To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

32

H8SX/1622 Group

Hardware Manual

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

H8SX/1622 R5F61622

Renesas Electronics
www.renesas.com

Rev.2.00 2009.09

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

5. Renesas has used reasonable care in compiling the information included in this document, but Renes
assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the inf
included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the informa
light of the total system before deciding about the applicability of such information to the intended app
Renesas makes no representations, warranties or guaranties regarding the suitability of its products f
particular application and specifically disclaims any liability arising out of the application and use of th
information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesa
products are not designed, manufactured or tested for applications or otherwise in systems the failure
malfunction of which may cause a direct threat to human life or create a risk of human injury or which
especially high quality and reliability such as safety systems, or equipment or systems for transportat
traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea comm
transmission. If you are considering the use of our products for such purposes, please contact a Ren
sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth a
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes list
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

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

RENESAS

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 ex
   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 part number
   that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part num
   differ because of the differences in internal memory capacity and layout pattern. W
   changing to products of different part 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.

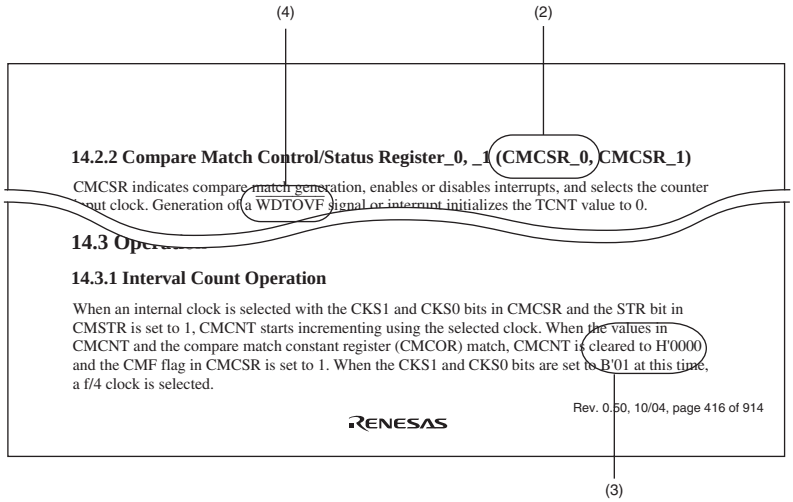
The list of revisions is a summary of major points of revision or addition for earlier v It does not cover all revised items. For details on the revised points, see the actual in the manual.

The following documents have been prepared for the H8SX/1622 Group. Before usi the documents, please visit our web site to verify that you have the most up-to-date a version of the document.

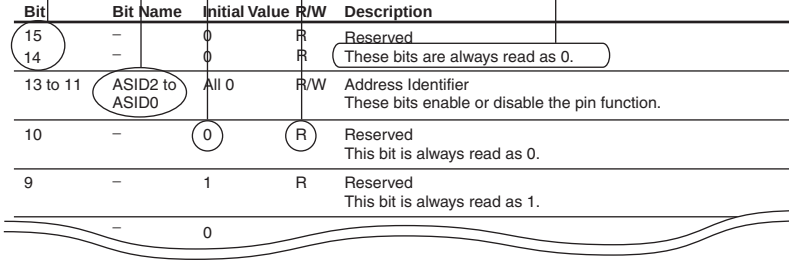
Document Type	Contents	Document Title	Docu
Data Sheet	Overview of hardware and electrical characteristics	—	—
Hardware Manual	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	H8SX/1622 Group Hardware Manual	This r
Software Manual	Detailed descriptions of the CPU and instruction set	H8SX Family Software Manual	REJ0
Application Note	Examples of applications and sample programs	The latest versions are available f web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.
[Examples] Binary: B'11 or 11
Hexadecimal: H'EFA0 or 0xEFA0
Decimal: 1234

(4) Notation for active-low
An overbar on the name indicates that a signal or pin is active-low.
[Example] $\overline{\text{WDTOVF}}$



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



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

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

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

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

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

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

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

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

SCI	Serial communication interface
TMR	8-bit timer
TPU	16-bit timer pulse unit
WDT	Watchdog timer
UBC	User break controller

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communication interface adapter
bps	Bits per second
DMA	Direct memory access
DMAC	Direct memory access controller
GSM	Global System for Mobile Communications
Hi-Z	High impedance
I/O	Input/output
LSB	Least significant bit
MSB	Most significant bit
NC	No connection
PLL	Phase-locked loop
PWM	Pulse width modulation
SIM	Subscriber Identity Module
UART	Universal asynchronous receiver/transmitter

speed data transfer, and a bus-state controller, which enables direct connection to differe
of memory. The LSI of the Group also includes a ΔΣ A/D converter specialized for sens
D/A converter, serial communication interfaces, and a multi-function timer that makes n
control easy. Together, the modules realize low-cost configurations for end systems. The
consumption of these modules are kept down dynamically by an on-chip power-manage
function.

1.1.1 Applications

Example of application: Consumer appliances

	Upward compatibility for H8/300, H8/300H, and H8S C object level • Sixteen 16-bit general registers • Eleven addressing modes • 4-Gbyte address space Program: 4 Gbytes available Data: 4 Gbytes available • 87 basic instructions, classifiable as bit arithmetic and l instructions, multiply and divide instructions, bit manipu instructions, multiply-and-accumulate instructions, and • Minimum instruction execution time: 20.0 ns (for an AD instruction when running with system clock Iϕ = 50 MHz = 3.0 to 3.6 V) • On-chip multiplier (16 × 16 → 32 bits) • Supports multiply-and-accumulate instructions (16 × 16 + 42 → 42 bits)
Operating mode	• Advanced mode (normal mode, middle mode, and maximum mode are unavailable)

		Mode 5: On-chip ROM disabled external extended mode, (selected by driving the MD1 pin low and driving and MD0 pins high) Mode 6: On-chip ROM enabled external extended mode (selected by driving the MD0 pin low and driving and MD1 pins high) Mode 7: Single-chip mode (can be externally extended) (selected by driving the MD2, MD1, and MD0 pir • Low power consumption state (transition driven by the instruction)
Interrupt (sources)	Interrupt controller (INTC)	• Seventeen external interrupt pins (NMI, and $\overline{\text{IRQ15}}$ to • 80 internal interrupt sources • Two interrupt control modes (specified by the interrupt register) • Eight priority orders specifiable (by setting the interrup register) • Independent vector addresses
	Break interrupt (UBC)	• Break points on four channels • Address break can be set at the CPU instruction fetch

	transfer controller (DTC)	activation sources) • Activated by interrupt sources (chain transfer enabled) • Three transfer modes (normal transfer, repeat transfer, transfer) • Short-address mode or full-address mode selectable
External bus extension	Bus controller (BSC)	• 16-Mbyte external address space
		• The external address space can be divided into eight a each of which is independently controllable — Chip-select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output — Access in two or three states can be selected for ea — Program wait cycles can be inserted — The period of $\overline{CS}$ assertion can be extended — Idle cycles can be inserted • Bus arbitration function (arbitrates bus mastership amo internal CPU, DMAC and DTC, and external bus maste
		Bus formats • External memory interfaces (for the connection of ROM ROM, SRAM, and byte control SRAM) • Address/data bus format: Support for both separate an multiplexed buses (8-bit access or 16-bit access)
		• Endian conversion function for connecting devices in lit endian format

		— External space modules are supplied with the exte clock (Bϕ): 8 to 50 MHz — ΔΣ A/D converter is run by the clock for ΔΣ A/D co (Aϕ): near 25 MHz • Includes a PLL frequency multiplier and frequency div (including a divider for Aϕ), so the operating frequency selectable • Five power-down modes: Sleep mode, all-module-cloc mode, software standby mode, deep software standby and hardware standby mode
A/D converter	10-bit A/D converter (ADC)	• 10-bit resolution × eight input channels • Sample and hold function included • Conversion time: 5.33 μs per channel (with ADCLK at operation) • Two operating modes: single mode and scan mode • Three ways to start A/D conversion: by software, time (TPU/TMR) trigger, and external trigger (Starting by TPU/TMR: This operation is available on t emulator but not available on other emulators.)
	16-bit ΔΣ A/D converter (ΔΣ AD)	• 16-bit resolution • Six input channels (differential inputs on two channels • Conversion time: 286 states • Two operating modes: single mode and scan mode • ΔΣ modulation • Three ways to start A/D conversion: by software, time (TPU/TMR) trigger, and external trigger (Starting by TPU/TMR: This operation is available on t emulator but not available on other emulators.) • Input voltage offset cancellation in two modes: by regi and by differential input

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

	• LQFP-144 package
Operating frequency/ Power supply voltage	• Operating frequency: 8 to 50 MHz • Power supply voltage: Vcc = PLLVcc = 3.0 to 3.6 V, A to 3.6 V, AVccP = AVccA = AVccD = 3.0 to 3.6 V • Flash program/erase voltage: 3.0 to 3.6 V • Supply current: — 48 mA (typ.) (Vcc = PLLVcc = 3.3 V, AVcc = 3.3 V AVccA = AVccD = 3.3 V, Iϕ = Bϕ = 50 MHz, Pϕ = 2 — 46 mA (typ.) (Vcc = PLLVcc = 3.3 V, AVcc = 3.3 V AVccA = AVccD = 3.0 V, Iϕ = Pϕ = Bϕ = 35 MHz)
Operating ambient temperature (°C)	• –20 to +75°C (regular specifications) • –40 to +85°C (wide-range specifications)

R5F61622D50LGV	256 Kbytes	24 Kbytes	LGA-145	

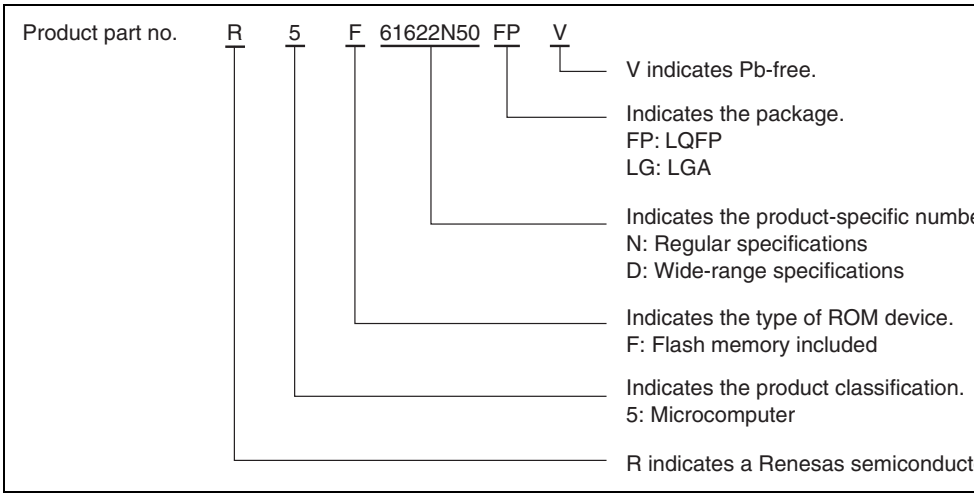
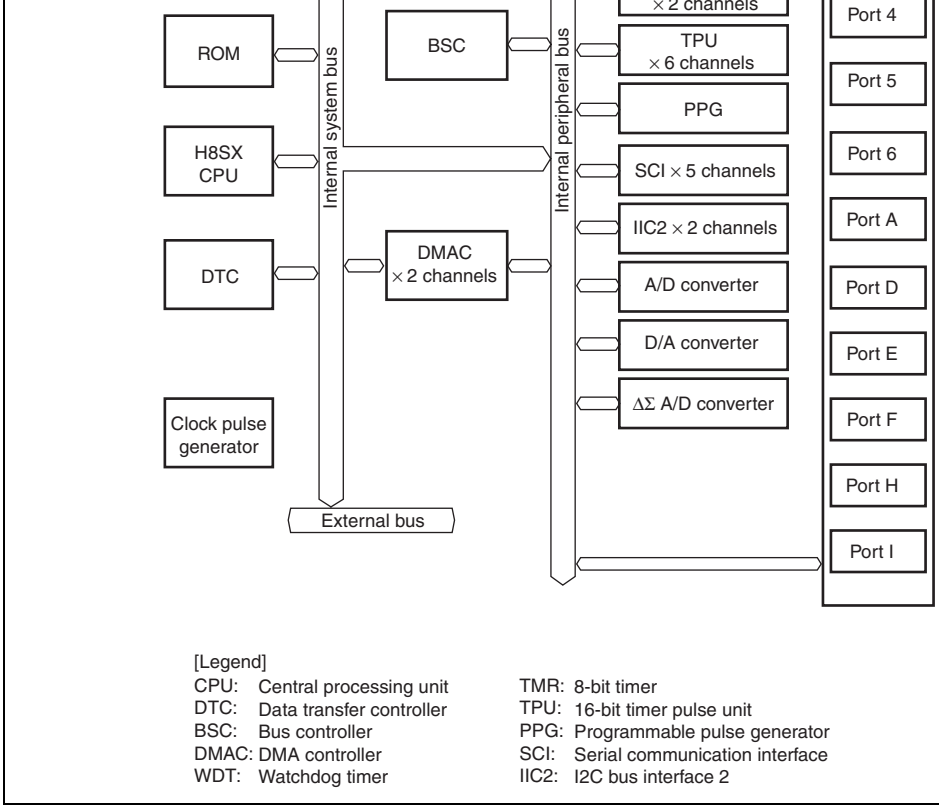


Figure 1.1 How to Read the Product Part No.

- Compact Package

Package	Code	Size	Pin Pitch
LGA-145	PTLG0145JB-A*	9.0 × 9.0 mm	0.65 mm
LQFP-144	PLQP0144KA-A*	20.0 × 20.0 mm	0.50 mm

Note: * Lead-free version



[Legend]

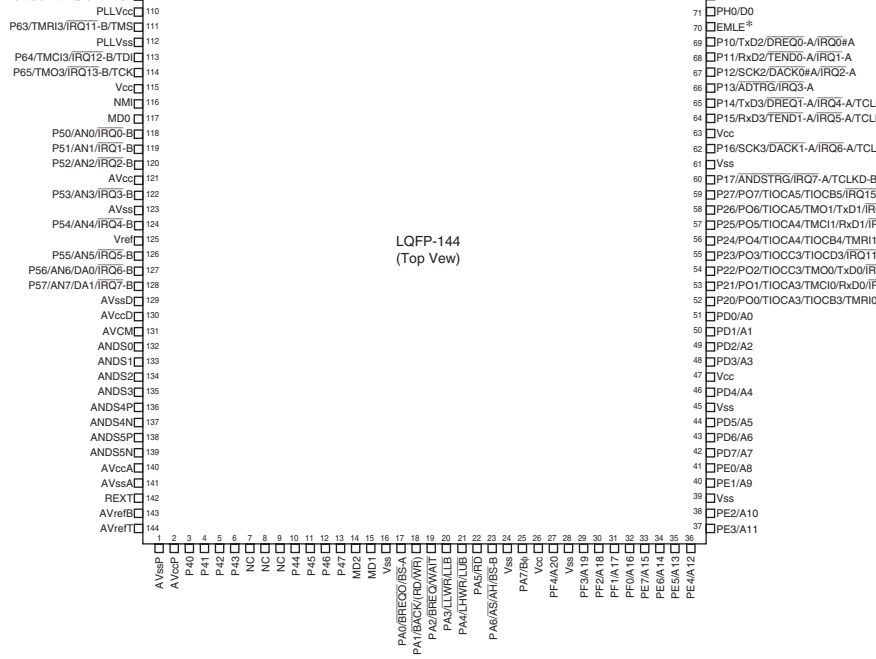
CPU:	Central processing unit	TMR:	8-bit timer
DTC:	Data transfer controller	TPU:	16-bit timer pulse unit
BSC:	Bus controller	PPG:	Programmable pulse generator
DMAC:	DMA controller	SCI:	Serial communication interface
WDT:	Watchdog timer	IIC2:	I2C bus interface 2

Figure 1.2 Block Diagram

D	NC	NC	NC	ANDS4P	ANDS0	P57	P56	P55	P53	$\overline{\text{WDTOVF}}$	P60	P31
E	P47	P44	P45	MD2	NC					Vss	$\overline{\text{STBY}}$	EXTA
F	PA0	P46	MD1	PA1						P33	P32	P36
G	PA4	Vss	PA6	PA2			LGA (Top view)			P37	P34	PI7
H	PA7	PA5	PF4	PA3						PI5	Vss	PI4
J	PF3	Vcc	PF1	Vss						Vcc	PI3	PI1
K	PF0	PF2	PE6	Vss	PD1	P22	P23	P24	P17	P15	PI0	PH7
L	PE5	PE7	PD7	PD5	Vcc	PD0	P27	Vcc	P12	EMLE*	PH5	Vss
M	PE4	PE2	Vss	PD6	PD4	PD3	P20	P26	P16	P13	P10	PH4
N	PE1	PE3	PE0	Vss	PD2	P21	P25	Vss	P14	P11	PH0	PH1

Note: * 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, P65, and $\overline{\text{WDTOVF}}$ pins are dedicated pins for the on-chip emulator. For details on a connection example with t see E10A Emulator User's Manual.

Figure 1.3 Pin Assignments (LGA-145)



Note: * 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 1.4 Pin Assignments (LQFP-144)

4	B2	P41	P41
5	C1	P42	P42
6	C3	P43	P43
7	D2	NC	NC
8	D3	NC	NC
9	D1	NC	NC
10	E2	P44	P44
11	E3	P45	P45
12	F2	P46	P46
13	E1	P47	P47
14	E4	MD2	MD2
15	F3	MD1	MD1
16	G2	V_{ss}	V_{ss}
17	F1	PA0/$\overline{\text{BREQO}}$/$\overline{\text{BS}}$-A	PA0/$\overline{\text{BREQO}}$/$\overline{\text{BS}}$-A
18	F4	PA1/$\overline{\text{BACK}}$/ (RD/$\overline{\text{WR}}$)	PA1/$\overline{\text{BACK}}$/ (RD/$\overline{\text{WR}}$)
19	G4	PA2/$\overline{\text{BREQ}}$/$\overline{\text{WAIT}}$	PA2/$\overline{\text{BREQ}}$/$\overline{\text{WAIT}}$
20	H4	PA3/$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$	PA3/$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$
21	G1	PA4/$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$	PA4/$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$
22	H2	PA5/$\overline{\text{RD}}$	PA5/$\overline{\text{RD}}$
23	G3	PA6/$\overline{\text{AS}}$/$\overline{\text{AH}}$/$\overline{\text{BS}}$-B	PA6/$\overline{\text{AS}}$/$\overline{\text{AH}}$/$\overline{\text{BS}}$-B
24	J4	V_{ss}	V_{ss}
25	H1	PA7/Bϕ	PA7/Bϕ
26	J2	V_{cc}	V_{cc}

34	K3	PE6/A14	A14
35	L1	PE5/A13	A13
36	M1	PE4/A12	A12
37	N2	PE3/A11	A11
38	M2	PE2/A10	A10
39	M3	V_{SS}	V_{SS}
40	N1	PE1/A9	A9
41	N3	PE0/A8	A8
42	L3	PD7/A7	A7
43	M4	PD6/A6	A6
44	L4	PD5/A5	A5
45	N4	V_{SS}	V_{SS}
46	M5	PD4/A4	A4
47	L5	V_{CC}	V_{CC}
48	M6	PD3/A3	A3
49	N5	PD2/A2	A2
50	K5	PD1/A1	A1
51	L6	PD0/A0	A0
52	M7	P20/PO0/TIOCA3/TIOCB3/TMRI0/SCK0/ $\overline{\text{IRQ8}}$-A	P20/PO0/TIOCA3/TIOCB3/TMRI0/S $\overline{\text{IRQ8}}$-A
53	N6	P21/PO1/TIOCA3/TMCI0/RxD0/$\overline{\text{IRQ9}}$-A	P21/PO1/TIOCA3/TMCI0/RxD0/$\overline{\text{IRQ}}$
54	K6	P22/PO2/TIOCC3/TMO0/TxD0/$\overline{\text{IRQ10}}$-A	P22/PO2/TIOCC3/TMO0/TxD0/$\overline{\text{IRQ}}$
55	K7	P23/PO3/TIOCC3/TIOCD3/$\overline{\text{IRQ11}}$-A	P23/PO3/TIOCC3/TIOCD3/$\overline{\text{IRQ11}}$-A

62	M9	P16/SCK3/$\overline{\text{DACK1}}$-A/$\overline{\text{IRQ6}}$-A/TCLKC-B/SDA0	P16/SCK3/$\overline{\text{DACK1}}$-A/$\overline{\text{IRQ6}}$-A/TCLKC
63	L8	V_{cc}	V_{cc}
64	K10	P15/RxD3/$\overline{\text{TEND1}}$-A/$\overline{\text{IRQ5}}$-A/TCLKB-B/SCL1	P15/RxD3/$\overline{\text{TEND1}}$-A/$\overline{\text{IRQ5}}$-A/TCLKB
65	N9	P14/TxD3/$\overline{\text{DREQ1}}$-A/$\overline{\text{IRQ4}}$-A/TCLKA-B/SDA1	P14/TxD3/$\overline{\text{DREQ1}}$-A/$\overline{\text{IRQ4}}$-A/TCLKA
66	M10	P13/$\overline{\text{ADTRG0}}$/$\overline{\text{IRQ3}}$-A	P13/$\overline{\text{ADTRG0}}$/$\overline{\text{IRQ3}}$-A
67	L9	P12/SCK2/$\overline{\text{DACK0}}$-A/$\overline{\text{IRQ2}}$-A	P12/SCK2/$\overline{\text{DACK0}}$-A/$\overline{\text{IRQ2}}$-A
68	N10	P11/RxD2/$\overline{\text{TEND0}}$-A/$\overline{\text{IRQ1}}$-A	P11/RxD2/$\overline{\text{TEND0}}$-A/$\overline{\text{IRQ1}}$-A
69	M11	P10/TxD2/$\overline{\text{DREQ0}}$-A/$\overline{\text{IRQ0}}$-A	P10/TxD2/$\overline{\text{DREQ0}}$-A/$\overline{\text{IRQ0}}$-A
70	L10	EMLE	EMLE
71	N11	PH0/D0	PH0/D0
72	N12	PH1/D1	PH1/D1
73	M13	PH2/D2	PH2/D2
74	N13	PH3/D3	PH3/D3
75	L12	V_{ss}	V_{ss}
76	M12	PH4/D4	PH4/D4
77	L11	PH5/D5	PH5/D5
78	L13	PH6/D6	PH6/D6
79	K12	PH7/D7	PH7/D7
80	K11	PI0/D8	PI0/D8
81	J12	PI1/D9	PI1/D9
82	K13	PI2/D10	PI2/D10
83	J10	V_{cc}	V_{cc}
84	J11	PI3/D11	PI3/D11

92	F12	P36/PO14/TIOCA2	P36/PO14/TIOCA2
93	G13	P35/PO13/TIOCA1/TIOCB1/TCLKC-A/ $\overline{\text{DACK1}}$-B	P35/PO13/TIOCA1/TIOCB1/TCLKC $\overline{\text{DACK1}}$-B
94	G11	P34/PO12/TIOCA1/$\overline{\text{TEND1}}$-B	P34/PO12/TIOCA1/$\overline{\text{TEND1}}$-B
95	F10	P33/PO11/TIOCC0/TIOCD0/TCLKB-A/ $\overline{\text{DREQ1}}$-B/$\overline{\text{CS3}}$/$\overline{\text{CS7}}$-A	P33/PO11/TIOCC0/TIOCD0/TCLKB $\overline{\text{DREQ1}}$-B/$\overline{\text{CS3}}$/$\overline{\text{CS7}}$-A
96	E10	V_{SS}	V_{SS}
97	F13	XTAL	XTAL
98	E12	EXTAL	EXTAL
99	E13	V_{CC}	V_{CC}
100	F11	P32/PO10/TIOCC0/TCLKA-A/$\overline{\text{DACK0}}$-B/ $\overline{\text{CS2}}$-A/$\overline{\text{CS6}}$-A	P32/PO10/TIOCC0/TCLKA-A/$\overline{\text{DACK}}$ $\overline{\text{CS2}}$-A/$\overline{\text{CS6}}$-A
101	D12	P31/PO9/TIOCA0/TIOCB0/$\overline{\text{TEND0}}$-B/ $\overline{\text{CS1}}$/$\overline{\text{CS2}}$-B/$\overline{\text{CS5}}$-A/$\overline{\text{CS6}}$-B/$\overline{\text{CS7}}$-B	P31/PO9/TIOCA0/TIOCB0/$\overline{\text{TEND0}}$- $\overline{\text{CS1}}$/$\overline{\text{CS2}}$-B/$\overline{\text{CS5}}$-A/$\overline{\text{CS6}}$-B/$\overline{\text{CS7}}$-B
102	E11	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$
103	D13	V_{CL}	V_{CL}
104	D10	$\overline{\text{WDTOVF}}$/TDO	$\overline{\text{WDTOVF}}$/TDO
105	C12	V_{SS}	V_{SS}
106	C13	P30/PO8/TIOCA0/$\overline{\text{DREQ0}}$-B/$\overline{\text{CS0}}$/$\overline{\text{CS4}}$/$\overline{\text{CS5}}$-B	P30/PO8/TIOCA0/$\overline{\text{DREQ0}}$-B/$\overline{\text{CS0}}$/$\overline{\text{C}}$
107	D11	P60/TMRI2/TxD4/$\overline{\text{IRQ8}}$-B	P60/TMRI2/TxD4/$\overline{\text{IRQ8}}$-B
108	B13	P61/TMCI2/RxD4/$\overline{\text{IRQ9}}$-B	P61/TMCI2/RxD4/$\overline{\text{IRQ9}}$-B
109	A12	P62/TMO2/SCK4/$\overline{\text{IRQ10}}$-B/$\overline{\text{TRST}}$	P62/TMO2/SCK4/$\overline{\text{IRQ10}}$-B/$\overline{\text{TRST}}$
110	A13	$PLLV_{CC}$	$PLLV_{CC}$
111	B11	P63/TMRI3/$\overline{\text{IRQ11}}$-B/TMS	P63/TMRI3/$\overline{\text{IRQ11}}$-B/TMS

119	C9	P51/AN1/$\overline{\text{IRQ1}}$-B	P51/AN1/$\overline{\text{IRQ1}}$-B
120	B8	P52/AN2/$\overline{\text{IRQ2}}$-B	P52/AN2/$\overline{\text{IRQ2}}$-B
121	A9	AV_{cc}	AV_{cc}
122	D9	P53/AN3/$\overline{\text{IRQ3}}$-B	P53/AN3/$\overline{\text{IRQ3}}$-B
123	C8	AV_{ss}	AV_{ss}
124	B7	P54/AN4/$\overline{\text{IRQ4}}$-B	P54/AN4/$\overline{\text{IRQ4}}$-B
125	A8	Vref	Vref
126	D8	P55/AN5/$\overline{\text{IRQ5}}$-B	P55/AN5/$\overline{\text{IRQ5}}$-B
127	D7	P56/AN6/DA0/$\overline{\text{IRQ6}}$-B	P56/AN6/DA0/$\overline{\text{IRQ6}}$-B
128	D6	P57/AN7/DA1/$\overline{\text{IRQ7}}$-B	P57/AN7/DA1/$\overline{\text{IRQ7}}$-B
129	A7	AV_{ss}D	AV_{ss}D
130	B6	AV_{cc}D	AV_{cc}D
131	C7	AVCM	AVCM
132	D5	ANDS0	ANDS0
133	A6	ANDS1	ANDS1
134	B5	ANDS2	ANDS2
135	C6	ANDS3	ANDS3
136	D4	ANDS4P	ANDS4P
137	A5	ANDS4N	ANDS4N
138	B4	ANDS5P	ANDS5P
139	C5	ANDS5N	ANDS5N
140	A4	AV_{cc}A	AV_{cc}A

	$PLLV_{SS}$	Input	Ground pin for the PLL circuit.
Clock	XTAL	Input	Pins for a crystal resonator. An external clock sign
	EXTAL	Input	input through the EXTAL pin. For an example of th connection, see section 23, Clock Pulse Generato
	Bϕ	Output	Outputs the system clock for external devices.
Operating mode control	MD2 to MD0	Input	Pins for setting the operating mode. The signal lev these pins must not be changed during operation.
System control	$\overline{RES}$	Input	Reset signal input pin. This LSI enters the reset st this signal goes low.
	$\overline{STBY}$	Input	This LSI enters hardware standby mode when this goes low.
	EMLE	Input	Input pin for the on-chip emulator enable signal. In level when using the on-chip emulator, and input a when not using the on-chip emulator.
On-chip emulator	$\overline{TRST}$	Input	Pins for the on-chip emulator
	TMS	Input	Driving the EMLE pin high makes these pins to fun on-chip emulator pins.
	TDI	Input	
	TCK	Input	
	TDO	Output	
Address bus	A20 to A0	Output	Output pins for the address bits.
Data bus	D15 to D0	Input/ output	Input and output for the bidirectional data bus. The also output addresses when accessing an address multiplexed I/O interface space.
Bus control	$\overline{BREQ}$	Input	External bus-master modules assert this signal to the bus.
	$\overline{BREQO}$	Output	Internal bus-master modules assert this signal to r access to the external space via the bus in the ext released state.

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

	$\overline{IRQ6}$-A/$\overline{IRQ6}$-B $\overline{IRQ5}$-A/$\overline{IRQ5}$-B $\overline{IRQ4}$-A/$\overline{IRQ4}$-B $\overline{IRQ3}$-A/$\overline{IRQ3}$-B $\overline{IRQ2}$-A/$\overline{IRQ2}$-B $\overline{IRQ1}$-A/$\overline{IRQ1}$-B $\overline{IRQ0}$-A/$\overline{IRQ0}$-B		
DMA controller (DMAC)	$\overline{DREQ0}$-A/$\overline{DREQ0}$-B $\overline{DREQ1}$-A/$\overline{DREQ1}$-B	Input	Requests DMAC activation.
	$\overline{DACK0}$-A/$\overline{DACK0}$-B $\overline{DACK1}$-A/$\overline{DACK1}$-B	Output	DMAC single address-transfer acknowledge signa
	$\overline{TEND0}$-A/$\overline{TEND0}$-B $\overline{TEND1}$-A/$\overline{TEND1}$-B	Output	Indicates end of data transfer by the DMAC.
16-bit timer pulse unit (TPU)	TCLKA-A/TCLKA-B TCLKB-A/TCLKB-B TCLKC-A/TCLKC-B TCLKD-A/TCLKD-B	Input	Input pins for the external clock signals.
	TIOCA0 TIOCB0 TIOCC0 TIOCD0	Input/ output	Signals for TGRA_0 to TGRD_0. These pins are u input capture inputs, output compare outputs, or P outputs.
	TIOCA1 TIOCB1	Input/ output	Signals for TGRA_1 and TGRB_1. These pins are input capture inputs, output compare outputs, or P outputs.

	TIOCA5 TIOCB5	Input/ output	Signals for TGRA_5 and TGRB_5. These pins ar input capture inputs, output compare outputs, or outputs.
Programmable pulse generator (PPG)	PO15 to PO0	Output	Output pins for the pulse signals.
8-bit timer (TMR)	TMO0 to TMO7	Output	Output pins for the compare match signals.
	TMCI0 to TMCI3	Input	Input pins for the external clock signals that drive counters.
	TMRI0 to TMRI3	Input	Input pins for the counter-reset signals.
Watchdog timer (WDT)	$\overline{\text{WDTOVF}}$	Output	Output pin for the counter-overflow signal in watc mode.
Serial communication interface (SCI)	TxD0 to TxD4	Output	Output pins for data transmission.
	RxD0 to RxD4	Input	Input pins for data reception.
	SCK0 to SCK4	Input/ output	Input/output pins for clock signals.
I^2C bus interface 2 (IIC2)	SCL0, SCL1	Input/ output	Input/output pins for clock signals for the IIC2. Th can drive the bus directly with NMOS open-drain
	SDA0, SDA1	Input/ output	Input/output pins for data signals for the IIC2. The can drive the bus directly with NMOS open-drain

			to the system power supply (0 V).
	Vref	Input	Reference power supply pin for the A/D and D/A c When the A/D and D/A converters are not in use, c this pin to the system power supply.
ΔΣ A/D converter	ANDS5N ANDS5P ANDS4N ANDS4P ANDS3 ANDS2 ANDS1 ANDS0	Input	Analog input pins for the ΔΣ A/D converter.
	$\overline{\text{ANDSTRG}}$	Input	External trigger input pin for starting ΔΣ A/D conve
	$AV_{CC}A$	Input	Analog power supply pin for the ΔΣ A/D converter. When not using the ΔΣ A/D converter, connect this the system power supply.
	$AV_{SS}A$	Input	Ground pin for the ΔΣ A/D converter. When not using the ΔΣ A/D converter, connect this the system power supply (0 V).
	$AV_{CC}D$	Input	Analog power supply pin for the ΔΣ A/D converter. When not using the ΔΣ A/D converter, connect this the system power supply.
	$AV_{SS}D$	Input	Ground pin for the ΔΣ A/D converter. When not using the ΔΣ A/D converter, connect this the system power supply (0 V).
	$AV_{ref}T$	Input	The same power as $AV_{CC}A$ and $AV_{SS}A$ is input to A $AV_{ref}B$, respectively. See section 19.2, Input/Output Pins, for details.
	$AV_{ref}B$	Input	

			When not using the ΔΣ A/D converter, connect th the system power supply (0 V).
I/O ports	P17 to P10	Input/ output	8 input/output pins.
	P27 to P20	Input/ output	8 input/output pins.
	P37 to P30	Input/ output	8 input/output pins.
	P47 to P40	Input	8 input pins.
	P57 to P50	Input	8 input/output pins.
	P65 to P60	Input/ output	6 input/output pins.
	PA7	Input	Input-only pin
	PA6 to PA0	Input/ output	7 input/output pins.
	PD7 to PD0	Input/ output	8 input/output pins.
	PE7 to PE0	Input/ output	8 input/output pins.
	PF4 to PF0	Input/ output	5 input/output pins.
	PH7 to PH0	Input/ output	8 input/output pins.
	PI7 to PI0	Input/ output	8 input/output pins.

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

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

Notes: 1. Advanced mode is only supported as the CPU operating mode of the H8SX/16
Group. Normal, middle, and maximum modes are not supported.
2. The multiplier and divider are supported by the H8SX/1622 Group.

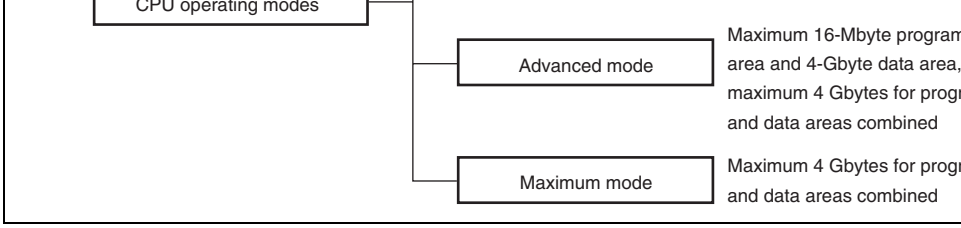


Figure 2.1 CPU Operating Modes

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

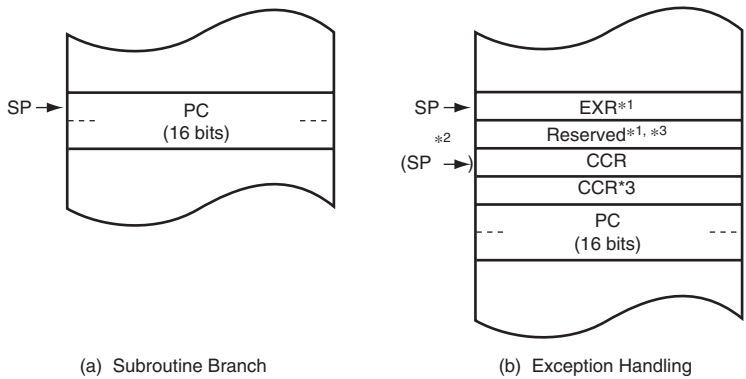
  All instructions and addressing modes can be used. Only the lower 16 bits of effectiv
  addresses (EA) are valid.

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 effecti
  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 lo
  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.

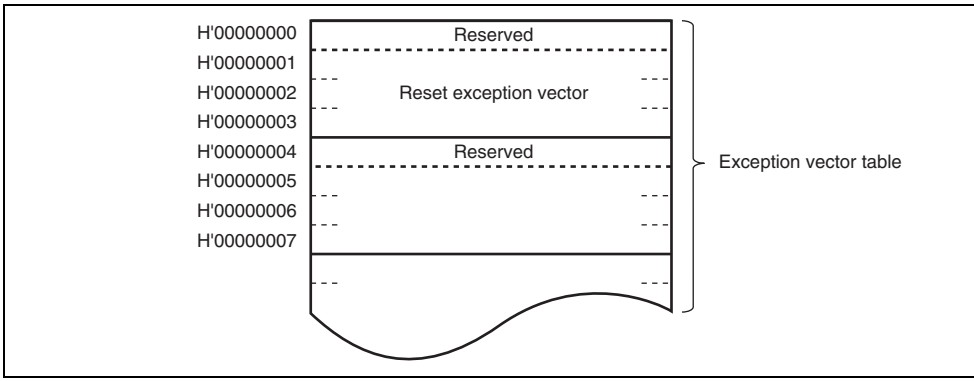


Figure 2.4 Exception Vector Table (Middle and Advanced Modes)

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.

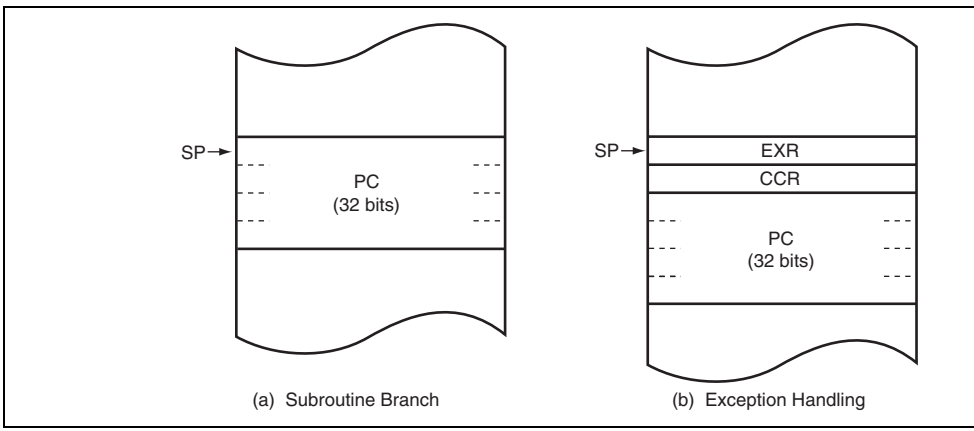


Figure 2.7 Stack Structure (Maximum Mode)

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

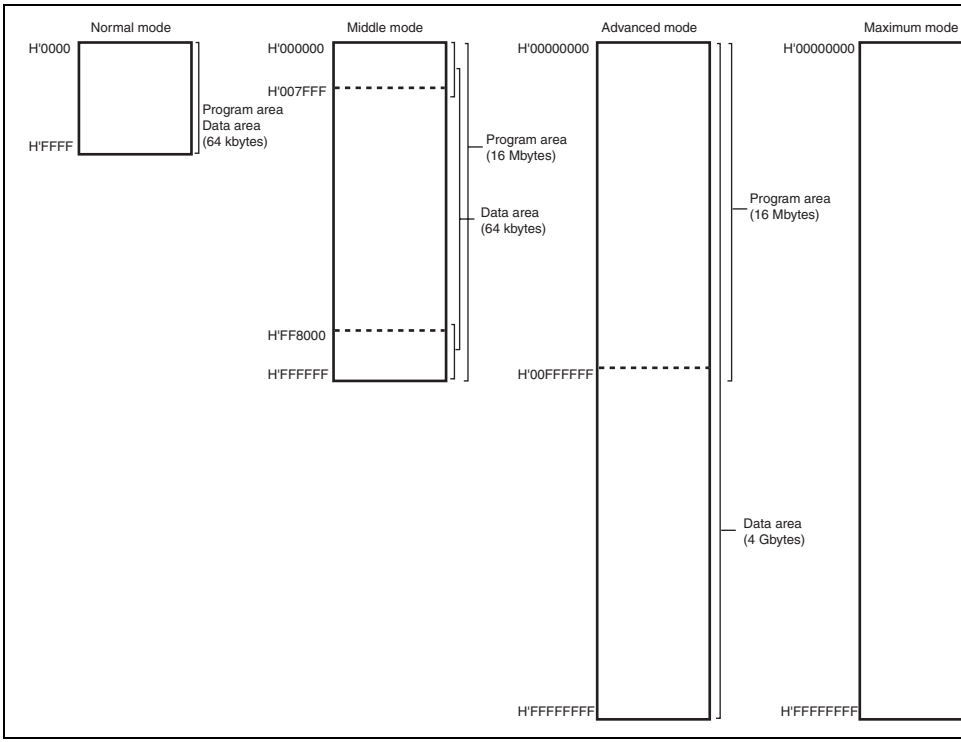
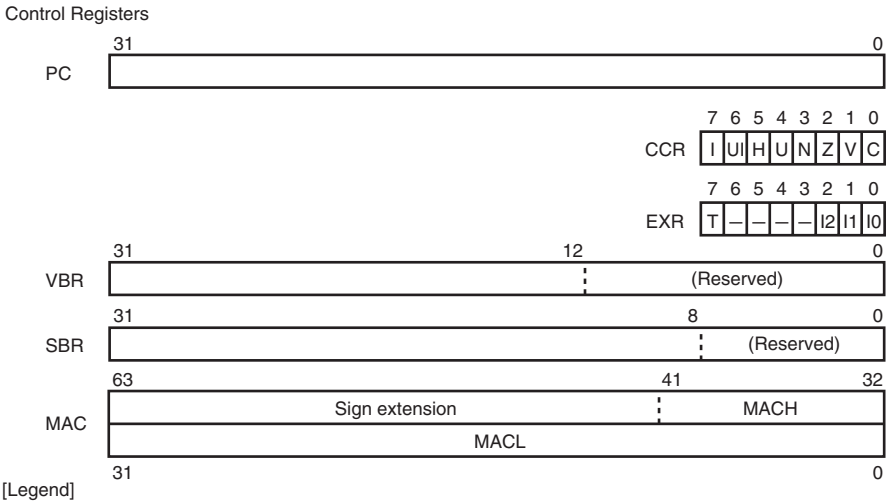


Figure 2.8 Memory Map

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



[Legend]

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

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.

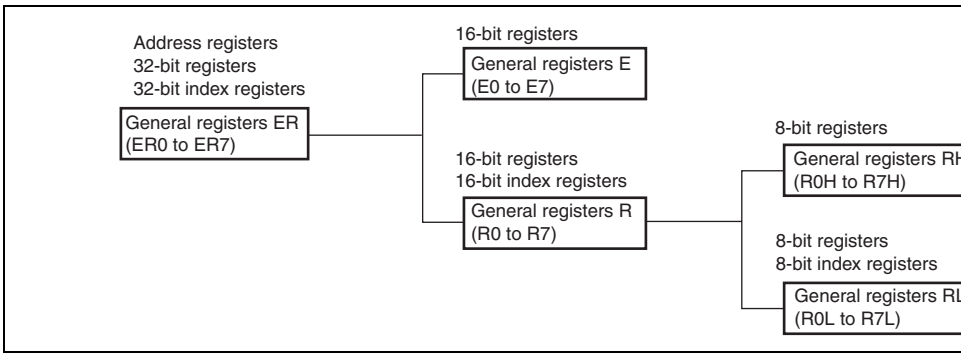


Figure 2.10 Usage of General Registers

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 start of an exception handling.
6	UI	Undefined	R/W	User 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 section 5, Exception Handling.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception 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 t
bits in CCR are not initialized. In particular, the initial value of the stack pointer (ER7) i
undefined. The SP should therefore be initialized using an MOV.L instruction executed
immediately after a reset.

Figure 2.12 shows the data formats in general registers.

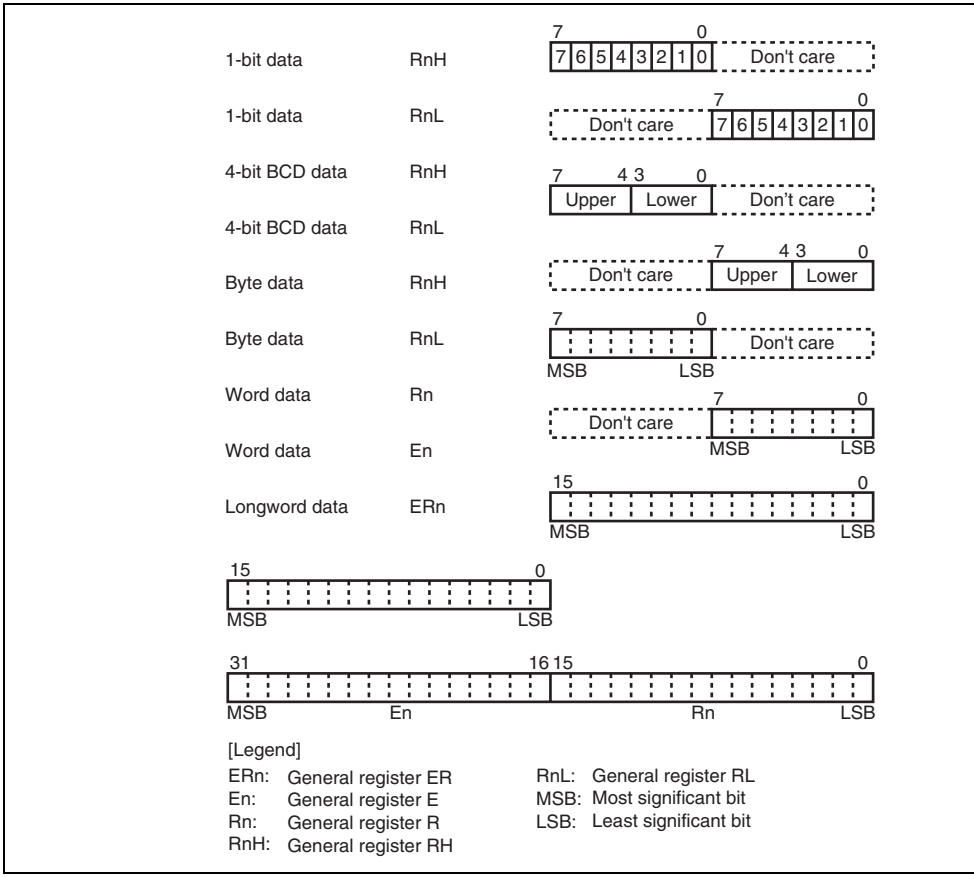


Figure 2.12 General Register Data Formats

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

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

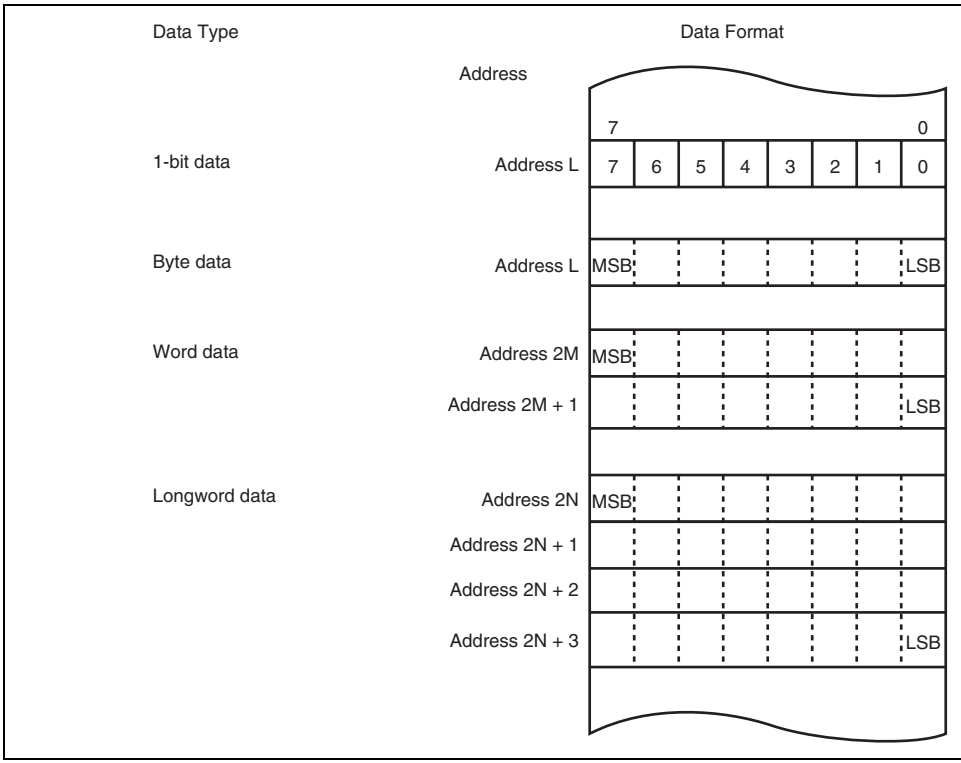


Figure 2.13 Memory Data Formats

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

[Legend]

B: Byte size

W: Word size

L: Longword size

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

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

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

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

	MAC	—								
	CLRMAC	—								
	LDMAC	—		S						
	STMAC	—		D						
Logic operations	AND, OR, XOR	B		S	D	D	D	D	D	D
		B		D	S	S	S	S	S	S
		B			SD	SD	SD	SD		SD
		W/L	S	SD	SD	SD	SD	SD		SD
	NOT	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D
Shift	SHLL, SHLR	B		D	D	D	D	D	D	D
		B/W/L*[6]		D	D	D	D	D		D
		B/W/L*[7]		D						
	SHAL, SHAR ROTL, ROTR ROTXL, ROTXR	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D
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 d 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 a 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. Not available in this LSI.

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

[Legend]

d: d:8 or d:16

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

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

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

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

Note: Not available in this LSI.

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

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

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

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

Table 2.8 Shift Operation Instructions

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

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

field in memory location contents.

Table 2.10 Branch Instructions

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

		location to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper 8 bits are val
	L	Rs → VBR, Rs → SBR Transfers the general register contents to VBR or SBR.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers the contents of CCR or EXR to a general register or r Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper 8 bits are val
	L	VBR → Rd, SBR → Rd Transfers the contents of VBR or SBR to a general register.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immedia
NOP	—	PC + 2 → PC Only increments the program counter.

