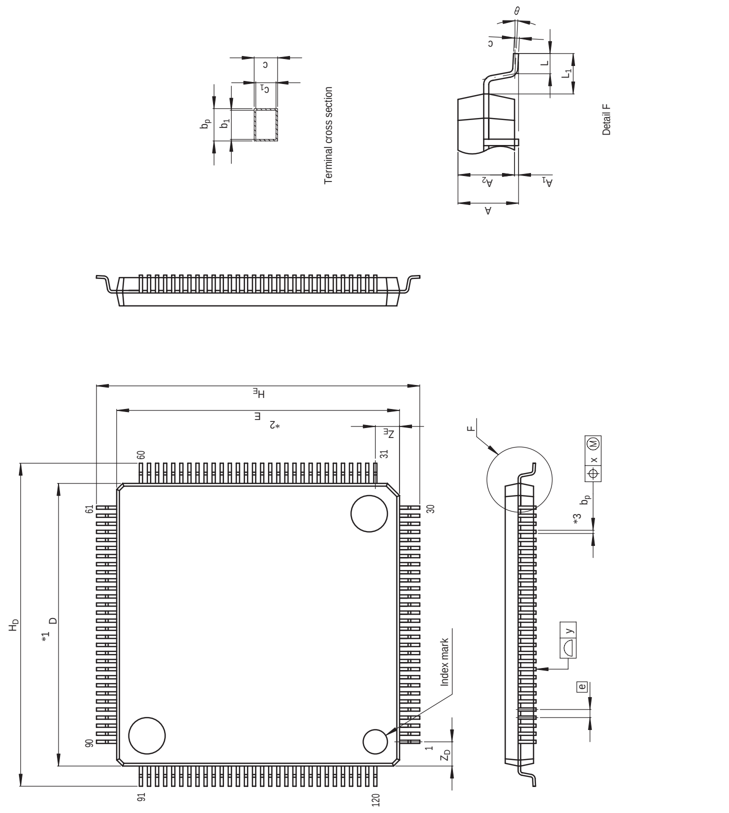
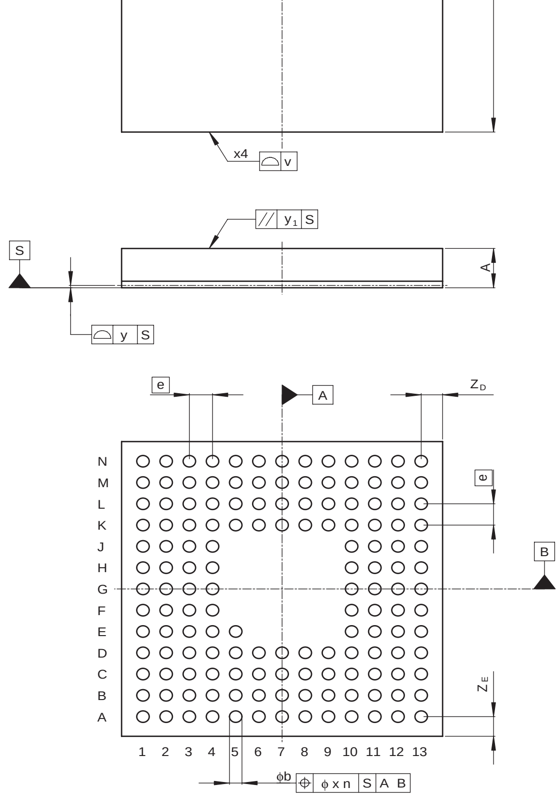


Figure C.1 Package Dimensions (FP-120BV)



Reference Symbol	Dimension	
	Min	
D	——	
E	——	
v	——	–
w	——	–
A	——	–
A_1	——	–
e	——	0
b	0.30	0
x	——	–
y	——	–
y_1	——	–
S_D	——	–
S_E	——	–
Z_D	——	
Z_E	——	

Figure C.2 Package Dimensions (TLP-145V)

MD_CLK	(Always used as mode pins)
MD2 to MD0	(Always used as mode pins)
NMI	• Connect this pin to VCC via a pull-up resistor
EXTAL	(Always used as a clock pin)
XTAL	• Leave this pin open
$\overline{\text{WDTOVF}}$	• Leave this pin open
USD+	• Leave this pin open
USD–	• Leave this pin open
VBUS	• Leave this pin open
Port 1	• Connect these pins to VCC via a pull-up resistor or to VSS via a pull-d resistor, respectively
Port 2	
Port 6	
PA2 to PA0	
PB3 to PB1	
Port J	
Port K	
Port M	
Port 5	• Connect these pins to AVcc via a pull-up resistor or to AVss via a pull resistor, respectively

Port D	• These pins are left open in the initial state for the address output.	
Port E		
PF4 to PF0		
Port H	(Used as a data bus)	
Port I	(Used as a data bus)	• Connect these pins to VCC via a pull-up resistor or to VSS via a pull-down resistor, respectively, in the initial state for the general input.
Vref	• Connect this pin to AVcc	

Notes: 1. Do not change the initial value (input-buffer disabled) of PnICR, where n corres an unused pin.