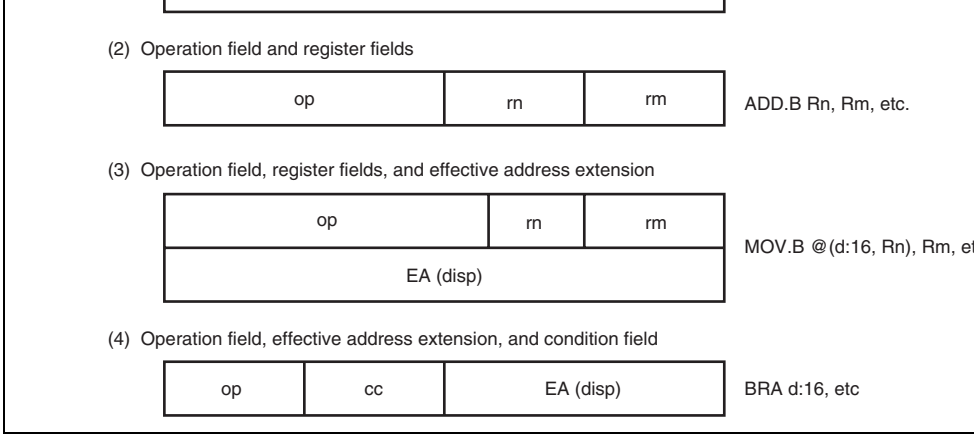


Figure 2.14 Instruction Formats

- Operation Field

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

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

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

  Specifies the branch condition of Bcc instructions.

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

2.8.1 Register Direct—Rn

The operand value is the contents of an 8-, 16-, or 32-bit general register which is specif 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 of contents of an address register (ERn) and a 16- or 32-bit displacement. ERn is specified b register field of the instruction code. The displacement is included in the instruction code 16-bit displacement is sign-extended when added to ERn.

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

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

The operand value is the contents of a memory location which is pointed to by the sum of 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 in code and the 16-bit displacement is sign-extended when added to ERn. If the operand is b ERn is multiplied by 1. If the operand is word or longword data, ERn is multiplied by 2 o respectively.

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

- Register indirect with pre-increment—@+ERn

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

- Register indirect with post-decrement—@ERn–

  The operand value is the contents of a memory location which is pointed to by the co
  an address register (ERn). ERn is specified by the register field of the instruction cod
  the memory location is accessed, 1, 2, or 4 is subtracted from the address register co
  the remainder is stored in the address register. The value subtracted is 1 for byte acce
  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 calcu
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 bit (@aa:32) is used. For an 8-bit absolute address, the upper 24 bits are specified by SBR. F bit absolute address, the upper 16 bits are sign-extended. A 32-bit absolute address can ac entire address space.

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

Table 2.13 shows the accessible absolute address ranges.

Table 2.13 Absolute Address Access Ranges

Absolute Address		Normal Mode	Middle Mode	Advanced Mode	Maximu Mode
Data area	8 bits (@aa:8)	A consecutive 256-byte area (the upper address is set in SBI			
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF,	H'00000000 to H'0000 H'FFFF8000 to H'FFFF	
	32 bits (@aa:32)		H'FF8000 to H'FFFFFF	H'00000000 to H'FFFF	
Program area	24 bits (@aa:24)		H'000000 to H'FFFFFF	H'00000000 to H'00FF	
	32 bits (@aa:32)			H'00000000 to H'00FFFFFF	H'00000 H'FFFF

manipulation instructions contain 3-bit immediate data in the instruction code, for specif
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 in
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 res
value should be an even number. In advanced mode, only the lower 24 bits of this branc
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 i
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 ar
vector address of an exception handling other than a reset or a CPU address error can be c
by VBR.

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

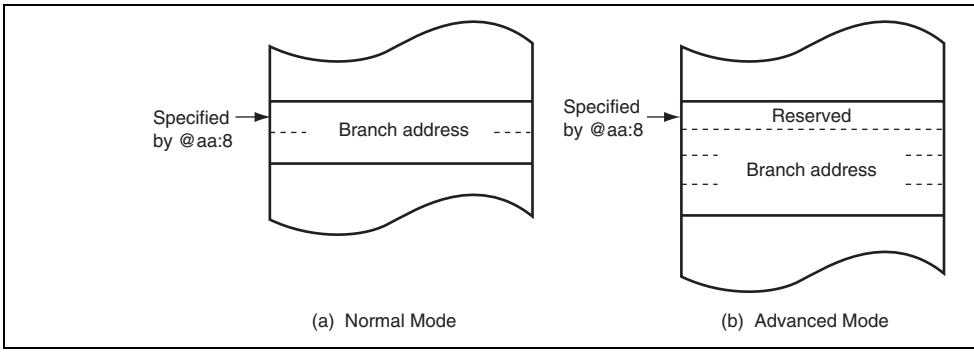


Figure 2.15 Branch Address Specification in Memory Indirect Mode

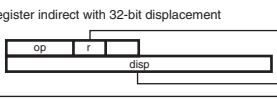
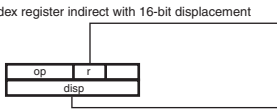
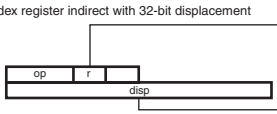
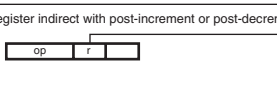
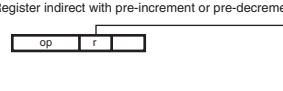
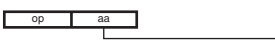
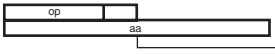
advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

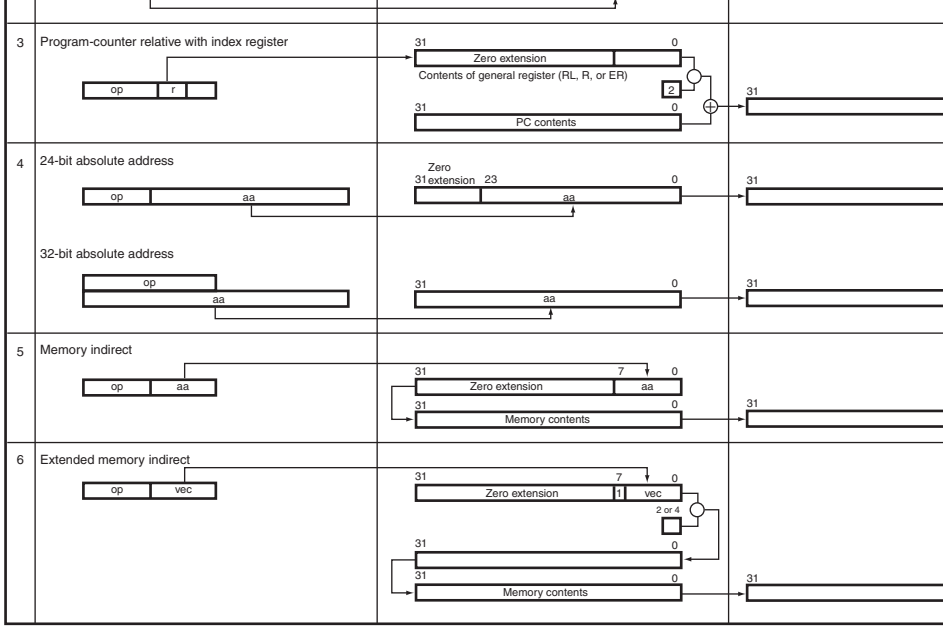
2.8.12 Effective Address Calculation

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

The valid bits in middle mode are as follows:

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

	Register indirect with 32-bit displacement 	
5	Index register indirect with 16-bit displacement  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  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 a
   mode shown in item 5 of table 2.14. The obtained data is used instead of the general
   The result is stored in a general register. For details, see H8SX Family Software Ma

- Exception-handling state

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

- Program execution state

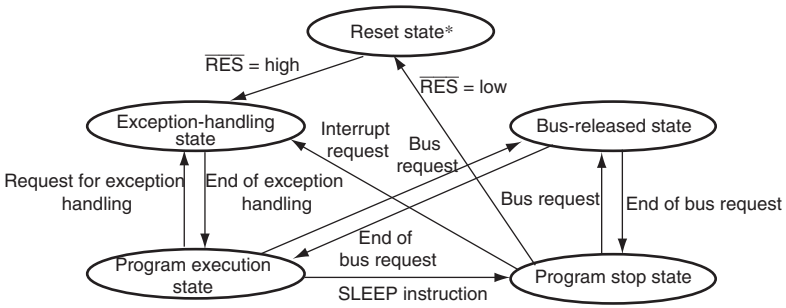
  In this state the CPU executes program instructions in sequence.

- Bus-released state

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

- Program stop state

  This is a power-down state in which the CPU stops operating. The program stop state when a SLEEP instruction is executed or the CPU enters hardware standby mode. For see section 24, Power-Down Modes.



Note: In any state, when the STBY signal goes low, a transition to the hardware standby mode occurs.
* In any state except the hardware standby mode, when the RES signal goes low, a transition to the reset state occurs. A transition to the reset state can also be made without driving the RES signal low. For details, see section 4, Resets.

Figure 2.16 State Transitions

Mode	MD2	MD1	MD0	Mode	Space	Mode	ROM	Defa
1	0	0	1	Advanced mode	16 Mbytes	User boot mode	Enabled	—
2	0	1	0			Boot 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 eight 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.

Mode 7 is a single-chip initiation mode. All I/O ports can be used as general input/outpu
The external address space cannot be accessed in the initial state, but setting the EXPE b
system control register (SYSCR) to 1 enables to use the external address space. After th
address space is enabled, ports H and I can be used as a data bus, and ports D, E, and F c
used as an address output bus by specifying the data direction register (DDR) for each p

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

3.2.1 Mode Control Register (MDCR)

MDCR indicates the current operating mode. When MDCR is read from, the states of sig
MD2 to MD0 are latched. 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*	Un
R/W	R	R	R	R	R	R	R	

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

Note: * Determined by pins MD2 to MD0.

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

Note: * Determined by pins MD2 to MD0.

Table 3.2 Settings of Bits MDS3 to MDS0

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

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

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

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

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

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

This is the boot mode for the flash memory. The LSI operates in the same way as in mode except for programming and erasing of the flash memory. For details, see section 22, Flas Memory.

3.3.3 Mode 4

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

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

3.3.4 Mode 5

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

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

3.3.6 Mode 7

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

All I/O ports can be used as general input/output ports. The external address space canno
accessed in the initial state, but setting the EXPE bit in the system control register (SYS
enables the external address space. After the external address space is enabled, ports H a
be used as a data bus, and ports D, E, and F can be used as an address output bus by spec
data direction register (DDR) for each port. For details, see section 11, I/O Ports.

Port 3	P33 to P31	P*/C	P*/C	P*/C	P*/C	P*/C	P*
	P30	P*/C	P*/C	P/C*	P/C*	P*/C	P*
Port D		P*/A	P*/A	A	A	P*/A	P*
Port E		P*/A	P*/A	A	A	P*/A	P*
Port F	PF4 to PF0	P*/A	P*/A	P/A*	P/A*	P*/A	P*
Port H		P*/D	P*/D	D	D	D	P*
Port I		P*/D	P*/D	P/D*	P*/D	P*/D	P*

[Legend]

P: I/O port

A: Address bus output

D: Data bus input/output

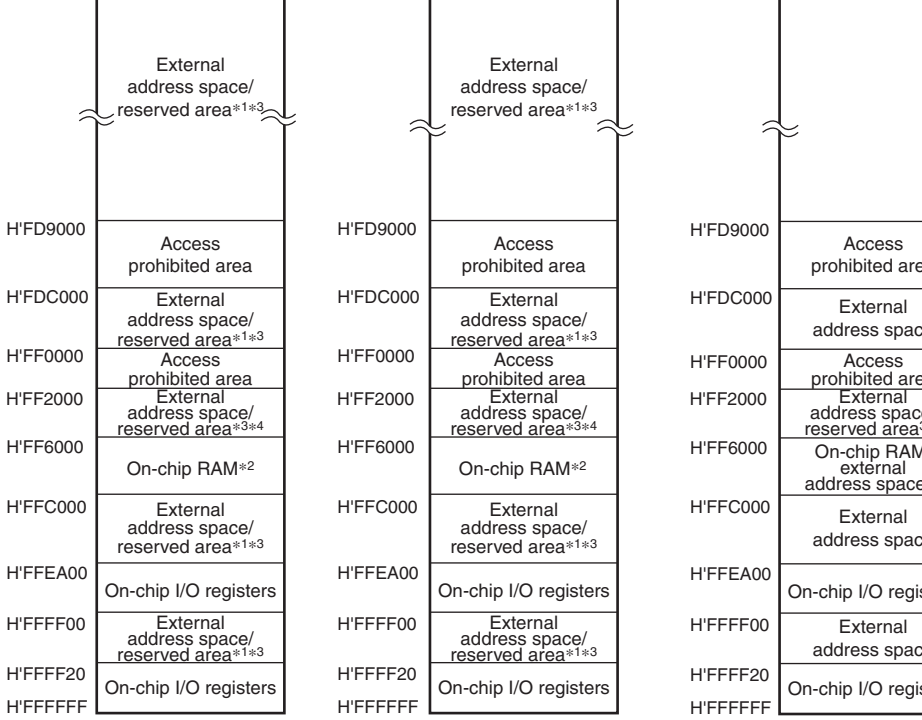
C: Control signals, clock input/output

*: Immediately after a reset

3.4 Address Map

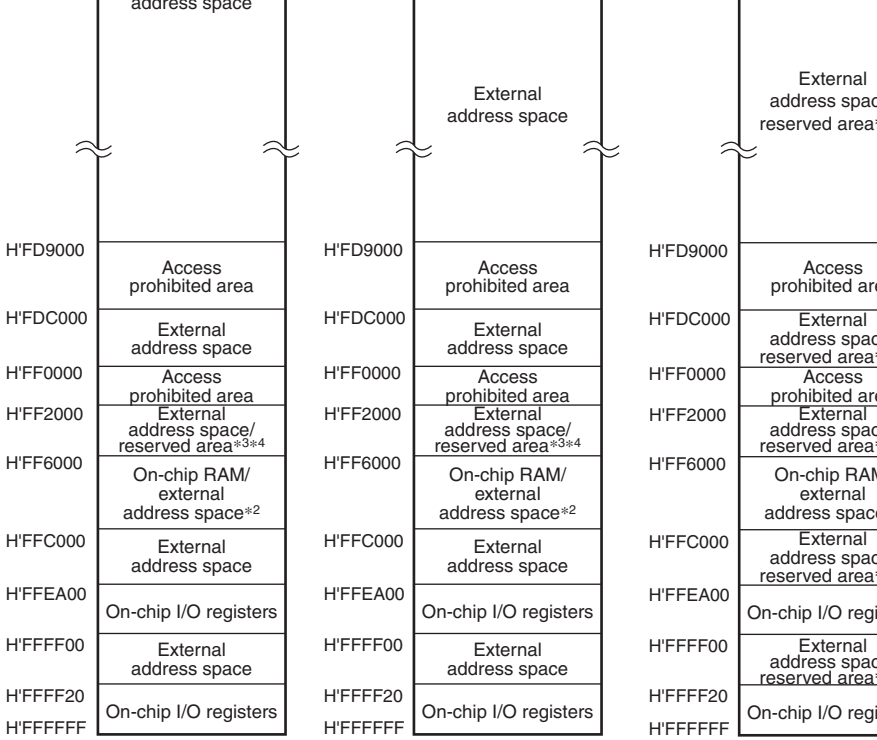
3.4.1 Address Map

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 as the reserved area when
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/1622 (1)



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

Figure 3.1 Address Map in Each Operating Mode of H8SX/1622 (2)

Table 4.1 Reset Names And Sources

Reset Name	Source
Pin reset	Voltage input to the $\overline{\text{RES}}$ pin is driven low.
Deep software standby reset	Deep software standby mode is canceled b interrupt.
Watchdog timer reset	The watchdog timer overflows.

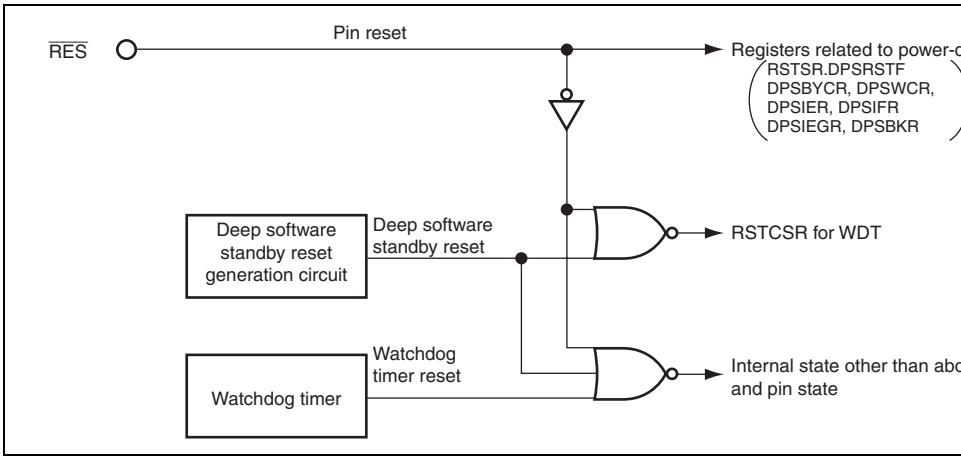


Figure 4.1 Block Diagram of Reset Circuit

When a reset is canceled, the reset exception handling is started. For the reset exception h
see section 5.3, Reset.

4.2 Input/Output Pin

Table 4.2 shows the pin related to resets.

Table 4.2 Pin Configuration

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	DPSRSTF	0	R/(W)*	Deep Software Standby Reset Flag Indicates that deep software standby mode is c by an external interrupt source specified with D DPSIEGR and an internal reset is generated. [Setting condition] When deep software standby mode is cancele external interrupt source. [Clearing conditions] • When this bit is read as 1 and then written • When a pin reset is generated.
6 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.

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

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 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 24, Power-Down Modes.

4.6 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 TCI overflow. After a certain time, the watchdog timer reset is canceled.

For details of the watchdog timer reset, see section 15, Watchdog Timer (WDT).

4.7 Determination of Reset Generation Source

Reading RSTCSR and RSTSR determines which reset was used to execute the reset exc handling. Figure 4.2 shows an example the flow to identify a reset generation source.

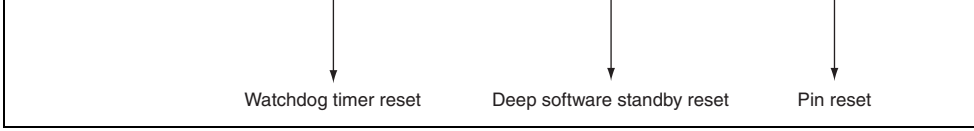


Figure 4.2 Example of Reset Generation Source Determination Flow

Table 5.1 Exception Types and Priority

Priority	Exception Type	Exception Handling Start Timing
High	Reset	Exception handling starts at the timing of low-to-high trans the $\overline{\text{RES}}$ pin, watchdog timer overflow, or input of an exter interrupt signal*4 in deep standby mode. The CPU enters state when the $\overline{\text{RES}}$ pin is low.
	Illegal instruction	Exception handling starts when an undefined code is exec
	Trace*1	Exception handling starts after execution of the current ins exception handling when the trace (T) bit in EXR has beer
	Address error	After an address error has occurred, exception handling s completion of instruction execution.
	Interrupt	When an interrupt request has occurred, exception handli after execution of the current instruction or exception hanc
	Sleep instruction	Exception handling starts by execution of a sleep instructi (SLEEP) when the SSBY bit in SBYCR has been cleared the SLPIE bit in SBYCR has been set to 1.
Low	Trap instruction*3	Exception handling starts by execution of a trap instructio (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.
4. The external interrupt input pins usable in deep software standby mode are IR IRQ0 (IRQnA pins only) and NMI.

section 3, MCU Operating Modes.

Table 5.2 Exception Handling Vector Table

Exception Source		Vector Number	Vector Table Address Offset: Normal Mode*2	Vector Table Address Offset: Advanced, 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	H'001E to H'001F	H'003C to H'
		17	H'0022 to H'0023	H'0044 to H'0
Sleep instruction		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
	IRQ12	76	H'0098 to H'0099	H'0130 to H
	IRQ13	77	H'009A to H'009B	H'0134 to H
	IRQ14	78	H'009C to H'009D	H'0138 to H
	IRQ15	79	H'009E to H'009F	H'013C to H
Internal interrupt*4		80 ⎮ 255	H'00A0 to H'00A1 ⎮ H'01FE to H'01FF	H'0140 to H ⎮ H'03FC to H

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. A DMA address error is generated by the DTC and DMAC.
4. For details of internal interrupt vectors, see section 6.5, 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.

In addition to the $\overline{\text{RES}}$ pin, it is also possible to establish the reset state by two operations internal circuit. One of them is to use an overflow in the watchdog timer. The other is to u external interrupt during deep software standby mode. For details, see section 4, Resets, s 15, Watchdog Timer (WDT), and section 24, Power-Down Modes.

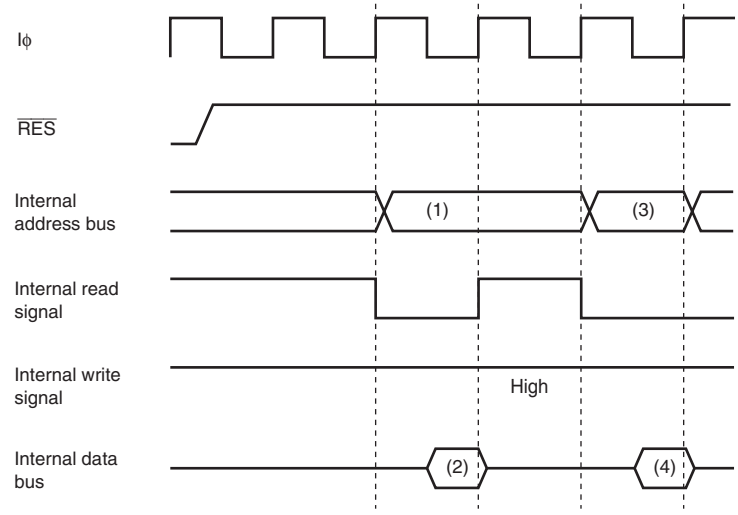
A reset initializes the internal state of the CPU and the registers of the on-chip peripheral The interrupt control mode is 0 immediately after a reset. However, there are registers tha be initialized by issuing an internal reset based on the watchdog timer or by issuing an int reset based on the external interrupt during deep software standby mode. For details, see Resets, section 15, Watchdog Timer (WDT), and section 24, Power-Down Modes.

has been issued based on the watchdog timer or the external interrupt during deep so
standby mode. For details, see section 4, Resets, section 15, Watchdog Timer (WDT
section 24, Power-Down Modes.

Figures 5.1 and 5.2 show examples of the reset sequence.

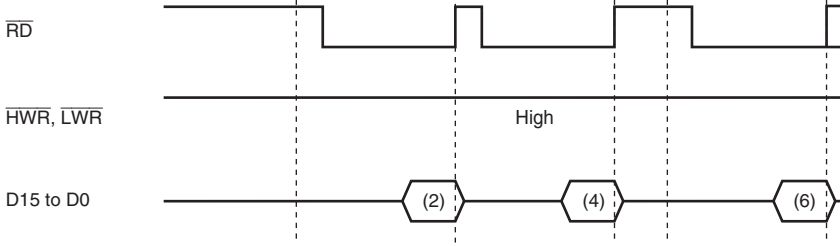
5.3.2 Interrupts after Reset

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



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

Figure 5.1 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 5.2 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 is carried out after execution of the RTE instruction.

Interrupts are accepted even within the trace exception handling routine.

Table 5.4 States 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	Li
Instruction fetch	CPU	Fetches instructions from even addresses	No
		Fetches instructions from odd addresses	Oc
		Fetches instructions from areas other than on-chip peripheral module space*1	No
		Fetches instructions from on-chip peripheral module space*1	Oc
		Fetches instructions from external memory space in single-chip mode	Oc
		Fetches instructions from access prohibited area.*2	Oc
Stack operation	CPU	Accesses stack when the stack pointer value is even address	No
		Accesses stack when the stack pointer value is odd	Oc
Data read/write	CPU	Accesses word data from even addresses	No
		Accesses word data from odd addresses	No
		Accesses external memory space in single-chip mode	Oc
		Accesses to access prohibited area*2	Oc
Data read/write	DTC or DMAC	Accesses word data from even addresses	No
		Accesses word data from odd addresses	No
		Accesses external memory space in single-chip mode	Oc
		Accesses to access prohibited area*2	Oc
Single address transfer	DMAC	Address access space is the external memory space for single address transfer	No
		Address access space is not the external memory space for single address transfer	Oc

Notes: 1. For on-chip peripheral module space, see section 8, Bus Controller (BSC).
2. For the access-prohibited area, see figure 3.1, Address Map (Advanced Mod section 3.4, Address Map.

program execution starts from that address.

Even though an address error occurs during a transition to an address error exception 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 and DMAC.

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

Table 5.6 shows the state of CCR and EXR after execution of the address error exception handling.

Table 5.6 States of CCR and EXR after Address Error Exception Handling

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

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

NMI	NMI pin (external input)	1
UBC break interrupt	User break controller (UBC)	1
IRQ0 to IRQ15	Pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$ (external input)	16
On-chip peripheral module	DMA controller (DMAC)	4
	Watchdog timer (WDT)	1
	A/D converter	1
	16-bit timer pulse unit (TPU)	26
	8-bit timer (TMR)	24
	Serial communications interface (SCI)	20
	IIC bus interface 2 (IIC2)	2
	ΔΣ A/D converter	1

Different vector numbers and vector table offsets are assigned to different interrupt sour
vector number and vector table offset, see table 6.2, Interrupt Sources, Vector Address C
and Interrupt Priority in section 6, Interrupt Controller.

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

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

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

Table 5.8 shows the state of CCR and EXR after execution of trap instruction exception

Table 5.8 States of CCR and EXR after Trap Instruction Exception Handling

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

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains the previous value.

the start address of the exception service routine is loaded from the vector table to PC
program execution starts from that address.

After execution of a sleep instruction, a bus master other than the CPU may have bus mas
In this case, the exception handling starts at the point when the CPU gets bus mastership
operation of the other bus master has ended.

Table 5.9 shows the state of CCR and EXR after execution of illegal instruction exceptio
handling. See section 24.10, Sleep Instruction Exception Handling, for details.

Table 5.9 States of CCR and EXR after Sleep Instruction Exception Handling

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

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains the previous value.

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

Table 5.10 shows the state of CCR and EXR after execution of illegal instruction except handling.

Table 5.10 States of CCR and EXR after Illegal Instruction Exception Handling

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

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

Interrupt control mode 0 | Interrupt control mode 2

Note: * Ignored on return.

Figure 5.3 Stack Status after Exception Handling

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

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

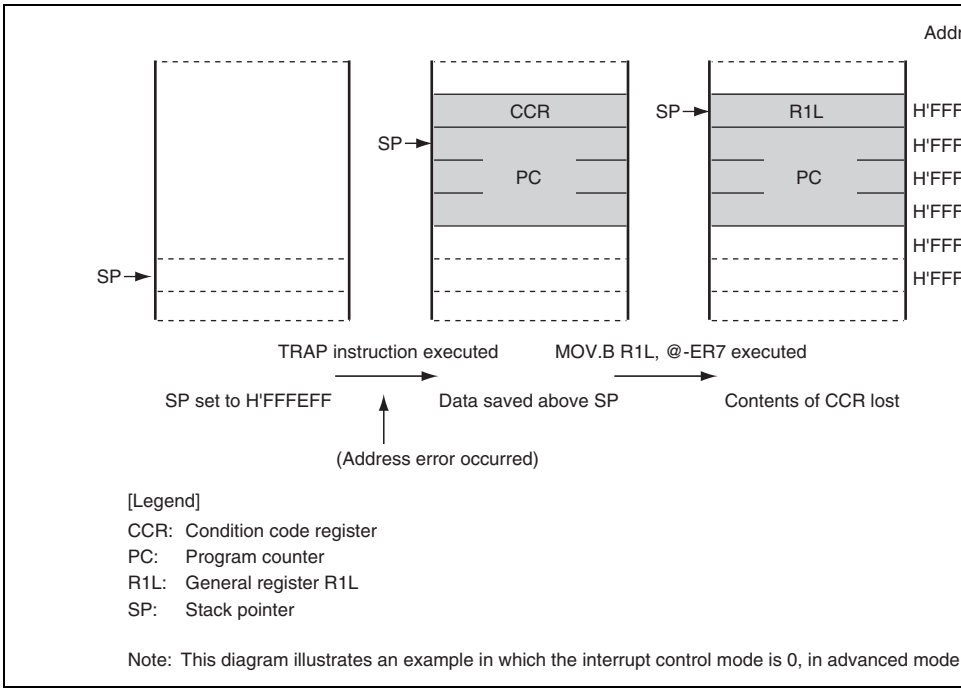


Figure 5.4 Operation when SP Value is Odd

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

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

- Independent vector addresses

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

- Seventeen external interrupts

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or fall
  detection can be selected for NMI. Falling edge, rising edge, or both edge detection,
  sensing, can be selected for $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$.

- DTC and DMAC control

  DTC and DMAC can be activated by means of interrupts.

- CPU priority control function

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

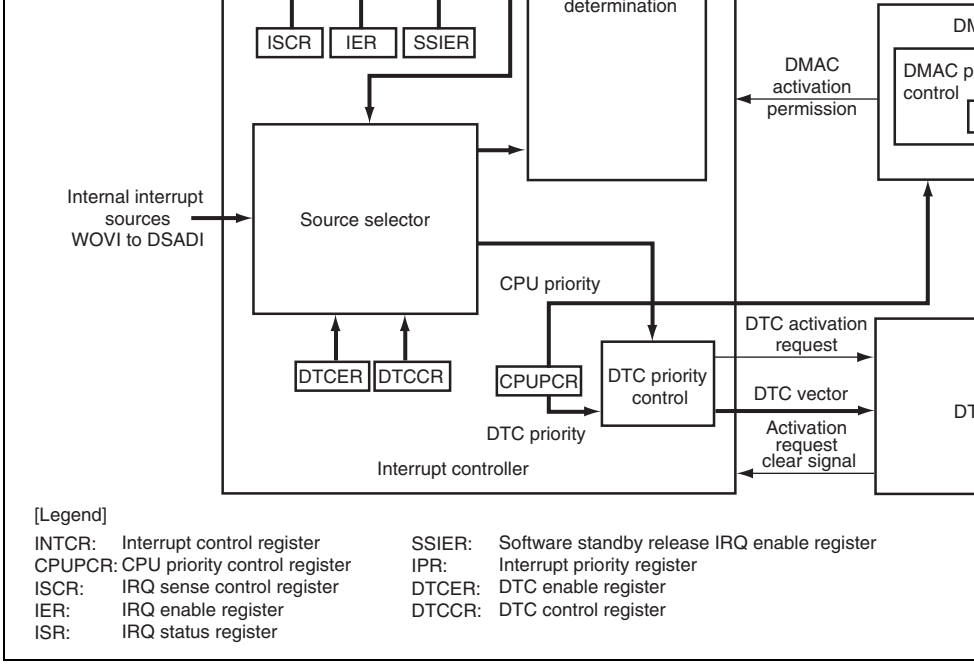


Figure 6.1 Block Diagram of Interrupt Controller

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

6.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt control register (INTCR)
- CPU priority control register (CPUPCR)
- Interrupt priority registers A to I, K, L, P to R (IPRA to IPRI, IPRK, IPRL, IPRP to I
- 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 I 1: Interrupt request generated at rising edge of N
2 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modified

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

Note: * When the IPSETE bit is set to 1, the CPU priority is automatically updated, so these bits cannot be 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. 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)
3	IPSETE	0	R/W	Interrupt Priority Set Enable Controls the function which automatically assig interrupt priority level of the CPU. Setting this b automatically sets bits CPUP2 to CPUP0 by the interrupt mask bit (I bit in CCR or bits I2 to I0 in 0: Bits CPUP2 to CPUP0 are not updated auto 1: The interrupt mask bit value is reflected in bi to CPUP0

011: Priority level 3
100: Priority level 4
101: Priority level 5
110: Priority level 6
111: Priority level 7 (highest)

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

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	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	IPR14	1	R/W	Sets the priority level of the corresponding inter
13	IPR13	1	R/W	source.
12	IPR12	1	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)
11	—	0	R	Reserved This is a read-only bit and cannot be modified.

				110: Priority level 6 111: Priority level 7 (highest)
7	—	0	R	Reserved This is a read-only bit and cannot be modified.
6	IPR6	1	R/W	Sets the priority level of the corresponding interr
5	IPR5	1	R/W	source.
4	IPR4	1	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)
3	—	0	R	Reserved This is a read-only bit and cannot be modified.
2	IPR2	1	R/W	Sets the priority level of the corresponding interr
1	IPR1	1	R/W	source.
0	IPR0	1	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)

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

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15E	0	R/W	IRQ15 Enable The IRQ15 interrupt request is enabled when t
14	IRQ14E	0	R/W	IRQ14 Enable The IRQ14 interrupt request is enabled when t
13	IRQ13E	0	R/W	IRQ13 Enable The IRQ13 interrupt request is enabled when t
12	IRQ12E	0	R/W	IRQ12 Enable The IRQ12 interrupt request is enabled when t
11	IRQ11E	0	R/W	IRQ11 Enable The IRQ11 interrupt request is enabled when t
10	IRQ10E	0	R/W	IRQ10 Enable The IRQ10 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
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when th

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: * The bits of this register cannot set the IRQ interrupt requests to exit from deep standby mode. For details, see section 24.2.6, Deep Standby Interrupt Enable (DPSIER).

software standby mode. For details, see section 24.2.8, Deep Standby Interrupt Edge Re
(DPSIEGR).

- ISCRH

Bit	15	14	13	12	11	10	9	
Bit Name	IRQ15SR	IRQ15SF	IRQ14SR	IRQ14SF	IRQ13SR	IRQ13SF	IRQ12SR	
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	

13	IRQ14SR	0	R/W	IRQ14 Sense Control Rise
12	IRQ14SF	0	R/W	IRQ14 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 11: Interrupt request generated at both falling a edges of $\overline{\text{IRQ14}}$
11	IRQ13SR	0	R/W	IRQ13 Sense Control Rise
10	IRQ13SF	0	R/W	IRQ13 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 11: Interrupt request generated at both falling a edges of $\overline{\text{IRQ13}}$
9	IRQ12SR	0	R/W	IRQ12 Sense Control Rise
8	IRQ12SF	0	R/W	IRQ12 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 11: Interrupt request generated at both falling a edges of $\overline{\text{IRQ12}}$

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

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 $\overline{IR}$
				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 $\overline{IRQ6}$
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 $\overline{IR}$
				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 $\overline{IRQ5}$
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 $\overline{IR}$
				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 $\overline{IRQ4}$

4	IRQ2SF	0	R/W	IRQ2 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 $\overline{IRQ2}$
3	IRQ1SR	0	R/W	IRQ1 Sense Control Rise*
2	IRQ1SF	0	R/W	IRQ1 Sense Control Fall*
				00: Interrupt request generated by low level of $\overline{I}$
				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 $\overline{IRQ0}$

Note: * The bits of this register cannot set the edge selections for IRQ interrupt reque from deep software standby mode. For details, see section 24.2.8, Deep Stan Interrupt Edge Register (DPSIEGR).

Bit	7	6	5	4	3	2	1	
Bit Name	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	I
Initial Value	0	0	0	0	0	0	0	
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R

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

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15F	0	R/(W)*[1]	[Setting condition]
14	IRQ14F	0	R/(W)*[1]	• When the interrupt selected by ISCR occur
13	IRQ13F	0	R/(W)*[1]	[Clearing conditions]
12	IRQ12F	0	R/(W)*[1]	• Writing 0 after reading IRQnF = 1 (n = 11 to
11	IRQ11F	0	R/(W)*[1]	• When interrupt exception handling is execu low-level sensing is selected and IRQn inpu
10	IRQ10F	0	R/(W)*[1]	
9	IRQ9F	0	R/(W)*[1]	• When IRQn interrupt exception handling is
8	IRQ8F	0	R/(W)*[1]	while falling-, rising-, or both-edge sensing
7	IRQ7F	0	R/(W)*[1]	selected
6	IRQ6F	0	R/(W)*[1]	• When the DTC is activated by an IRQn inte
5	IRQ5F	0	R/(W)*[1]	and the DISEL bit in MRB of the DTC is cle
4	IRQ4F	0	R/(W)*[1]	(n = 15 to 0)
3	IRQ3F*[2]	0	R/(W)*[1]	
2	IRQ2F*[2]	0	R/(W)*[1]	
1	IRQ1F*[2]	0	R/(W)*[1]	
0	IRQ0F*[2]	0	R/(W)*[1]	

Notes: 1. Only 0 can be written, to clear the flag.
2. The bits of this register cannot set the IRQ interrupt request flags, IRQnF (n = exit from deep software standby mode. For details, see section 24.2.7, Deep S Interrupt Flag Register (DPSIFR).

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	SSI15	0	R/W	Software Standby Release IRQ Setting
14	SSI14	0	R/W	These bits select the IRQn interrupt used to le software standby mode (n = 15 to 0).
13	SSI13	0	R/W	
12	SSI12	0	R/W	0: An IRQn request is not sampled in software mode
11	SSI11	0	R/W	
10	SSI10	0	R/W	1: When an IRQn request occurs in software mode, this LSI leaves software standby mo the oscillation settling time has elapsed
9	SSI9	0	R/W	
8	SSI8	0	R/W	
7	SSI7	0	R/W	
6	SSI6	0	R/W	
5	SSI5	0	R/W	
4	SSI4	0	R/W	
3	SSI3	0	R/W	
2	SSI2	0	R/W	
1	SSI1	0	R/W	
0	SSI0	0	R/W	

(1) NMI Interrupts

Nonmaskable interrupt request (NMI) is the highest-priority interrupt, and is always acce
the CPU regardless of the interrupt control mode or the settings of the CPU interrupt mas
The NMIEG bit in INTCR selects whether an interrupt is requested at the rising or falling
the NMI pin.

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

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

(2) IRQn Interrupts

An IRQn interrupt is requested by a signal input on pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$. $\overline{\text{IRQn}}$ (n = 15 to
the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, fa
  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 0
  software. The bit manipulation instructions and memory operation instructions should
  to clear the flag.

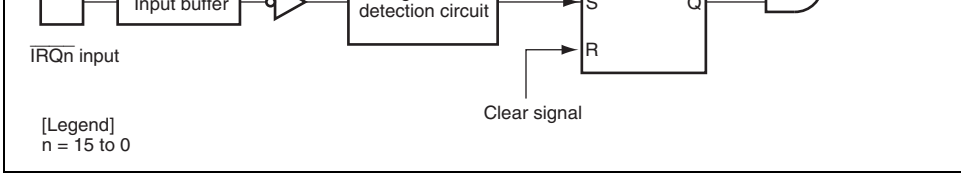


Figure 6.2 Block Diagram of Interrupts IRQn

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 si to high in the interrupt handling routine and clear the IRQnF to 0. Interrupts may not be when the corresponding input signal $\overline{\text{IRQn}}$ is set to high before the interrupt handling be

6.4.2 Internal Interrupts

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

- For each on-chip peripheral module there are flags that indicate the interrupt request and enable bits that enable or disable these interrupts. They can be controlled indepe When the enable bit is set to 1, an interrupt request is issued to the interrupt controlle
- 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 priority control functions.

Classification	Interrupt Source	Vector Number	Vector Address Offset*	IPR	Priority	DTC Activation	D A
External pin	NMI	7	H'001C	—	High	—	–
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	–
	IRQ12	76	H'0130	IPRD14 to IPRD12		O	–
	IRQ13	77	H'0134	IPRD10 to IPRD8		O	–
	IRQ14	78	H'0138	IPRD6 to IPRD4		O	–
	IRQ15	79	H'013C	IPRD2 to IPRD0		O	–
—	Reserved for system use	80	H'0140	—		—	–
WDT	WOVI	81	H'0144	IPRE10 to IPRE8	Low	—	–

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

	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	–
	Reserved for system use	130	H'0208	—		—	–
		131	H'020C	—		—	–
—	Reserved for system use	132	H'0210	—		—	–
		133	H'0214			—	–
		134	H'0218			—	–
		135	H'021C			—	–
DMAC	DMEEND0	136	H'0220	IPRK14 to IPRK12		O	–
	DMEEND1	137	H'0224			O	–
	Reserved for system use	138	H'0228			—	–
		139	H'022C		Low	—	–

	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			—
SCI_3	ERI3	156	H'0270	IPRL10 to IPRL8		—
	RXI3	157	H'0274			O
	TXI3	158	H'0278			O
	TEI3	159	H'027C			—
SCI_4	ERI4	160	H'0280	IPRL6 to IPRL4		—
	RXI4	161	H'0284			O
	TXI4	162	H'0288			O
	TEI4	163	H'028C			—
—	Reserved for system use	164 | 199	H'0290 | H'031C	—		— | —
TMR_4	CMI4A	200	H'0320	IPRP10 to IPRP8		O
	CMI4B	201	H'0324			O
	OV4I	202	H'0328		Low	—

TMR_7	CMI7A	209	H'0344	IPRQ14 to IPRQ12	O	–
	CMI7B	210	H'0348		O	–
	OV7I	211	H'034C		—	–
—	Reserved for system use	212	H'0350	—	—	–
		|	|		|	|
		215	H'035C		—	–
IIC2	IICI0	216	H'0360	IPRQ6 to IPRQ4	—	–
	Reserved for system use	217	H'0364		—	–
	IICI1	218	H'0368		—	–
	Reserved for system use	219	H'036C		—	–
A/D	ADI0	220	H'0370	IPRQ2 to IPRQ0	—	C
	Reserved for system use	221	H'0374		—	–
		222	H'0378		—	–
		223	H'037C		—	–
ΔΣ A/D	DSADI	224	H'0380	IPRR14 to IPRR12	—	C
	Reserved for system use	225	H'0384		—	–
		226	H'0388		—	–
		227	H'038C		—	–
—	Reserved for system use	228	H'0390	—	—	–
		|	|		|	|
		255	H'03FC		—	–

Low

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

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.

6.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI are masked by the I bit in the CPU. Figure 6.3 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.

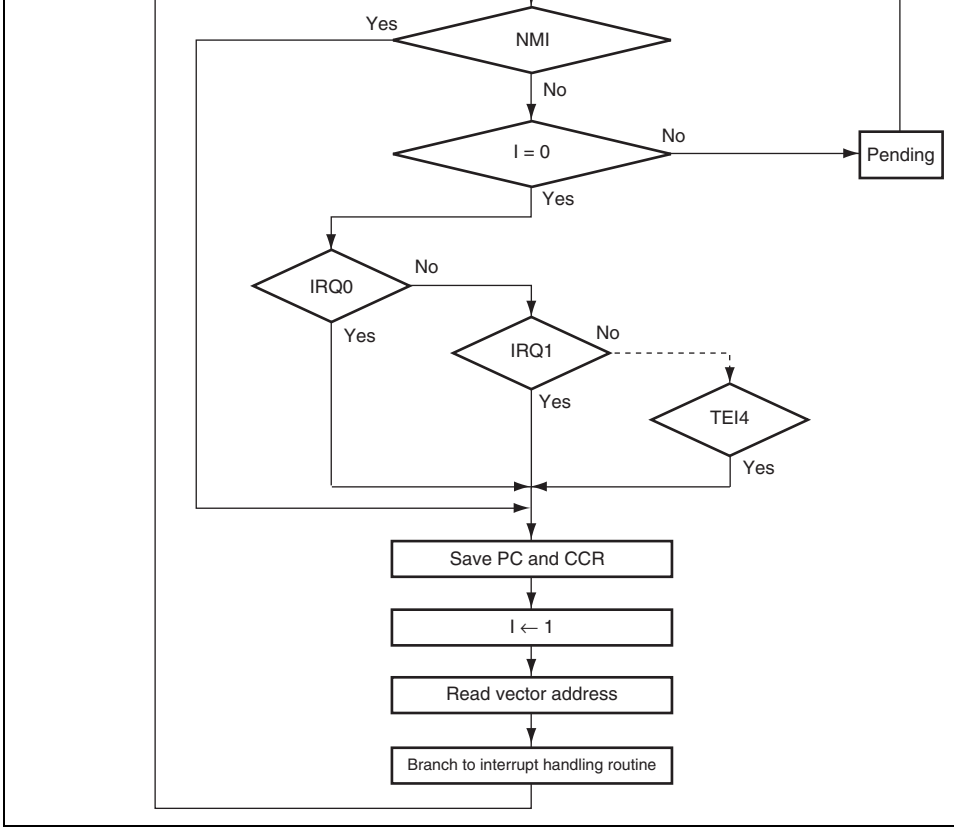


Figure 6.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

multiple interrupt requests have the same priority, an interrupt request is selected acc
the default setting shown in table 6.2.

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.

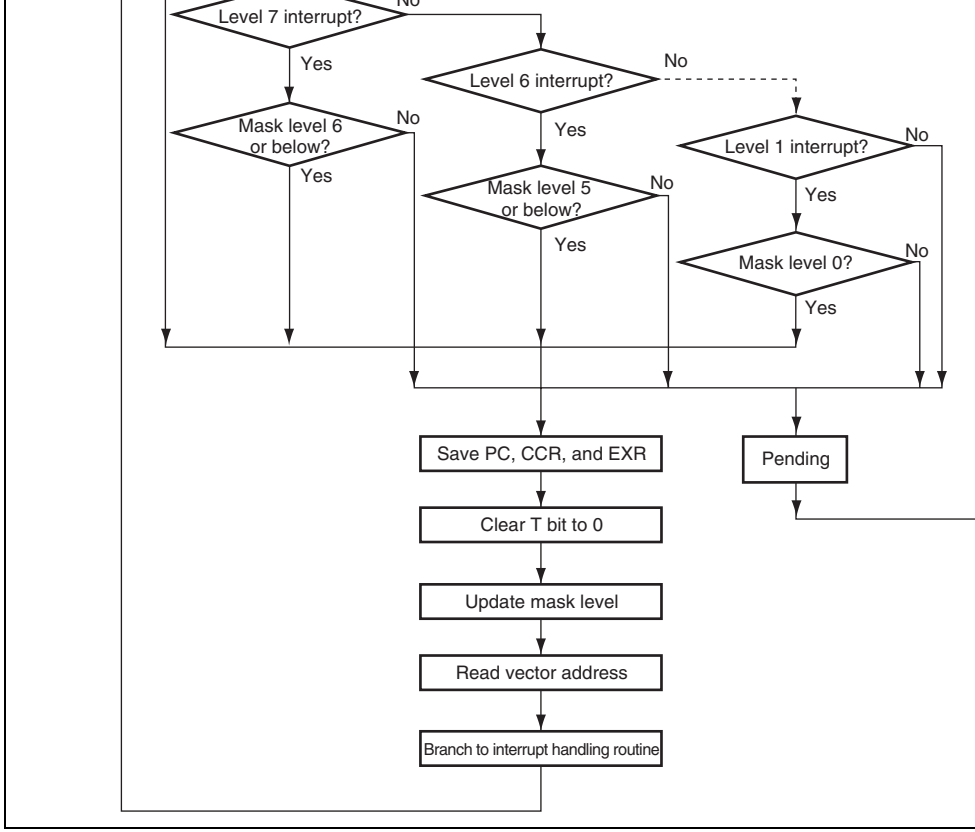


Figure 6.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

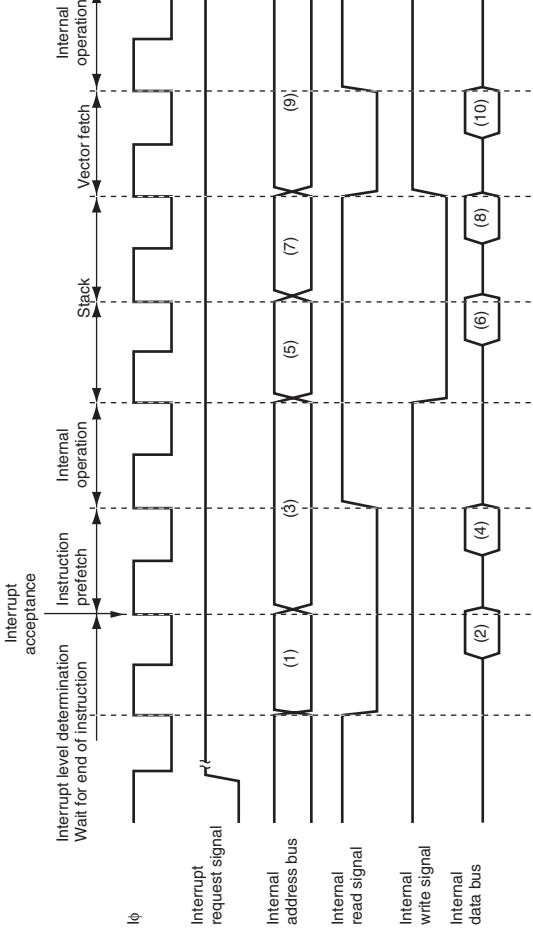


Figure 6.5 Interrupt Exception Handling

Execution State	Normal Mode*		Advanced Mode		Maximum
	Interrupt Control Mode 0	Interrupt Control Mode 2	Interrupt Control Mode 0	Interrupt Control Mode 2	Interrupt Control Mode 0
Interrupt 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.

6.6.5 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 9, DMA Controller (DMAC), and section 10, Data Transfer Controller (DTC).

Figure 6.6 shows a block diagram of the DTC, DMAC, and interrupt controller.

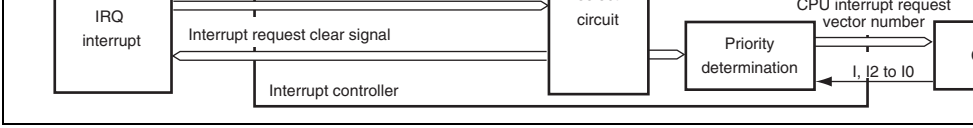


Figure 6.6 Block Diagram of DTC, DMAC, and Interrupt Controller

(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 DTCERG 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 IPSET
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 6.6 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 DTCERG of the DTC,
DISEL bit in MRB of the DTC.

Table 6.6 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 6.6, 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.

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 and DMAC activation sources are co 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 DM 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.

There are two methods for assigning the priority level to the CPU by the IPSETE bit in C Setting the IPSETE bit to 1 enables a function to automatically assign the value of the int mask bit of the CPU to the CPU priority level. Clearing the IPSETE bit to 0 disables the f to automatically assign the priority level. Therefore, the priority level is assigned directly software rewriting bits CPUP2 to CPUP0. Even if the IPSETE bit is 1, the priority level o CPU is software assignable by rewriting the interrupt mask bit of the CPU (I bit in CCR o bits in EXR).

Interrupt Control Mode	Interrupt Priority	Interrupt Mask Bit	IPSETE in CPUPCR	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

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

be executed on completion of the instruction. However, if there is an interrupt request w
over that interrupt, interrupt exception handling will be executed for the interrupt with p
and another interrupt will be ignored. The same also applies when an interrupt source fla
cleared to 0. Figure 6.7 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.

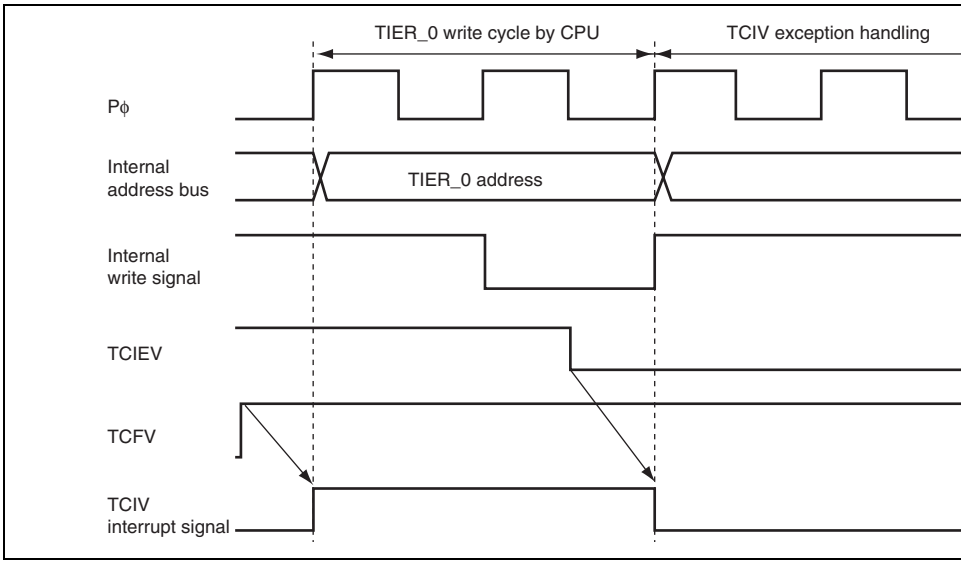


Figure 6.7 Conflict between Interrupt Generation and Disabling

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.

6.8.4 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.

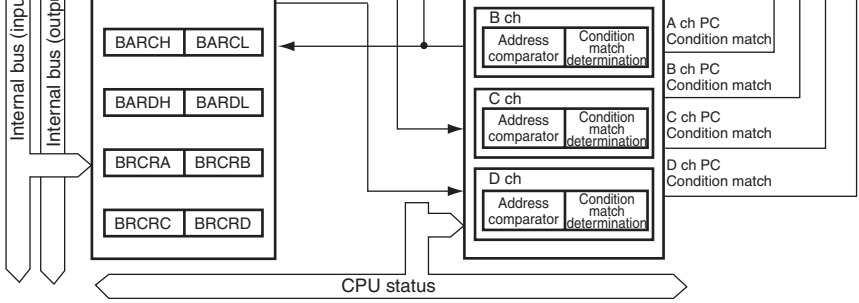
```
L1:    EEPMOV.W
       MOV.W  R4,R4
       BNE    L1
```

6.8.5 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.

7.1 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))
- After a break condition is satisfied, UBC break interrupt exception handling is execu
  immediately before execution of the instruction fetched from the specified address (I
- Module stop state specifiable



[Legend]

BARAH, BARAL:	Break address register A
BARBH, BARBL:	Break address register B
BARCH, BARCL:	Break address register C
BARDH, BARDL:	Break address register D
BRCRA:	Break control register A
BRCRB:	Break control register B
BRCRC:	Break control register C
BRCRD:	Break control register D

Figure 7.1 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]

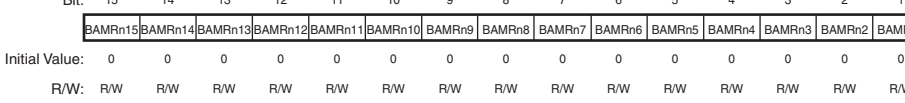
n = Channels A to D

- 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



- 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
9	CPn0	0	R/W	condition for the given channel. 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
6	—	0	R/W	should always be 0.

				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

7.4.2 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

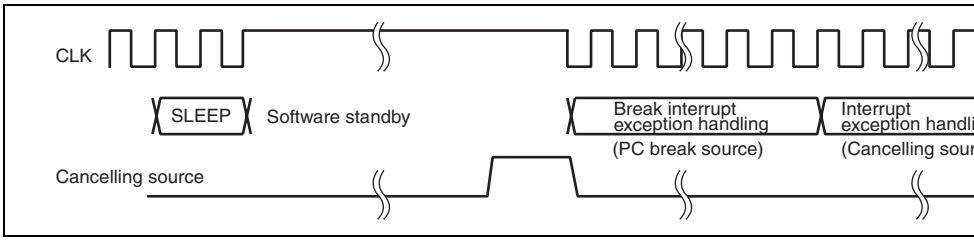


Figure 7.2 Contention between SLEEP Instruction (Software Standby) and PC

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 7.3 Flag Bit Clearing Sequence (Condition Match Flag)

4. The valid range of break addresses in the MCU and CPU address modes is given in t
   — MCU operating mode/CPU mode and valid address range
   In setting break addresses, MCU address mode and CPU mode need to be taken i
   account as shown below.

   The mask must be set in the address mask register.

Table 7.3 Valid Range of Break/Branch Addresses for MCU/CPU Address Mod

	Advanced Mode	
	256 MB	**16 MB**
PC break address	The lower 24 bits are valid and the upper 8 bits are maske	

5. If an illegal instruction is executed after setting break conditions for the UBC, an unc
   UBC break interrupt may occur depending on the value of the program counter and t
   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, DTC, a external bus master
- Multi-clock function

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

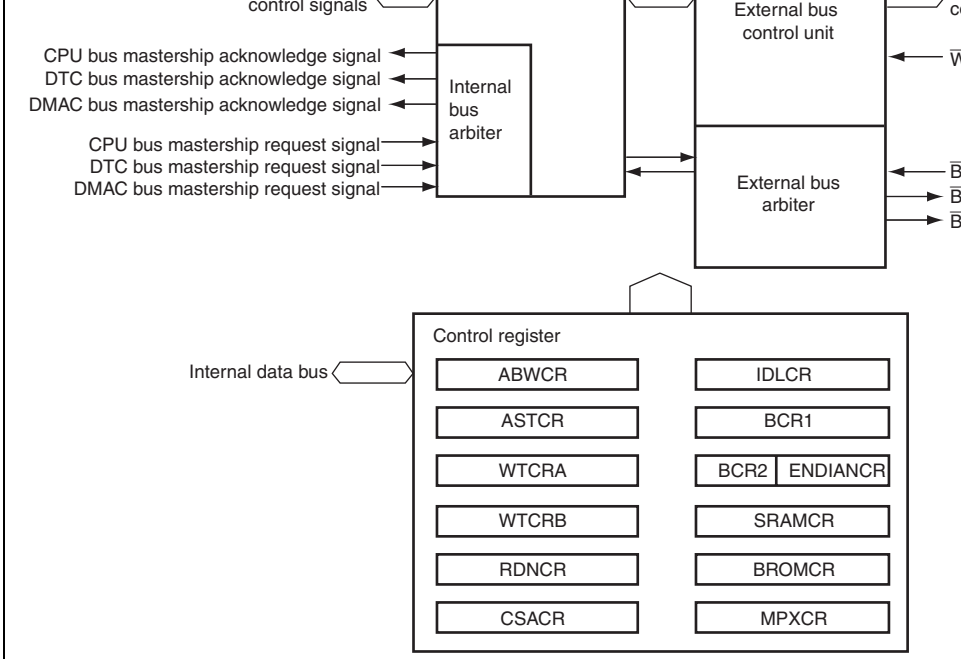


Figure 8.1 Block Diagram of Bus Controller

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

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

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

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

[Legend]

×: Don't care

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

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

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

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

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

• WTCRB

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

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

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

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

				001: 1 program wait cycle inserted
				010: 2 program wait cycles inserted
				011: 3 program wait cycles inserted
				100: 4 program wait cycles inserted
				101: 5 program wait cycles inserted
				110: 6 program wait cycles inserted
				111: 7 program wait cycles inserted
11	—	0	R	Reserved
				This is a read-only bit and cannot be modified.
10	W22	1	R/W	Area 2 Wait Control 2 to 0
9	W21	1	R/W	These bits select the number of program wait c
8	W20	1	R/W	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	W02	1	R/W	Area 0 Wait Control 2 to 0
1	W01	1	R/W	These bits select the number of program wait
0	W00	1	R/W	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	
12	RDN4	0	R/W	As shown in figure 8.2, 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.
11	RDN3	0	R/W	
10	RDN2	0	R/W	
9	RDN1	0	R/W	
8	RDN0	0	R/W	0: In an area n read access, the $\overline{RD}$ signal is n at the end of the read cycle 1: In an area n read access, the $\overline{RD}$ signal is n one half-cycle before the end of the read cy (n = 7 to 0)
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 RI setting is ignored during CPU read accesses and the same operation when RD performed.

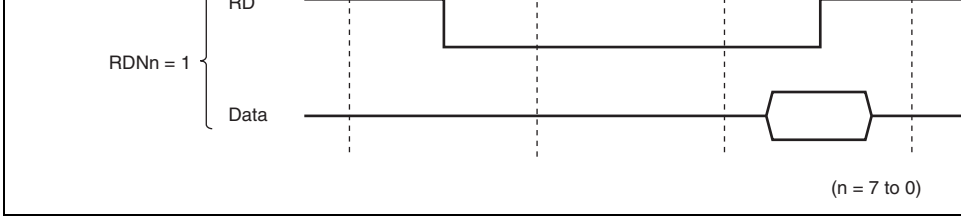


Figure 8.2 Read Strobe Negation Timing (Example of 3-State Access Spac

8.2.5 $\overline{CS}$ Assertion Period Control Registers (CSACR)

CSACR selects whether or not the assertion periods of the chip select signals ($\overline{CSn}$) 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 $\overline{CSn}$ and address signa the setup time and hold time of read strobe ($\overline{RD}$) and write strobe ($\overline{LHWR}/\overline{LLWR}$) 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 8.3). 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
2	CSXT2	0	R/W	after the normal access cycle.
1	CSXT1	0	R/W	0: In access to area n, the $\overline{CSn}$ and address as
0	CSXT0	0	R/W	period (Tt) is not extended 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 CPU read acce

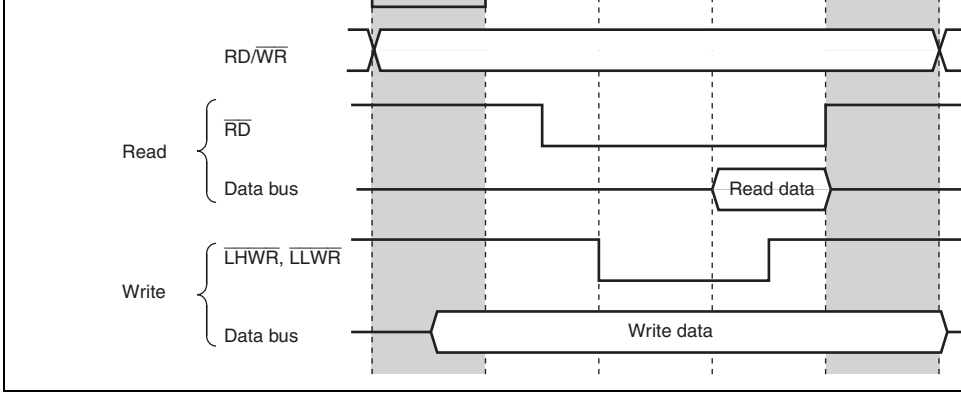


Figure 8.3 $\overline{CS}$ and Address Assertion Period Extension (Example of Basic Bus Interface, 3-State Access Space, and RDNn = 0)

Bit Name	IDLSEL7	IDLSEL6	IDLSEL5	IDLSEL4	IDLSEL3	IDLSEL2	IDLSEL1	ID
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 single address transfer (write cycle) is fo 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	IDLCA1	1	R/W	Idle Cycle State Number Select A
8	IDLCA0	1	R/W	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	IDLSEL7	0	R/W	Idle Cycle Number Select
6	IDLSEL6	0	R/W	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)
5	IDLSEL5	0	R/W	
4	IDLSEL4	0	R/W	
3	IDLSEL3	0	R/W	
2	IDLSEL2	0	R/W	
1	IDLSEL1	0	R/W	
0	IDLSEL0	0	R/W	

Bit Name	DKC	—	—	—	—	—	—	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
15	BRLE	0	R/W	External Bus Release Enable Enables/disables external bus release. 0: External bus release disabled $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ pins can be us ports 1: External bus release enabled* For details, see section 11, I/O Ports.
14	BREQOE	0	R/W	$\overline{\text{BREQO}}$ Pin Enable Controls outputting the bus request signal ($\overline{\text{BR}}$ 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 11, 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 ec 1: $\overline{\text{DACK}}$ signal is asserted at the Bϕ rising ed
6	—	0	R/W	Reserved This bit is always read as 0. The write value sl always be 0.
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 11, I/O Ports.

Bit	Bit Name	Value	R/W	Description
7, 6	—	All 0	R	Reserved These are read-only bits and cannot be modifie
5	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.
4	IBCCS	0	R/W	Internal Bus Cycle Control Select Selects the internal bus arbiter function. 0: Releases the bus mastership according to th 1: Executes the bus cycles alternatively when a 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

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 en
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
12	BCSEL4	0	R/W	BROMCR and MPXCR must be cleared to 0.
11	BCSEL3	0	R/W	0: Area n is basic bus interface
10	BCSEL2	0	R/W	1: Area n is byte control SRAM interface
9	BCSEL1	0	R/W	(n = 7 to 0)
8	BCSEL0	0	R/W	
7 to 0	—	All 0	R	Reserved
				These are read-only bits and cannot be modifie

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

Bit	Bit Name	Initial Value	R/W	Description
15	BSRM0	0	R/W	Area 0 Burst ROM Interface Select Specifies the area 0 bus interface. To set this 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, 2	—	All 0	R	Reserved These are read-only bits and cannot be modifie
1	BSWD11	0	R/W	Area 1 Burst Word Number Select
0	BSWD10	0	R/W	Selects the number of words in burst access to 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)

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
12	MPXE4	0	R/W	0.
11	MPXE3	0	R/W	0: Area n is specified as a basic interface or a control SRAM interface. 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

registers of peripheral modules such as SCI and timer.

- External access cycle

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

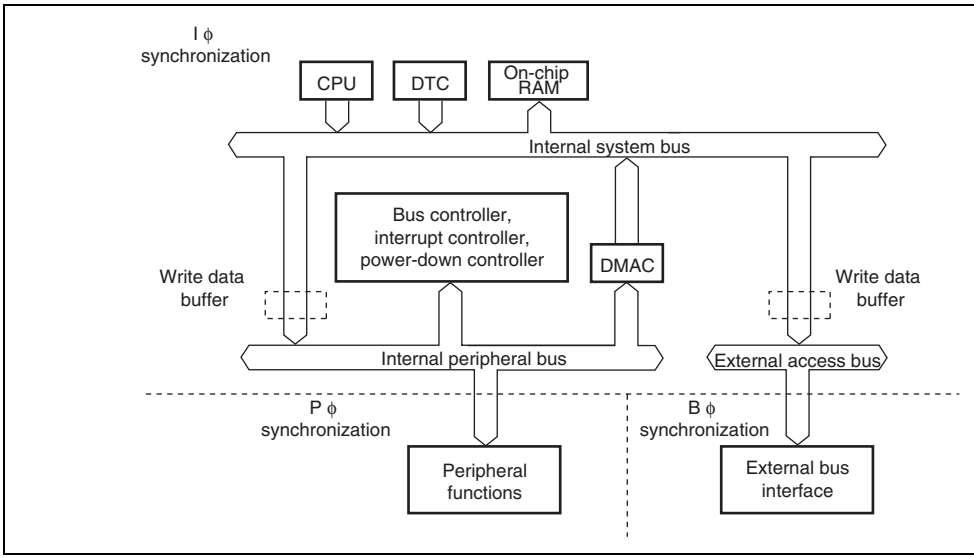


Figure 8.4 Internal Bus Configuration

	Bus controller CPU DTC DMAC Internal memory Clock pulse generator Power down control UBC
Pϕ	I/O ports TPU PPG TMR WDT SCI A/D D/A IIC2 ΔΣ A/D
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 23, 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 peripheral 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 8.5 shows the external 2-state access timing when the frequency rate of Iϕ and Bϕ
Figure 8.6 shows the external 3-state access timing when the frequency rate of Iϕ and Bϕ

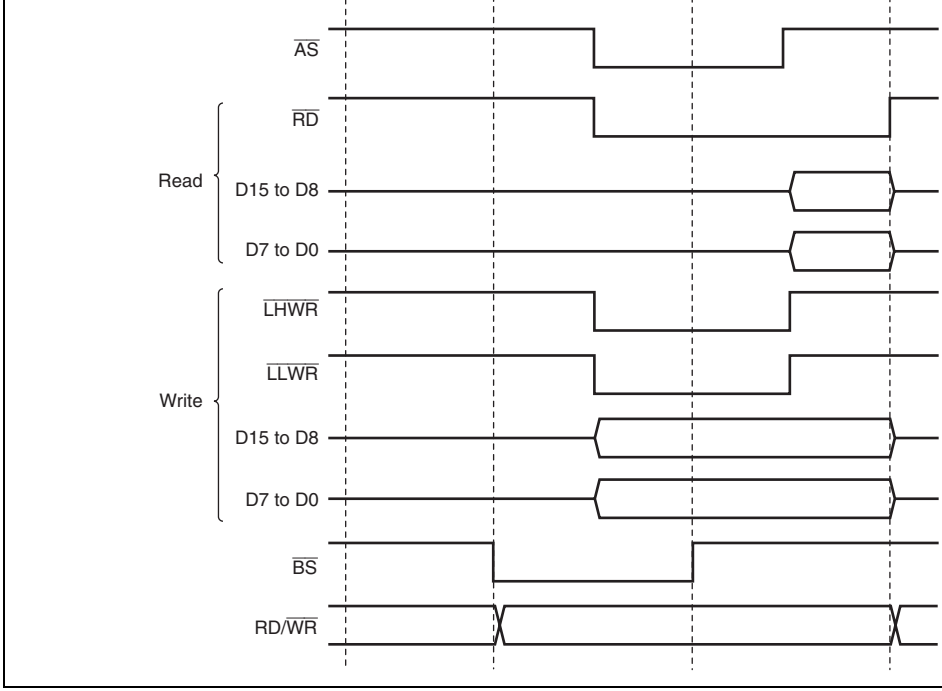


Figure 8.5 System Clock: External Bus Clock = 4:1, External 2-State Acce

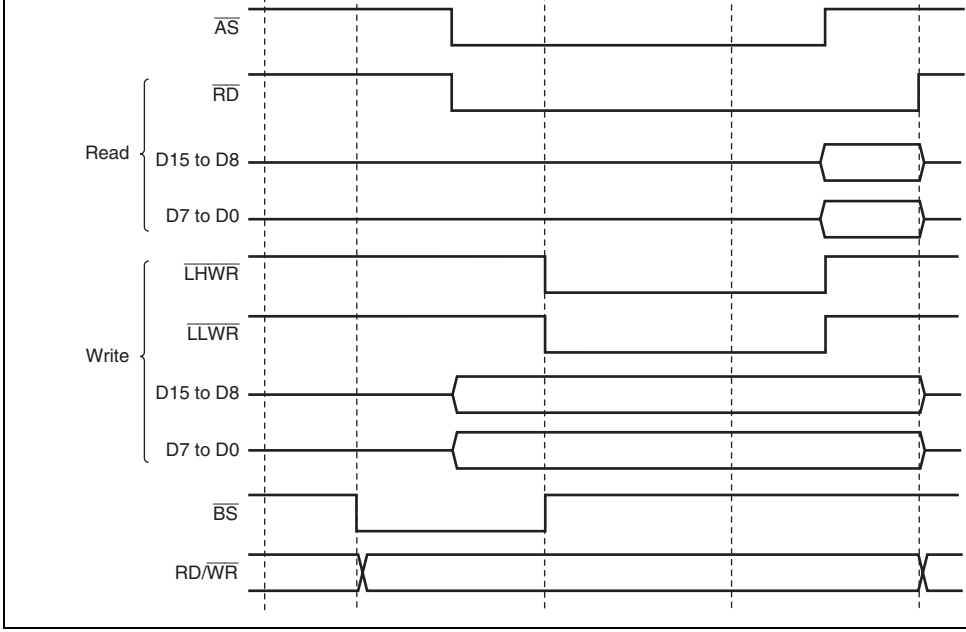


Figure 8.6 System Clock: External Bus Clock = 2:1, External 3-State Acces

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 up (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	$\overline{\text{WAIT}}$	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 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
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)

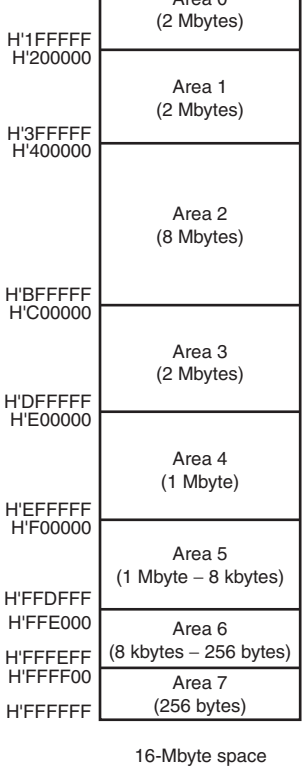


Figure 8.7 Address Space Area Division

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

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

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

Figure 8.9 shows the signal output timing when the $\overline{CS}$ signals to be output to areas 5 an
output to the same pin.

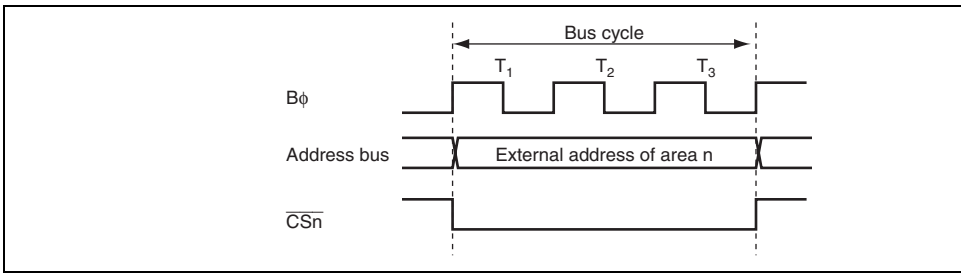


Figure 8.8 $\overline{CSn}$ Signal Output Timing (n = 0 to 7)

Figure 8.9 Timing When $\overline{CS}$ Signal is Output to the Same Pin

8.5.4 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 8.5 areas that can be specified for each interface. The initial state of each area is a basic bus i

Table 8.4 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 desig
16-bit access space, 16-bit bus mode is set.

(3) Endian Format

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

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

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

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

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

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

(b) Byte Control SRAM Interface

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

(c) Burst ROM Interface

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

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

Table 8.6 lists the number of access cycles for each interface.

Table 8.6 Number of Access Cycles

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

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

(5) Strobe Assert/Negate Timings

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

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

selected for area 0 by bit BSRM0 in BROMCR and bit BCSEL0 in SRAMCR. Table 8.7 the external interface of area 0.

Table 8.7 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

	Register Setting	
Interface	**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 8.9 shows the external interface of area 2.

Table 8.9 Area 2 External Interface

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

	Register Setting	
Interface	**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 $\overline{\text{CS4}}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexe interface can be selected for area 4 by bit MPXE4 in MPXCR and bit BCSEL4 in SRAM Table 8.11 shows the external interface of area 4.

Table 8.11 Area 4 External Interface

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

SRAMCR. Table 8.12 shows the external interface of area 5.

Table 8.12 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

	Register Setting	
Interface	**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 $\overline{CS7}$ 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 8.14 shows the external interface of area 7.

Table 8.14 Area 7 External Interface

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

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

Figures 8.10 and 8.11 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	
Word	n	2	1st	Byte	
			2nd	Byte	
Longword	n	4	1st	Byte	
			2nd	Byte	
			3rd	Byte	
			4th	Byte	

**Figure 8.10 Access Sizes and Data Alignment Control for 8-Bit Access Spa
(Big Endian)**

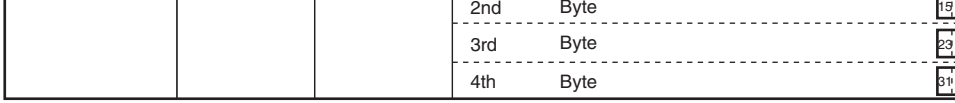


Figure 8.11 Access Sizes and Data Alignment Control for 8-Bit Access Spac (Little Endian)

(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 8.12 and 8.13 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 8 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	Strobe s LHWR/LUB 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 8.12 Access Sizes and Data Alignment Control for 16-Bit Access Sp (Big Endian)

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 8.13 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 8.5.6, Endian and Data Alignm

8.6.2 I/O Pins Used for Basic Bus Interface

Table 8.15 shows the pins used for basic bus interface.

Table 8.15 I/O Pins for Basic Bus Interface

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

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

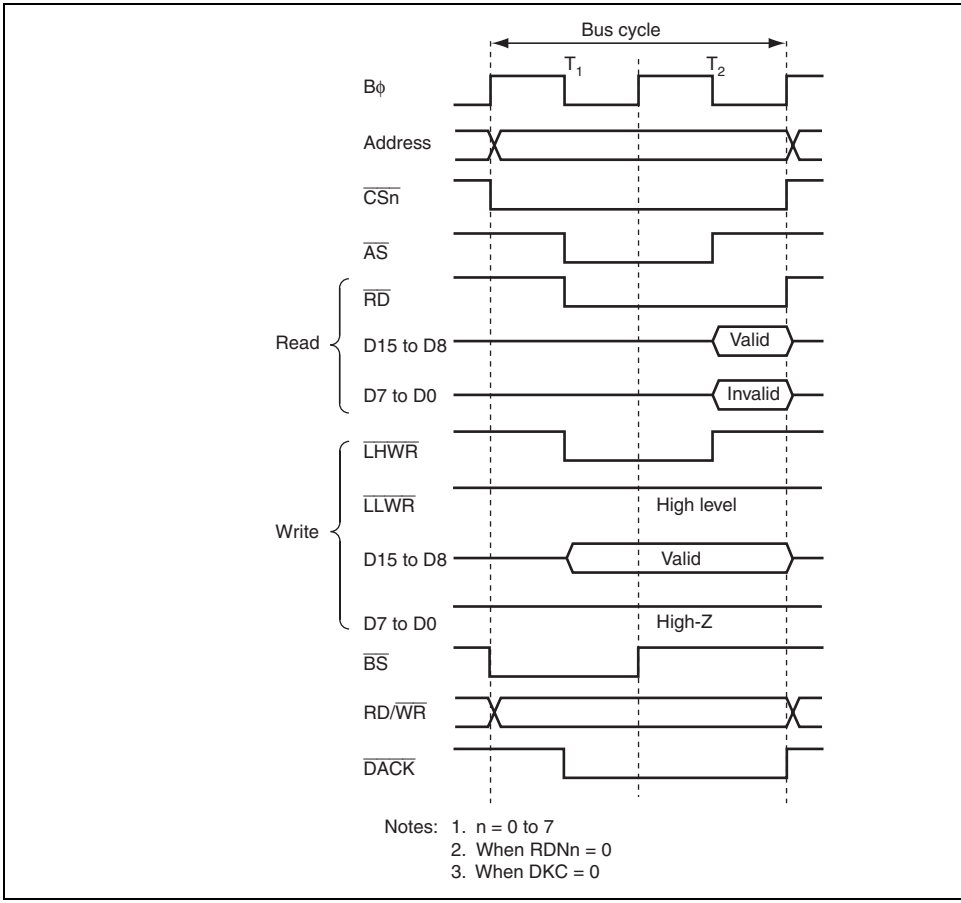
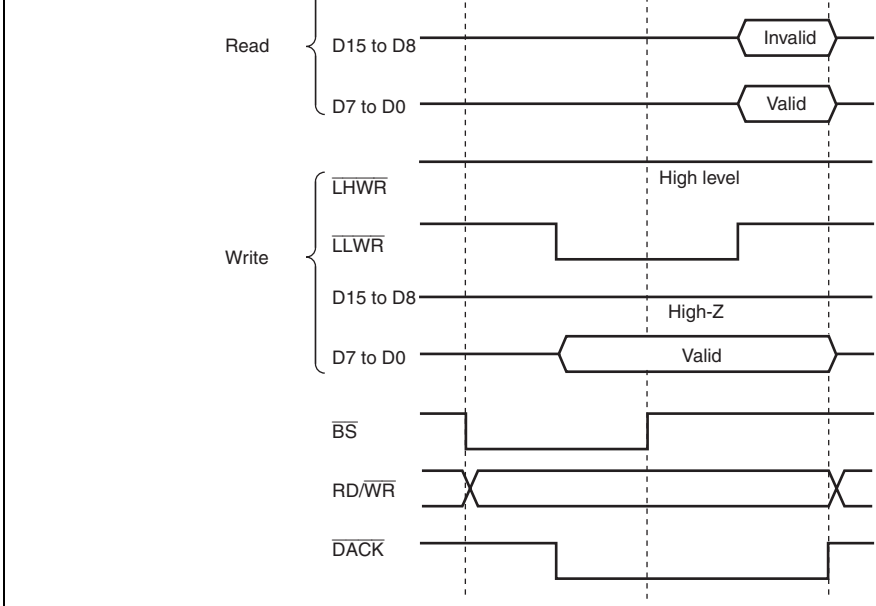
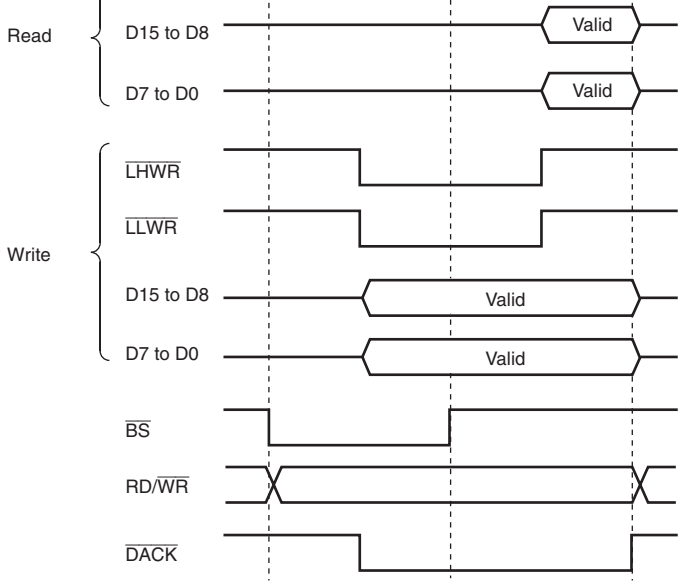


Figure 8.14 16-Bit 2-State Access Space Bus Timing (Byte Access for Even Ad



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

Figure 8.15 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 = 0

Figure 8.16 16-Bit 2-State Access Space Bus Timing (Word Access for Even Ad

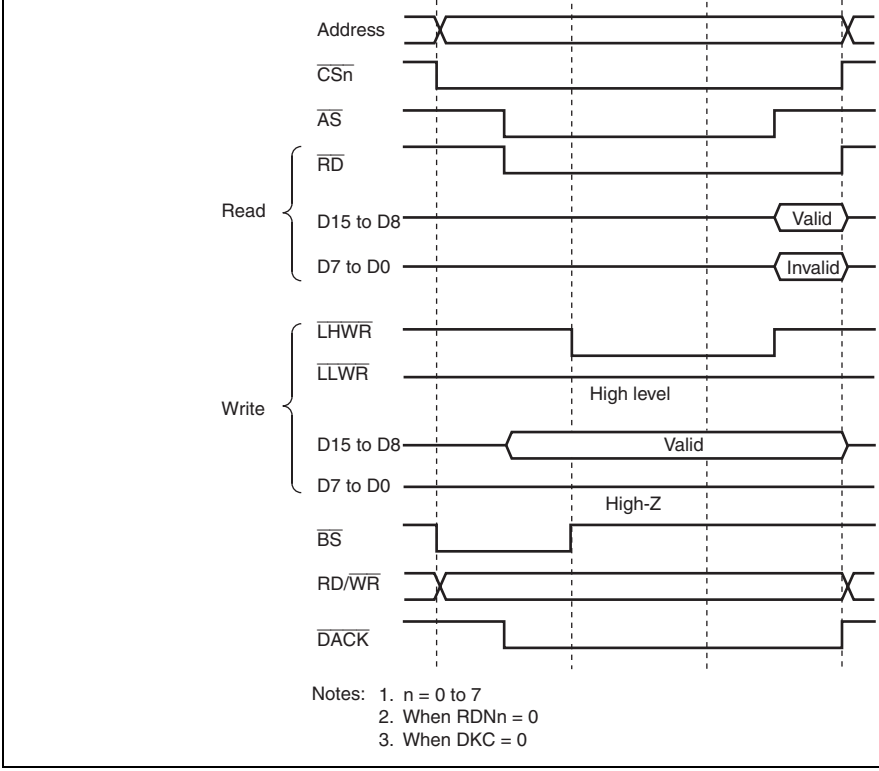
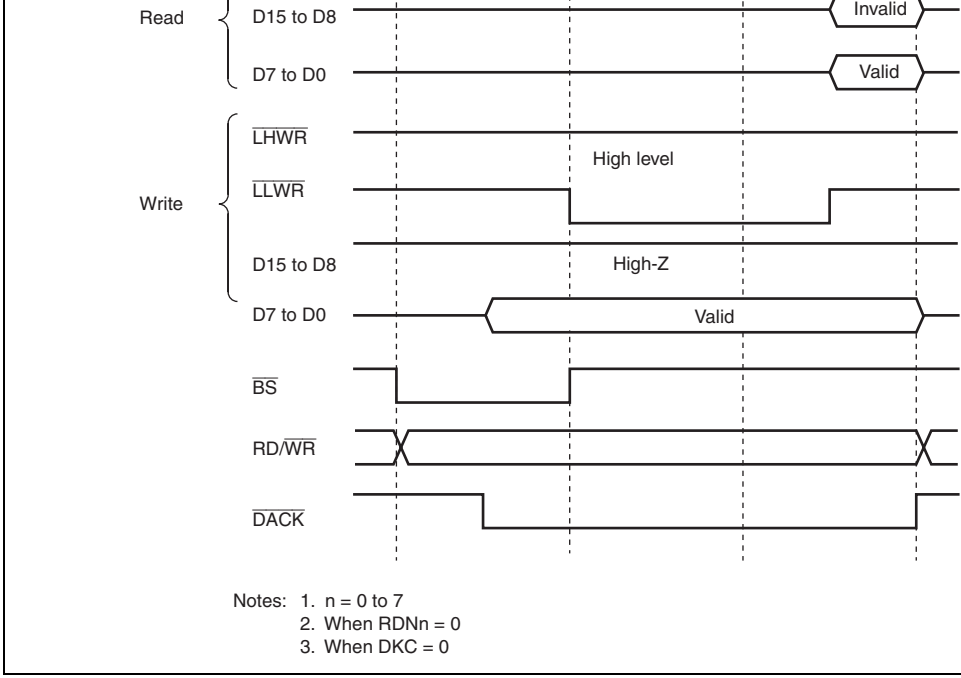


Figure 8.17 16-Bit 3-State Access Space Bus Timing (Byte Access for Even Add



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

Figure 8.18 16-Bit 3-State Access Space Bus Timing (Word Access for Odd Ac

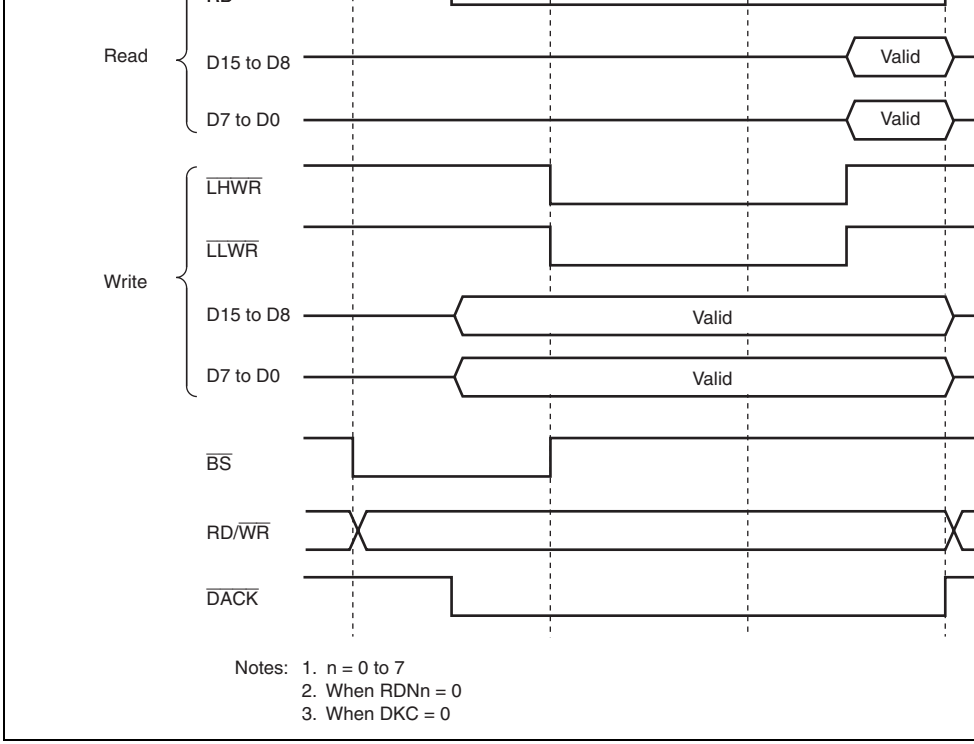
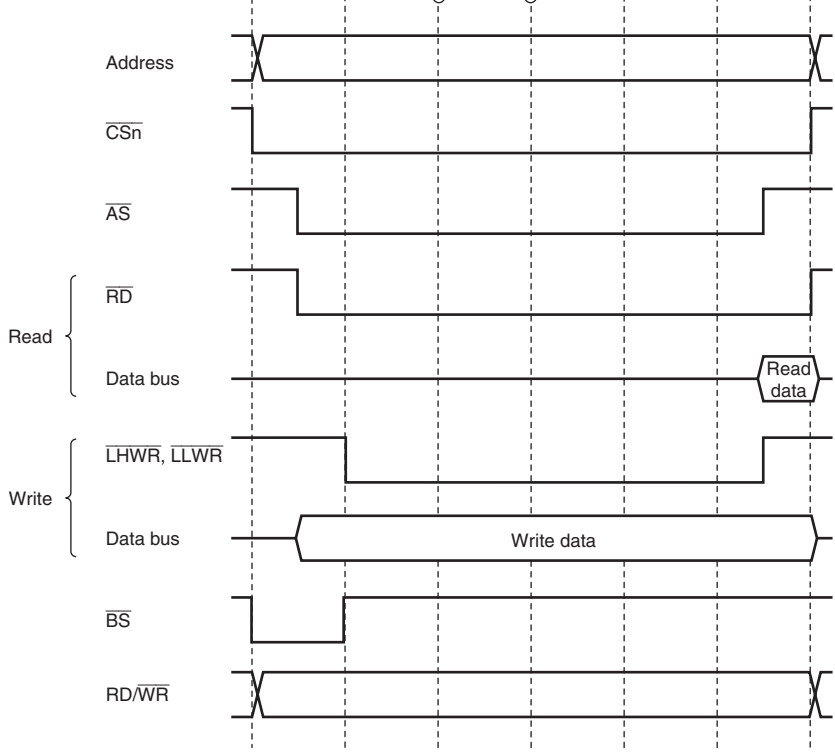


Figure 8.19 16-Bit 3-State Access Space Bus Timing (Word Access for Even Ad

(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 o
cycles to be inserted is changed according to the external devices. The WAITE bit is con
all areas. For details on ICR, see section 11, I/O Ports.



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

Figure 8.20 Example of Wait Cycle Insertion Timing

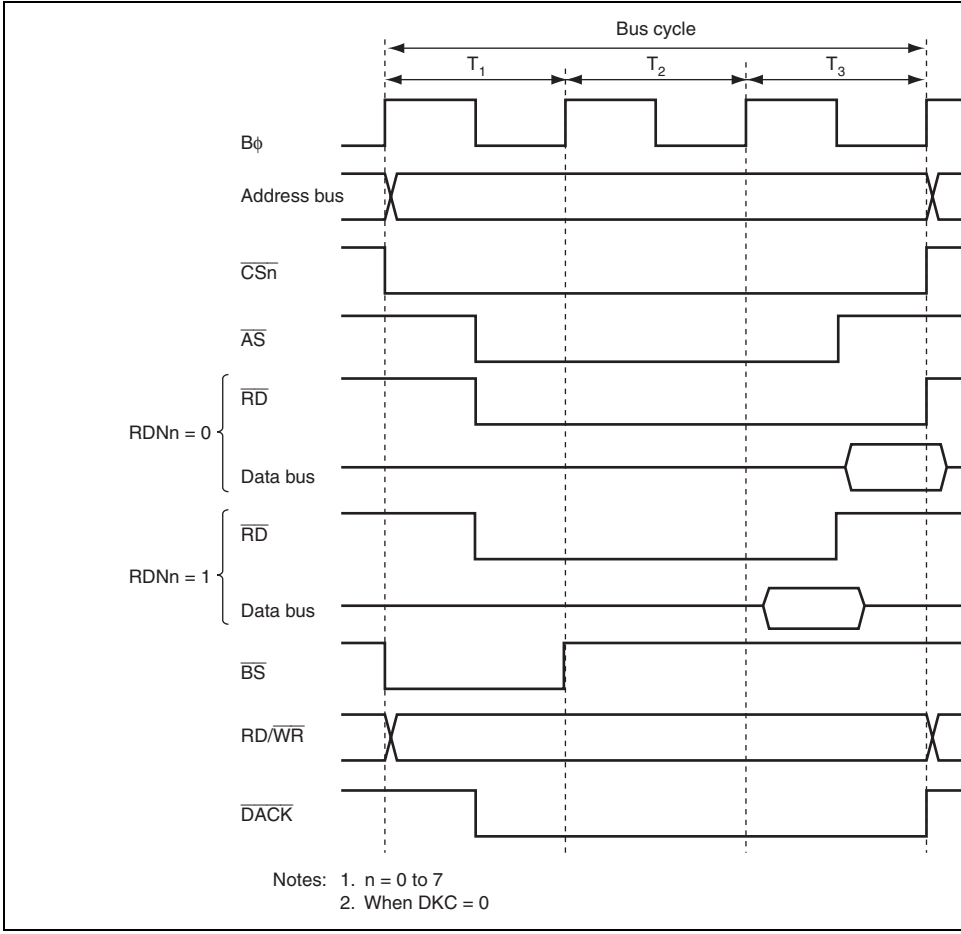
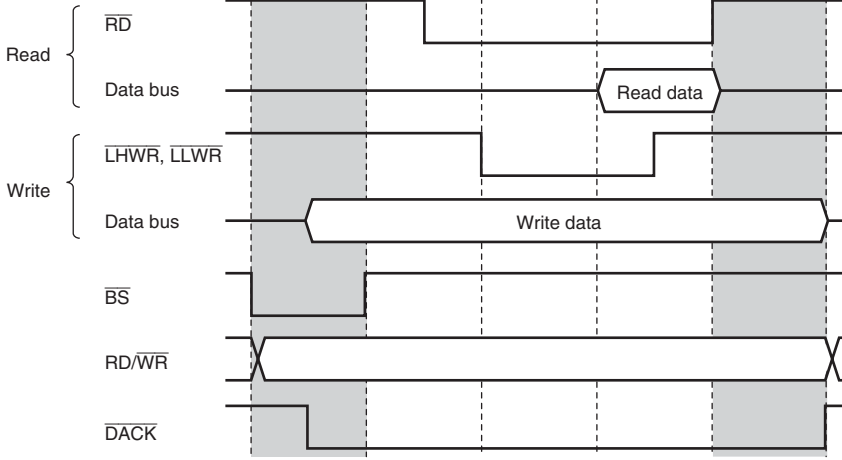


Figure 8.21 Example of Read Strobe Timing

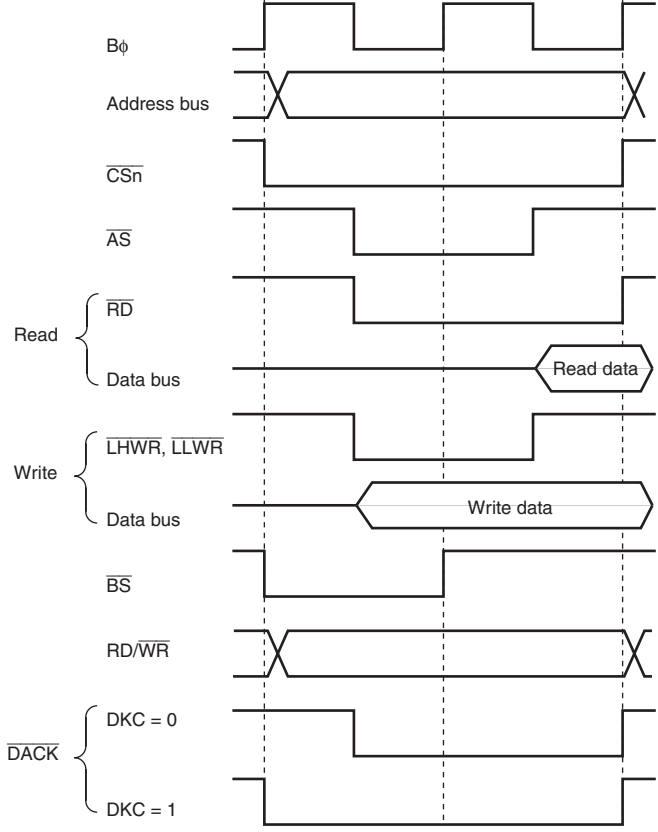
3-state access space.

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



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

Figure 8.22 Example of Timing when Chip Select Assertion Period is Exten



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

Figure 8.23 DACK Signal Output Timing

8.7.1 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.

8.7.2 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
8.5.6, Endian and Data Alignment.

AS/AH	AS	Address strobe	Output	Strobe signal indicating that the ad output on the address bus is valid v 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 v 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

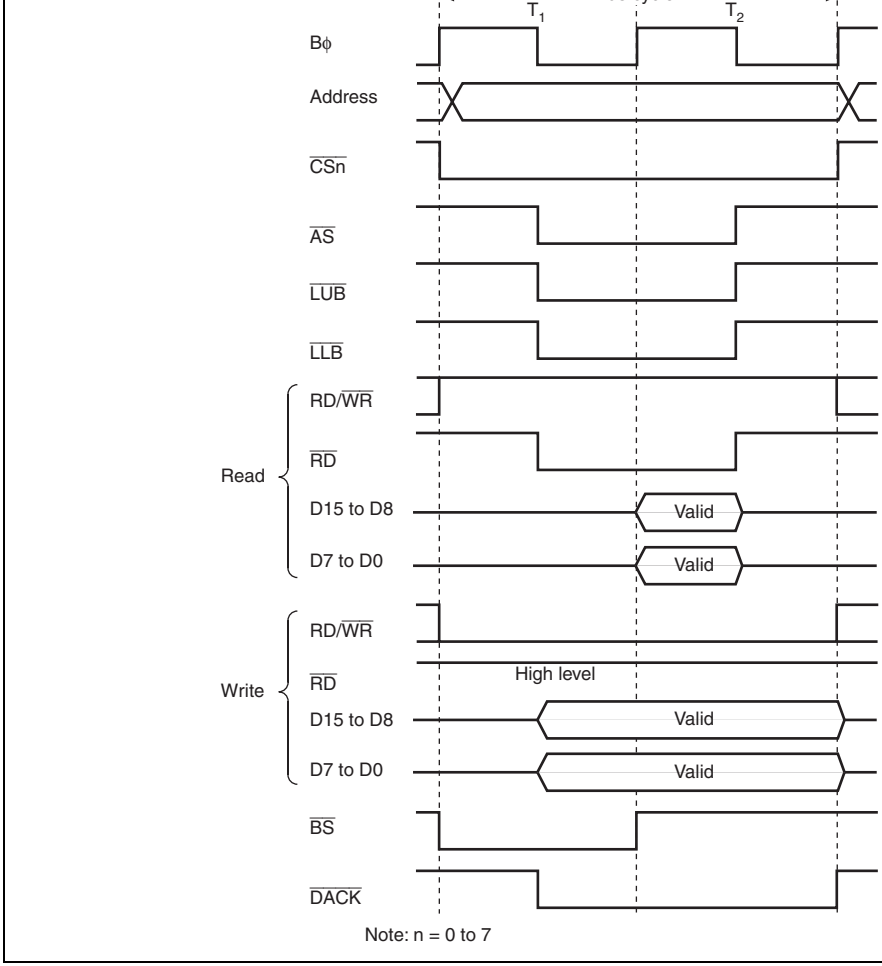


Figure 8.24 16-Bit 2-State Access Space Bus Timing

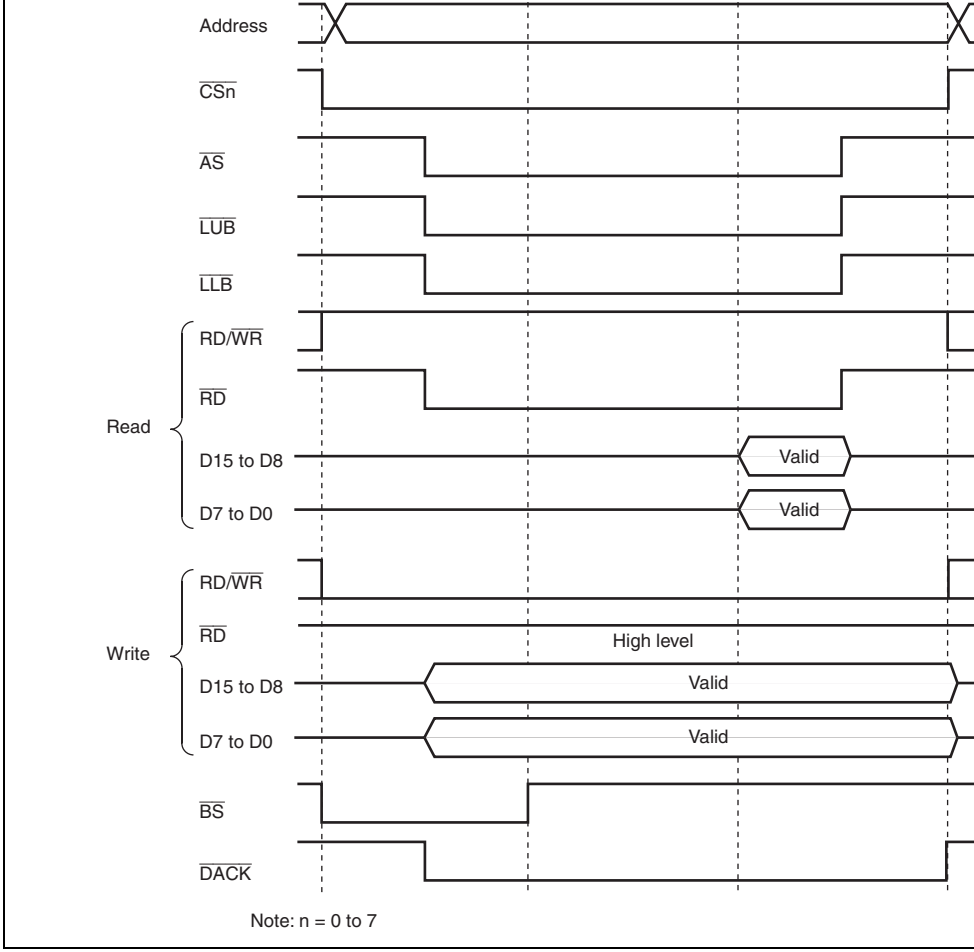
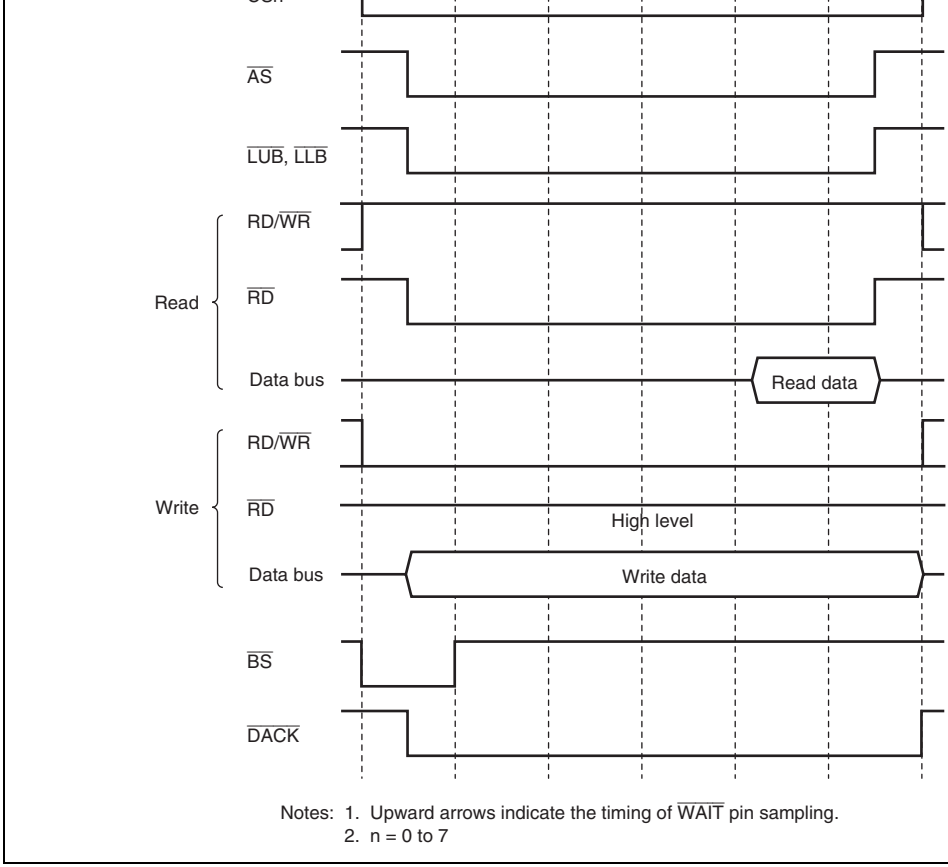


Figure 8.25 16-Bit 3-State Access Space Bus Timing

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 11, I/O Ports.