2. When the pin function is changed from its initial state, use a pull-up or pull-dow resistor as needed.

6. Exception Handling

7. Interrupt Controller

8. User Break Controller (UBC)

9. Bus Controller (BSC)

10. DMA Controller (DMAC)

11. EXDMA Controller (EXDMAC)

12. Data Transfer Controller (DTC)

13. I/O Ports

14. 16-Bit Timer Pulse Unit (TPU)

15. Programmable Pulse Generator (PPG)

16. 8-Bit Timers (TMR)

17. Watchdog Timer (WDT)

18. Serial Communication Interface (SCI, IrDA, CRC)

19. USB Function Module (USB)

20. I^2C Bus Interface 2 (IIC2)

21. A/D Converter

22. D/A Converter

23. RAM

24. Flash Memory

25. Boundary Scan

26. Clock Pulse Generator

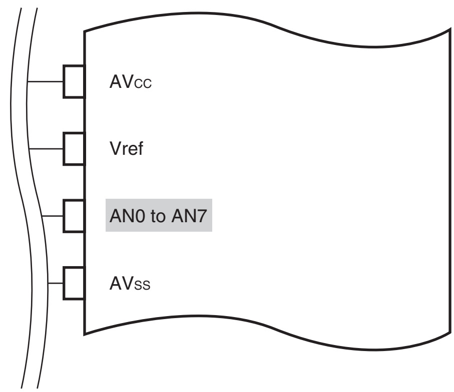
27. Power-Down Modes

28. List of Registers

29. Electrical Characteristics

Appendix

		5. The lower five bits are valid and the upper th are reserved for port F registers. The write va should be the same as the initial value. 6. The lower five bits are valid and the upper th are reserved for port M registers. The write v should be the same as the initial value.
Section 21 A/D Converter 21.4 Operation Input Sampling and 21.4.3 A/D Conversion Time	997 to 999	The following tables replaced Table 21.3 Characteristics of A/D Conversion (Unit 0: w EXCKS* = 0, ICKSEL = 0, and ADSSTR* = H'0F) (1) Table 21.3 Characteristics of A/D Conversion (Unit 0: w EXCKS* = 1, ICKSEL = 0, and ADSSTR* = H'0F) (2) Table 21.4 Characteristics of A/D Conversion (Unit 1: w EXCKS = 0, ICKSEL = 0, and ADSSTR* = H'0F) (1) Table 21.4 Characteristics of A/D Conversion (Unit 1: w EXCKS = 1, ICKSEL = 0, and ADSSTR* = H'0F) (2) Table 21.5 Characteristics of A/D Conversion (When E = 1, ICKSEL*1 = 0, and ADSSTR*2 = H'19)
Table 21.6 Period of A/D Conversion (Scan Mode) (Units 0 and 1)	1000	Notes added Notes: 1. Make the sampling setting15 (ADSSRT = D' 2. When Pϕ = Iϕ/2, make the sampling setting 2 (ADSSRT = D'25). 3. Unit 0: The full-spec emulator (E6000H) sho used, but the on-chip emulator (E10A-USB) 4. Unit 1: Access to the full-spec emulator (E60 prohibited but the on-chip emulator (E10A-U usable.

		reference power supply pin (V_{ref}), analog power suppl (AV_{CC}), and analog ground pin (AV_{SS}) by shielding the input pins (AN0 to AN7) with the analog ground pin (A
21.7.9 Notes on Noise Countermeasures	1011	Amended A protection circuit connected to prevent damage due to abnormal voltage such as an excessive surge at the ana input pins (AN0 to AN7) should be connected to AV_{CC} ar as shown in figure 21.14. Also, the bypass capacitors co to AV_{CC} and V_{ref} and the filter capacitor connected to the AN7 pins must be connected to AV_{SS}. … If a filter capacitor is connected, the input currents at the AN7 pins are averaged, and so an error may arise. …
Figure 21.14 Example of Analog Input Protection Circuit	1012	Amended 

J

L

M

N

O

Q

R

S

Renesas 32-Bit CISC Microcomputer
Hardware Manual
H8SX/1655 Group, H8SX/1655M Group

Publication Date: Rev.1.00, Mar. 25, 2009
Rev.2.00, Oct. 20, 2009
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, J

RENESAS SALES OFFICES

http://www.ren

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

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, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Ma
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8SX/1655 Group, H8SX/1655M Group Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0499-0200