Figure 8.26 shows an example of wait cycle insertion timing.



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

Figure 8.26 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 8.6.6, Extensior
Select ($\overline{CS}$) Assertion Period.

8.7.8 $\overline{DACK}$ Signal Output Timing

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

Figure 8.27 shows the $\overline{DACK}$ signal output timing. Setting the DKC bit to 1 asserts the
signal a half cycle earlier.

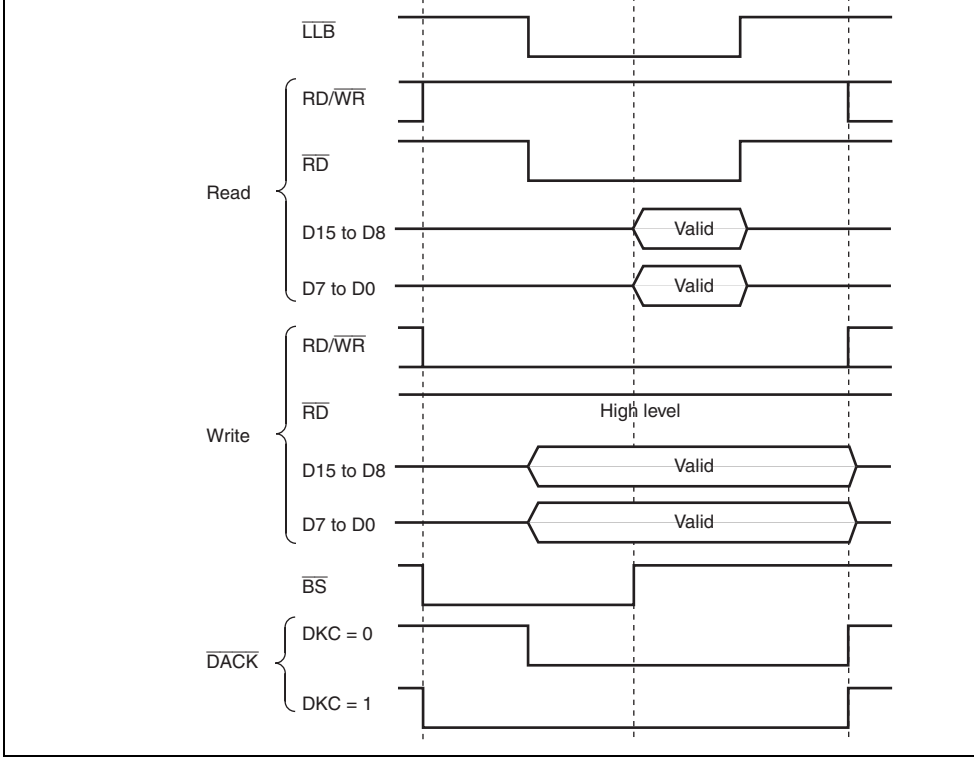


Figure 8.27 $\overline{\text{DACK}}$ Signal Output Timing

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

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

8.8.1 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.

8.8.2 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
8.5.6, 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

The basic access timing for burst ROM space is shown in figures 8.28 and 8.29.

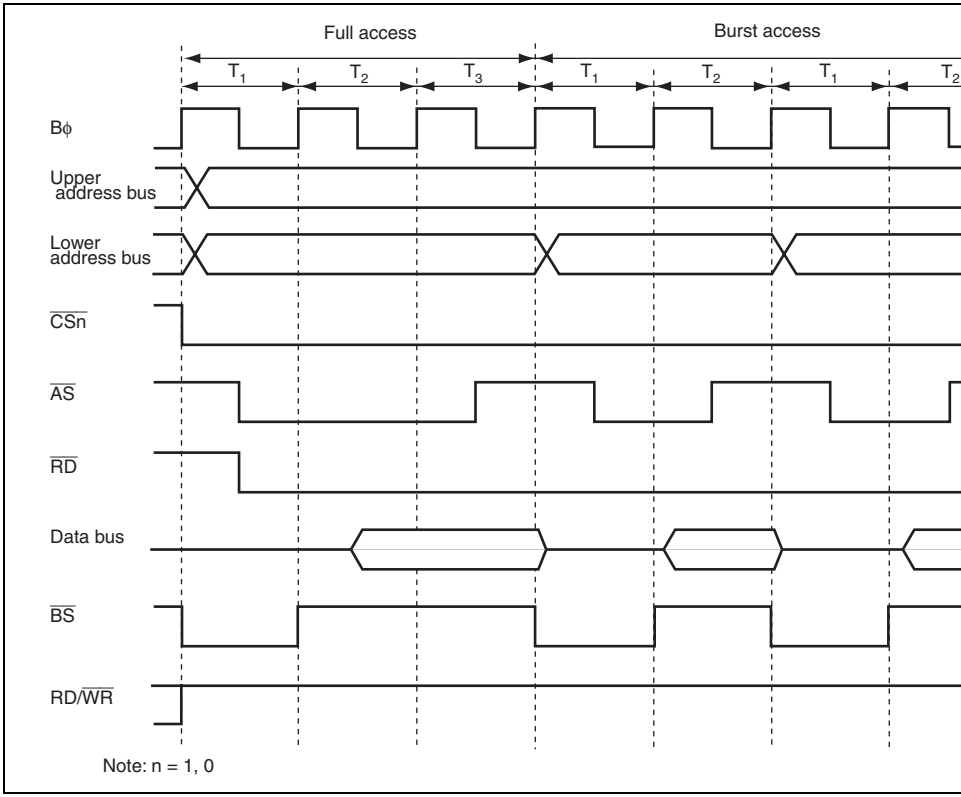


Figure 8.28 Example of Burst ROM Access Timing (ASTn = 1, Two Burst Cy

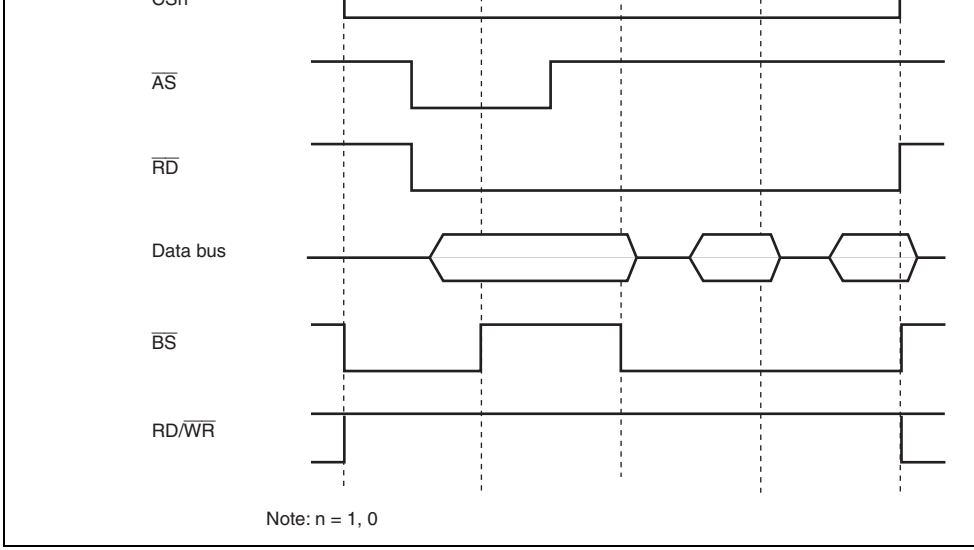


Figure 8.29 Example of Burst ROM Access Timing (ASTn = 0, One Burst Cy

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

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

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

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

8.9.2 Address/Data Multiplex

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

Table 8.18 Address/Data Multiplex

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

8.9.3 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 8.5.6, Endian and Data Alignme

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

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

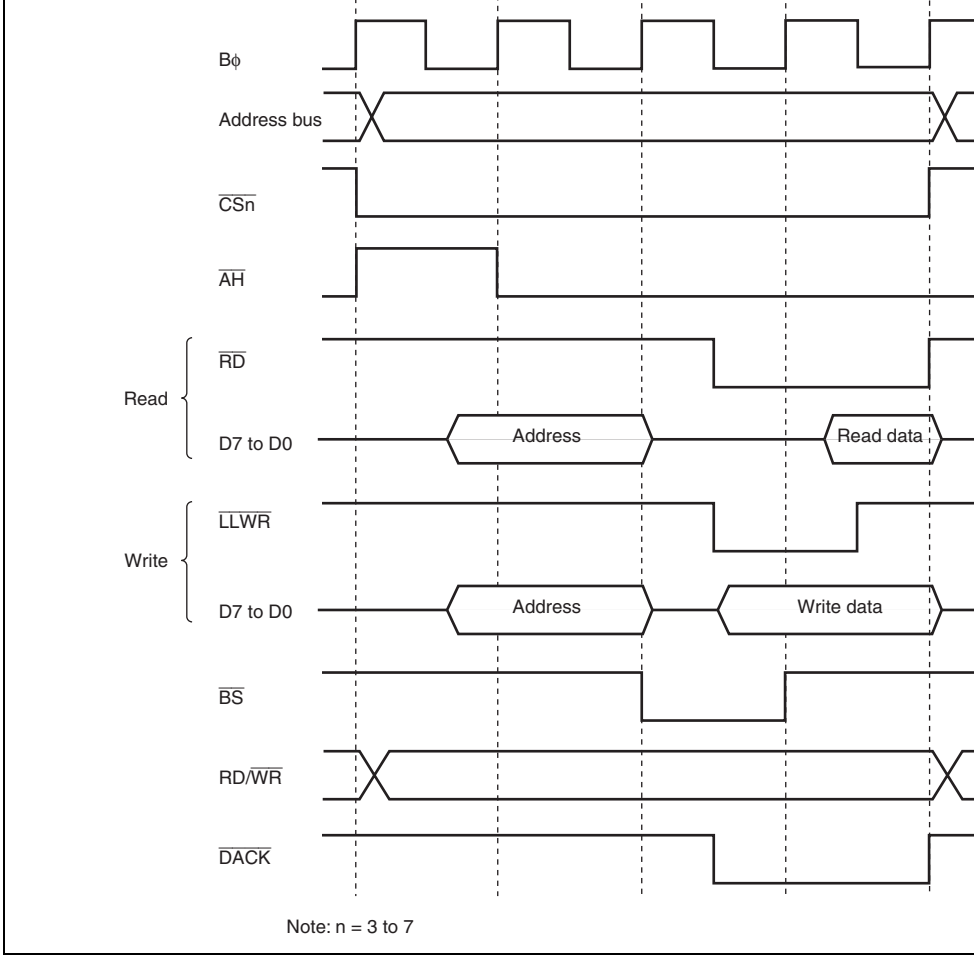


Figure 8.30 8-Bit Access Space Access Timing (ABWHn = 1, ABWLn = 1)

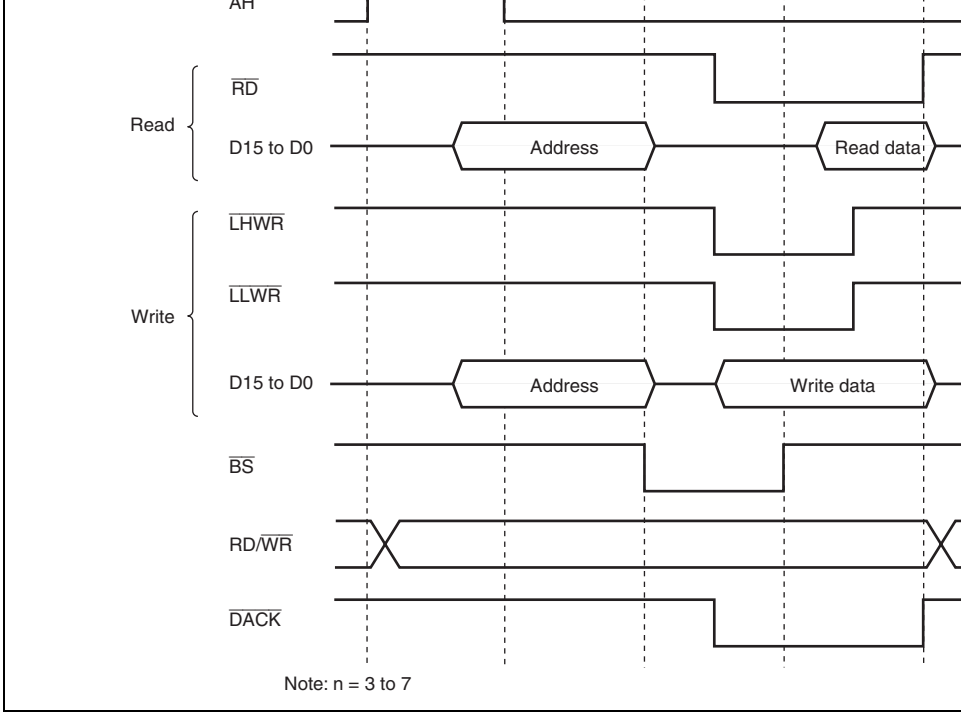


Figure 8.31 16-Bit Access Space Access Timing (ABWHn = 0, ABWLn =

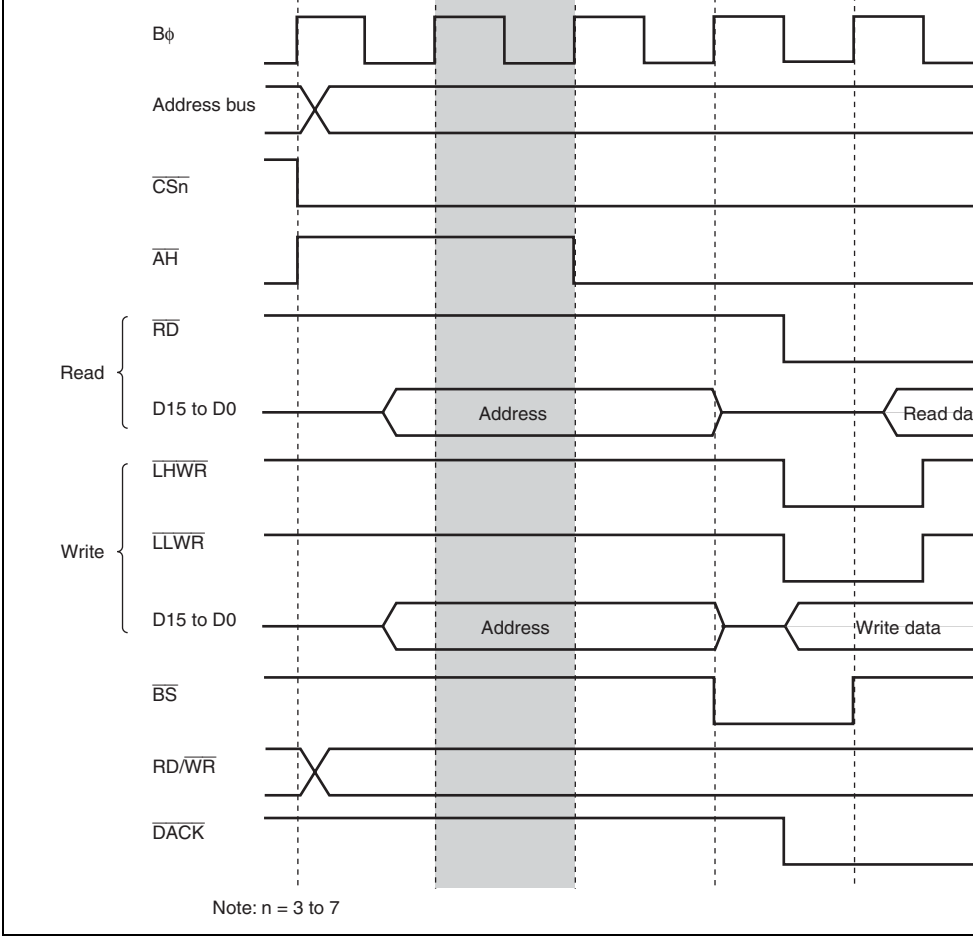


Figure 8.32 Access Timing of 3 Address Cycles (ADDEX = 1)

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 8.6.5, Read Strobe ($\overline{RD}$

Figure 8.33 shows an example when the read strobe timing is modified.

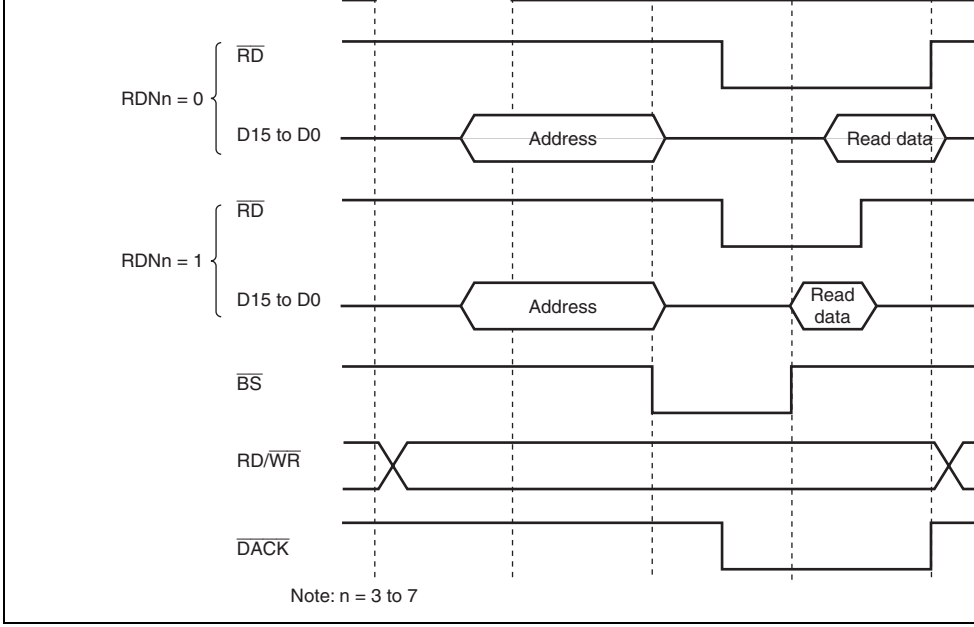


Figure 8.33 Read Strobe Timing

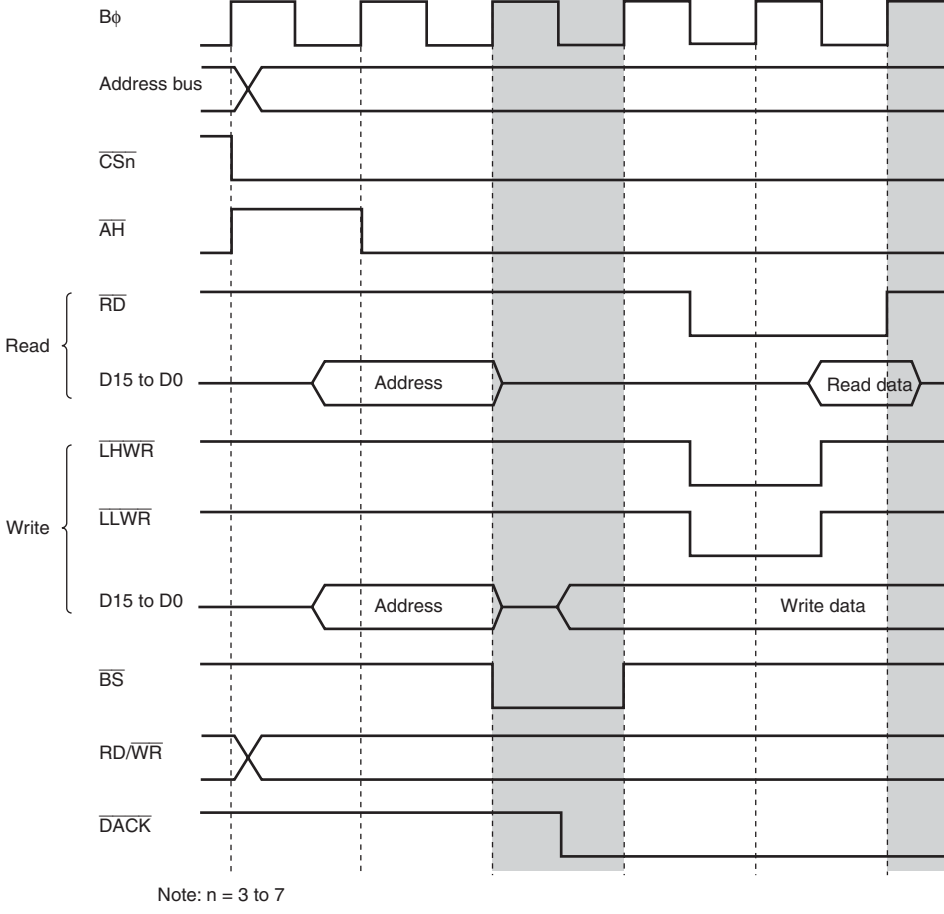
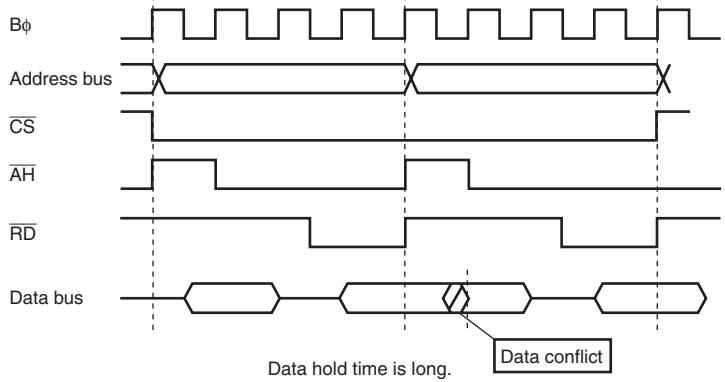
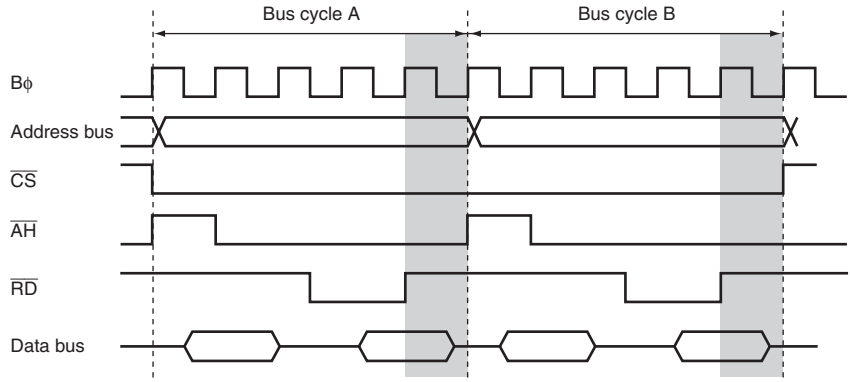


Figure 8.34 Chip Select (CS) Assertion Period Extension Timing in Data Cy



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



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

Figure 8.35 Consecutive Read Accesses to Same Area (Address/Data Multiplexed I/O Space)

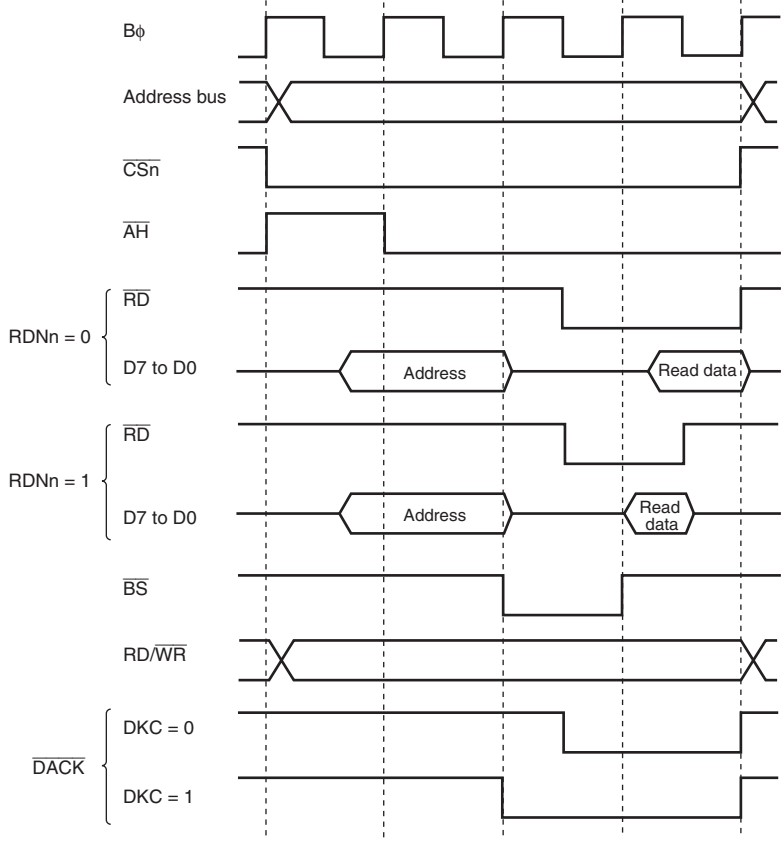


Figure 8.36 DACK Signal Output Timing

and write and previously accessed area.

1. When read cycles of different areas in the external address space occur consecutively
2. When an external write cycle occurs immediately after an external read cycle
3. When an external read cycle occurs immediately after an external write cycle
4. When an external access occurs immediately after a DMAC single address transfer (w 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 t idle cycles after write, can be determined by setting A as described above.

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

Table 8.20 shows the correspondence between conditions 1 to 4 and number of idle cycle inserted for each area. Table 8.21 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 8.21 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.

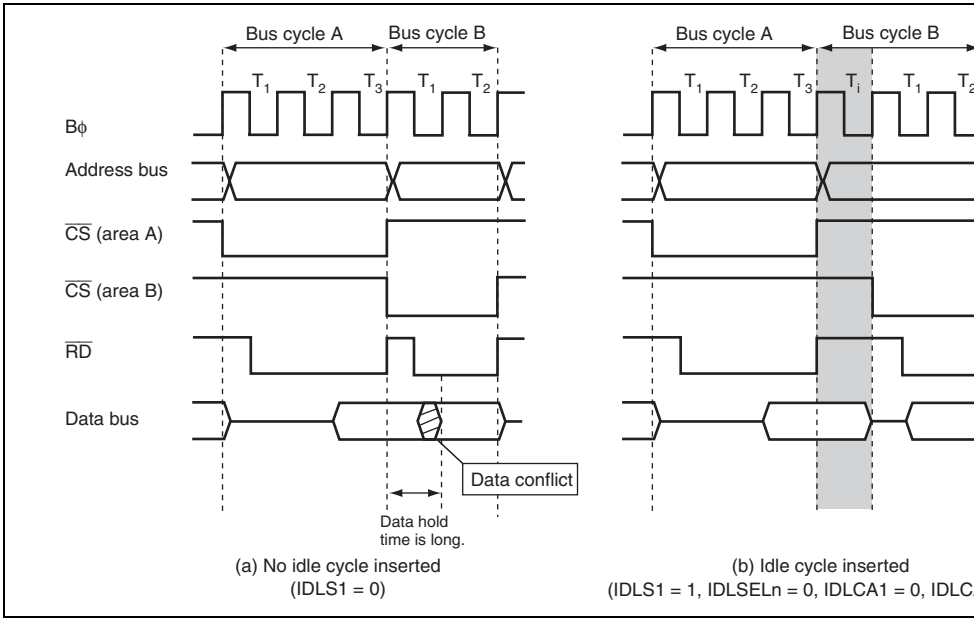


Figure 8.37 Example of Idle Cycle Operation (Consecutive Reads in Different A

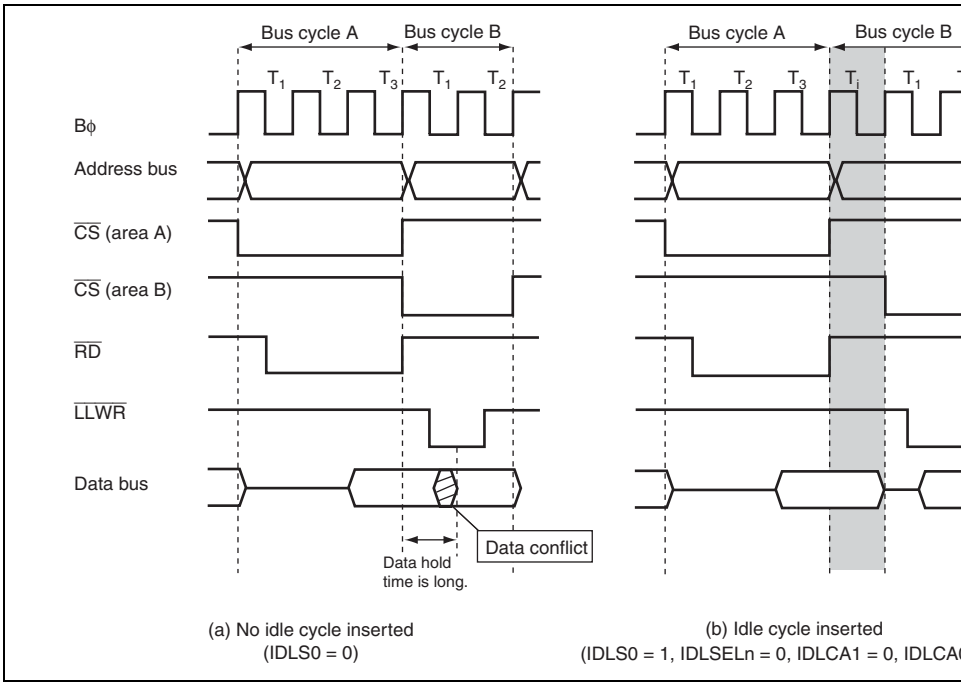


Figure 8.38 Example of Idle Cycle Operation (Write after Read)

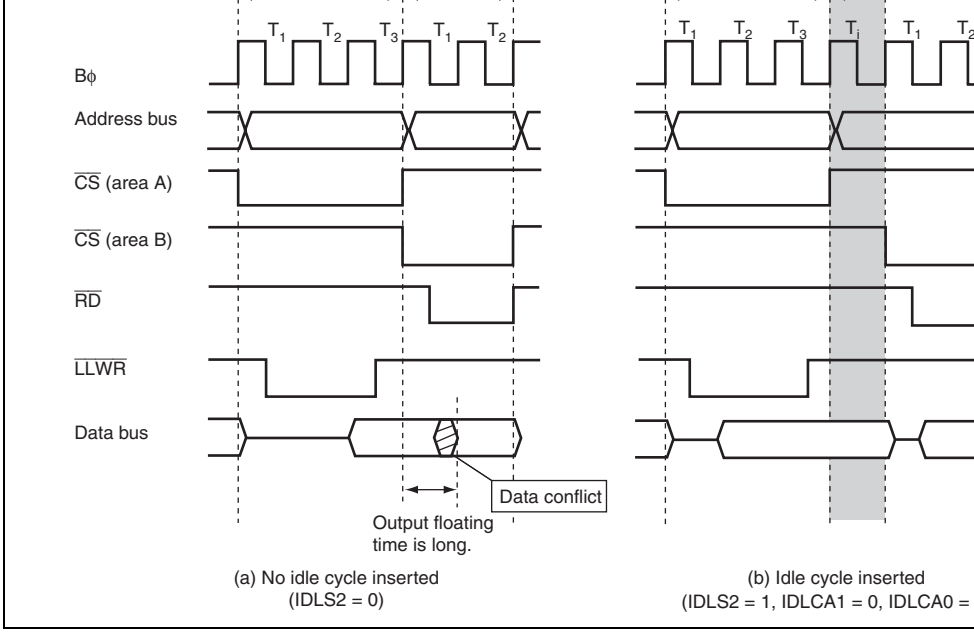


Figure 8.39 Example of Idle Cycle Operation (Read after Write)

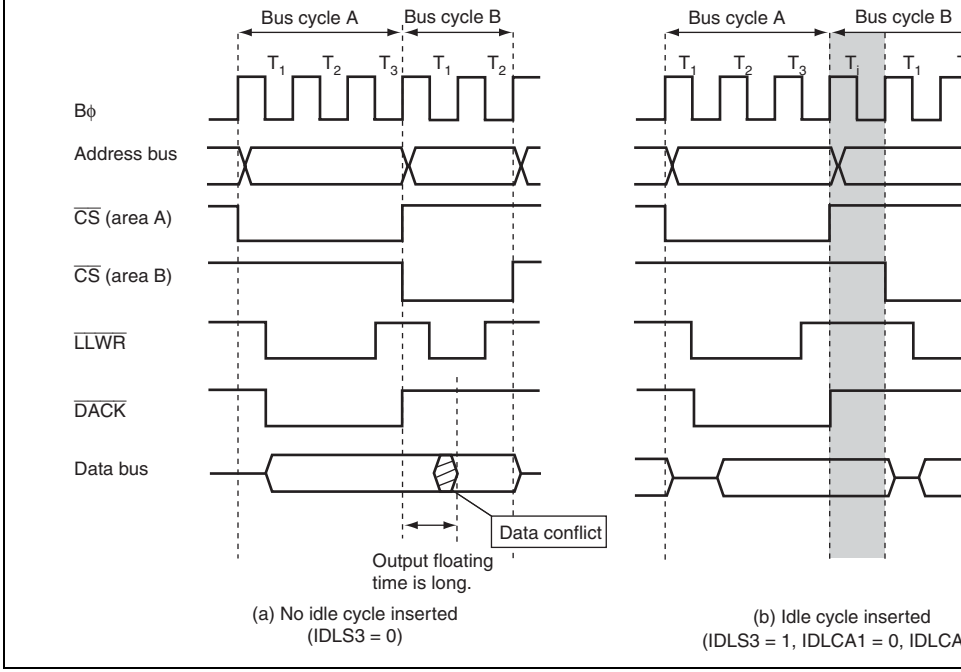


Figure 8.40 Example of Idle Cycle Operation (Write after Single Address Transf

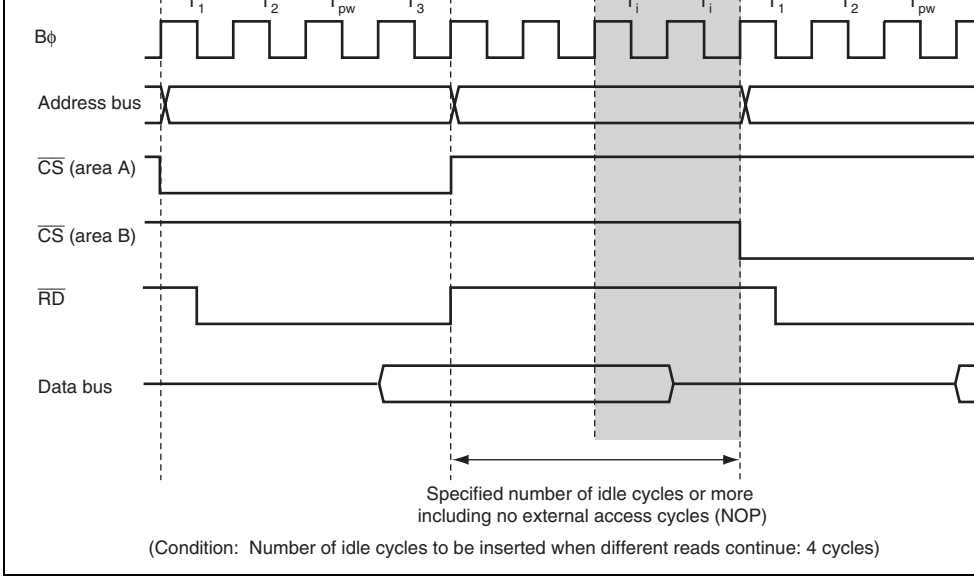


Figure 8.41 Idle Cycle Insertion Example

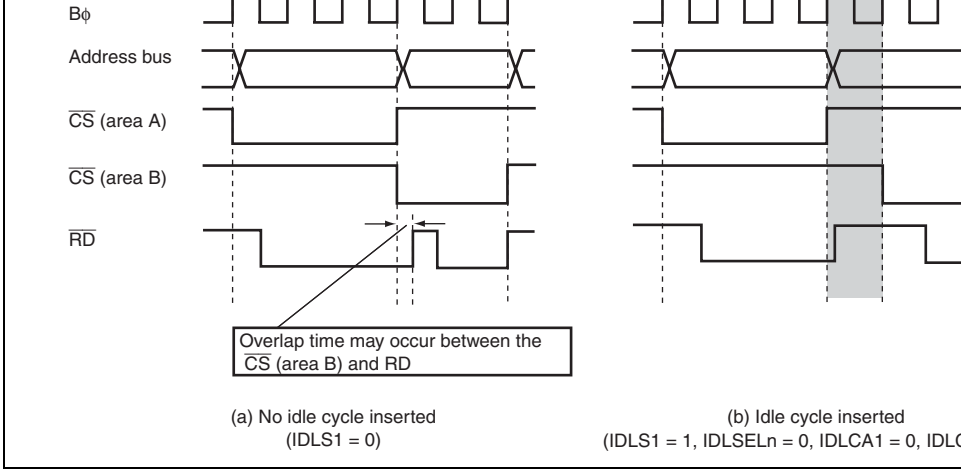


Figure 8.42 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

									0	1	2 cycle in
									1	0	3 cycles i
									1	1	4 cycles i
Normal space read	Normal space write	—	—	—	0	—	—	—	—	—	Disabled
		—	—	—	1	0	0	0	—	—	1 cycle in
							0	1			2 cycles i
							1	0			3 cycles i
							1	1			4 cycles i
						1	—	—	0	0	0 cycle in
									0	1	2 cycle in
									1	0	3 cycles i
									1	1	4 cycles i
Normal space write	Normal space read	—	0	—	—	—	—	—	—	—	Disabled
		—	1	—	—	—	0	0	—	—	1 cycle in
							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 in
							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 = 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 11, 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, or DMAC attempts to access the external address temporarily defers initiation of the bus cycle, and waits for the bus request from the exter master to be canceled.

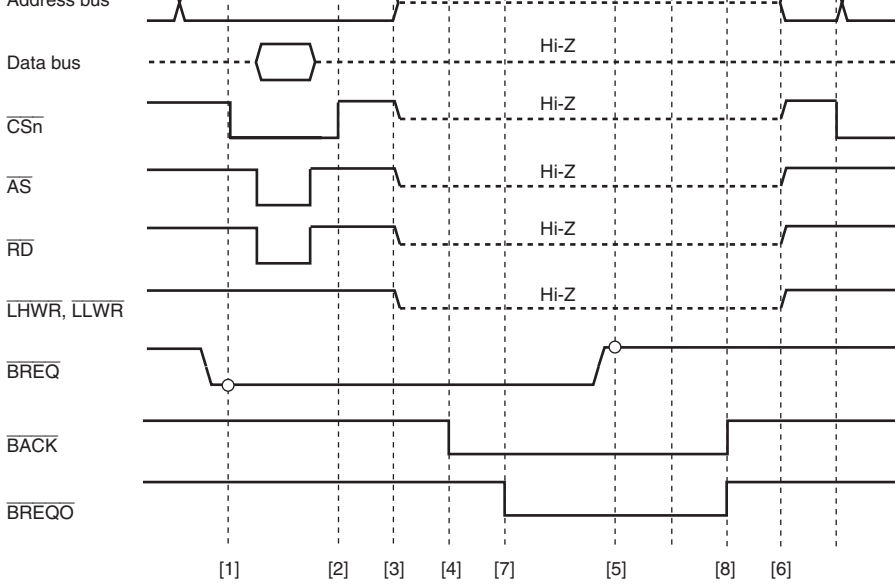
If the BREQOE bit in BCR1 is set to 1, the $\overline{\text{BREQO}}$ pin can be driven low when any of th following requests are issued, to request cancellation of the bus request externally.

- When the CPU, DTC, or DMAC attempts to access the external address space
- When a SLEEP instruction is executed to place the chip in software standby mode or module-clock-stop mode
- When 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) External bus release > External access by CPU, DTC, or DMAC (Low)

$\overline{CS}$n (n = 7 to 0)	High impedance
$\overline{AS}$	High impedance
$\overline{AH}$	High impedance
RD/$\overline{WR}$	High impedance
$\overline{RD}$	High impedance
$\overline{LUB}$, $\overline{LLB}$	High impedance
$\overline{LHWR}$, $\overline{LLWR}$	High impedance
$\overline{DACK}$n (n = 1 to 0)	High



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

Figure 8.43 Bus Released State Transition Timing

Table 8.25 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 c
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 8.26 lists the number of access cycles for registers of on-chip peripheral modules.

Table 8.26 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 UBC registers	Two Iϕ		Disabled
MCU operating mode, clock pulse generator, power-down control registers, interrupt controller, bus controller, 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, TPU, PPG, TMR, SCI0 to SCI4, and D/A registers	Two Pϕ		Enabled
A/D, ΔΣ A/D	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 me
internal I/O register read/write) and the external address space write rather than waiting u
ends are executed in parallel.

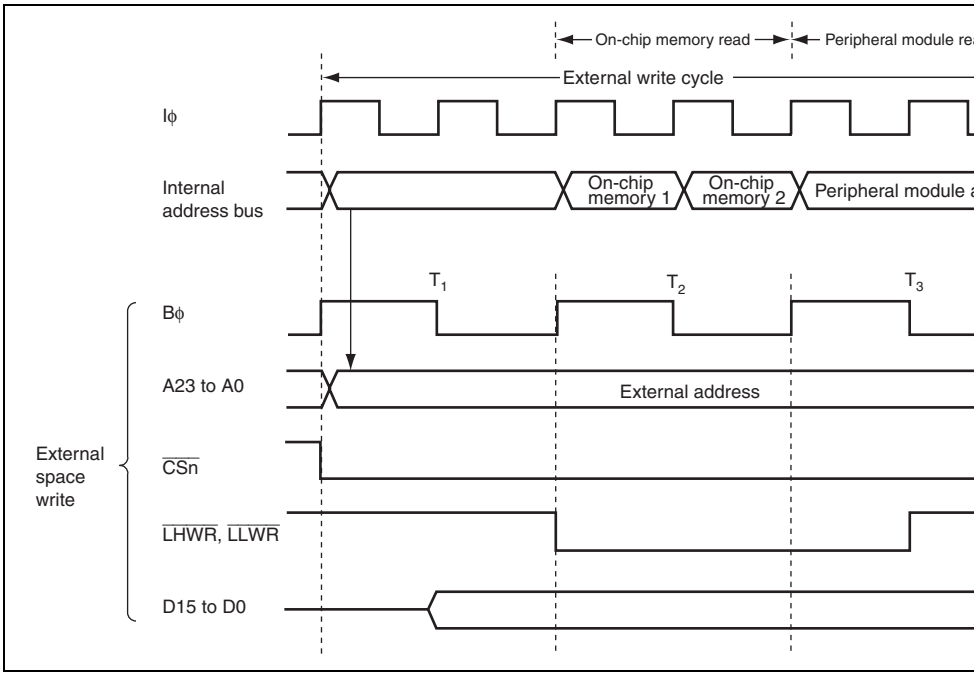


Figure 8.44 Example of Timing when Write Data Buffer Function is Used

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.

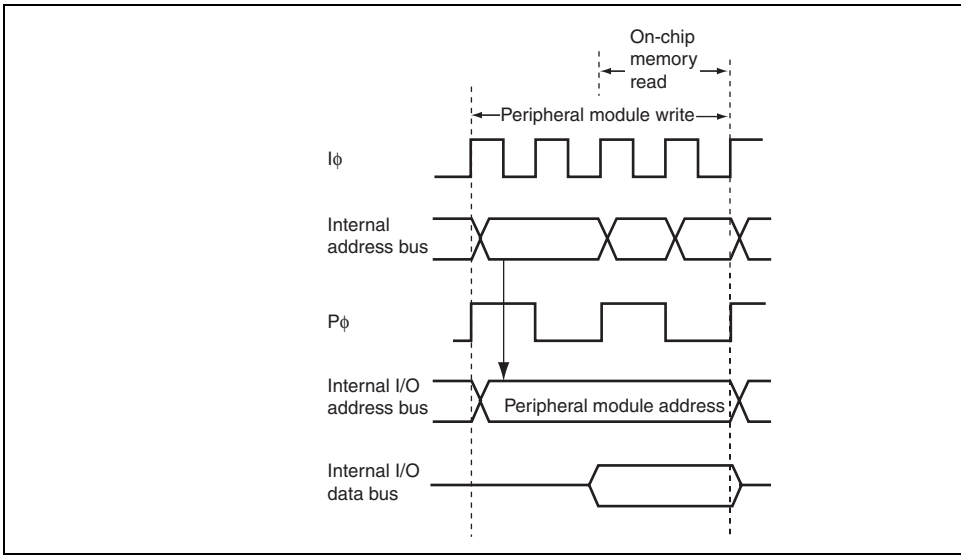


Figure 8.45 Example of Timing when Peripheral Module Write Data Buffer Function is Used

8.14.1 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) External bus release request > External access by the CPU, DTC, and DMAC

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.

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

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

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

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

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

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 ex
with the setting for all peripheral module clocks to be stopped (MSTPCRA and MSTPC
H'FFFFFFFF) or for operation of the 8-bit timer module alone (MSTPCRA = H'F[E to
0]FFFFFF), and a transition is made to the sleep state, the all-module-clock-stop mode i
in which the clock is also stopped for the bus controller and I/O ports. For details, see se
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 MSTPCR must be cleared to 0. Conversely, if a SLEEP ins
place the chip in all-module-clock-stop mode is executed in the external bus released sta
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 lo
mode, indicating an external bus release request, the request cannot be answered until th
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
mode.

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

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

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

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

  In block transfer mode, low level detection is only avail

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

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

  Single address mode: Either source or destination is specified by the $\overline{\text{DREQ}}$ signal 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 add returns to the transfer start address

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

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

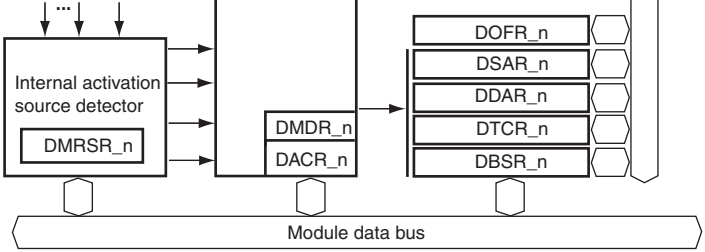
  Up to 65,536 bytes/words/longwords can be set as block

respective boundary

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

- Two types of interrupts can be requested to the CPU

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



[Legend]

DSAR_n:	DMA source address register	$\overline{\text{DREQ}}$n:	DMA transfer request
DDAR_n:	DMA destination address register	$\overline{\text{DACK}}$n:	DMA transfer acknowledge
DOFR_n:	DMA offset register	$\overline{\text{TEND}}$n:	DMA transfer end
DTCR_n:	DMA transfer count register	n = 0, 1	
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 9.1 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

- 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)

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

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

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

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

Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	23	22	21	20	19	18	17	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

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

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

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

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

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

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BKSZH31 to BKSZH16	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 9.2). 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 remain becomes 0, the value of the BKSZH bits is load the same value as the BKSZH bits.

DMDR controls the DMAC operation.

- DMDR_0

Bit	31	30	29	28	27	26	25	
Bit Name	DTE	DACKE	TENDE	—	DREQS	NRD	—	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	
Bit	23	22	21	20	19	18	17	
Bit Name	ACT	—	—	—	ERRF	—	ESIF	
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R/(W)*	R	R/(W)*	R
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	D
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	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	R/W	R

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

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

transfer.

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

If an event which stops (sustains) a transfer 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 mo

				1: Disables TEND signal output
28	—	0	R/W	Reserved Initial value should not be changed.
27	DREQS	0	R/W	DREQ Select Selects whether a low level or the falling edge o DREQ signal used in external request mode is When a block transfer is performed in external mode, clear this bit to 0. 0: Low level detection 1: Falling edge detection (the first transfer 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 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 state, t 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 interru request. When the ESIF bit is set to 1 with this 1, a transfer escape end interrupt is requested CPU or DTC. The transfer end interrupt 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 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 m 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 the 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 $\overline{\text{DACK}}$ pin is according to the DACKE bit. 0: Dual address mode 1: Single address mode
30	DIRS	0	R/W	Single Address Direction Select Specifies the data transfer direction in single ad mode. This bit s ignored in dual address mode. 0: Specifies DSAR as source address 1: Specifies DDAR as destination address
29 to 27	—	0	R/W	Reserved These bits are always read as 0 and cannot be modified.

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

				10: Destination address is updated by adding 1 according to the data access size 11: Destination address is updated by subtracti or 4 according to the data access size
15	SARIE	0	R/W	Interrupt Enable for Source Address Extended Overflow Enables/disables an interrupt request for an ex area overflow on the source address. When an extended repeat area overflow on the address occurs while this bit is set to 1, the DT DMDR is cleared to 0. At this time, the ESIF bit DMDR is set to 1 to indicate an interrupt by an repeat area overflow on the source address is requested. When block transfer mode is used with the exte repeat area function, an interrupt is requested a completion of a 1-block size transfer. When set DTE bit in DMDR of the channel for which a tra been stopped to 1, the transfer is resumed from state when the transfer is stopped. When the extended repeat area is not 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 9.3 shows the settings and a the extended repeat area.
7	DARIE	0	R/W	Destination Address Extended Repeat Area O Interrupt Enable Enables/disables an interrupt request for an e 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 ex repeat area function, an interrupt is requested completion of a 1-block size transfer. When se DTE bit in DMDR of the channel for which the has been stopped to 1, the transfer is resume state when the transfer is stopped. When the extended repeat area is not specifie is ignored. 0: Disables an interrupt request for an extende overflow on the destination address 1: Enables an interrupt request for an extende overflow on the destination address
6, 5	—	All 0	R	Reserved These bits are always read as 0 and cannot b modified.

area for address addition and subtraction, resp

When an overflow in the extended repeat area with the DARIE bit set to 1, an interrupt can be requested. Table 9.3 shows the settings and ar the extended repeat area.

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

[Legend]

×: Don't care

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 9.2 shows an example of the signal timing in dual address mode and figure 9.3 sho
operation in dual address mode.

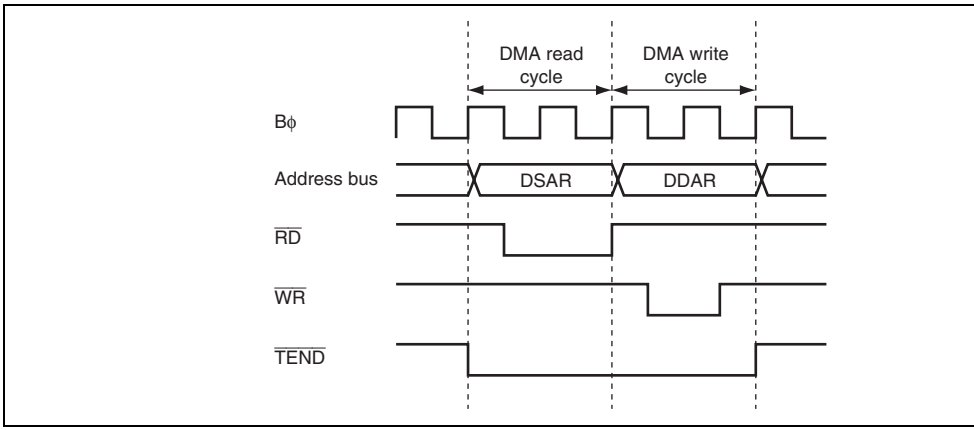


Figure 9.2 Example of Signal Timing in Dual Address Mode

Figure 9.3 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 8, 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.

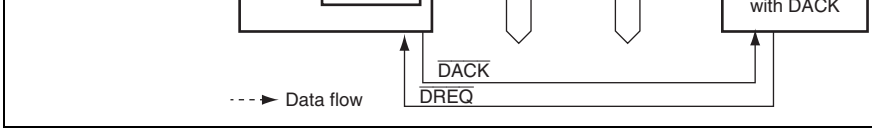


Figure 9.4 Data Flow in Single Address Mode

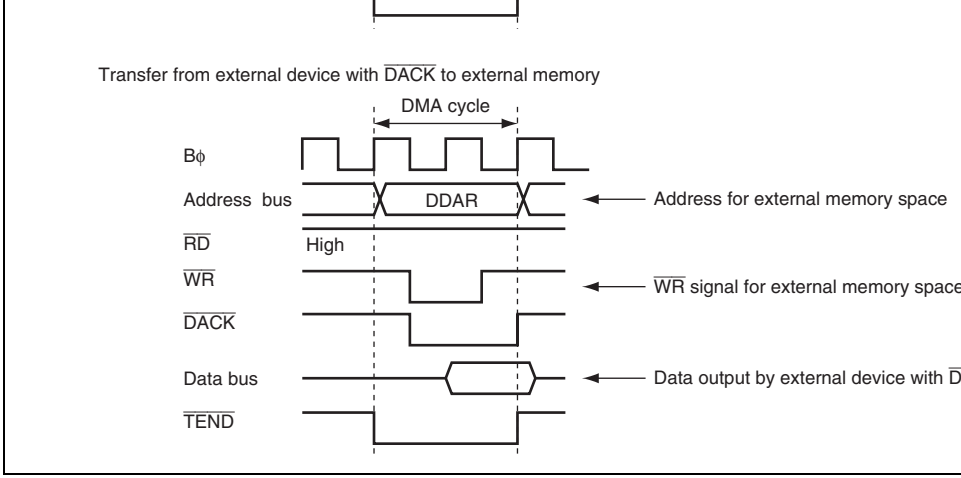


Figure 9.5 Example of Signal Timing in Single Address Mode

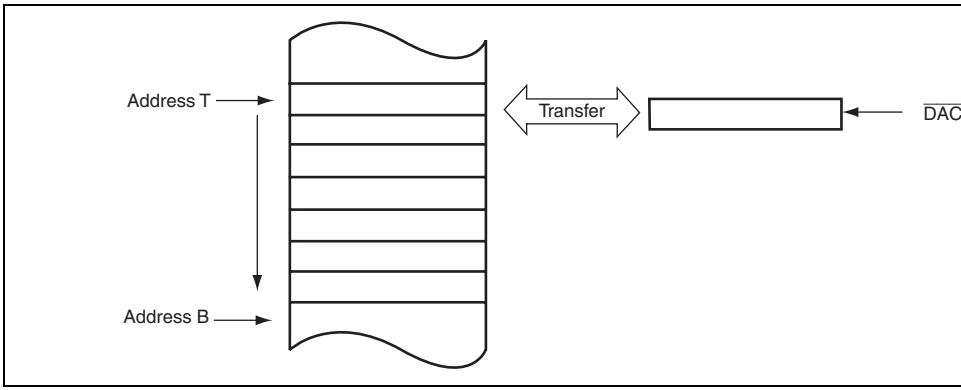


Figure 9.6 Operations in Single Address Mode

the operation in normal transfer mode.

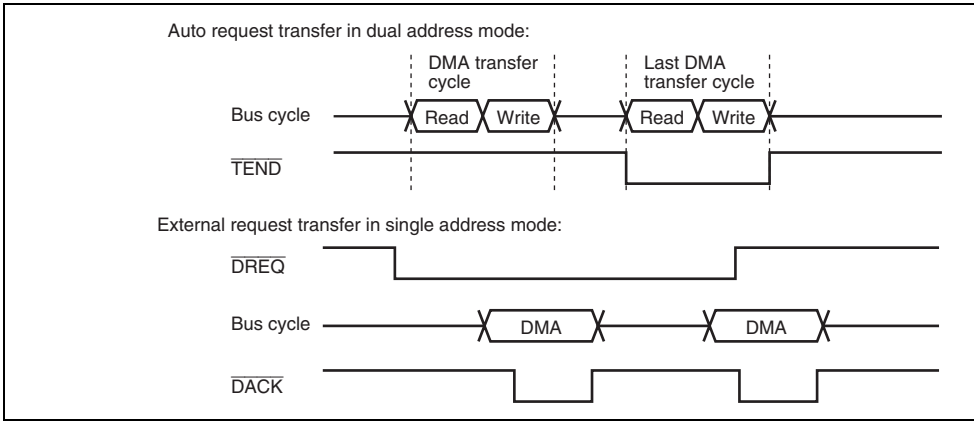


Figure 9.7 Example of Signal Timing in Normal Transfer Mode

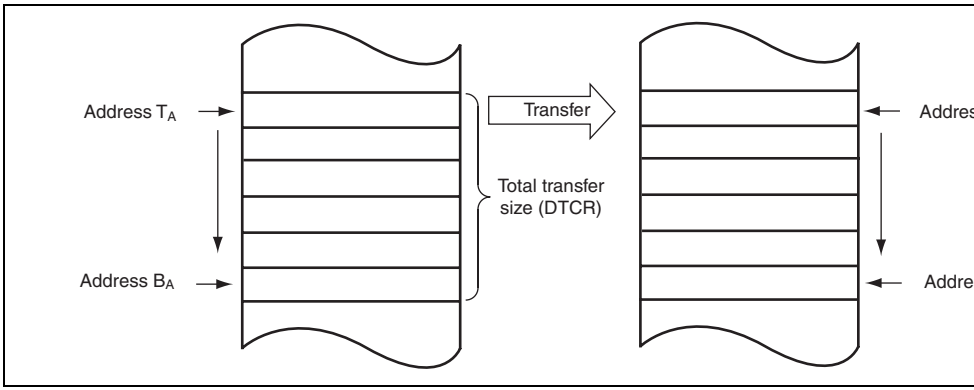


Figure 9.8 Operations in Normal Transfer Mode

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

Figure 9.9 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 9.8. In this case, a repea
interrupt can also be requested to the CPU when the repeat size of transfers is completed

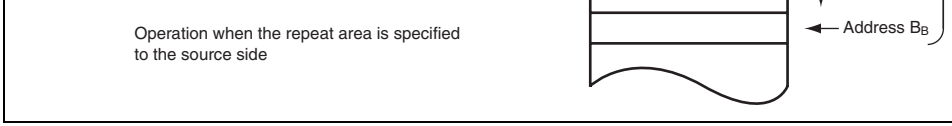


Figure 9.9 Operations in Repeat Transfer Mode

(3) Block Transfer Mode

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

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

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

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

When an interrupt request by an extended repeat area overflow is used in block transfer m
settings should be selected carefully. For details, see section 9.5.5, Extended Repeat Area
Function.

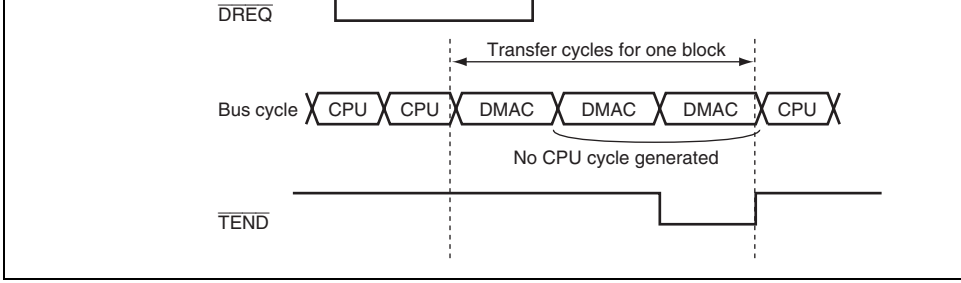


Figure 9.10 Operations in Block Transfer Mode

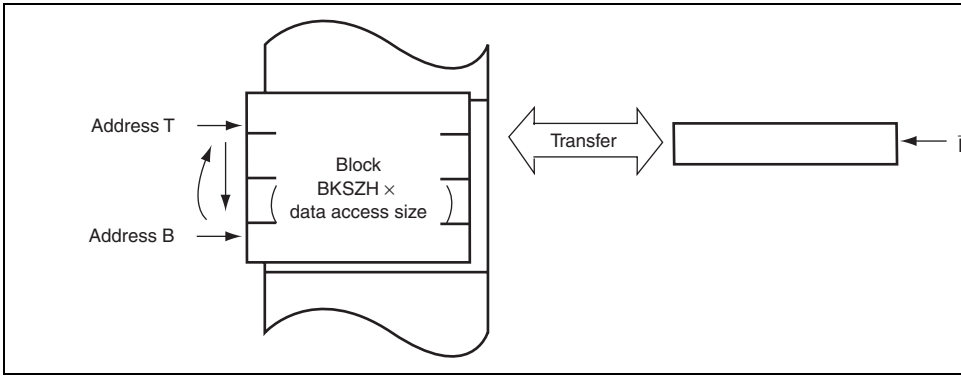


Figure 9.11 Operation in Single Address Mode in Block Transfer Mode (Block Area Specified)

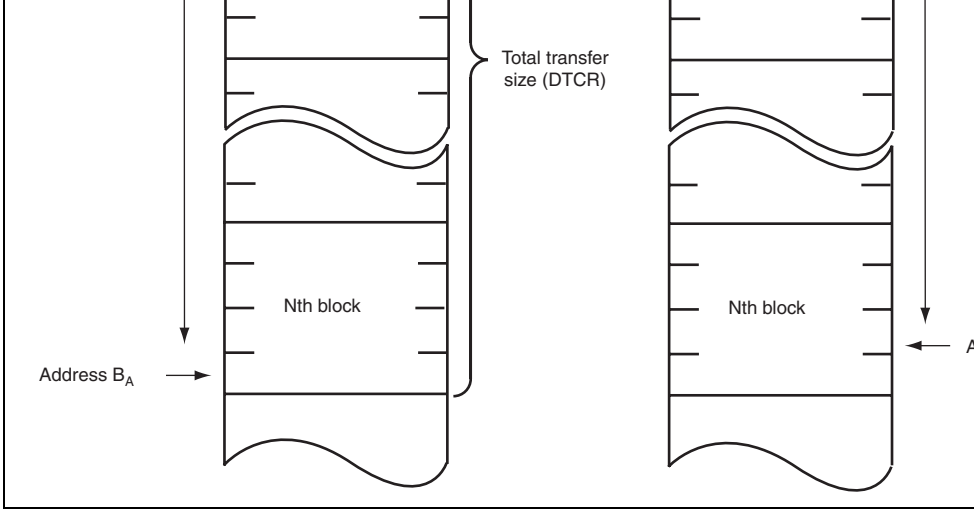


Figure 9.12 Operation in Dual Address Mode in Block Transfer Mode (Block Area Not Specified)

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

(2) Activation by On-Chip Module Interrupt

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

The activation source of the on-chip module interrupt is selected by the DMA module re select register (DMRSR). The activation sources are specified to the individual channels 9.5 is a list of on-chip module interrupts for the DMAC. The interrupt request selected a activation source can generate an interrupt request simultaneously to the CPU or DTC. F refer to section 6, 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 source 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.

RXI0 (receive data full interrupt from SCI channel 0)	SCI_0	14
TXI0 (transmit data empty interrupt from SCI channel 0)	SCI_0	14
RXI1 (receive data full interrupt from SCI channel 1)	SCI_1	14
TXI1 (transmit data empty interrupt from SCI channel 1)	SCI_1	15
RXI2 (receive data full interrupt from SCI channel 2)	SCI_2	15
TXI2 (transmit data empty interrupt from SCI channel 2)	SCI_2	15
RXI3 (receive data full interrupt from SCI channel 3)	SCI_3	15
TXI3 (transmit data empty interrupt from SCI channel 3)	SCI_3	15
RXI4 (receive data full interrupt from SCI channel 4)	SCI_4	16
TXI4 (transmit data empty interrupt from SCI channel 4)	SCI_4	16
ADIO (conversion end interrupt from A/D converter)	ADIO	22
DSADI (conversion end interrupt from $\Delta\Sigma$ A/D converter)	DSADI	22

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 11, I/O Ports.

9.5.4 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 9.5.8, Priority of Channels.

Bus released temporarily for the CPU

Figure 9.13 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 IBCR of the bus controller makes the D 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 DT cleared to 0 by the DMAC is executed. When an interrupt by a transfer size error, a repea end, or an extended repeat area overflow occurs, the DTE bit is cleared to 0 and the trans

Figure 9.14 shows an example of timing in burst mode.

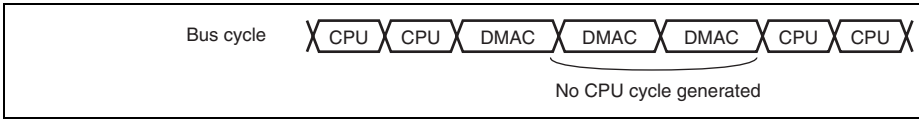


Figure 9.14 Example of Timing in Burst Mode

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 SAR
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

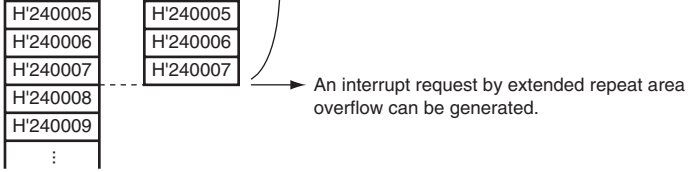


Figure 9.15 Example of Extended Repeat Area Operation

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

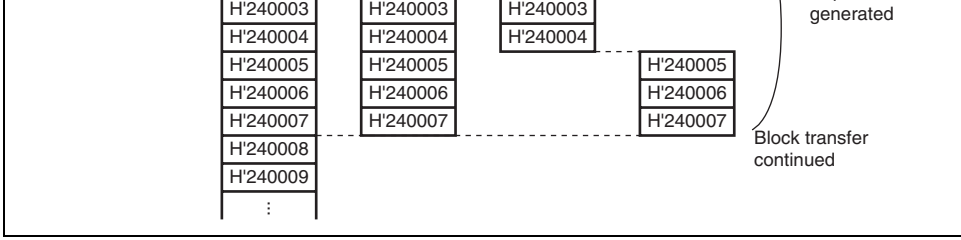


Figure 9.16 Example of Extended Repeat Area Function in Block Transfer M

9.5.6 Address Update Function using Offset

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

Figure 9.17 shows the address update method.

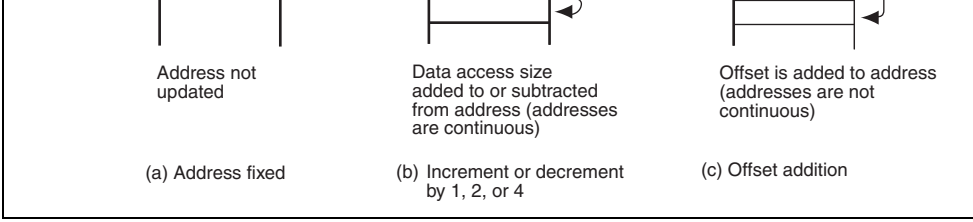


Figure 9.17 Address Update Method

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.

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

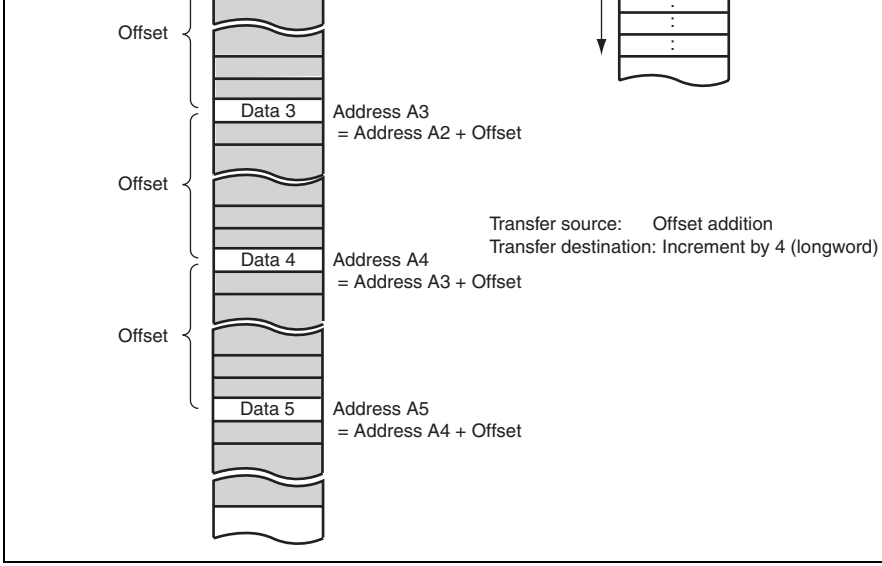


Figure 9.18 Operation of Offset Addition

In figure 9.18, the offset addition is selected as the transfer source address update and in
decrement by 1, 2, or 4 is selected as the transfer destination address. The address updat
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.

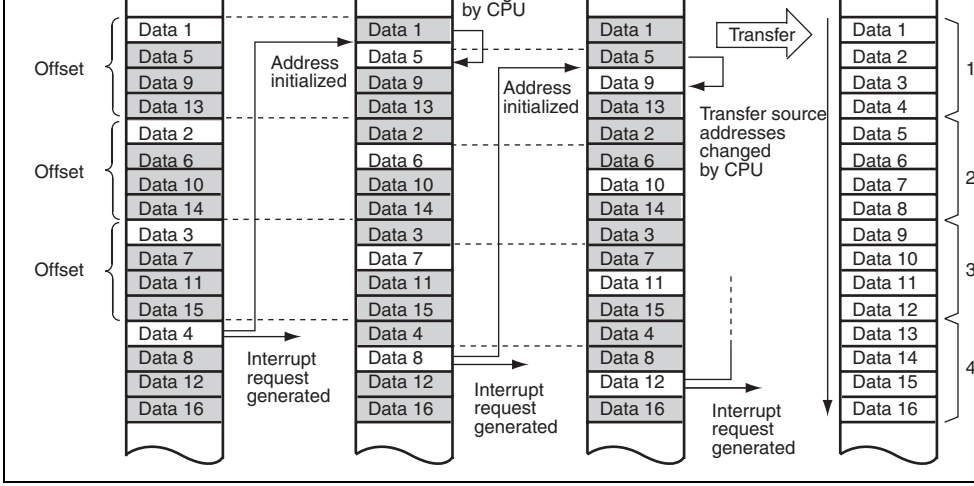


Figure 9.19 XY Conversion Operation Using Offset Addition in Repeat Transfer

In figure 9.19, the source address side is specified to the repeat area by DACR and the of 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 destination A repeat size end interrupt is requested when the repeat size of transfers is completed.

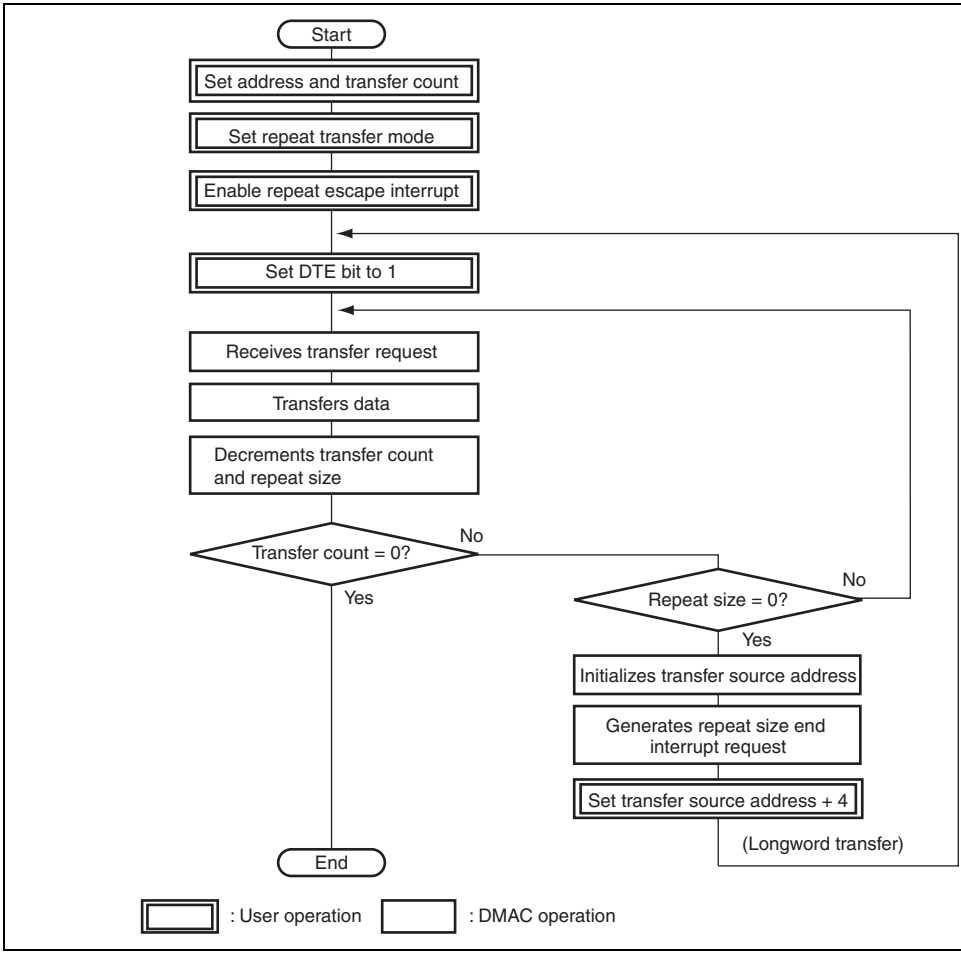


Figure 9.20 XY Conversion Flowchart Using Offset Addition in Repeat Transfe

9.3.7 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 outp
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 SAT0
the address is decremented. The size of increment or decrement depends on the data acce

The data access size is specified by bits DTSZ1 and DTSZ0 in DMDR. When DTSZ1 an
= B'00, the data access size is byte and the address is incremented or decremented by 1. W
DTSZ1 and DTSZ0 = B'01, the data access size is word and the address is incremented o
decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data access size is longword and
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 long
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 ac 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 comp 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 9.21 show the procedure for changing the register settings for the channel being
transferred.

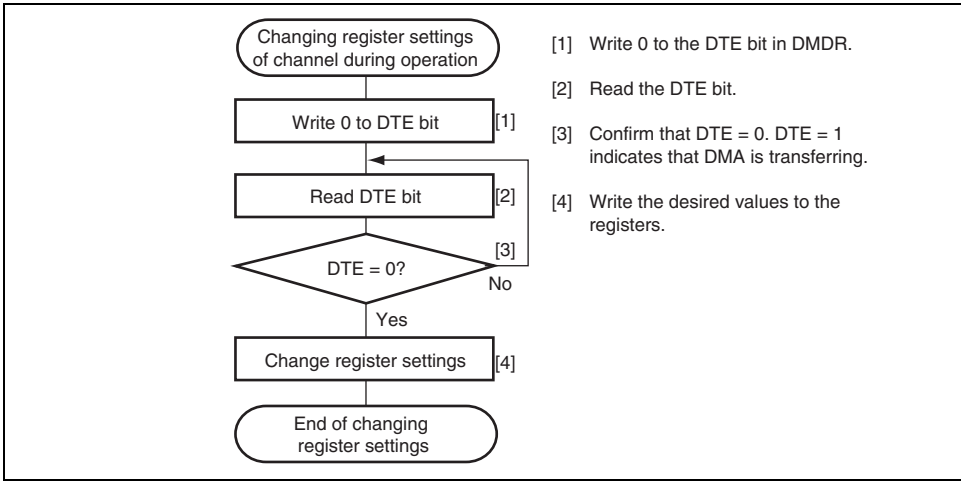


Figure 9.21 Procedure for Changing Register Setting For Channel being Trans

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

(7) ERRF Bit in DMDR

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

(8) ESIF Bit in DMDR

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

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

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

For details on interrupts, see section 9.8, Interrupt Sources.

For details on interrupts, see section 9.8, Interrupt Sources.

9.5.8 Priority of Channels

The channels of the DMAC are given following priority levels: channel 0 > channel 1. T
shows the priority levels among the DMAC channels.

Table 9.6 Priority among DMAC Channels

Channel	Priorit
Channel 0	High
Channel 1	Low

The channel having highest priority other than the channel being transferred is selected
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.

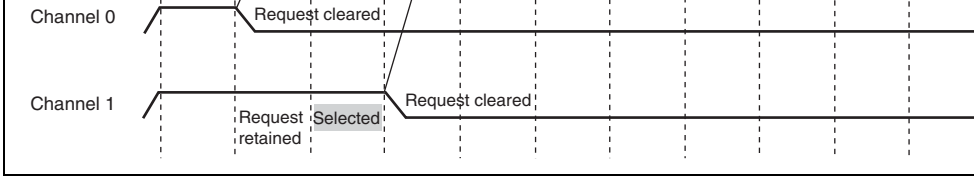


Figure 9.22 Example of Timing for Channel Priority

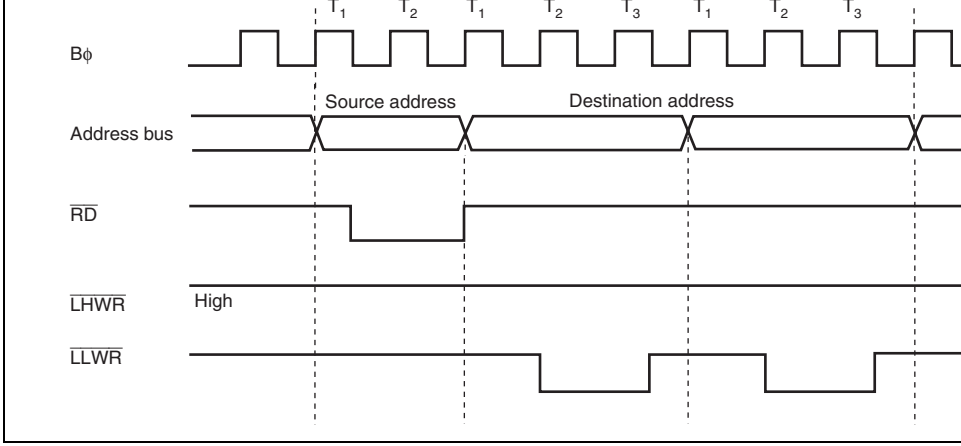


Figure 9.23 Example of Bus Timing of DMA Transfer

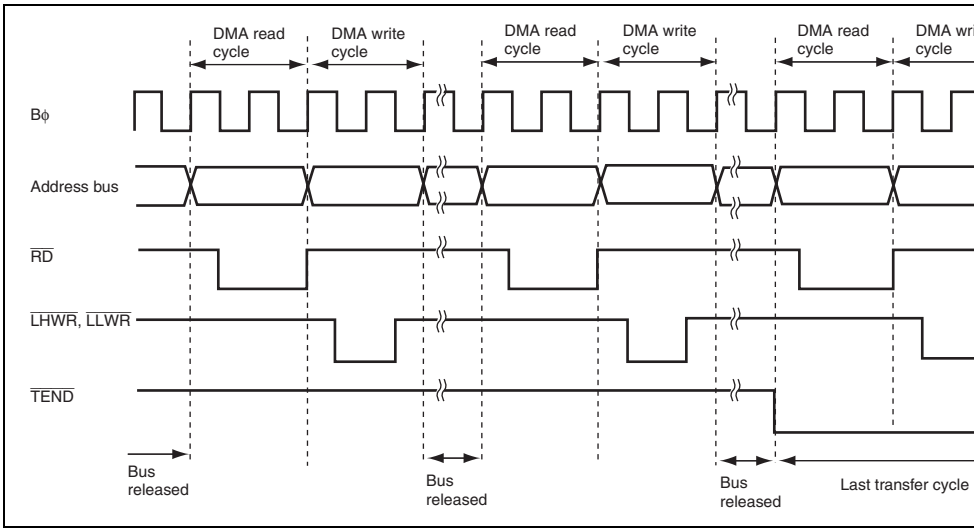


Figure 9.24 Example of Transfer in Normal Transfer Mode by Cycle Steali

In figures 9.25 and 9.26, the $\overline{\text{TEND}}$ signal output is enabled and data is transferred in lon from the external 16-bit 2-state access space to the 16-bit 2-state access space in normal t mode by cycle stealing.

In figure 9.25, the transfer source (DSAR) is not aligned with a longword boundary and t transfer destination (DDAR) is aligned with a longword boundary.

In figure 9.26, the transfer source (DSAR) is aligned with a longword boundary and the t destination (DDAR) is not aligned with a longword boundary.

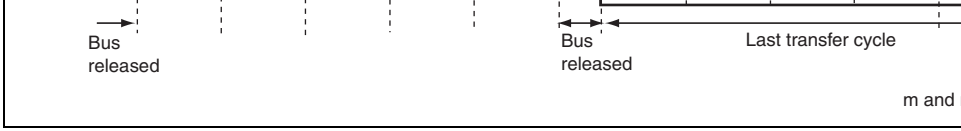


Figure 9.25 Example of Transfer in Normal Transfer Mode by Cycle Steal (Transfer Source DSAR = Odd Address and Source Address Increment)

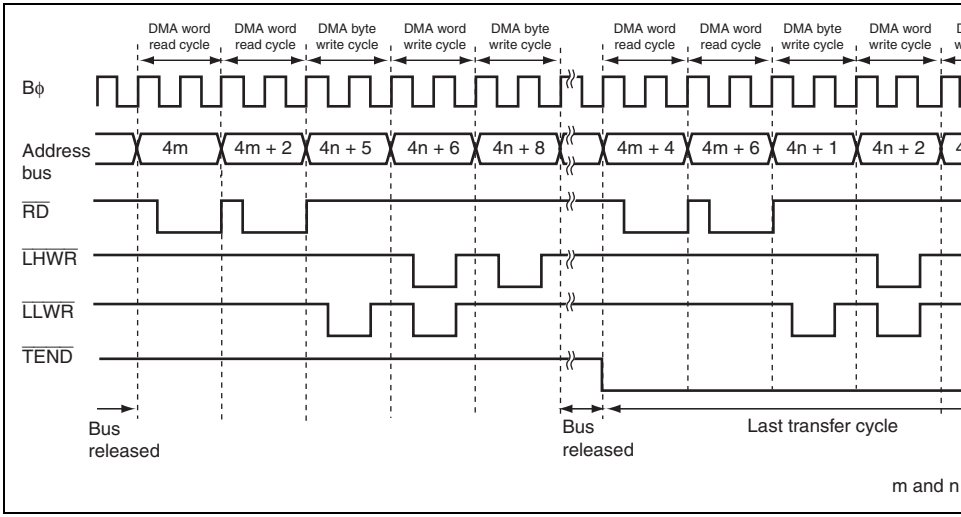


Figure 9.26 Example of Transfer in Normal Transfer Mode by Cycle Steal (Transfer Destination DDAR = Odd Address and Destination Address Decren

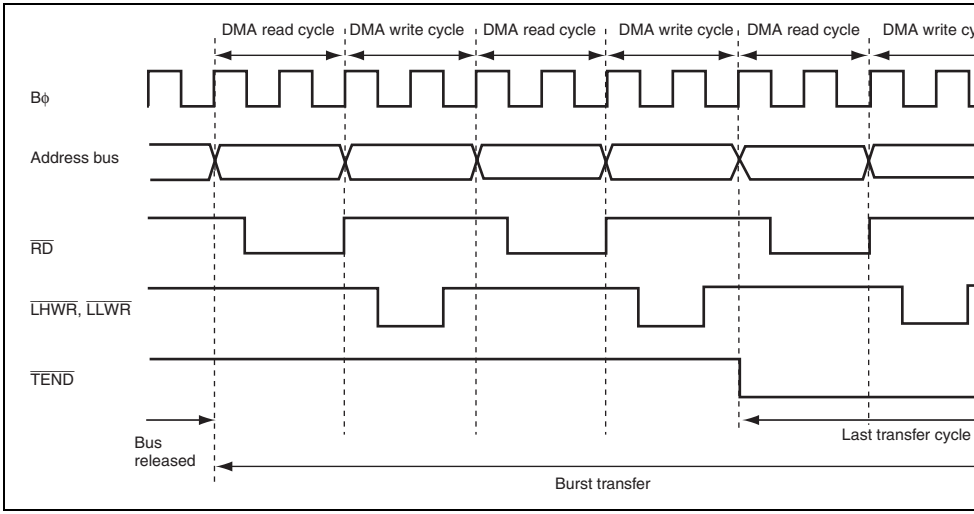


Figure 9.27 Example of Transfer in Normal Transfer Mode by Burst Acces

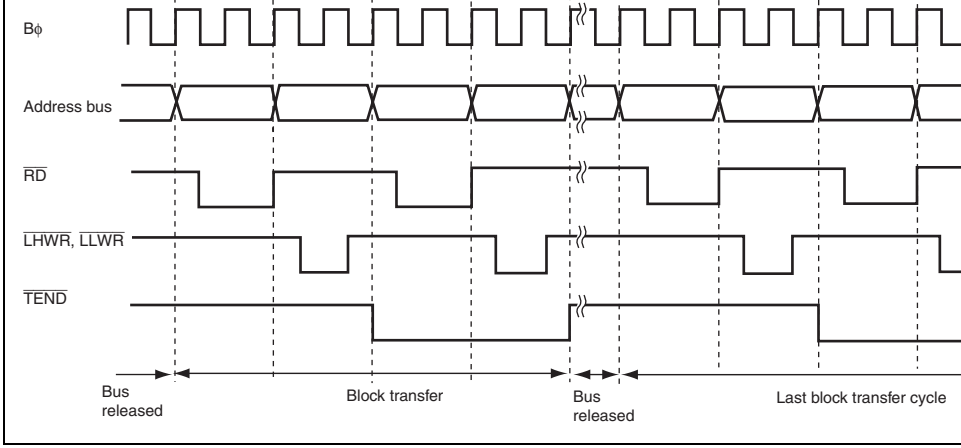
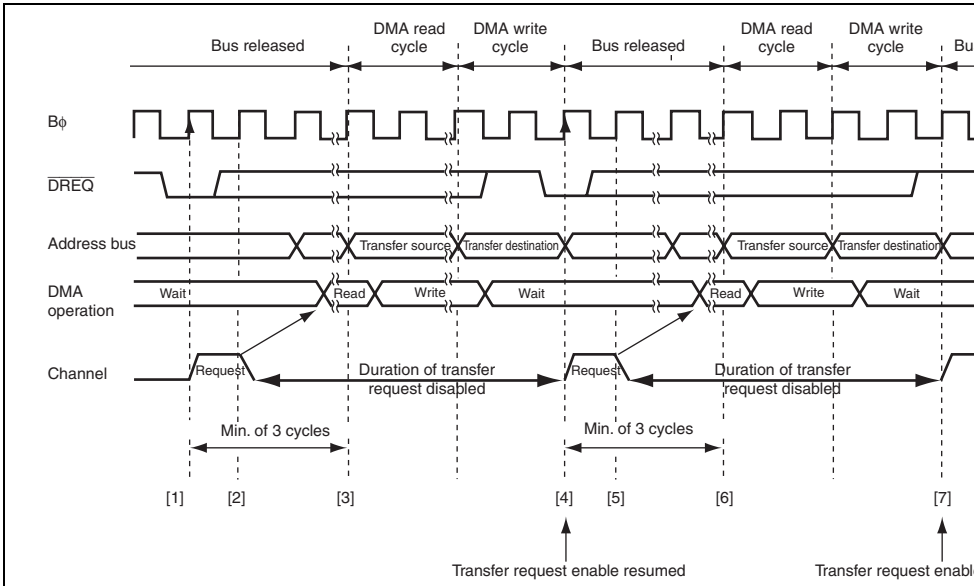


Figure 9.28 Example of Transfer in Block Transfer Mode

This operation is repeated until the transfer is completed.



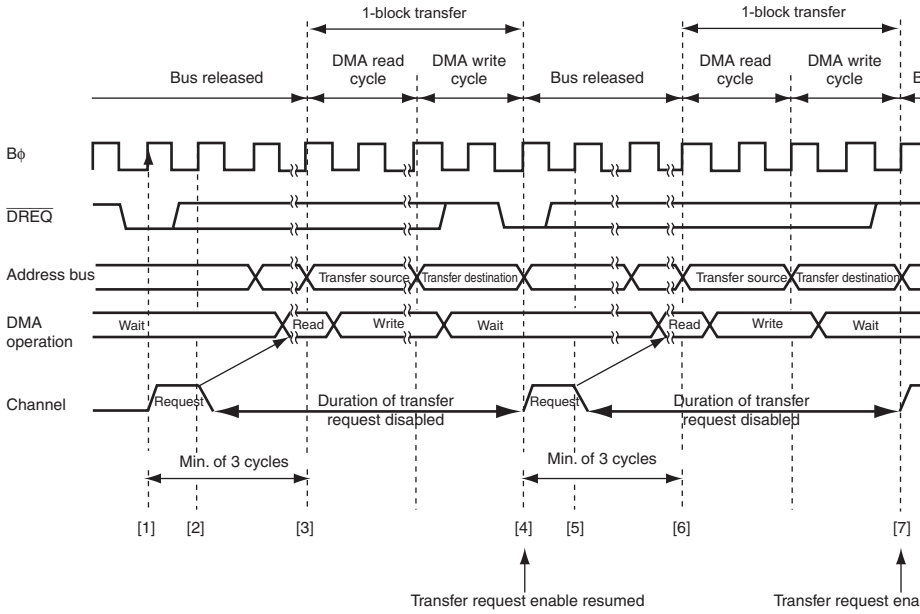
[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bφ signal and a request is held.

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

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

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

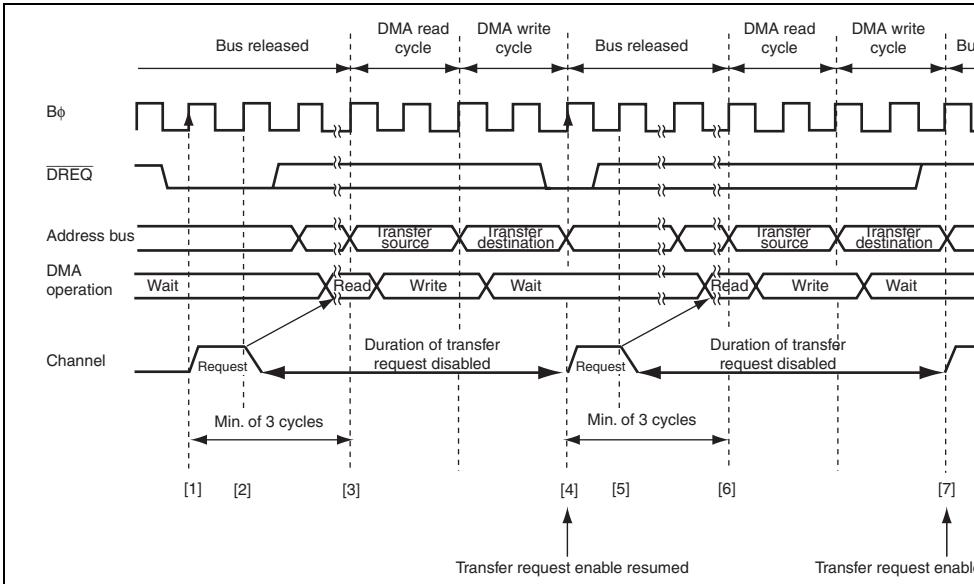
Figure 9.29 Example of Transfer in Normal Transfer Mode Activated by $\overline{\text{DREQ}}$ Falling Edge



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

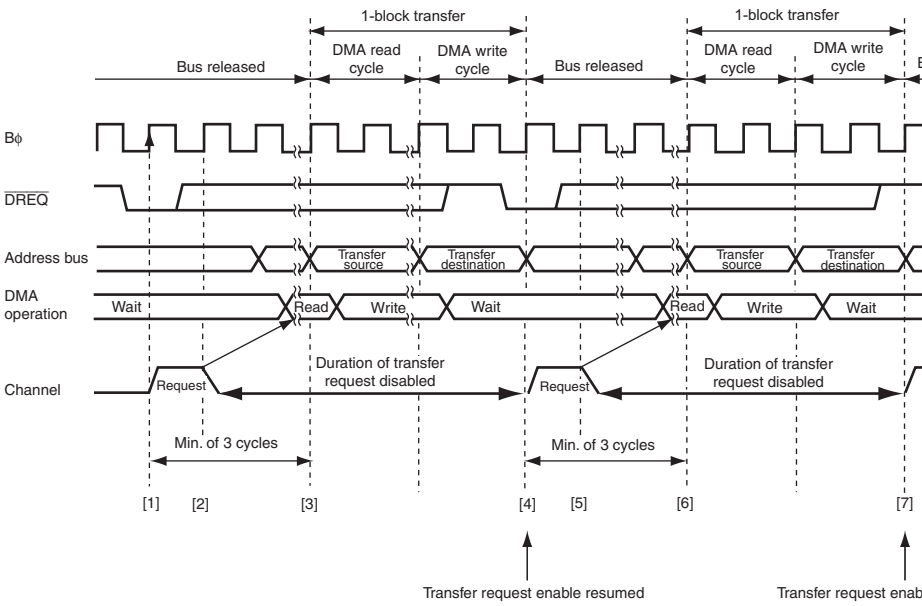
Figure 9.30 Example of Transfer in Block Transfer Mode Activated by $\overline{\text{DREQ}}$ Falling Edge

completed.



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

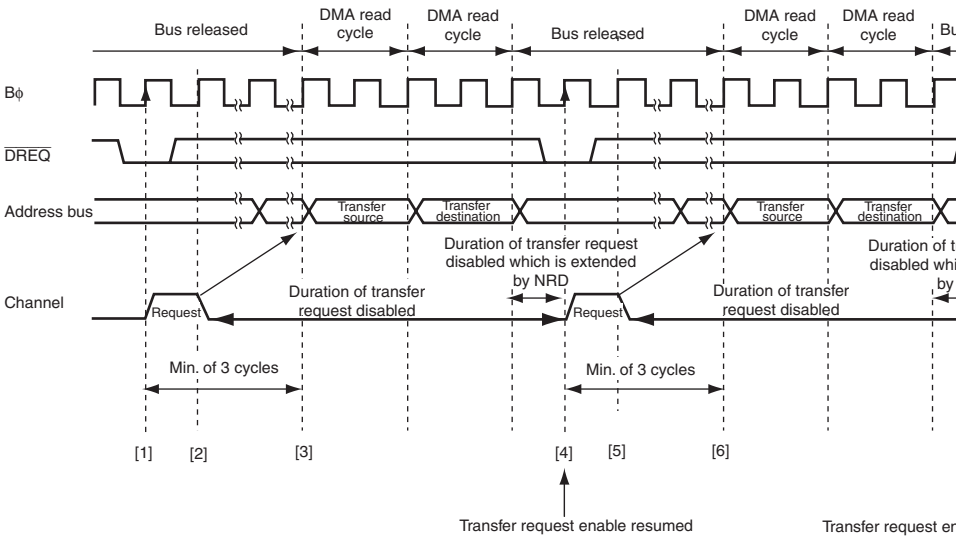
Figure 9.31 Example of Transfer in Normal Transfer Mode Activated by $\overline{\text{DREQ}}$ Low Level



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

Figure 9.32 Example of Transfer in Block Transfer Mode Activated by $\overline{\text{DREQ}}$ Low Level

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



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

Figure 9.33 Example of Transfer in Normal Transfer Mode Activated by $\overline{\text{DREQ}}$ Low Level with NRD = 1

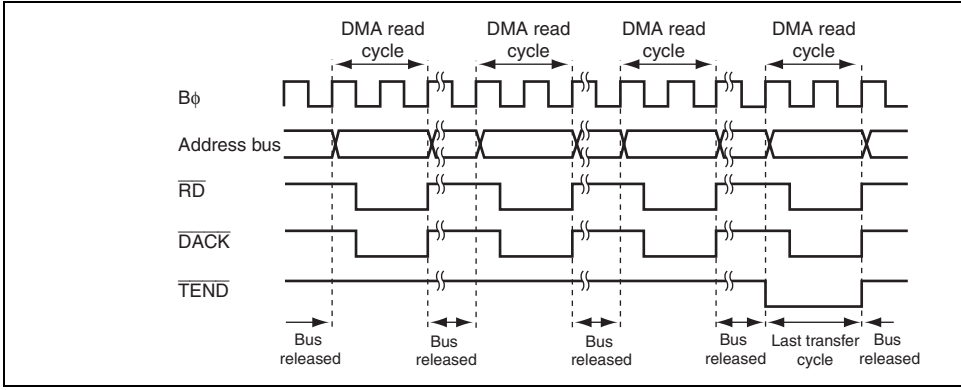


Figure 9.34 Example of Transfer in Single Address Mode (Byte Read)

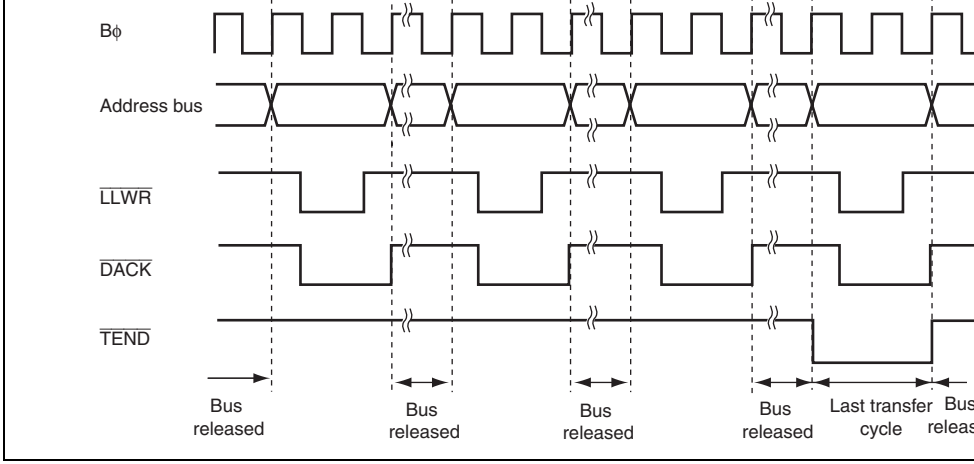
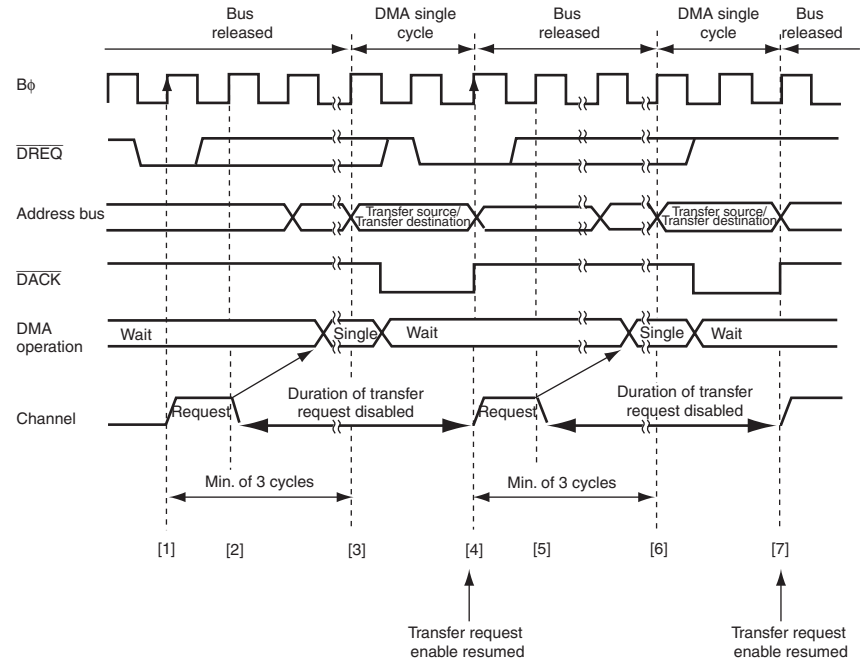


Figure 9.35 Example of Transfer in Single Address Mode (Byte Write)

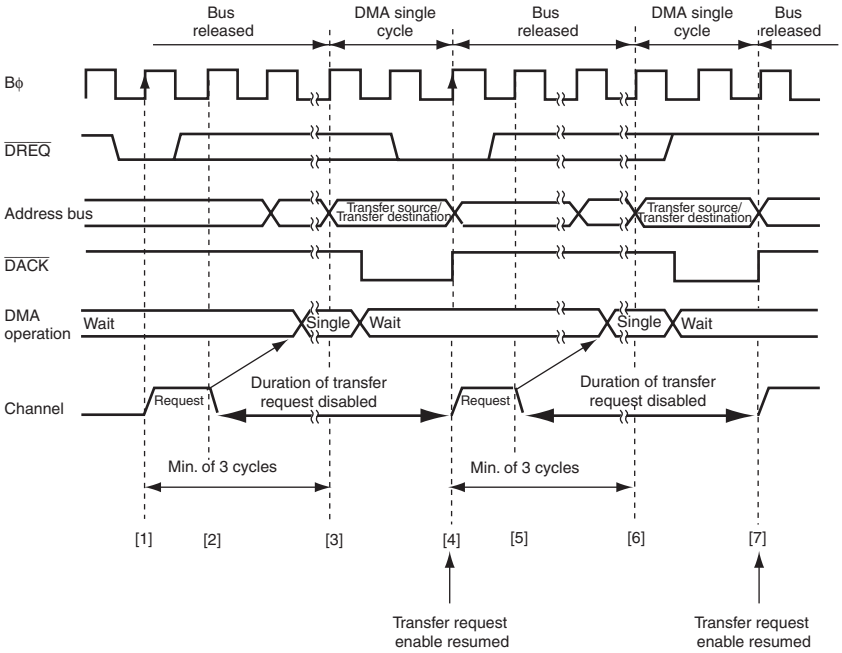
operation is repeated until the transfer is completed.



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

Figure 9.36 Example of Transfer in Single Address Mode Activated by DREQ Falling Edge

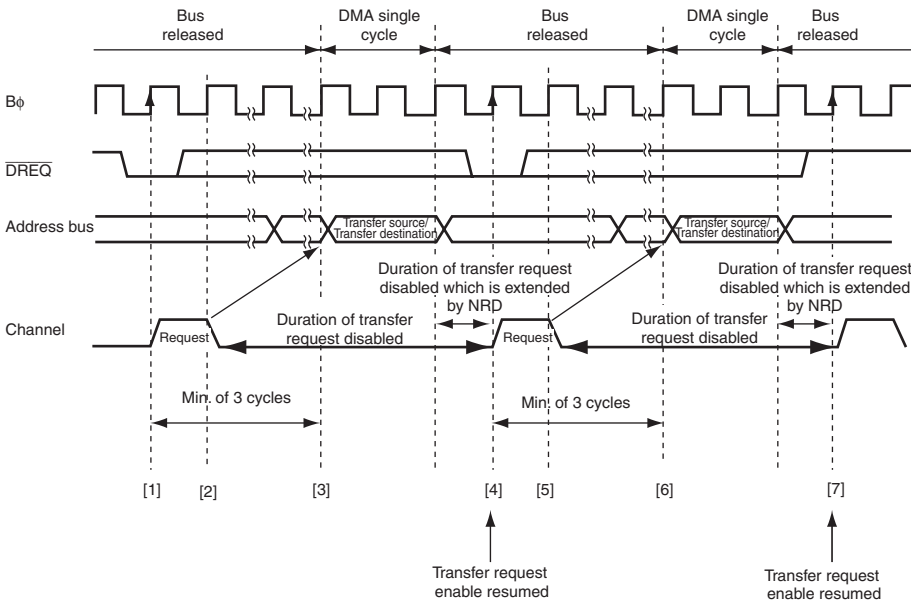
completed.



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

Figure 9.37 Example of Transfer in Single Address Mode Activated by $\overline{\text{DREQ}}$ Low Level

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



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

(2) Transfer End by Transfer Size Error Interrupt

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

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

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 8, Bus Controller (BSC).

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

In dual address mode, the DMAC releases the external bus after the external space write Since the read and write cycles are not separated, the bus is not released.

An internal space (on-chip memory and internal I/O registers) access of the DMAC and a external bus release cycle may be performed at the same time.

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	L

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 b
DMDR. A DMEEND interrupt is generated by the combination of the ESIF and ESIE bi
DMDR. The DMEEND interrupt sources are not distinguished. The priority among chan
decided by the interrupt controller and it is shown in table 9.7. For details, see section 6,
Controller.

An interrupt other than the transfer end interrupt by the transfer counter is generated when 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.

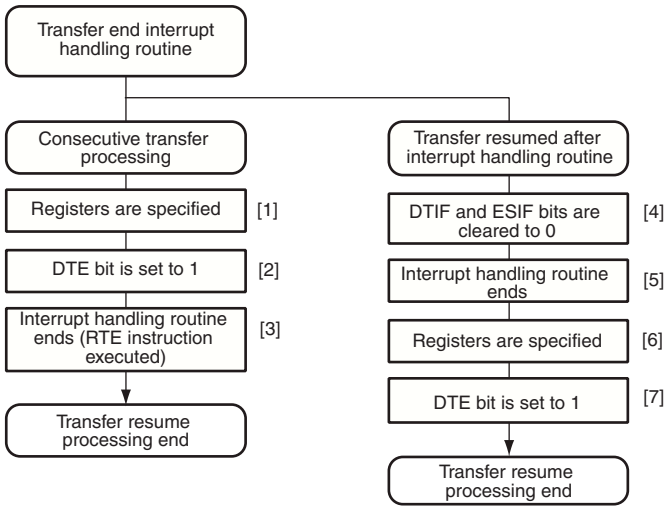
A transfer size error interrupt is generated when the next transfer cannot be performed be DTCR value is less than the data access size, meaning that the data access size of transfer be performed. In block transfer mode, the block size is compared with the DTCR value fo transfer error decision.

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 9.39 is a block diagram of interrupts and interrupt flags. To clear an interrupt, clea 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 9.40 shows procedure to re transfer by clearing a interrupt.

Figure 9.39 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 9.40 Procedure Example of Resuming Transfer by Clearing Interrupt S

The DMAC operation can be enabled or disabled by the module stop control register. DMAC is enabled by the initial value.

Setting bit MSTPA13 in MSTPCRA stops the clock supplied to the DMAC and the D 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 TEND signal output enabled)

— DACKE bit in DMDR is 1 (the DACK signal output enabled)

9.9.3 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 $\overline{\text{DREQ}}$ signa sampled by low level detection at the first activation after a DMAC transfer enabled.

- 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

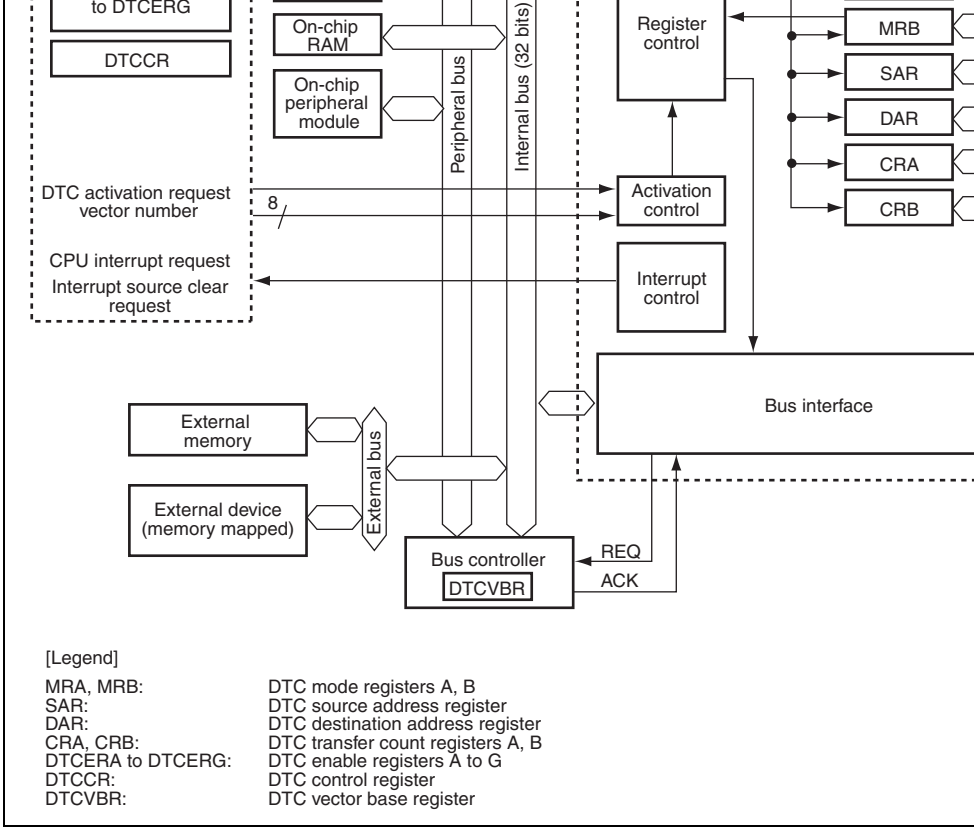
    The transfer source and destination addresses can be specified by 32 bits to selec Gbyte address space directly
- Size of data for data transfer can be specified as byte, word, or longword

  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 DTCERG:	DTC enable registers A to G
DTCCR:	DTC control register
DTCVBR:	DTC vector base register

Figure 10.1 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. W
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 G (DTCERA to DTCERG)
- DTC control register (DTCCR)
- DTC vector base register (DTCVBR)

Bit	Bit Name	Value	R/W	Description
7	MD1	Undefined	—	DTC Mode 1 and 0
6	MD0	Undefined	—	Specify DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
5	Sz1	Undefined	—	DTC Data Transfer Size 1 and 0
4	Sz0	Undefined	—	Specify the size of data to be transferred. 00: Byte-size transfer 01: Word-size transfer 10: Longword-size transfer 11: Setting prohibited
3	SM1	Undefined	—	Source Address Mode 1 and 0
2	SM0	Undefined	—	Specify an SAR operation after a data transfer. 0x: SAR is fixed (SAR writeback is skipped) 10: SAR is incremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10 11: SAR is decremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10
1, 0		Undefined		Reserved The write value should always be 0.

[Legend]
X: Don't care

Bit	Bit Name	Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable Specifies the chain transfer. For details, see s 10.5.7, 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 end 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 when Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]

X: Don't care

SAR cannot be accessed directly from the CPU.

10.2.4 DTC Destination Address Register (DAR)

DAR is a 32-bit register that designates the destination address of data to be transferred
DTC.

In full address mode, 32 bits of DAR are valid. In short address mode, the lower 24 bits
valid and bits 31 to 24 are ignored. At this time, the upper eight bits are filled with the v
bit 23.

If a word or longword access is performed while an odd address is specified in DAR or
longword access is performed while address 4n + 2 is specified in DAR, the bus cycle is
into multiple cycles to transfer data. For details, see section 10.5.1, Bus Cycle Division.

DAR cannot be accessed directly from the CPU.

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

In block transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and t
eight bits (CRAL). CRAH holds the block size while CRAL functions as an 8-bit block-s
counter (1 to 256 for byte, word, or longword). CRAL is decremented by 1 every time a b
(word or longword) data is transferred, and the contents of CRAH are sent to CRAL whe
count reaches H'00. The block size is 1 byte (word or longword) when CRAH = CRAL =
255 bytes (words or longwords) when CRAH = CRAL = H'FF, and 256 bytes (words or
longwords) when CRAH = CRAL =H'00.

CRA cannot be accessed directly from the CPU.

10.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by th
block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decreme
every time data is transferred, and bit DTCEn (n = 15 to 0) corresponding to the activatio
is cleared and then an interrupt is requested to the CPU when the count reaches H'0000. T
transfer count is 1 when CRB = H'0001, 65,535 when CRB = H'FFFF, and 65,536 when
H'0000.

CRB is not available in normal and repeat modes and cannot be accessed directly by the

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

Bit	7	6	5	4	3	2	1	
Bit Name	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
15	DTCE15	0	R/W	DTC Activation Enable 15 to 0
14	DTCE14	0	R/W	Setting this bit to 1 specifies a relevant interrup
13	DTCE13	0	R/W	a DTC activation source.
12	DTCE12	0	R/W	[Clearing conditions]
11	DTCE11	0	R/W	• When writing 0 to the bit to be cleared after
10	DTCE10	0	R/W	• When the DISEL bit is 1 and the data transf ended
9	DTCE9	0	R/W	
8	DTCE8	0	R/W	• When the specified number of transfers hav
7	DTCE7	0	R/W	These bits are not cleared when the DISEL bit i 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	

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 valu always be 0.
4	RRS	0	R/W	DTC Transfer Information Read Skip Enable Controls the vector address read and transfer information read. A DTC vector number is alwa compared with the vector number for the previ activation. If the vector numbers match and thi set to 1, the DTC data transfer is started witho reading a vector address and transfer informat previous DTC activation is a chain transfer, the address read and transfer information read are performed. 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 tran not occur when CRAL is 0. If this bit is set to 1, chain transfer is enabled when CRAH is written CRAL. 0: Disables the chain transfer after repeat trans 1: Enables the chain transfer after repeat trans

[Clearing condition]

- When writing 0 after reading 1

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

10.2.9 DTC Vector Base Register (DTCVBR)

DTCVBR is a 32-bit register that specifies the base address for vector table address calc
Bits 31 to 28 and bits 11 to 0 are fixed 0 and cannot be written to. The initial value of D
H'00000000.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
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

10.3 Activation Sources

The DTC is activated by an interrupt request. The interrupt source is selected by DTCE
activation source can be selected by setting the corresponding bit in DTCER; the CPU i
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 inter
corresponding DTCER bit is cleared.

activation source, and then reads the transfer information from the start address. Figure 10
correspondences between the DTC vector address and transfer information.

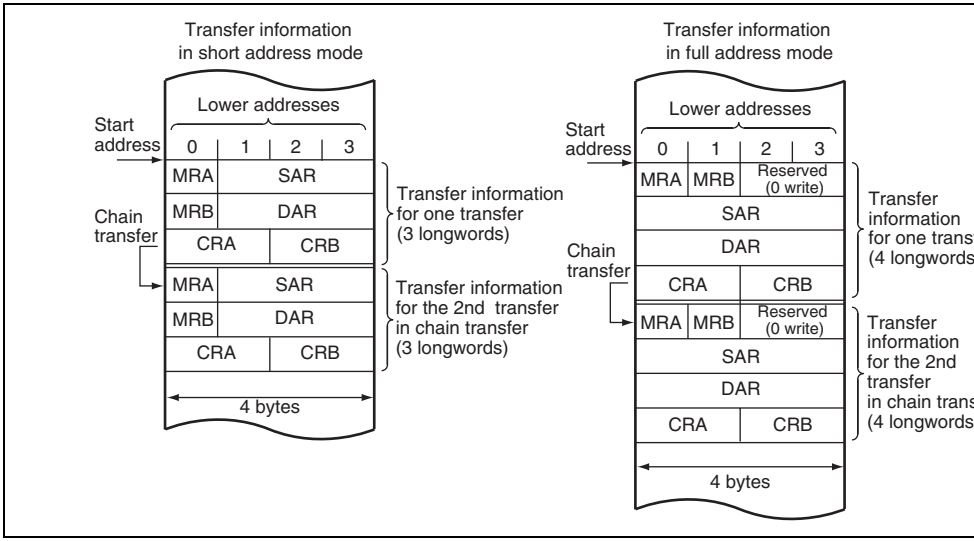


Figure 10.2 Transfer Information on Data Area

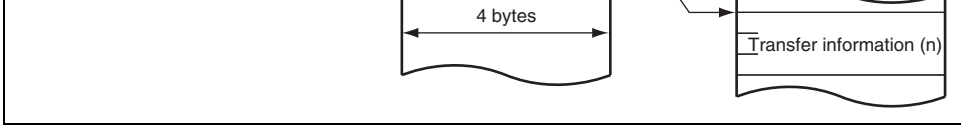


Figure 10.3 Correspondence between DTC Vector Address and Transfer Infor

	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	
	IRQ12	76	H'0530	DTCEA3	
	IRQ13	77	H'0534	DTCEA2	
	IRQ14	78	H'0538	DTCEA1	
	IRQ15	79	H'053C	DTCEA0	
TPU_0	TGI0A	88	H'560	DTCEB13	
	TGI0B	89	H'564	DTCEB12	
	TGI0C	90	H'568	DTCEB11	
	TGI0D	91	H'56C	DTCEB10	
TPU_1	TGI1A	93	H'574	DTCEB9	
	TGI1B	94	H'578	DTCEB8	
TPU_2	TGI2A	97	H'584	DTCEB7	
	TGI2B	98	H'588	DTCEB6	
TPU_3	TGI3A	101	H'594	DTCEB5	
	TGI3B	102	H'598	DTCEB4	
	TGI3C	103	H'59C	DTCEB3	
	TGI3D	104	H'5A0	DTCEB2	Lo

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
DMAC	DMEEND0	136	H'620	DTCED13
	DMEEND1	137	H'624	DTCED12
SCI_0	RXI0	145	H'644	DTCED5
	TXI0	146	H'648	DTCED4
SCI_1	RXI1	149	H'654	DTCED3
	TXI1	150	H'658	DTCED2
SCI_2	RXI2	153	H'664	DTCED1
	TXI2	154	H'668	DTCED0
SCI_3	RXI3	157	H'674	DTCEE15
	TXI3	158	H'678	DTCEE14
SCI_4	RXI4	161	H'684	DTCEE13
	TXI4	162	H'688	DTCEE12

	CMI7B	207	H'F748	DTCEG12	EG

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 10.2 shows the DTC transfer modes.

Table 10.2 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 10.4 shows a flowchart of DTC operation, and table 10.3 summarizes the chain tr conditions (combinations for performing the second and third transfers are omitted).

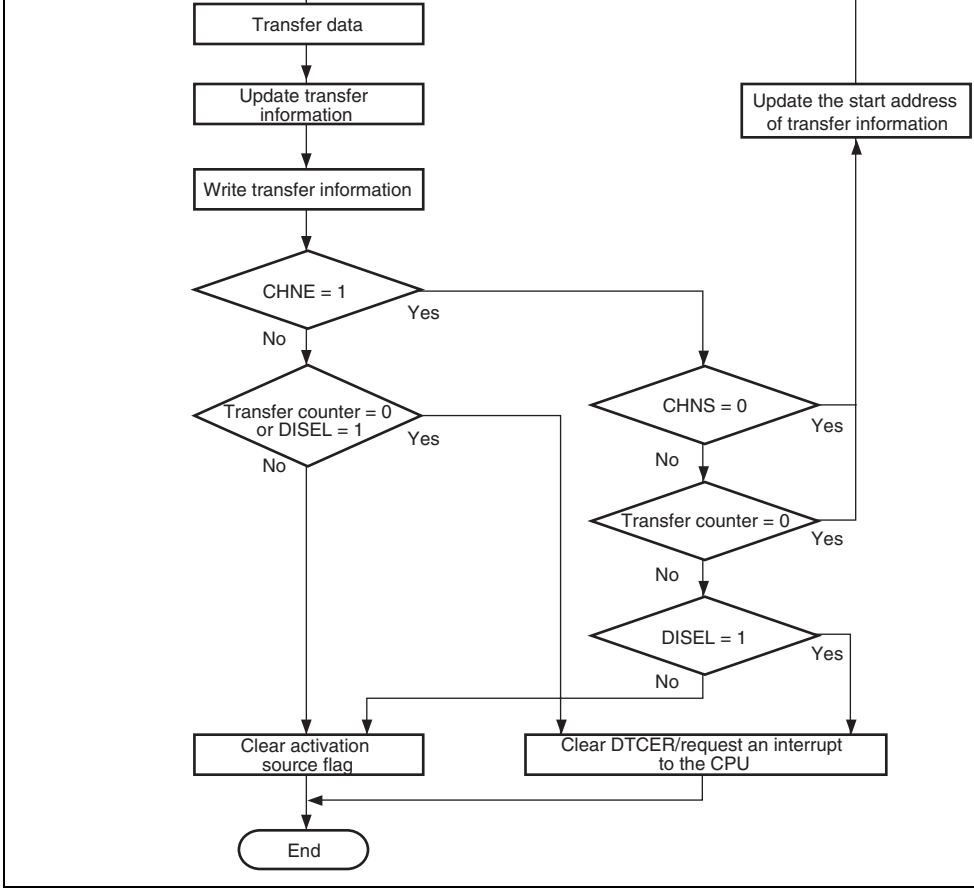


Figure 10.4 Flowchart of DTC Operation

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 mod

10.5.1 Bus Cycle Division

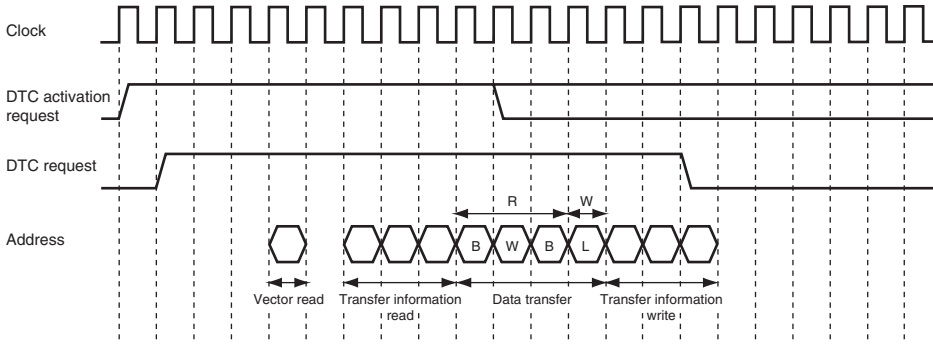
When the transfer data size is word and the SAR and DAR values are not a multiple of 2 cycle is divided and the transfer data is read from or written to in bytes. Similarly, when transfer data size is longword and the SAR and DAR values are not a multiple of 4, the b is divided and the transfer data is read from or written to in words.

Table 10.4 shows the relationship among, SAR, DAR, transfer data size, bus cycle divis access data size. Figure 10.5 shows the bus cycle division example.

Table 10.4 Number of Bus Cycle Divisions and Access Size

	Specified Data Size		
SAR and DAR Values	**Byte (B)**	**Word (W)**	**Longword (L**
Address 4n	1 (B)	1 (W)	1 (LW)
Address 2n + 1	1 (B)	2 (B-B)	3 (B-W-B)
Address 4n + 2	1 (B)	1 (W)	2 (W-W)

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



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

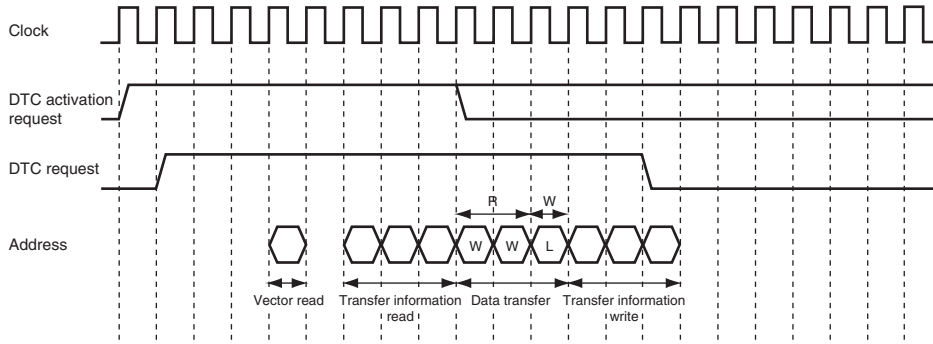


Figure 10.5 Bus Cycle Division Example

cleared to 0, the stored vector number is deleted, and the updated vector table and transf
information are read at the next activation.

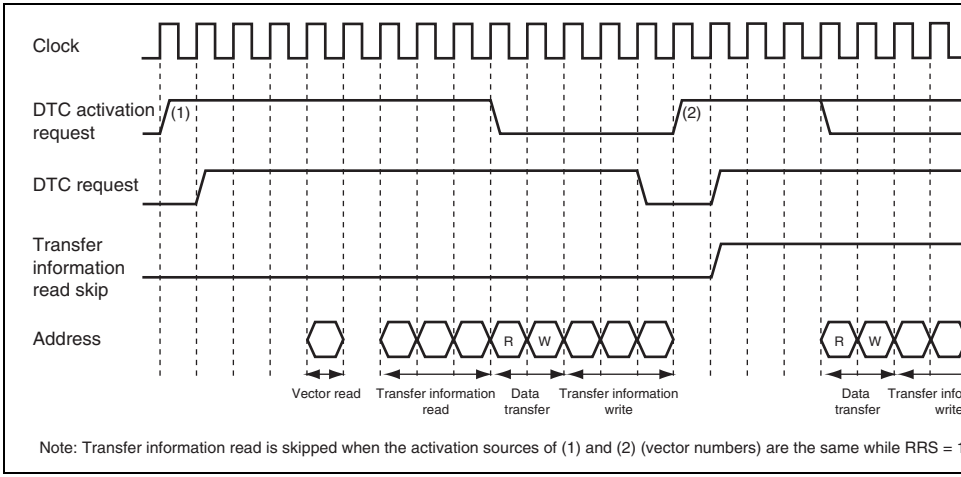


Figure 10.6 Transfer Information Read Skip Timing

SM1	DM1	SAR	DAR
0	0	Skipped	Skipped
0	1	Skipped	Written back
1	0	Written back	Skipped
1	1	Written back	Written back

10.5.4 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 addresses specified as incremented, decremented, or fixed. When the specified number of transfers interrupt can be requested to the CPU.

Table 10.6 lists the register function in normal transfer mode. Figure 10.7 shows the mem in normal transfer mode.

Table 10.6 Register Function in Normal Transfer Mode

Register	Function	Written Back Value
SAR	Source address	Incremented/decremented/fixed
DAR	Destination address	Incremented/decremented/fixed
CRA	Transfer count A	CRA – 1
CRB	Transfer count B	Not updated

Note: * Transfer information writeback is skipped.

Figure 10.7 Memory Map in Normal Transfer Mode

10.5.5 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 transfe repeated. The other address register is then incremented, decremented, or left fixed. In re transfer mode, the transfer counter (CRAL) is updated to the value specified in CRAH v CRAL becomes H'00. Thus the transfer counter value does not reach H'00, and therefore interrupt cannot be requested when DISEL = 0.

Table 10.7 lists the register function in repeat transfer mode. Figure 10.8 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.

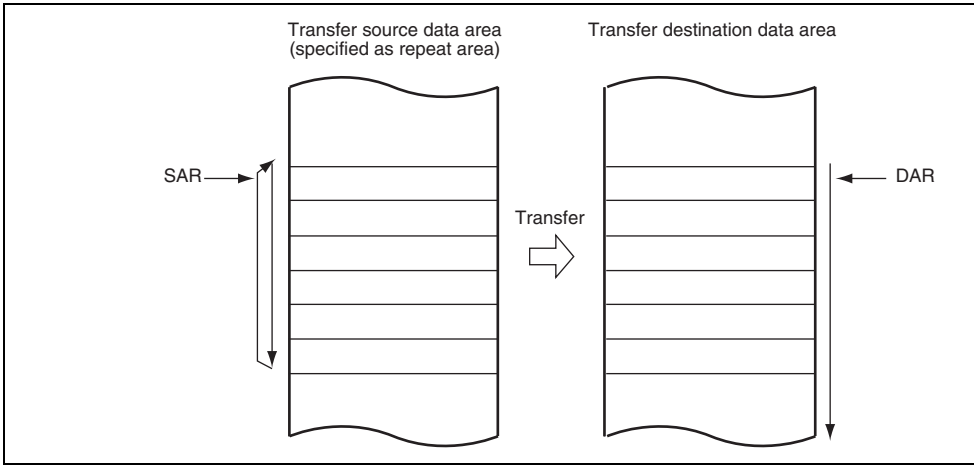


Figure 10.8 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

Table 10.8 lists the register function in block transfer mode. Figure 10.9 shows the mem
in block transfer mode.

Table 10.8 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.

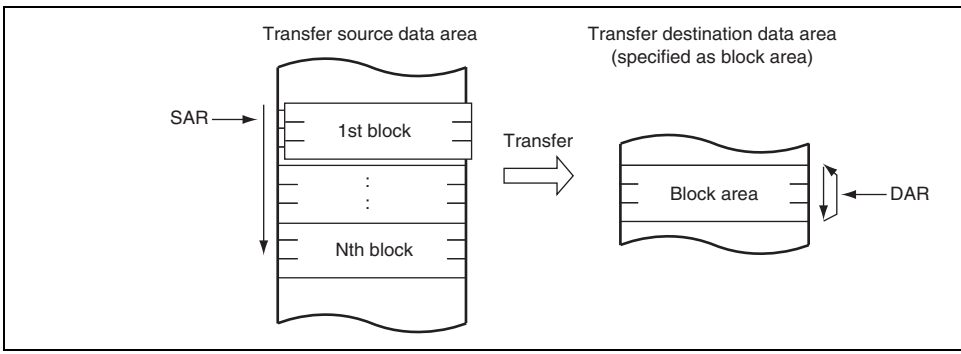


**Figure 10.9 Memory Map in Block Transfer Mode
(When Transfer Destination is Specified as Block Area)**

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.

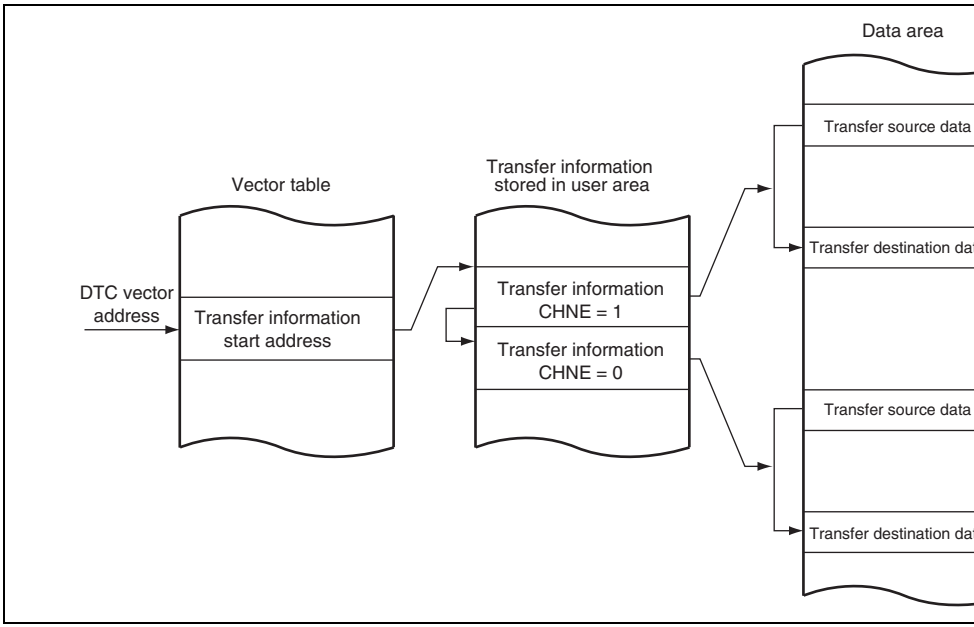


Figure 10.10 Operation of Chain Transfer

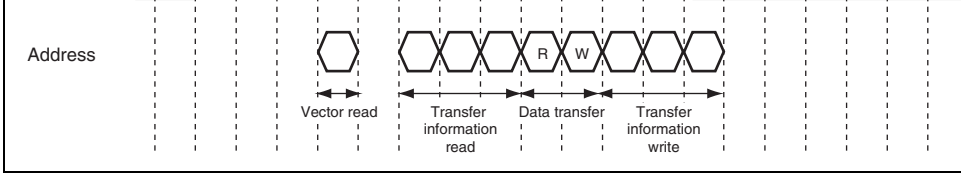


Figure 10.11 DTC Operation Timing
(Example of Short Address Mode in Normal Transfer Mode or Repeat Transfer

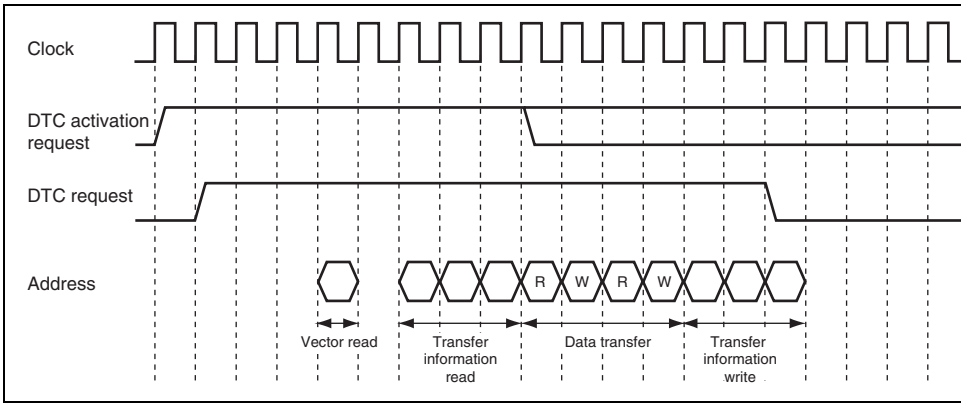


Figure 10.12 DTC Operation Timing
(Example of Short Address Mode in Block Transfer Mode with Block Size o

Figure 10.13 DTC Operation Timing (Example of Short Address Mode in Chain T

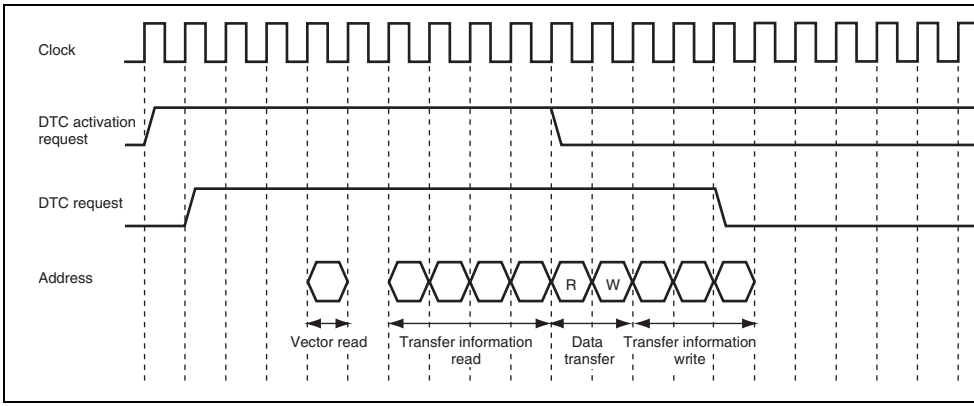


Figure 10.14 DTC Operation Timing
(Example of Full Address Mode in Normal Transfer Mode or Repeat Transfer M

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 addr register
7. When a word is transferred while an odd address is specified in the address r when a longword is transferred while address 4n + 2 is specified

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 8, Bus Controller (BSC).)

The number of execution cycles is calculated from the formula below. Note that Σ means
of all transfers activated by one activation event (the number in which the CHNE bit is se
plus 1).

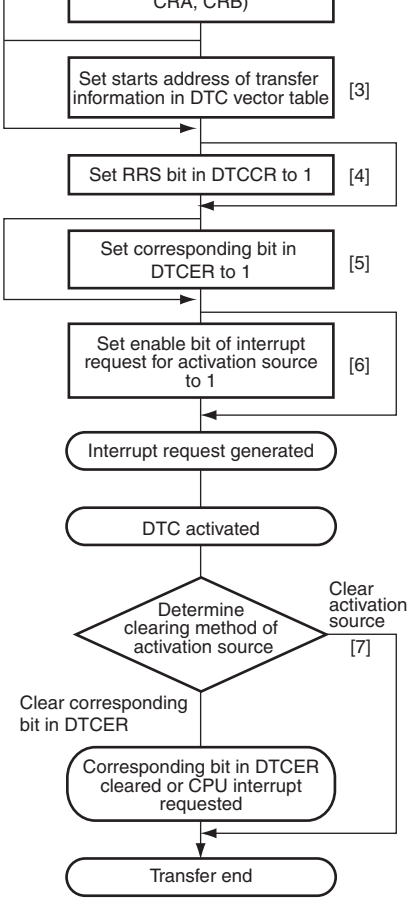
$$\text{Number of execution cycles} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L + M \cdot S_M) + N \cdot S$$

10.5.10 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.

10.5.11 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 6, Interrupt Controller.



information in the data area. For details on setting tra information, see section 10.2, Register Descriptions. F on location of transfer information, see section 10.4, 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 10.4, Location of Transfer Information and DTC Vector

[4] Setting the RRS bit to 1 performs a read skip of secon later transfer information when the DTC is activated co 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 10.1. 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 used activation sources to 1. The DTC is activated when an used as an activation source is generated. For details settings of the interrupt enable bits, see the correspon descriptions of the corresponding module.

[7] After the end of one data transfer, the DTC clears the source flag or clears the corresponding bit in DTCER requests an interrupt to the CPU. The operation after t depends on the transfer information. For details, see s 10.2, Register Descriptions and figure 10.4.

Figure 10.15 DTC with Interrupt Activation

the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any va

2. Set the start address of the transfer information for an RXI interrupt at the DTC vecto
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the end (RXI) interrupt. Since the generation of a receive error during the SCI reception c will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set RXI interrupt is generated, and the DTC is activated. The receive data is transferred fr to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Terminatio processing should be performed in the interrupt handling routine.

10.7.2 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 compare be used as the output trigger.
8. Set the CST bit in TSTR to 1, and start the TCNT count operation.
9. Each time a TGRA compare match occurs, the next output value is transferred to ND set value of the next output trigger period is transferred to TGRA. The activation sou flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is CPU. Termination processing should be performed in the interrupt handling routine.

10.7.3 Chain Transfer when Counter = 0

By executing a second data transfer and performing re-setting of the first data transfer or 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 to have been set to start at lower address H'0000. Figure 10.16 shows the chain transfer counter value is 0.

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.

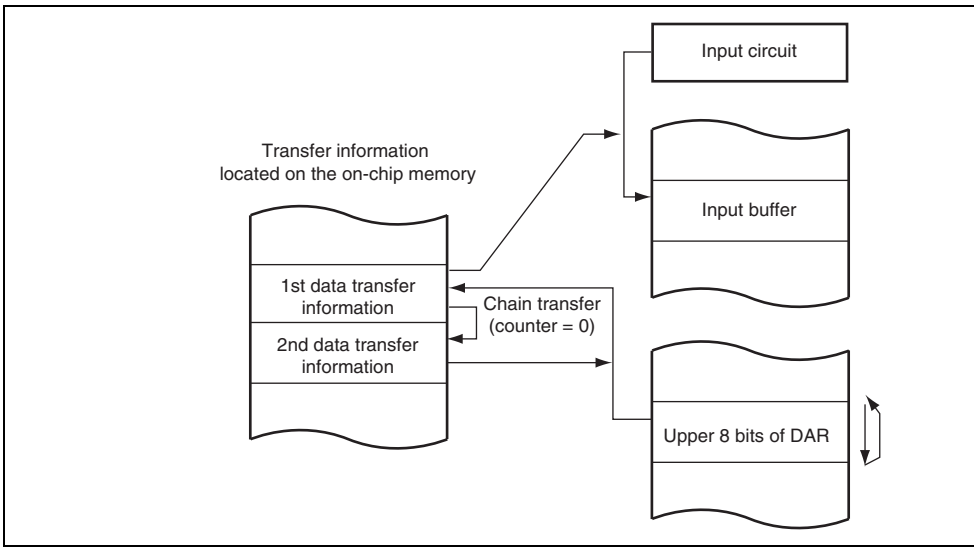


Figure 10.16 Chain Transfer when Counter = 0

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 24, Power-Down Modes.

10.9.2 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.

10.9.3 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.

10.9.4 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.

10.9.6 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.

10.9.7 Transfer Information Modification

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

10.9.8 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.

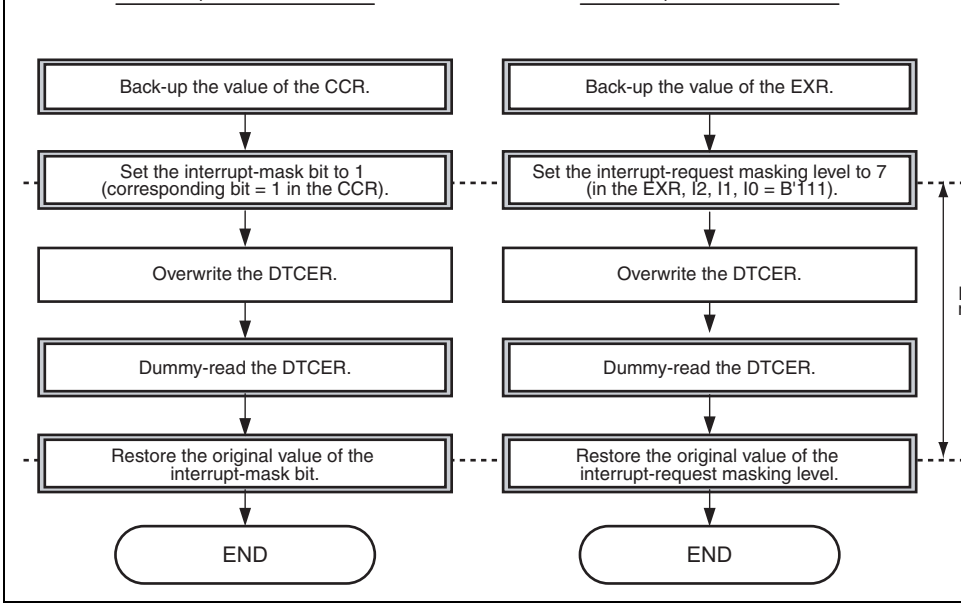


Figure 10.17 Example of Procedures for Overwriting DTCER

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 with a capacitive component of up to 30
drive Darlington transistors when functioning as output ports.

Port 2 have pins for Schmitt-trigger inputs. Schmitt-trigger input is enabled for pins of o
when they are used as IRQ, TPU, TMR, or IIC2 inputs.

converter input, TPU input, and IIC2 I/O	5	P15/SCL1	$\overline{\text{IRQ5}}$-A/ TCLKB-B/ RxD3	$\overline{\text{TEND1}}$-A	$\overline{\text{IRQ5}}$-A, TCLKB-B, SCL1
	4	P14/SDA1	$\overline{\text{DREQ1}}$-A/ $\overline{\text{IRQ4}}$-A/ TCLKA-B	TxD3	$\overline{\text{IRQ4}}$-A, TCLKA-B, SDA1
	3	P13	$\overline{\text{ADTRG0}}$/ $\overline{\text{IRQ3}}$-A	—	$\overline{\text{IRQ3}}$-A
	2	P12/SCK2	$\overline{\text{IRQ2}}$-A	$\overline{\text{DACK0}}$-A	$\overline{\text{IRQ2}}$-A
	1	P11	RxD2/$\overline{\text{IRQ1}}$-A	$\overline{\text{TEND0}}$-A	$\overline{\text{IRQ1}}$-A
	0	P10	$\overline{\text{DREQ0}}$-A/$\overline{\text{IRQ0}}$-A	TxD2	$\overline{\text{IRQ0}}$-A

4	P24/TIOCB4/SCK1	TIOCA4/TMRI1/ $\overline{\text{IRQ12}}$-A	PO4	P24, TIOCB4, TIOCA4, TMRI1, $\overline{\text{IRQ12}}$-A
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

		3	P33/TIOCD0	TIOCC0/ TCLKB-A/ DREQ1-B	PO11/ CS3 /CS7-A	P33, TIOCD0, TIOCC0, TCKB-A	
		2	P32/TIOCC0	TCLKA-A	PO10/DACK0-B/ CS2-A/CS6-A	P32, TIOCC0, TCLKA-A	
		1	P31/TIOCB0	TIOCA0	PO9/TEND0-B/ CS1/CS2-B/ CS5-A/CS6-B/CS7-B	P31, TIOCB0, TIOCAO	
		0	P30/TIOCA0	DREQ0-B	PO8/CS0/ CS4/CS5-B	P30, TIOCA0	
Port 4	General input port	7	—	P47	—	—	—
		6	—	P46	—	—	
		5	—	P45	—	—	
		4	—	P44	—	—	
		3	—	P43	—	—	
		2	—	P42	—	—	
		1	—	P41	—	—	
		0	—	P40	—	—	

	output						
		1	—	P51/AN1/IRQ1-B	—	IRQ1-B	
		0	—	P50/AN0/IRQ0-B	—	IRQ0-B	
Port 6	General I/O port function multiplexed with TMR I/O, SCI I/O, H-UDI input, and interrupt input	7	—	—	—	—	—
		6	—	—	—	—	
		5	P65	IRQ13-B	TMO3	IRQ13-B	
		4	P64	TMCI3/ IRQ12-B	—	TMCI3, IRQ12-B	
		3	P63	TMRI3/ IRQ11-B	—	TMRI3, IRQ11-B	
		2	P62/SCK4	IRQ10-B	TMO2	IRQ10-B	
		1	P61	TMCI2/RxD4/ IRQ9-B	—	TMCI2, IRQ9-B	
		0	P60	TMRI2/ IRQ8-B	TxD4	TMRI2, IRQ8-B	
Port A	General I/O port function multiplexed with system clock output and bus control I/O	7	—	PA7	Bϕ	—	—
		6	PA6	—	AS/AH/BS-B		
		5	PA5	—	RD		
		4	PA4		LHWR/LUB		
		3	PA3		LLWR/LLB		
		2	PA2	BREQ/WAIT	—		
		1	PA1		BACK/ (RD/WR)		
		0	PA0		BREQO/BS-A		

		1	PD1	—	A1	—	
		0	PD0	—	A0	—	
Port E	General I/O port function multiplexed with address output	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		
Port F	General I/O port function multiplexed with address output	7	—	—	—	—	O
		6	—	—	—		
		5	—	—	—		
		4	PF4	—	A20		
		3	PF3	—	A19		
		2	PF2	—	A18		
		1	PF1	—	A17		
		0	PF0	—	A16		

		1	PH1/D1*2	—	—		
		0	PH0/D0*2	—	—		
Port I	General I/O port function multiplexed with bi-directional data bus	7	PI7/D15*2	—	TMO7	—	O
		6	PI6/D14*2	—	TMO6		
		5	PI5/D13*2	—	TMO5		
		4	PI4/D12*2	—	TMO4		
		3	PI3/D11*2	—	—		
		2	PI2/D10*2	—	—		
		1	PI1/D9*2	—	—		
		0	PI0/D8*2	—	—		

Notes: 1. Pins without Schmitt-trigger input buffer have CMOS input buffer.
2. Addresses are also output when accessing to the address/data multiplexed I/O space.

Port 3	8	O	O	O	O	—	—
Port 4	8	—	—	O	O	—	—
Port 5	8	—	—	O	O	—	—
Port 6*[1]	6	O	O	O	O	—	—
Port A	8	O	O	O	O	—	—
Port D	8	O	O	O	O	O	—
Port E	8	O	O	O	O	O	—
Port F*[2]	5	O	O	O	O	O	O
Port H	8	O	O	O	O	O	—
Port I	8	O	O	O	O	O	—

[Legend]

O: Register exists

—: No register exists

Notes: 1. The lower six bits are valid and the upper two bits are reserved. The write valu always be the initial value.

2. The lower five bits are valid and the upper three bits are reserved. The write va should always be the initial value.

Bit	7	6	5	4	3	2	1	
Bit Name	Pn7DDR	Pn6DDR	Pn5DDR	Pn4DDR	Pn3DDR	Pn2DDR	Pn1DDR	
Initial Value	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower five bits are valid and the upper three bits are reserved for port F registers.

Table 11.3 Startup Mode and Initial Value

	Startup Mode	
Port	**External Extended Mode**	**Single-Chip Mode**
Port A	H'80	H'00
Other ports	H'00	

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower five bits are valid and the upper three bits are reserved for port F registers.

11.1.3 Port Register (PORTn) (n = 1 to 6, A, D to F, H, and I)

PORT is an 8-bit read-only register that reflects the port pin status. A write to PORT is in

When PORT is read, the DR bits that correspond to the respective DDR bits set to 1 are r
the status of each pin whose corresponding DDR bit is cleared to 0 is also read regardless
ICR value.

The initial value of PORT is undefined and is determined based on the port pin status.

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	U
R/W	R	R	R	R	R	R	R	

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower five bits are valid and the upper three bits are reserved for port F registers.

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

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

If ICR is modified, an internal edge may occur depending on the pin status. Accordingly should be modified when the corresponding input pins are not used. For example, in $\overline{\text{IRQ}}$ 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 five bits are valid and the upper three bits are reserved for port F registers.

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

	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	Peripheral module output	OFF	
	Data input/output	OFF	
	Port output	OFF	
	Port input	OFF	ON/OFF

[Legend]

OFF: The input pull-up MOS is always off.

ON/OFF: If PCR is set to 1, the input pull-up MOS is on; if PCR is cleared to 0, the inpu MOS is off.

Bit Name	Pn7ODR	Pn6ODR	Pn5ODR	Pn4ODR	Pn3ODR	Pn2ODR	Pn1ODR	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: The lower five bits are valid and the upper three bits are reserved for port F registers.

For a pin whose initial value changes according to the activation mode, "Initial value E" the initial value when the LSI is started up in external extended mode and "Initial value indicates the initial value when the LSI is started in single-chip mode.

11.2.1 Port 1

(1) P17/$\overline{\text{ANDSTRG}}$/$\overline{\text{IRQ7}}$-A/TCLKD-B/SCL0

The pin function is switched as shown below according to the combination of the IIC2 r P17DDR bit settings.

		Setting	
		IIC2	I/O Port
Module Name	Pin Function	SCL0_OE	P17DDR
IIC2	SCL0 I/O	1	—
I/O port	P17 output	0	1
	P17 input (initial value)	0	0

SCI	SCK3 I/O	0	0	1	—
I/O port	P16 output	0	0	0	1
	P16 input (initial value)	0	0	0	0

(3) P15/RxD3/$\overline{\text{TEND1}}$-A/$\overline{\text{IRQ5}}$-A/TCLKB-B/SCL1

The pin function is switched as shown below according to the combination of the DMAC register settings and P15DDR bit setting.

		Setting		
		DMAC	IIC2	I/O Port
Module Name	Pin Function	$\overline{\text{TEND1A}}$_OE	SCL1_OE	P15DDR
DMAC	$\overline{\text{TEND1}}$-A output	1	—	—
IIC2	SCL1 I/O	0	1	—
I/O port	P15 output	0	0	1
	P15 input (initial value)	0	0	0

I/O port	P14 output	0	0	1
	P14 input (initial value)	0	0	0

(5) P13/$\overline{\text{ADTRG0}}$/$\overline{\text{IRQ3}}$-A

The pin function is switched as shown below according to the P13DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	P13DDR
I/O port	P13 output	1
	P13 input (initial value)	0

I/O port	P12 output	0	0	1
	P12 input (initial value)	0	0	0

(7) P11/RxD2/$\overline{\text{TEND0}}$-A/$\overline{\text{IRQ1}}$-A

The pin function is switched as shown below according to the combination of the DMAC setting and P11DDR bit setting.

		Setting	
		DMAC	I/O Port
Module Name	Pin Function	$\overline{\text{TEND0A}}$_OE	P11DDR
DMAC	$\overline{\text{TEND0}}$-A output	1	—
I/O port	P11 output	0	1
	P11 input (initial value)	0	0

	P10 input (initial value)	0		0

11.2.2 Port 2

(1) P27/PO7/TIOCA5/TIOCB5/$\overline{\text{IRQ15}}$

The pin function is switched as shown below according to the combination of the TPU a register settings and P27DDR bit setting.

		Setting		
		TPU	PPG	I/O 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	1
	P26 input (initial value)	0	0	0	0	0

(3) P25/PO5/TIOCA4/TMCI1/RxD1/$\overline{\text{IRQ13}}$-A

The pin function is switched as shown below according to the combination of the TPU ar 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 a 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	1
	P22 input (initial value)	0	0	0	0	0

(7) P21/PO1/TIOCA3/TMCI0/RxD0/$\overline{\text{IRQ9}}$-A

The pin function is switched as shown below according to the combination of the TPU ar 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

11.2.3 Port 3

(1) P37/PO15/TIOCA2/TIOCB2/TCLKD-A

The pin function is switched as shown below according to the combination of the TPU a register settings and P37DDR bit setting.

		Setting		
		TPU	PPG	I/O Port
Module Name	Pin Function	TIOCB2_OE	PO15_OE	P37DDR
TPU	TIOCB2 output	1	—	—
PPG	PO15 output	0	1	—
I/O port	P37 output	0	0	1
	P37 input (initial value)	0	0	0

I/O port	P36 output	0	0	1	
	P36 input (initial value)	0	0	0	

(3) P35/PO13/TIOCA1/TIOCB1/TCLKC-A/$\overline{\text{DACK1}}$-B

The pin function is switched as shown below according to the combination of the DMAC and PPG register settings and P35DDR bit setting.

		Setting			
		DMAC	TPU	PPG	I/O P
Module Name	Pin Function	$\overline{\text{DACK1B}}$_OE	TIOCB1_OE	PO13_OE	P35D
DMAC	$\overline{\text{DACK1}}$-B output	1	—	—	—
TPU	TIOCB1 output	0	1	—	—
PPG	PO13 output	0	0	1	—
I/O port	P35 output	0	0	0	1
	P35 input (initial value)	0	0	0	0

PPG	PO12 output	0	0	1	—
I/O port	P34 output	0	0	0	1
	P34 input (initial value)	0	0	0	0

(5) P33/PO11/TIOCC0/TIOCD0/TCLKB-A/$\overline{\text{DREQ1}}$-B/$\overline{\text{CS3}}$/$\overline{\text{CS7}}$-A

The pin function is switched as shown below according to the combination of the TPU a register settings and P33DDR bit setting.

		Setting				
		I/O Port		TPU	PPG	I/
Module Name	**Pin Function**	**$\overline{\text{CS3}}$_OE**	**$\overline{\text{CS7A}}$_OE**	**TIOCD0_OE**	**PO11_OE**	**P**
Bus controller	$\overline{\text{CS3}}$ output*	1	—	—	—	–
	$\overline{\text{CS7}}$A output*	—	1	—	—	–
TPU	TIOCD0 output	0	0	1	—	–
PPG	PO11 output	0	0	0	1	–
I/O port	P33 output	0	0	0	0	1
	P33 input (initial value)	0	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

DMAC	DACK0-B output	0	0	1	—	—
TPU	TIOCC0 output	0	0	0	1	—
PPG	PO10 output	0	0	0	0	1
I/O port	P32 output	0	0	0	0	0
	P32 input (initial value)	0	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

	output*								
	$\overline{CS5}$-A output*	—	—	1	—	—	—	—	—
	$\overline{CS6}$-B output*	—	—	—	1	—	—	—	—
	$\overline{CS7}$-B output*	—	—	—	—	1	—	—	—
DMAC	$\overline{TEND0}$-B output	0	0	0	0	0	1	—	—
TPU	TIOCB0 output	0	0	0	0	0	0	1	—
PPG	PO9 output	0	0	0	0	0	0	0	1
I/O port	P31 output	0	0	0	0	0	0	0	0
	P31 input (initial value)	0	0	0	0	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

	$\overline{CS5}$-B output*	—	—	1	—	—	–
TPU	TIOCA0 output	0	0	0	1	—	–
PPG	PO8 output	0	0	0	0	1	–
I/O port	P30 output	0	0	0	0	0	1
	P30 input (initial value S)	0	0	0	0	0	0

[Legend]
Initial value E: Initial value in on-chip ROM disabled external extended mode
Initial value S: Initial value in other modes
Note: * Valid in external extended mode (EXPE = 1)

11.2.4 Port 5

(1) P57/AN7/DA1/$\overline{IRQ7}$-B

Module Name	Pin Function
D/A converter	DA1 output

(2) P56/AN6/DA0/$\overline{IRQ6}$-B

Module Name	Pin Function
D/A converter	DA0 output

TMR	TMO3 output	1	—
I/O port	P65 output	0	1
	P65 input (initial value)	0	0

(2) P64/TMCI3/ $\overline{\text{IRQ12}}$-B

The pin function is switched as shown below according to the P64DDR bit setting.

		Setting
		I/O Port
Module Name	**Pin Function**	**P64DDR**
I/O port	P64 output	1
	P64 input (initial value)	0

(4) P62/TMO2/SCK4/$\overline{\text{IRQ10}}$

The pin function is switched as shown below according to the combination of the TMR a register settings and P62DDR bit setting.

		Setting		
		TMR	SCI	I/O Port
Module Name	Pin Function	TMO2_OE	SCK4_OE	P62DDR
TMR	TMO2 output	1	—	—
SCI	SCK4 output	0	1	—
I/O port	P62 output	0	0	1
	P62 input (initial value)	0	0	0

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

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

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

	(initial value E)	
	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 the operati
EXPE bit, bus controller register, port function control register (PFCR), and PA6DDR bi

		Setting			
		Bus Controller		I/O Port	
Module Name	Pin Function	$\overline{\text{AH}}$_OE	$\overline{\text{BS}}$-B_OE	$\overline{\text{AS}}$_OE	PA6D
Bus controller	$\overline{\text{AH}}$ output*	1	—	—	—
	$\overline{\text{BS}}$-B output*	0	1	—	—
	$\overline{\text{AS}}$ output* (initial 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 the operat
EXPE bit, bus controller register, port function control register (PFCR), and PA4DDR b

		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; otherwi $\overline{\text{LHWR}}$ 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 cor register setting and 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

	RD/$\overline{WR}$ 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/$\overline{BREQO}$/$\overline{BS}$-A

The pin function is switched as shown below according to the combination of the operat
EXPE bit, bus controller register, port function control register (PFCR), and PA0DDR b

		Setting		
		I/O Port	Bus Controller	I/O Port
Module Name	**Pin Function**	**$\overline{BSA}$_OE**	**$\overline{BREQO}$_OE**	**PA0DDI**
Bus controller	$\overline{BS}$-A output*	1	—	—
	$\overline{BREQO}$ 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)

		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).

11.2.8 Port E

(1) PE7/A15

The pin function is switched as shown below according to the combination of the operati EXPE bit, and PE7DDR bit settings.

		Setting	
			I/O Port
Module Name	**Pin Function**	**MCU Operating Mode**	**PE7DDR**
Bus controller	Address output	On-chip ROM disabled extended mode	—
		On-chip ROM enabled extended mode	1
I/O port	PE7 output	Single-chip mode*	1
	PE7 input (initial value)	Modes other than on-chip ROM disabled extended mode	0

Note: * Address output is enabled by setting PE7DDR = 1 in external extended mode (EXPE = 1).

	PE6 input (initial value)	Modes other than on-chip ROM disabled extended mode	0

Note: * Address output is enabled by setting PE6DDR = 1 in external extended mode (EXPE = 1).

(3) PE5/A13

The pin function is switched as shown below according to the combination of the operat EXPE bit, and PE5DDR bit settings.

		Setting	
			I/O Por
Module Name	**Pin Function**	**MCU Operating Mode**	**PE5DD**
Bus controller	Address output	On-chip ROM disabled extended mode	—
		On-chip ROM enabled extended mode	1
I/O port	PE5 output	Single-chip mode*	1
	PE5 input (initial value)	Modes other than on-chip ROM disabled extended mode	0

Note: * Address output is enabled by setting PE5DDR = 1 in external extended mode (EXPE = 1).

	PEn input (initial value)	Modes other than on-chip ROM disabled extended mode	0

[Legend]

n: 0 to 4

Note: * Address output is enabled by setting PEnDDR = 1 in external extended mode (EXPE = 1).

11.2.9 Port F

(1) PF4/A20

The pin function is switched as shown below according to the combination of the operati
EXPE bit, port function control register (PFCR), and PF4DDR bit settings.

			Setting	
			I/O Port	I/O Port
MCU Operating Mode	Module Name	Pin Function	A20_OE	PF4DDR
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)

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)

(3) PF2/A18

The pin function is switched as shown below according to the combination of the operat
EXPE bit, port function control register (PFCR), and PF2DDR bit settings.

			Setting	
			I/O Port	I/O Port
MCU Operating Mode	**Module Name**	**Pin Function**	**A18_OE**	**PF2DDR**
On-chip ROM disabled extended mode	Bus controller	A18 output	—	—
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)

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)

(5) PF0/A16

The pin function is switched as shown below according to the combination of the operati
EXPE bit, port function control register (PFCR), and PF0DDR bit settings.

			Setting	
			I/O Port	I/O Port
MCU Operating Mode	**Module Name**	**Pin Function**	**A16_OE**	**PF0DDR**
On-chip ROM disabled extended mode	Bus controller	A16 output	—	—
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)

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	0	0
TMR	TMOn output	0	1	—
I/O port	PIn output	0	0	1
	PIn 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

n: 4 to 7

Note: * Valid in external extended mode (EXPE = 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 3

Note: * Valid in external extended mode (EXPE = 1)

				SMR.GM = 0, SCR.CKE[1, 0] = 01 or while SMF When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE[1, 0] = 01 or while SMF SCR.CKE1 = 0
5	$\overline{\text{TEND1A}}$_OE	$\overline{\text{TEND1}}$	FPCR7.DMAS1[A,B] = 00	DMDR.TENDE = 1
	SCL1_OE	SCL1		ICCRA.ICE = 1
4	TxD3_OE	TxD3		SCR.TE = 1, IrCR.IrE = 0
	SDA1_OE	SDA1		ICCRA.ICE = 1
3	—	—	—	—
2	$\overline{\text{DACK0A}}$_OE	$\overline{\text{DACK0}}$	FPCR7.DMAS0[A,B] = 00	DACR.AMS = 1, DMDR.DACKE = 1
	SCK2_OE	SCK2		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE[1, 0] = 01 or while SMF When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE[1, 0] = 01 or while SMF SCR.CKE1 = 0
1	$\overline{\text{TEND0A}}$_OE	$\overline{\text{TEND0}}$	PFCR7.DMAS0[A,B] = 00	DMDR.TENDE = 1
0	TxD2_OE	TxD2		SCR.TE = 1

	PO5_OE	PO5	NDERL.NDER5 = 1
4	TIOCB4_OE	TIOCB4	TPU.TIOR4.IOB3 = 0, TPU.TIOR4.IOB[1,0] =
	SCK1_OE	SCK1	When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE[1, 0] = 01 or while SM When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE[1, 0] = 01 or while SM SCR.CKE1 = 0
	PO4_OE	PO4	NDERL.NDER4 = 1
3	TIOCD3_OE	TIOCD3	TPU.TMDR.BFB = 0, TPU.TIORL3.IOD3 = 0, TPU.TIORL3.IOD[1,0] = 01/10/11
	PO3_OE	PO3	NDERL.NDER3 = 1
2	TIOCC3_OE	TIOCC3	TPU.TMDR.BFA = 0, TPU.TIORL3.IOC3 = 0, TPU.TIORL3.IOD[1,0] = 01/10/11
	TMO0_OE	TMO0	TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] =
	TxD0_OE	TxD0	SCR.TE = 1
	PO2_OE	PO2	NDERL.NDER2 = 1
1	TIOCA3_OE	TIOCA3	TPU.TIORH3.IOA3 = 0, TPU.TIORH3.IOA[1,0
	PO1_OE	PO1	NDERL.NDER1 = 1

		PO9_OE	PO9		NDERH.NDER9 = 1
P3	7	TIOCB2_OE	TIOCB2		TPU.TIOR2.IOB3 = 0, TPU.TIOR2.IOB[1,0] = 01
		PO15_OE	PO15		NDERH.NDER15 = 1
	6	TIOCA2_OE	TIOCA2		TPU.TIOR2.IOA3 = 0, TPU.TIOR2.IOA[1,0] = 01
		PO14_OE	PO14		NDERH.NDER14 = 1
	5	$\overline{\text{DACK1B}}$_OE	$\overline{\text{DACK1}}$	PFCR7.DMAS1[A,B] = 01	DACR.AMS = 1, DMDR.DACKE = 1
		TIOCB1_OE	TIOCB1		TPU.TIOR1.IOB3 = 0, TPU.TIOR1.IOB[1,0] = 01
		PO13_OE	PO13		NDERH.NDER13 = 1
	4	$\overline{\text{TEND1B}}$_OE	$\overline{\text{TEND1}}$	PFCR7.DMAS1[A,B] = 01	DMDR.TENDE = 1
		TIOCA1_OE	TIOCA1		TPU.TIOR1.IOA3 = 0, TPU.TIOR1.IOA[1,0] = 01
		PO12_OE	PO12		NDERH.NDER12 = 1
	3	$\overline{\text{CS3}}$_OE	$\overline{\text{CS3}}$		SYSCR.EXPE = 1, PFCR0.CS3E = 1
		$\overline{\text{CS7A}}$_OE	$\overline{\text{CS7}}$	PFCR1.CS7S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS7E = 1
		TIOCD0_OE	TIOCD0		TPU.TMDR.BFB = 0, TPU.TIORL0.IOD3 = 0, TPU.TIORL0.IOD[1,0] = 01/10/11
		PO11_OE	PO11		NDERH.NDER11 = 1
	2	$\overline{\text{CS2A}}$_OE	$\overline{\text{CS2}}$	PFCR2.CS2S = 0	SYSCR.EXPE = 1, PFCR0.CS2E = 1
		$\overline{\text{CS6A}}$_OE	$\overline{\text{CS6}}$	PFCR1.CS6S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS6E = 1
		$\overline{\text{DACK0B}}$_OE	$\overline{\text{DACK0}}$	PFCR7.DMAS0[A,B] = 01	DACR.AMS = 1, DMDR.DACKE = 1
		TIOCC0_OE	TIOCC0		TPU.TMDR.BFA = 0, TPU.TIORL0.IOC3 = 0, TPU.TIORL0.IOD[1,0] = 01/10/11
		PO10_OE	PO10		NDERH.NDER10 = 1

		$\overline{\text{TEND0B}}$_OE	$\overline{\text{TEND0}}$	PFCR7.DMAS0[A,B] = 01	DMDR.TENDE = 1
		TIOCB0_OE	TIOCB0		TPU.TIORH0.IOB3 = 0, TPU.TIORH0.IOB[1,0]
		PO9_OE	PO9		NDERH.NDER9 = 1
	0	$\overline{\text{CS0}}$_OE	$\overline{\text{CS0}}$		SYSCR.EXPE = 1, PFCR0.CS0E = 1
		$\overline{\text{CS4}}$_OE	$\overline{\text{CS4}}$		SYSCR.EXPE = 1, PFCR0.CS4E = 1
		$\overline{\text{CS5B}}$_OE	$\overline{\text{CS5}}$	PFCR1.CS5S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS5E = 1
		TIOCA0_OE	TIOCA0		TPU.TIORH0.IOA3 = 0, TPU.TIORH0.IOA[1,0]
		PO8_OE	PO8		NDERH.NDER8 = 1
P6	5	TMO3_OE	TMO3		TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] =
	2	TMO2_OE	TMO2		TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] =
		SCK4_OE	SCK4		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE[1, 0] = 01 or while SM When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE[1, 0] = 01 or while SM SCR.CKE1 = 0
	0	TxD4_OE	TxD4		SCR.TE = 1

		$\overline{\text{LHWR}}$_OE	$\overline{\text{LHWR}}$		SYSCR.EXPE = 1, PFCR6.LHWROE = 1
	3	$\overline{\text{LLB}}$_OE	$\overline{\text{LLB}}$		SYSCR.EXPE = 1, SRAMCR.BCSELn = 1
		$\overline{\text{LLWR}}$_OE	$\overline{\text{LLWR}}$		SYSCR.EXPE = 1
	1	$\overline{\text{BACK}}$_OE	$\overline{\text{BACK}}$		SYSCR.EXPE = 1, BCR1.BRLE = 1
		(RD/$\overline{\text{WR}}$)_OE	RD/$\overline{\text{WR}}$		SYSCR.EXPE = 1, PFCR2.RDWRE = 1, or SRAMCR.BCSELn = 1
	0	$\overline{\text{BSA}}$_OE	$\overline{\text{BS}}$	PFCR2.BSS = 0	SYSCR.EXPE = 1, PFCR2.BSE = 1
		$\overline{\text{BREQO}}$_OE	$\overline{\text{BREQO}}$		SYSCR.EXPE = 1, BCR1.BRLE = 1, BCR1.BRE
PD	7	A7_OE	A7		SYSCR.EXPE = 1, PDDDR.PD7DDR = 1
	6	A6_OE	A6		SYSCR.EXPE = 1, PDDDR.PD6DDR = 1
	5	A5_OE	A5		SYSCR.EXPE = 1, PDDDR.PD5DDR = 1
	4	A4_OE	A4		SYSCR.EXPE = 1, PDDDR.PD4DDR = 1
	3	A3_OE	A3		SYSCR.EXPE = 1, PDDDR.PD3DDR = 1
	2	A2_OE	A2		SYSCR.EXPE = 1, PDDDR.PD2DDR = 1
	1	A1_OE	A1		SYSCR.EXPE = 1, PDDDR.PD1DDR = 1
	0	A0_OE	A0		SYSCR.EXPE = 1, PDDDR.PD0DDR = 1
PE	7	A15_OE	A15		SYSCR.EXPE = 1, PEDDR.PE7DDR = 1
	6	A14_OE	A14		SYSCR.EXPE = 1, PEDDR.PE6DDR = 1
	5	A13_OE	A13		SYSCR.EXPE = 1, PEDDR.PE5DDR = 1
	4	A12_OE	A12		SYSCR.EXPE = 1, PEDDR.PE4DDR = 1
	3	A11_OE	A11		SYSCR.EXPE = 1, PEDDR.PE3DDR = 1
	2	A10_OE	A10		SYSCR.EXPE = 1, PEDDR.PE2DDR = 1
	1	A9_OE	A9		SYSCR.EXPE = 1, PEDDR.PE1DDR = 1
	0	A8_OE	A8		SYSCR.EXPE = 1, PEDDR.PE0DDR = 1

	5	D5_E	D5	SYSCR.EXPE = 1
	4	D4_E	D4	SYSCR.EXPE = 1
	3	D3_E	D3	SYSCR.EXPE = 1
	2	D2_E	D2	SYSCR.EXPE = 1
	1	D1_E	D1	SYSCR.EXPE = 1
	0	D0_E	D0	SYSCR.EXPE = 1
PI	7	D15_E	D15	SYSCR.EXPE = 1, ABWCR.ABW[H,L]n = 01
		TMO7_OE	TMO7	TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] =
	6	D14_E	D14	SYSCR.EXPE = 1, ABWCR.ABW[H,L]n = 01
		TMO6_OE	TMO6	TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] =
	5	D13_E	D13	SYSCR.EXPE = 1, ABWCR.ABW[H,L]n = 01
		TMO5_OE	TMO5	TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] =
	4	D12_E	D12	SYSCR.EXPE = 1, ABWCR.ABW[H,L]n = 01
		TMO4_OE	TMO4	TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] =
	3	D11_E	D11	SYSCR.EXPE = 1, ABWCR.ABW[H,L]n = 01
	2	D10_E	D10	SYSCR.EXPE = 1, ABWCR.ABW[H,L]n = 01
	1	D9_E	D9	SYSCR.EXPE = 1, ABWCR.ABW[H,L]n = 01
	0	D8_E	D8	SYSCR.EXPE = 1, ABWCR.ABW[H,L]n = 01

- Port function control register 6 (PFCR6)
- Port function control register 7 (PFCR7)
- Port function control register 9 (PFCR9)
- Port function control register B (PFCRB)
- Port function control register C (PFCRC)

11.3.1 Port Function Control Register 0 (PFCR0)

PFCR0 enables/disables the $\overline{CS}$ output.

Bit	7	6	5	4	3	2	1	
Bit Name	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	
Initial Value	0	0	0	0	0	0	0	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: * 1 in external extended mode, 0 in other modes.

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

Note: * 1 in external extended mode, 0 in other modes.

7	CS7SA*	0	R/W	$\overline{CS7}$ Output Pin Select
6	CS7SB*	0	R/W	Selects the output pin for $\overline{CS7}$ when $\overline{CS7}$ outpu 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 $\overline{CS6}$ outpu 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 $\overline{CS5}$ outpu 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)

select bits (n = 4 to 7), multiple CS signals are output from the pin. For details, section 8.5.3, Chip Select Signals.

11.3.3 Port Function Control Register 2 (PFCR2)

PFCR1 selects the $\overline{CS}$ output pin, enables/disables bus control I/O, and selects the bus co pins.

Bit	7	6	5	4	3	2	1	
Bit Name	—	CS2S	BSS	BSE	—	RDWRE	ASOE	
Initial Value	0	0	0	0	0	0	1	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0. The write value sho always be 0.
6	CS2S*[1]	0	R/W	$\overline{CS2}$ Output Pin Select Selects the output pin for $\overline{CS2}$ when $\overline{CS2}$ output enabled (CS2E = 1) 0: Specifies pin PB2 as $\overline{CS2}$-A output pin 1: Specifies pin PB1 as $\overline{CS2}$-B output pin

3	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.
2	RDWRE*2	0	R/W	RD/$\overline{WR}$ Output Enable Enables/disables the RD/$\overline{WR}$ output 0: Disables the RD/$\overline{WR}$ output 1: Enables the RD/$\overline{WR}$ output
1	ASOE	1	R/W	$\overline{AS}$ Output Enable Enables/disables the $\overline{AS}$ output 0: Specifies pin PA6 as I/O port 1: Specifies pin PA6 as $\overline{AS}$ output pin
0	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.

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

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

7 to 5	—	0	R/W	Reserved This bit is always read as 0. The write value shou 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: * The initial value differs according to the set operating mode: 1 for operating mo which on-chip ROM is disabled, and 0 for those in which on-chip ROM is enab

7	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.
6	LHWROE	1	R/W	$\overline{\text{LHWR}}$ Output Enable Enables/disables $\overline{\text{LHWR}}$ output (valid in externa extended mode). 0: Specifies pin PA4 as I/O port 1: Specifies pin PA4 as $\overline{\text{LHWR}}$ output pin
5	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.
4	—	0	R	Reserved This is a read-only bit and cannot be modified.
3	TCLKS	0	R/W	TPU External Clock Input Pin Select Selects the TPU external clock input pins. 0: Specifies pins P32, P33, P35, and P37 as ex clock inputs 1: Specifies pins P14 to P17 as external clock i
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.

7 to 4	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
3	DMAS1A	0	R/W	DMAC Control Pin Select
2	DMAS1B	0	R/W	Selects the I/O port to control DMAC_1. 00: Specifies pins P14 to P16 as DMAC control 01: Specifies pins P33 to P35 as DMAC control 10: Setting prohibited 11: Setting prohibited
1	DMAS0A	0	R/W	DMAC Control Pin Select
0	DMAS0B	0	R/W	Selects the I/O port to control DMAC_0. 00: Specifies pins P10 to P12 as DMAC control 01: Specifies pins P30 to P32 as DMAC control 10: Setting prohibited 11: Setting prohibited

Bit	Bit Name	Value	R/W	Description
7	TPUMS5	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA5 function 0: Specifies pin P26 as output compare output capture 1: Specifies P27 as input capture input and P26 compare
6	TPUMS4	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA4 function 0: Specifies P25 as output compare output and capture 1: Specifies P24 as input capture input and P25 compare
5	TPUMS3A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA3 function 0: Specifies P21 as output compare output and capture 1: Specifies P20 as input capture input and P21 compare
4	TPUMS3B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC3 function 0: Specifies P22 as output compare output and capture 1: Specifies P23 as input capture input and P22 compare

				0: Specifies P34 as output compare output and i capture 1: Specifies P35 as input capture input and P34 compare
1	TPUMS0A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA0 function 0: Specifies P30 as output compare output and i capture 1: Specifies P31 as input capture input and P30 compare
0	TPUMS0B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC0 function 0: Specifies P32 as output compare output and i capture 1: Specifies P33 as input capture input and P32 compare

Bit	Bit Name	Value	R/W	Description
7 to 6	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
5	ITS13	0	R/W	$\overline{\text{IRQ13}}$ Pin Select Selects an input pin for $\overline{\text{IRQ13}}$. 0: Selects pin P23 as $\overline{\text{IRQ13}}$-A input 1: Selects pin P63 as $\overline{\text{IRQ13}}$-B input
4	ITS12	0	R/W	$\overline{\text{IRQ12}}$ Pin Select Selects an input pin for $\overline{\text{IRQ12}}$. 0: Selects pin P23 as $\overline{\text{IRQ12}}$-A input 1: Selects pin P63 as $\overline{\text{IRQ12}}$-B input
3	ITS11	0	R/W	$\overline{\text{IRQ11}}$ Pin Select Selects an input pin for $\overline{\text{IRQ11}}$. 0: Selects pin P23 as $\overline{\text{IRQ11}}$-A input 1: Selects pin P63 as $\overline{\text{IRQ11}}$-B input
2	ITS10	0	R/W	$\overline{\text{IRQ10}}$ Pin Select Selects an input pin for $\overline{\text{IRQ10}}$. 0: Selects pin P22 as $\overline{\text{IRQ10}}$-A input 1: Selects pin P62 as $\overline{\text{IRQ10}}$-B input

11.3.9 Port Function Control Register C (PFCRC)

PFCRC selects input pins for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

Bit	7	6	5	4	3	2	1	
Bit Name	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	
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	ITS7	0	R/W	$\overline{\text{IRQ7}}$ Pin Select Selects an input pin for $\overline{\text{IRQ7}}$. 0: Selects pin P17 as $\overline{\text{IRQ7}}$-A input 1: Selects pin P57 as $\overline{\text{IRQ7}}$-B output
6	ITS6	0	R/W	$\overline{\text{IRQ6}}$ Pin Select Selects an input pin for $\overline{\text{IRQ6}}$. 0: Selects pin P16 as $\overline{\text{IRQ6}}$-A input 1: Selects pin P56 as $\overline{\text{IRQ6}}$-B output
5	ITS5	0	R/W	$\overline{\text{IRQ5}}$ Pin Select Selects an input pin for $\overline{\text{IRQ5}}$. 0: Selects pin P15 as $\overline{\text{IRQ5}}$-A input 1: Selects pin P55 as $\overline{\text{IRQ5}}$-B output

2	ITS2	0	R/W	$\overline{\text{IRQ2}}$ Pin Select Selects an input pin for $\overline{\text{IRQ2}}$. 0: Selects pin P12 as $\overline{\text{IRQ2}}$-A input 1: Selects pin P52 as $\overline{\text{IRQ2}}$-B output
1	ITS1	0	R/W	$\overline{\text{IRQ1}}$ Pin Select Selects an input pin for $\overline{\text{IRQ1}}$. 0: Selects pin P11 as $\overline{\text{IRQ1}}$-A input 1: Selects pin P51 as $\overline{\text{IRQ1}}$-B output
0	ITS0	0	R/W	$\overline{\text{IRQ0}}$ Pin Select Selects an input pin for $\overline{\text{IRQ0}}$. 0: Selects pin P10 as $\overline{\text{IRQ0}}$-A input 1: Selects pin P50 as $\overline{\text{IRQ0}}$-B output

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

11.4.2 Notes on Port Function Control Register (PFCR) Settings

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

- The following operations can be set for each channel:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Synchronous operations:
    - Multiple timer counters (TCNT) can be written to simultaneously
    - Simultaneous clearing by compare match and input capture possible
    - Simultaneous input/output for registers possible by counter synchronous oper
    - Maximum of 15-phase PWM output possible by combination with synchronc operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- Programmable pulse generator (PPG) output trigger can be generated
- Conversion start trigger for the A/D converter and ΔΣ A/D converter can be generate
- Module stop state specifiable

		TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TG
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TI TI
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TG ma cap
Compare match output	0 output	O	O	O	O	O	O
	1 output	O	O	O	O	O	O
	Toggle output	O	O	O	O	O	O
Input capture function		O	O	O	O	O	O
Synchronous operation		O	O	O	O	O	O
PWM mode		O	O	O	O	O	O
Phase counting mode		—	O	O	—	O	O
Buffer operation		O	—	—	O	—	—

PPG trigger	TGRA_0/ TGRB_0 compare match or input capture	TGRA_1/ TGRB_1 compare match or input capture	TGRA_2/ TGRB_2 compare match or input capture	TGRA_3/ TGRB_3 compare match or input capture	—	–
Interrupt sources	5 sources	4 sources	4 sources	5 sources	4 sources	4
	Compare match or input capture 0A	Compare match or input capture 1A	Compare match or input capture 2A	Compare match or input capture 3A	Compare match or input capture 4A	C m c
	Compare match or input capture 0B	Compare match or input capture 1B	Compare match or input capture 2B	Compare match or input capture 3B	Compare match or input capture 4B	C m c
	Compare match or input capture 0C	Overflow Underflow	Overflow Underflow	Compare match or input capture 3C	Overflow Underflow	C U
	Compare match or input capture 0D			Compare match or input capture 3D		
	Overflow			Overflow		

[Legend]

O : Possible

— : Not possible

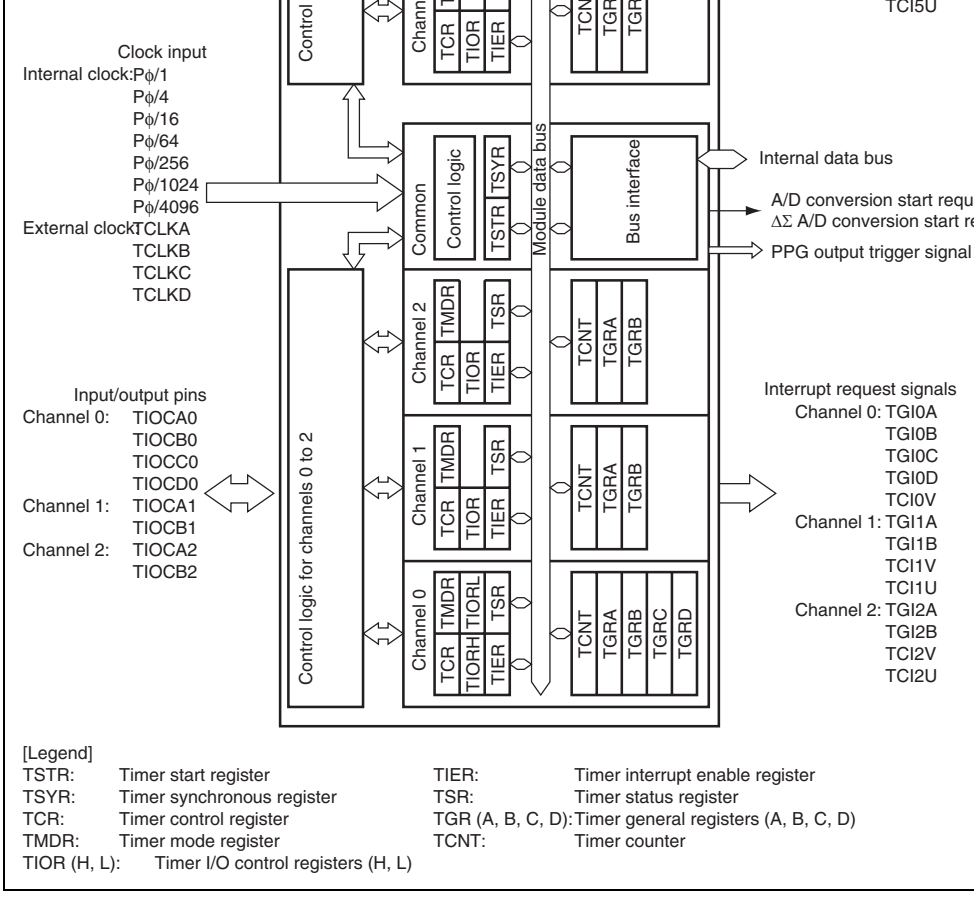


Figure 12.1 Block Diagram of TPU

			(Channel 1 and 3 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM o
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM o
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM o
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM o
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM o
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM o
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM o
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM o
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM o
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM o
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM o
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM o
4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM o
	TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM o
5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM o
	TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM o

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

Channel 1:

- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)

Channel 3:

- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register H_3 (TIORH_3)
- Timer I/O control register L_3 (TIORL_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)

Channel 4:

- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register _4 (TIOR_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)

Common Registers:

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

Bit	Bit Name	Value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	These bits select the TCNT counter clearing so
5	CCLR0	0	R/W	tables 12.3 and 12.4 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 12.5. When the input clock is counted usi edges, the input clock period is halved (e.g. Pϕ/ edges = Pϕ/2 rising edge). If phase counting m used on channels 1, 2, 4, and 5, this setting is i and the phase counting mode setting has priori clock edge selection is valid when the input clo or slower. This setting is ignored if the input clo or when overflow/underflow of another channel selected.
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The
0	TPSC0	0	R/W	source can be selected independently for each See tables 12.6 to 12.11 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 11, I/O Ports.

1	0	0	TCNT clearing disabled
1	0	1	TCNT cleared by TGRC compare matcl capture*[2]
1	1	0	TCNT cleared by TGRD compare matcl capture*[2]
1	1	1	TCNT cleared by counter clearing for ar channel performing synchronous clearir synchronous operation*[1]

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared becau buffer register setting has priority, and compare match/input capture does not

Table 12.4 CCLR2 to CCLR0 (Channels 1, 2, 4, and 5)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare matcl capture
	0	1	0	TCNT cleared by TGRB compare matcl capture
	0	1	1	TCNT cleared by counter clearing for ar channel performing synchronous clearir synchronous operation*[1]

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot b modified.

Table 12.6 TPSC2 to TPSC0 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin i
	1	0	1	External clock: counts on TCLKB pin i
	1	1	0	External clock: counts on TCLKC pin i
	1	1	1	External clock: counts on TCLKD pin i

Table 12.7 TPSC2 to TPSC0 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin i
	1	0	1	External clock: counts on TCLKB pin i
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

	1	1	0	External clock: counts on TCLKC pin in
	1	1	1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 12.9 TPSC2 to TPSC0 (Channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin in
	1	0	1	Internal clock: counts on Pϕ/1024
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	Internal clock: counts on Pϕ/4096

	1	1	0	Internal clock: counts on Pϕ/1024
	1	1	1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 12.11 TPSC2 to TPSC0 (Channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin i
	1	0	1	External clock: counts on TCLKC pin i
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	External clock: counts on TCLKD pin i

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

Bit	Bit Name	Value	R/W	Description
7, 6	—	All 1	R	Reserved These are read-only bits and cannot be modifie
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to normally operate TGRB and TGRD are to be used together for b operation. When TGRD is used as a buffer reg TGRD input capture/output compare is not ger In channels 1, 2, 4, and 5, which have no TGR reserved. It is always read as 0 and cannot be 0: TGRB operates normally 1: TGRB and TGRD used together for buffer op
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to normally operate TGRA and TGRC are to be used together for b operation. When TGRC is used as a buffer reg TGRC input capture/output compare is not ger In channels 1, 2, 4, and 5, which have no TGR reserved. It is always read as 0 and cannot be 0: TGRA operates normally 1: TGRA and TGRC used together for buffer op
3	MD3	0	R/W	Modes 3 to 0 Set the timer operating mode. MD3 is a reserved bit. The write value should a 0. See table 12.12 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

0	1	1	0	Phase counting mode 3
0	1	1	1	Phase counting mode 4
1	X	X	X	—

[Legend]

X: Don't care

Notes: 1. MD3 is a reserved bit. The write value should always be 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 sho be written to MD2.

12.3.3 Timer I/O Control Register (TIOR)

TIOR controls TGR. The TPU has eight TIOR registers, two each for channels 0 and 3, each for channels 1, 2, 4, and 5. Care is required since TIOR is affected by the TMDR s

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

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

To designate the input capture pin in TIOR, the DDR bit and ICR bit for the correspond should be set to 0 and 1, respectively. For details, see section 11, I/O Ports.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIOR_4, TIOR_5

Bit	Bit Name	Initial Value	R/W	Description
7	IOB3	0	R/W	I/O Control B3 to B0
6	IOB2	0	R/W	Specify the function of TGRB.
5	IOB1	0	R/W	For details, see tables 12.13, 12.15, 12.16, 12.17
4	IOB0	0	R/W	and 12.20.
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	For details, see tables 12.21, 12.23, 12.24, 12.25
0	IOA0	0	R/W	and 12.28.

- TIORL_0, TIORL_3

Bit	Bit Name	Initial Value	R/W	Description
7	IOD3	0	R/W	I/O Control D3 to D0
6	IOD2	0	R/W	Specify the function of TGRD.
5	IOD1	0	R/W	For details, see tables 12.14 and 12.18.
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O Control C3 to C0
2	IOC2	0	R/W	Specify the function of TGRC.
1	IOC1	0	R/W	For details, see tables 12.22 and 12.26.
0	IOC0	0	R/W	

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB0 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB0 pin Input capture at both edges
1	1	x	x		Capture input source is channel 1/coun Input capture at TCNT_1 count-up/cou

[Legend]

X: Don't care

Note: * When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and Pϕ/1 is used as t
TCNT_1 count clock, this setting is invalid and input capture is not generated

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCD0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCD0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/count Input capture at TCNT_1 count-up/count

[Legend]

X: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and Pϕ/1 is used as th TCNT_1 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer regis setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB1 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCB1 pin Input capture at both edges
1	1	X	X		TGRC_0 compare match/input capture Input capture at generation of TGRC_0 match/input capture

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
1	X	0	1		Capture input source is TIOCB2 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCB2 pin Input capture at both edges

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/coun Input capture at TCNT_4 count-up/cou

[Legend]

X: Don't care

Note: When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and Pϕ/1 is used as the T count clock, this setting is invalid and input capture is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCD3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCD3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/count Input capture at TCNT_4 count-up/count

[Legend]

X: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and Pϕ/1 is used as th TCNT_4 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer regis setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB4 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCB4 pin Input capture at both edges
1	1	X	X		Capture input source is TGRC_3 comp match/input capture Input capture at generation of TGRC_3 match/input capture

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCB5 pin Input capture at rising edge
1	X	0	1		Capture input source is TIOCB5 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCB5 pin Input capture at both edges

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/coun Input capture at TCNT_1 count-up/cou

[Legend]

X: Don't care

Note: * When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and Pϕ/1 is used as t
TCNT_1 count clock, this setting is invalid and input capture is not generated

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCC0 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCC0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCC0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/count Input capture at TCNT_1 count-up/count

[Legend]

X: Don't care

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 regis setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA1 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA1 pin Input capture at both edges
1	1	X	X		Capture input source is TGRA_0 comp match/input capture Input capture at generation of channel compare match/input capture

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
1	X	0	1		Capture input source is TIOCA2 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCA2 pin Input capture at both edges

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA3 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA3 pin Input capture at both edges
1	1	X	X		Capture input source is channel 4/coun Input capture at TCNT_4 count-up/cou

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

[Legend]

X: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and Pϕ/1 is used as th TCNT_4 count clock, this setting is invalid and input capture is not generated.

2. When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer regis setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA4 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA4 pin Input capture at both edges
1	1	X	X		Capture input source is TGRA_3 comp match/input capture Input capture at generation of TGRA_3 match/input capture

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	X	0	0	Input capture register	Input capture source is TIOCA5 pin Input capture at rising edge
1	X	0	1		Input capture source is TIOCA5 pin Input capture at falling edge
1	X	1	X		Input capture source is TIOCA5 pin Input capture at both edges

[Legend]
X: Don't care

Bit	Bit Name	value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables/disables generation of A/D conversion ΔΣ A/D conversion start requests by TGRA inpu capture/compare match. 0: A/D conversion and ΔΣ A/D conversion start generation disabled 1: A/D conversion and ΔΣ A/D conversion start generation enabled
6	—	1	R	Reserved This is a read-only bit and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables/disables interrupt requests (TCIU) by t flag when the TCFU flag in TSR is set to 1 in ch 2, 4, and 5. In channels 0 and 3, bit 5 is reserved. It is alwa 0 and cannot be modified. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables/disables interrupt requests (TCIV) by t flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled

2	TGIEC	0	R/W	TGR Interrupt Enable C Enables/disables interrupt requests (TGIC) by th bit when the TGFC bit in TSR is set to 1 in chan and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disable 1: Interrupt requests (TGIC) by TGFC bit enable
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables/disables interrupt requests (TGIB) by th bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disable 1: Interrupt requests (TGIB) by TGFB bit enable
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables/disables interrupt requests (TGIA) by th bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disable 1: Interrupt requests (TGIA) by TGFA bit enable

Bit	Bit Name	value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TC in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is alwa 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This is a read-only bit and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that a TCNT underflow occurred when channels 1, 2, 4, and 5 are set t counting mode. In channels 0 and 3, bit 5 is reserved. It is alwa 0 and cannot be modified. [Setting condition] When the TCNT value underflows (changes fro to H'FFFF) [Clearing condition] When a 0 is written to TCFU after reading TCF (When the CPU is used to clear this flag by writ 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 writi while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGR capture or compare match in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD while TGRD is functior output compare register • When TCNT value is transferred to TGRD by capture signal while TGRD is functioning as i capture register [Clearing conditions] • When DTC is activated by a TGID interrupt w DISEL bit in MRB of DTC is 0 • When 0 is written to TGFD after reading TGF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)

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

[Clearing conditions]

- When DTC is activated by a TGIA interrupt w
  DISEL bit in MRB of DTC is 0
- When DMAC is activated by a TGIA interrupt
  the DTA bit in DMDR of DMAC is 1
- When 0 is written to TGFA after reading TGF
  (When the CPU is used to clear this flag by w
  while the corresponding interrupt is enabled,
  to read the flag after writing 0 to it.)

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

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

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

12.3.7 Timer General Register (TGR)

TGR is a 16-bit readable/writable register with a dual function as output compare and in
capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two
channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated fo
operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they m
always be accessed in 16-bit units. TGR and buffer register combinations during buffer
are TGRA–TGRC and TGRB–TGRD.

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
5	CST5	0	R/W	Counter Start 5 to 0
4	CST4	0	R/W	These bits select operation or stoppage for TCN
3	CST3	0	R/W	If 0 is written to the CST bit during operation with
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 retained 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 stoppe
				1: TCNT_5 to TCNT_0 performs count operation

Bit	Bit Name	value	R/W	Description
7, 6	——	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
5	SYNC5	0	R/W	Timer Synchronization 5 to 0
4	SYNC4	0	R/W	These bits select whether operation is indepen synchronized with other channels.
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	When synchronous operation is selected, synch presetting of multiple channels, and synchronou
1	SYNC1	0	R/W	through counter clearing on another channel ar
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 CCLR CCLR0 in TCR. 0: TCNT_5 to TCNT_0 operate independently (presetting/clearing is unrelated to other char 1: TCNT_5 to TCNT_0 perform synchronous op (TCNT synchronous presetting/synchronous is possible)

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 12.2 shows an example of the count operation setting procedure.

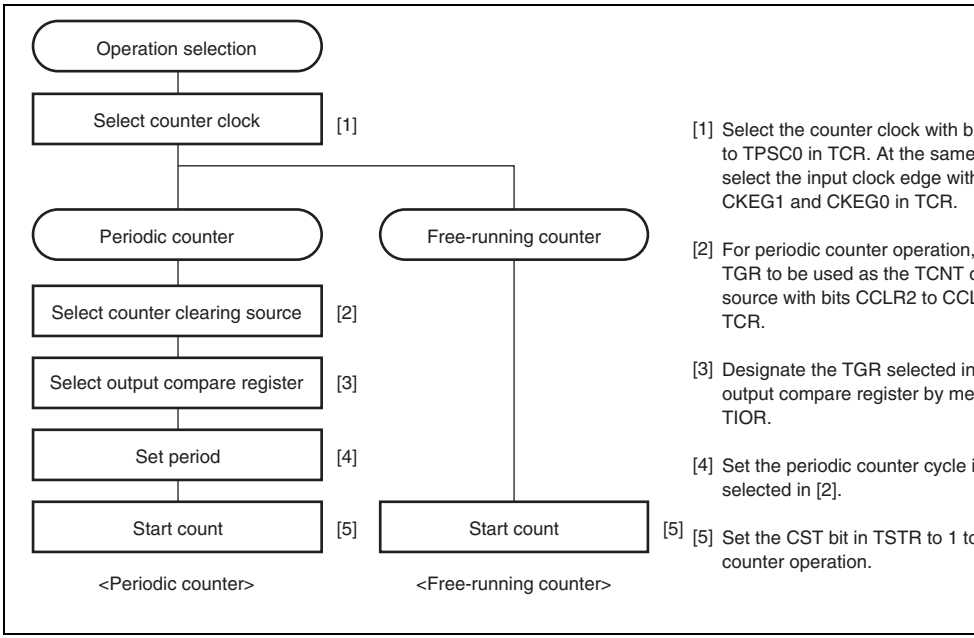


Figure 12.2 Example of Counter Operation Setting Procedure

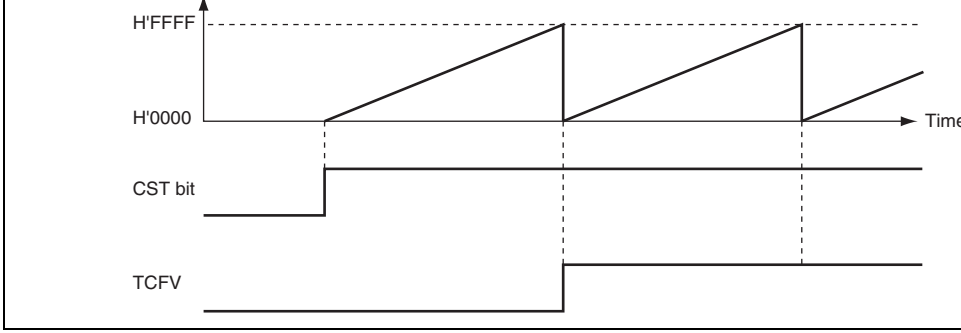


Figure 12.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for th channel performs periodic count operation. The TGR register for setting the period is de as an output compare register, and counter clearing by compare match is selected by me CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up op a periodic counter when the corresponding bit in TSTR is set to 1. When the count value the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests a After a compare match, TCNT starts counting up again from H'0000.

Figure 12.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

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

(a) Example of setting procedure for waveform output by compare match

Figure 12.5 shows an example of the setting procedure for waveform output by a compar

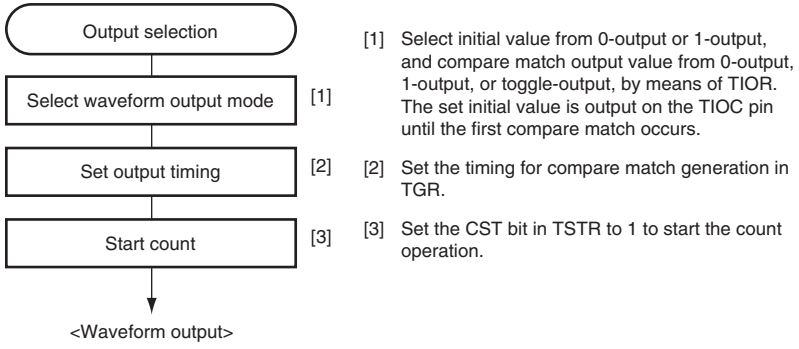


Figure 12.5 Example of Setting Procedure for Waveform Output by Compare M

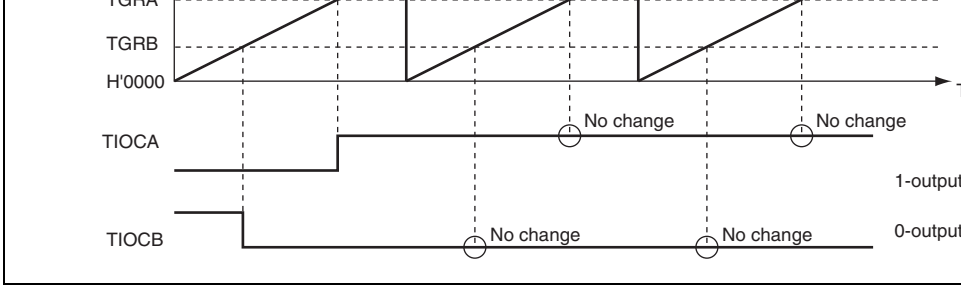


Figure 12.6 Example of 0-Output/1-Output Operation

Figure 12.7 shows an example of toggle output.

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

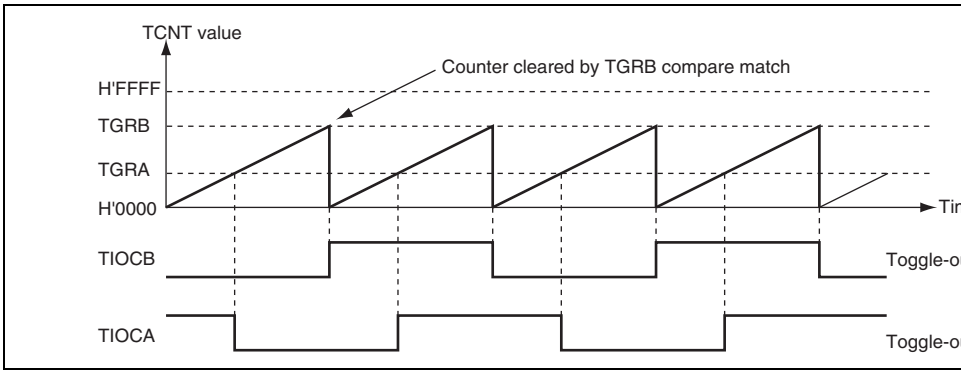


Figure 12.7 Example of Toggle Output Operation

(a) Example of setting procedure for input capture operation

Figure 12.8 shows an example of the setting procedure for input capture operation.

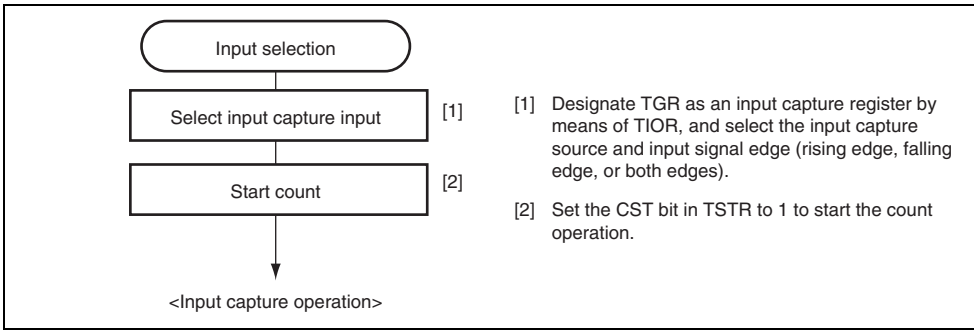


Figure 12.8 Example of Setting Procedure for Input Capture Operation

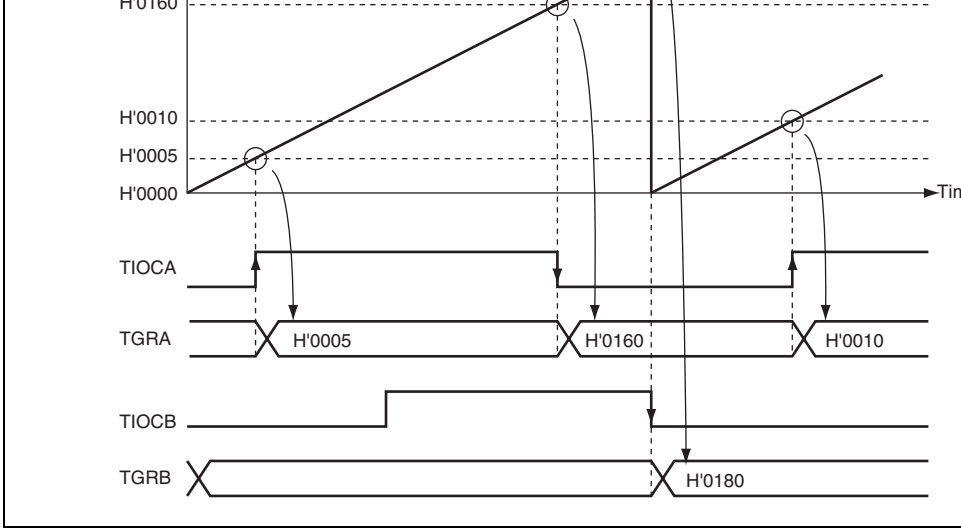
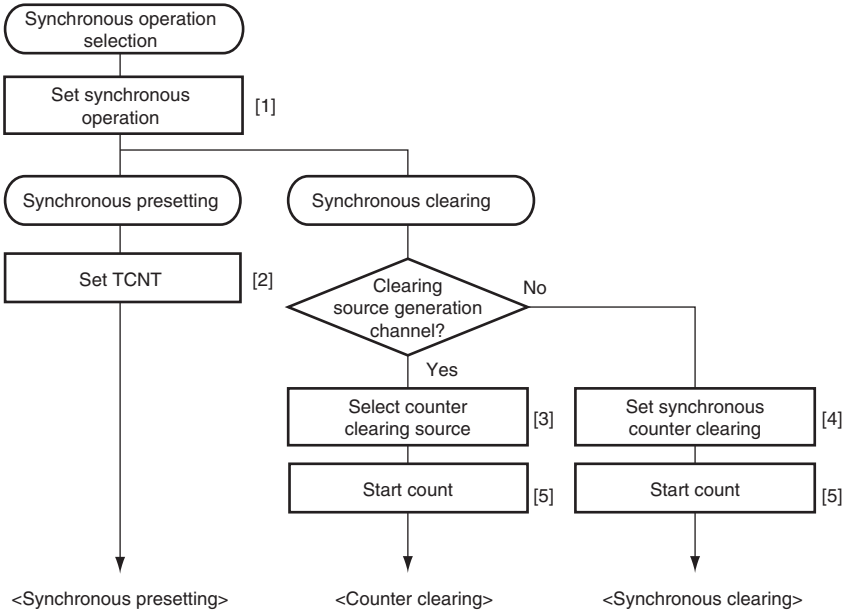


Figure 12.9 Example of Input Capture Operation

Figure 12.10 shows an example of the synchronous operation setting procedure.



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

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

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

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

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

Figure 12.10 Example of Synchronous Operation Setting Procedure

For details on PWM modes, see section 12.4.5, PWM Modes.

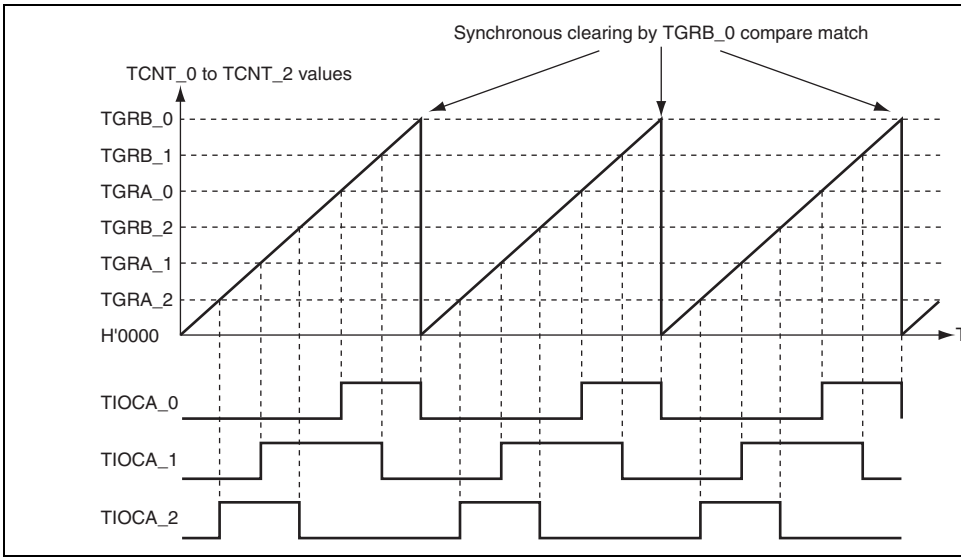


Figure 12.11 Example of Synchronous Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding c
  transferred to the timer general register.

  This operation is illustrated in figure 12.12.

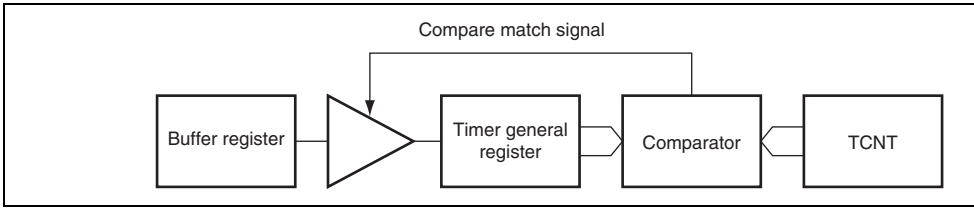


Figure 12.12 Compare Match Buffer Operation

Figure 12.13 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 12.14 shows an example of the buffer operation setting procedure.

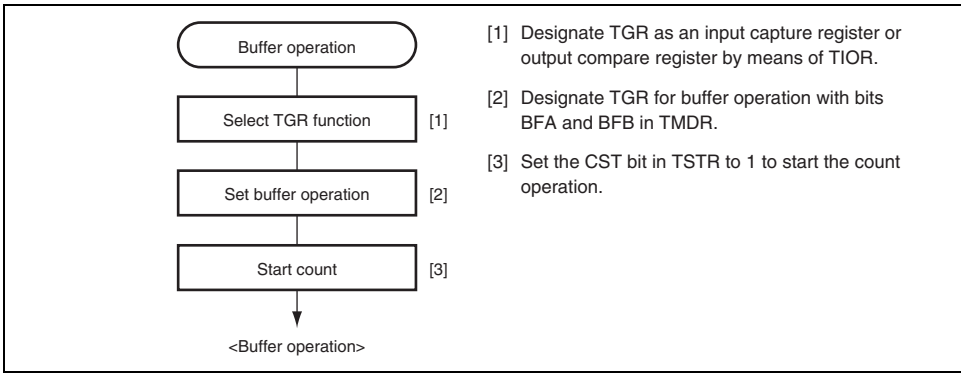


Figure 12.14 Example of Buffer Operation Setting Procedure

For details on PWM modes, see section 12.4.5, PWM Modes.

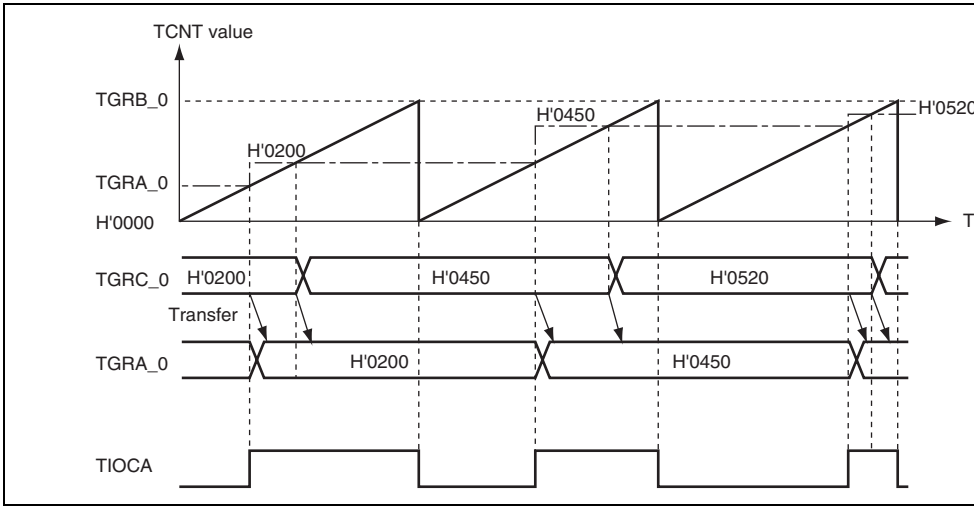


Figure 12.15 Example of Buffer Operation (1)

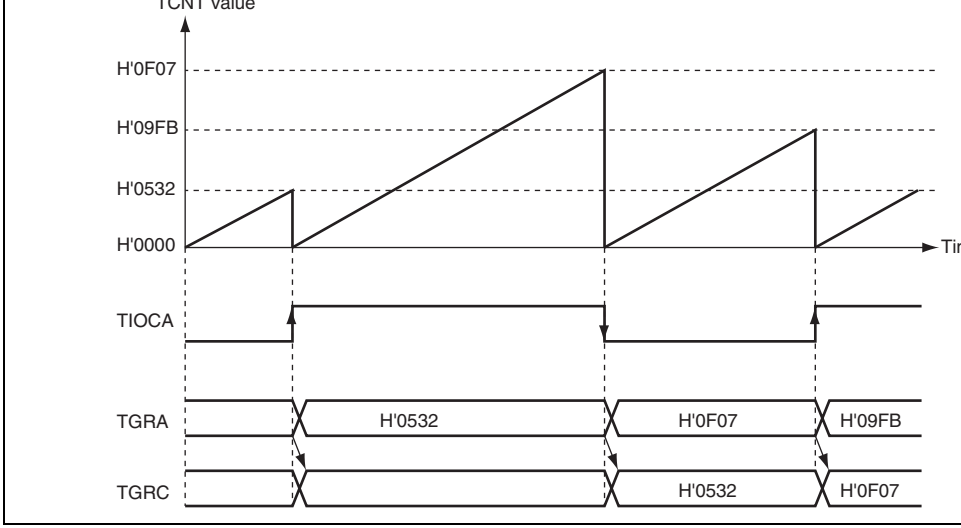


Figure 12.16 Example of Buffer Operation (2)

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

Table 12.30 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5

(1) Example of Cascaded Operation Setting Procedure

Figure 12.17 shows an example of the setting procedure for cascaded operation.

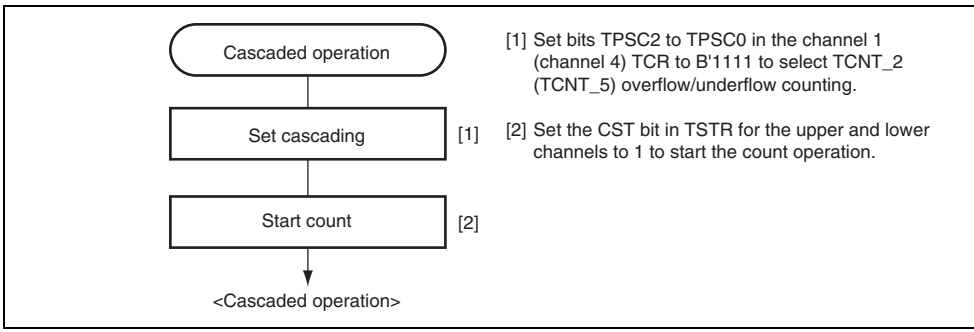


Figure 12.17 Example of Cascaded Operation Setting Procedure

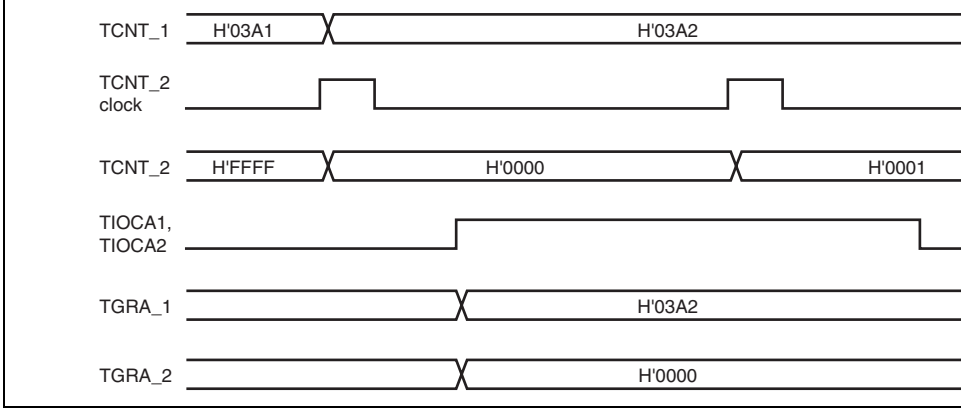


Figure 12.18 Example of Cascaded Operation (1)

Figure 12.19 illustrates the operation when counting upon TCNT_2 overflow/underflow set for TCNT_1, and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow

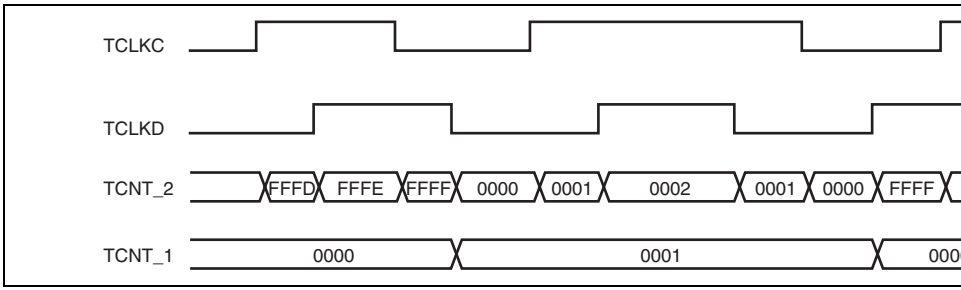


Figure 12.19 Example of Cascaded Operation (2)

There are two PWM modes, as described below.

(a) PWM mode 1

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

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

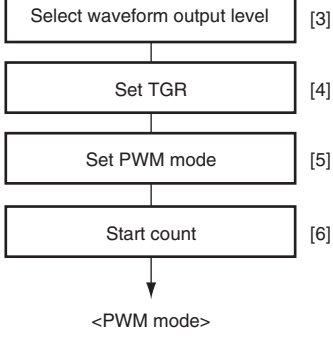
(b) PWM mode 2

PWM output is generated using one TGR as the cycle register and the others as duty cycl
registers. The output specified in TIOR is performed by means of compare matches. Upo
clearing by a synchronous register compare match, the output value of each pin is the init
set in TIOR. If the set values of the cycle and duty cycle registers are identical, the output
does not change when a compare match occurs.

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

1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGRA_4	TIOCA4	TIOCA4
	TGRB_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the cy



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

[4] Set the cycle in TGR selected in [2], and set the duty in the other TGRs.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

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

Figure 12.20 Example of PWM Mode Setting Procedure

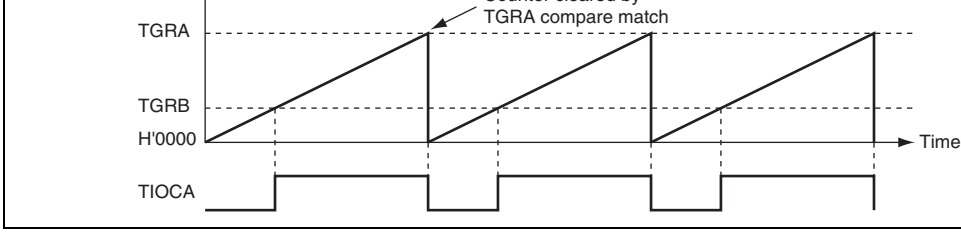


Figure 12.21 Example of PWM Mode Operation (1)

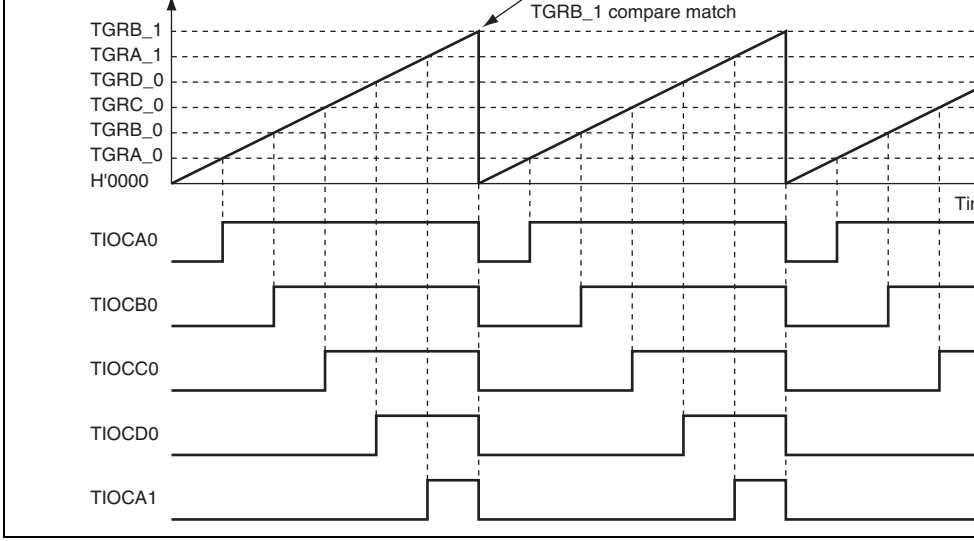


Figure 12.22 Example of PWM Mode Operation (2)

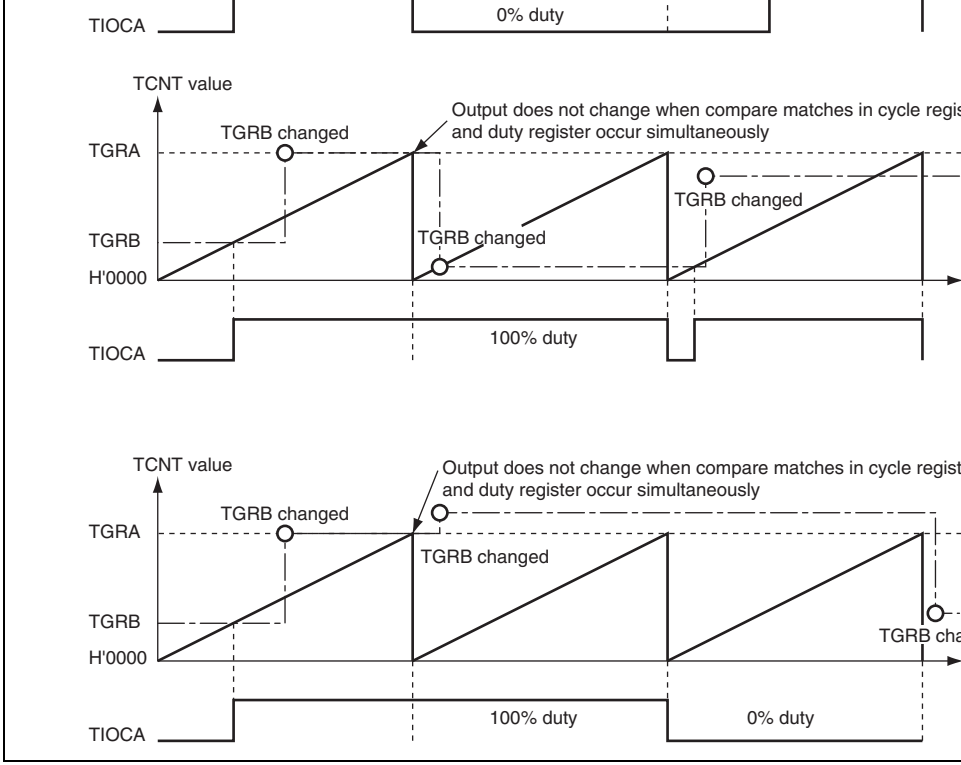


Figure 12.23 Example of PWM Mode Operation (3)

This can be used for two-phase encoder pulse input.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when u
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 12.32 shows the correspondence between external clock pins and channels.

Table 12.32 Clock Input Pins in Phase Counting Mode

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 12.24 shows an example of the phase counting mode setting procedure.

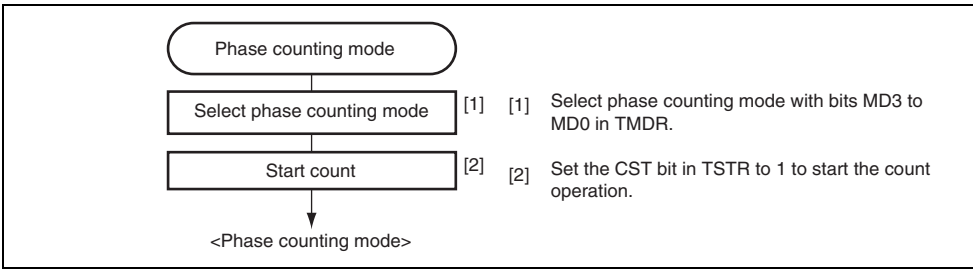


Figure 12.24 Example of Phase Counting Mode Setting Procedure

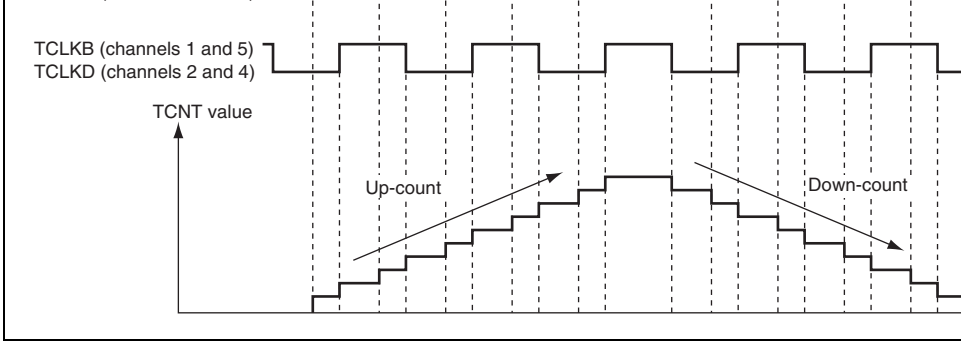


Figure 12.25 Example of Phase Counting Mode 1 Operation

Table 12.33 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

[Legend]

↑: Rising edge

↓: Falling edge

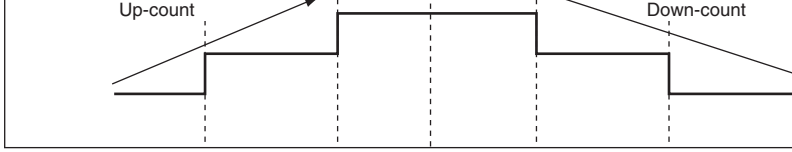


Figure 12.26 Example of Phase Counting Mode 2 Operation

Table 12.34 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	⌐↑	Don't care
Low level	⌐↓	Don't care
⌐↑	Low level	Don't care
⌐↓	High level	Up-count
High level	⌐↓	Don't care
Low level	⌐↑	Don't care
⌐↑	High level	Don't care
⌐↓	Low level	Down-count

[Legend]

⌐↑: Rising edge

⌐↓: Falling edge

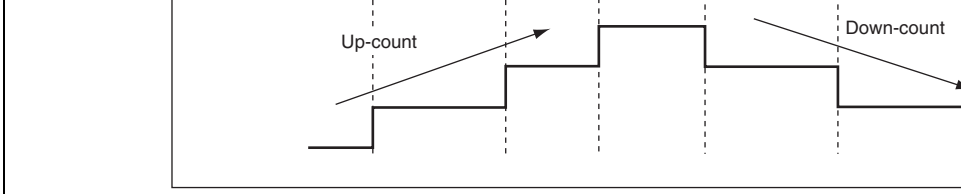


Figure 12.27 Example of Phase Counting Mode 3 Operation

Table 12.35 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

[Legend]

↑: Rising edge

↓: Falling edge

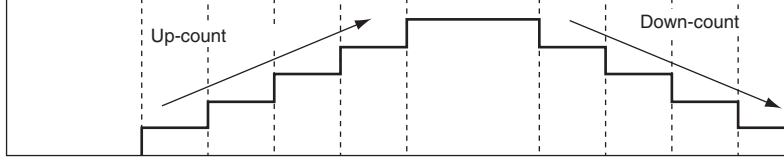


Figure 12.28 Example of Phase Counting Mode 4 Operation

Table 12.36 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]

↑: Rising edge

↓: Falling edge

in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input ca
source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, channel 0 TGRA_
TGRC_0 compare matches are selected as the input capture source, and the up/down-co
values for the control cycles are stored.

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

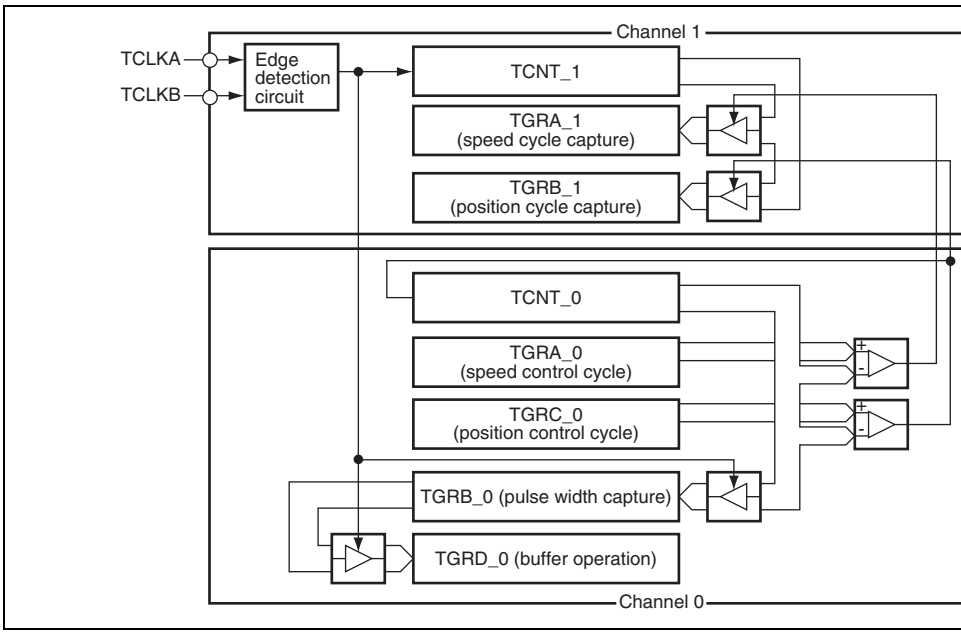


Figure 12.29 Phase Counting Mode Application Example

channel is fixed. For details, see section 6, Interrupt Controller.

Table 12.37 lists the TPU interrupt sources.

Table 12.37 TPU Interrupts

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

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

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

the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR
1 by the occurrence of a TCNT underflow on a channel. The interrupt request is cleared b
clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channe
and 5.

The DMAC can be activated by the TGRA input capture/compare match interrupt for a
For details, see section 9, DMA Controller (DMAC).

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

12.8 A/D Converter Activation

The TGRA input capture/compare match for each channel can activate the A/D converte
A/D converter.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurre
TGRA input capture/compare match on a particular channel, a request to start A/D conv
sent to the A/D converter and ΔΣ A/D converter. If the TPU conversion start trigger has
selected on the A/D converter or ΔΣ A/D converter side at this time, A/D conversion is s

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as
converter conversion start sources, one for each channel.

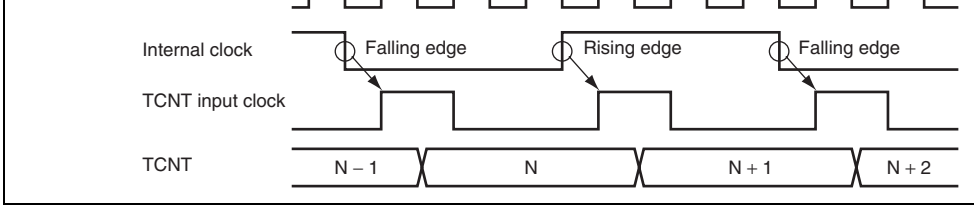


Figure 12.30 Count Timing in Internal Clock Operation

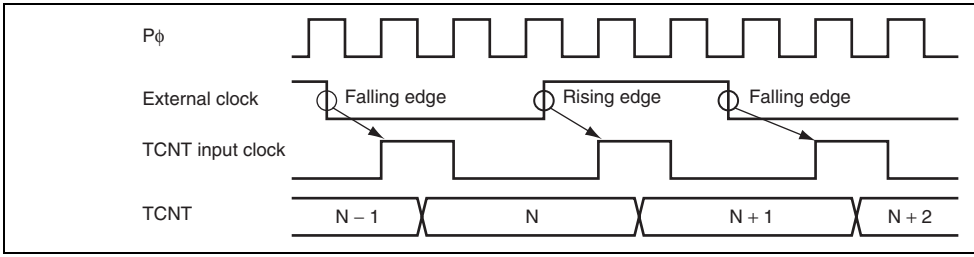


Figure 12.31 Count Timing in External Clock Operation

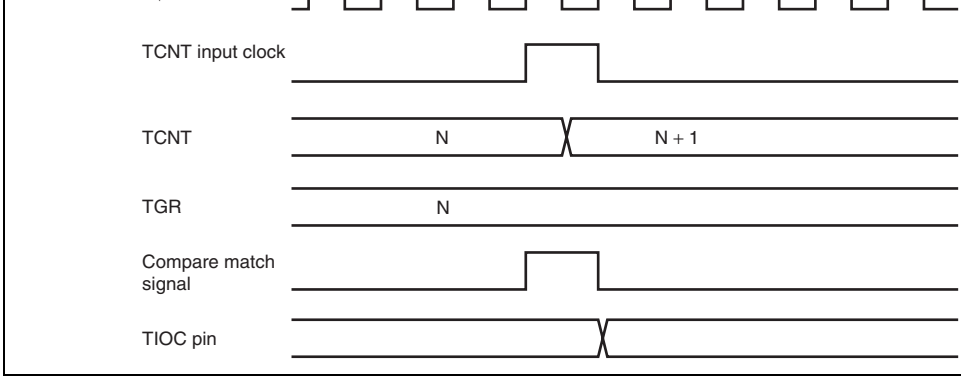


Figure 12.32 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 12.33 shows input capture signal timing.

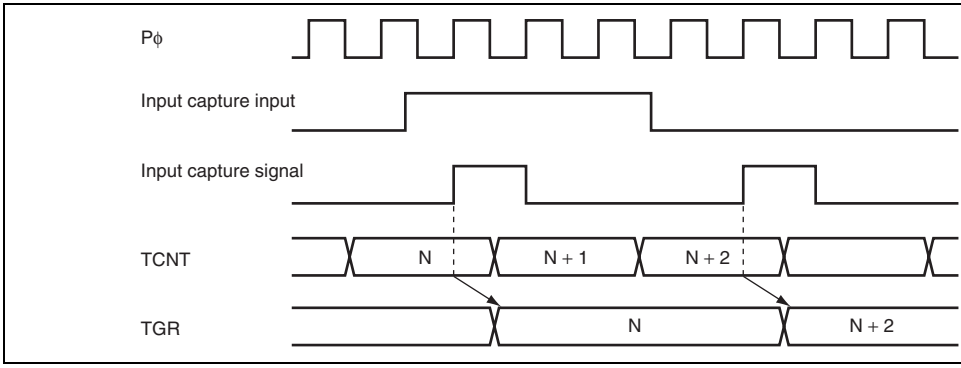


Figure 12.33 Input Capture Input Signal Timing

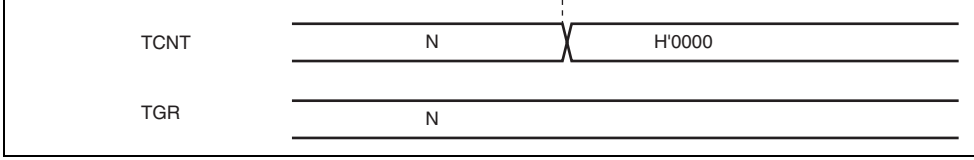


Figure 12.34 Counter Clear Timing (Compare Match)

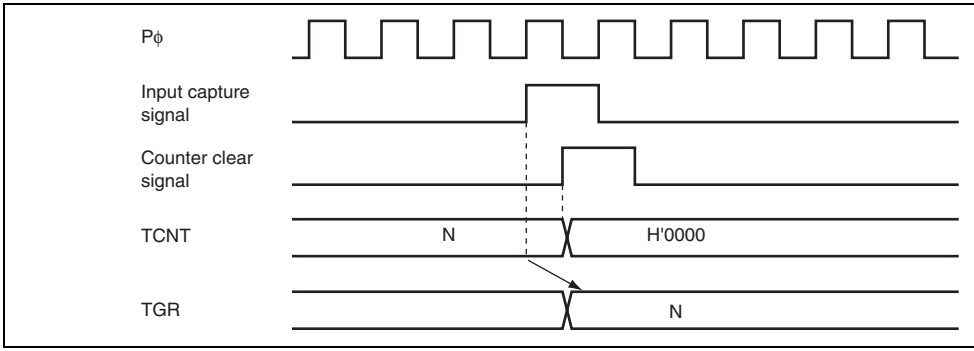


Figure 12.35 Counter Clear Timing (Input Capture)

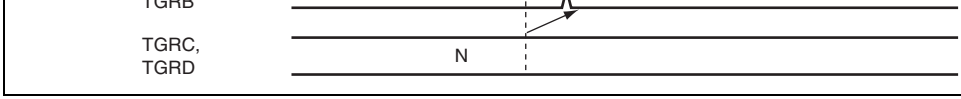


Figure 12.36 Buffer Operation Timing (Compare Match)

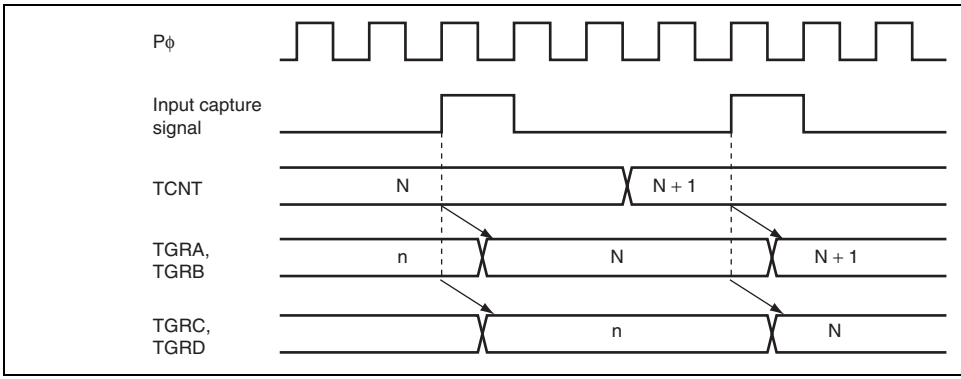


Figure 12.37 Buffer Operation Timing (Input Capture)

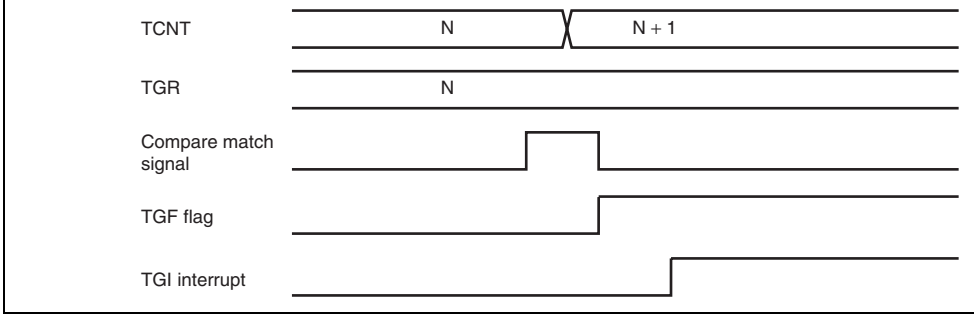


Figure 12.38 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 12.39 shows the timing for setting of the TGF flag in TSR by input capture occurr
the TGI interrupt request signal timing.

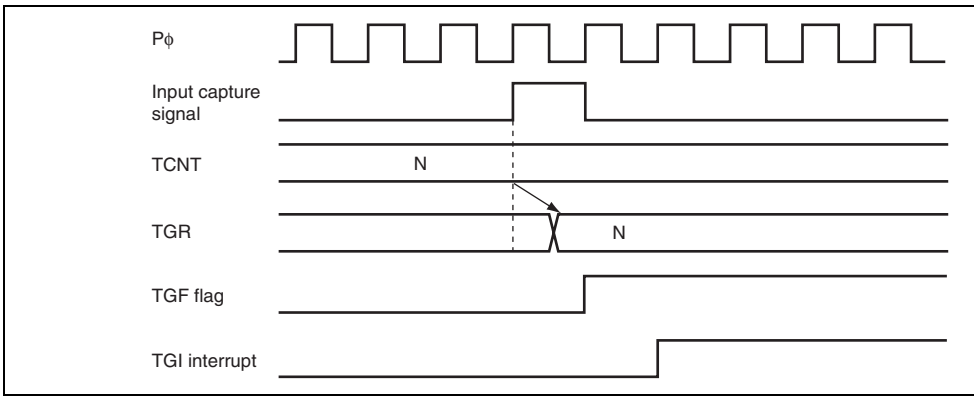


Figure 12.39 TGI Interrupt Timing (Input Capture)

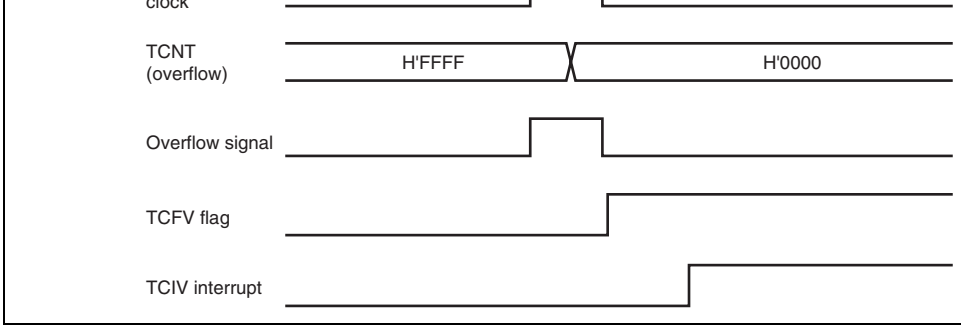


Figure 12.40 TCIV Interrupt Setting Timing

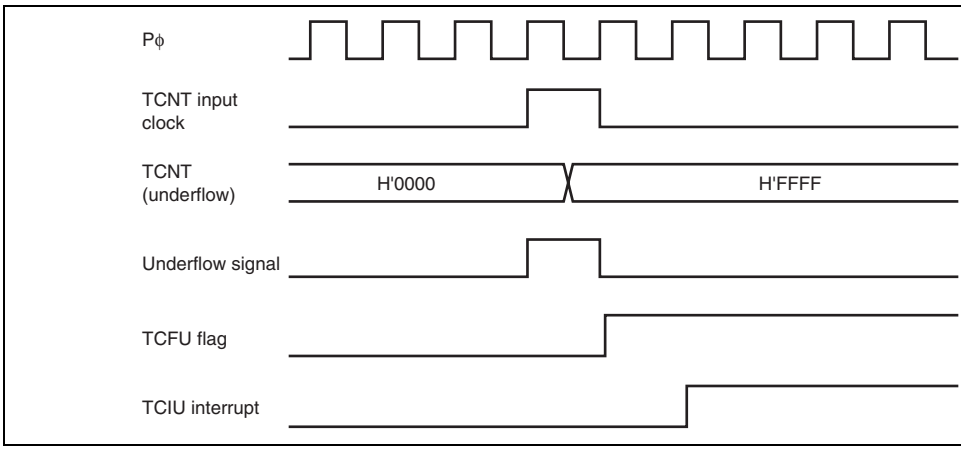


Figure 12.41 TCIU Interrupt Setting Timing

Figure 12.42 Timing for Status Flag Clearing by CPU

The status flag and interrupt request signal are cleared in synchronization with Pϕ after th
DMAC transfer has started, as shown in figure 12.43. If conflict occurs for clearing the st
and interrupt request signal due to activation of multiple DTC or DMAC transfers, it will
to five clock cycles (Pϕ) for clearing them, as shown in figure 12.44. The next transfer re
masked for a longer period of either a period until the current transfer ends or a period fo
clock cycles (Pϕ) from the beginning of the transfer. Note that in the DTC transfer, the st
may be cleared during outputting the destination address.

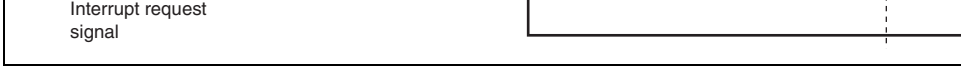


Figure 12.43 Timing for Status Flag Clearing by DTC or DMAC Activation

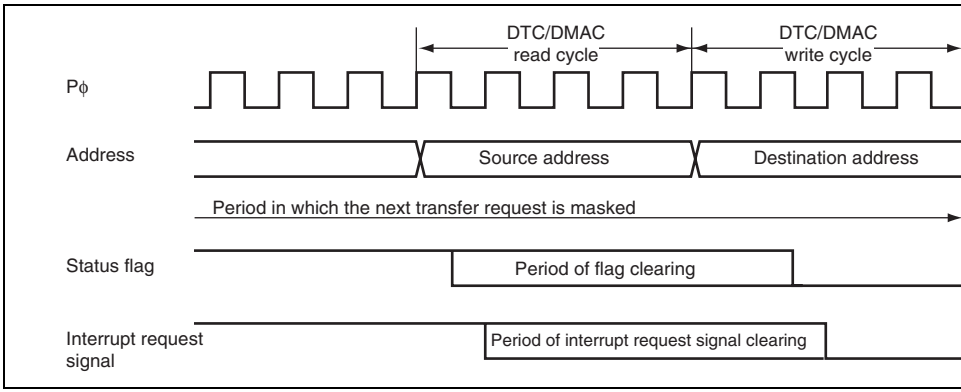


Figure 12.44 Timing for Status Flag Clearing by DTC or DMAC Activation

The input clock pulse width must be at least 1.5 states in the case of single-edge detection least 2.5 states in the case of both-edge detection. The TPU will not operate properly with narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks m least 1.5 states, and the pulse width must be at least 2.5 states. Figure 12.45 shows the inp conditions in phase counting mode.

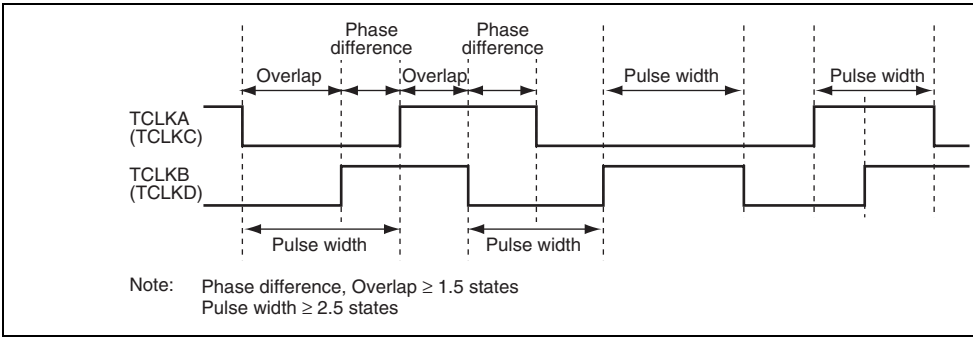


Figure 12.45 Phase Difference, Overlap, and Pulse Width in Phase Counting M

N: TGR set value

12.10.4 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 12.46 shows the timing i
case.

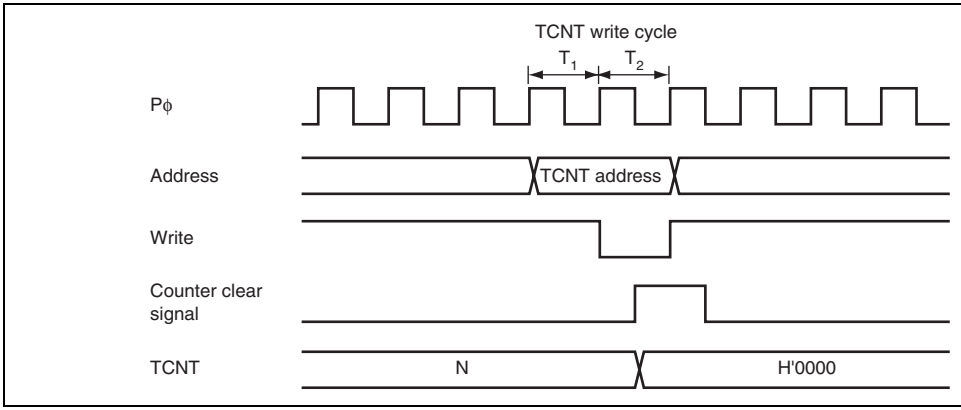


Figure 12.46 Conflict between TCNT Write and Clear Operations

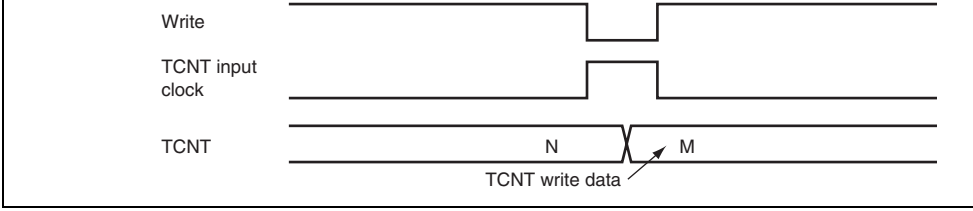


Figure 12.47 Conflict between TCNT Write and Increment Operations

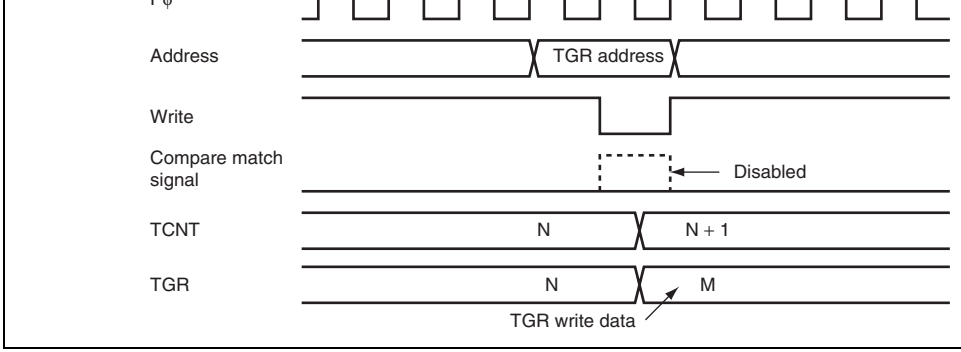


Figure 12.48 Conflict between TGR Write and Compare Match

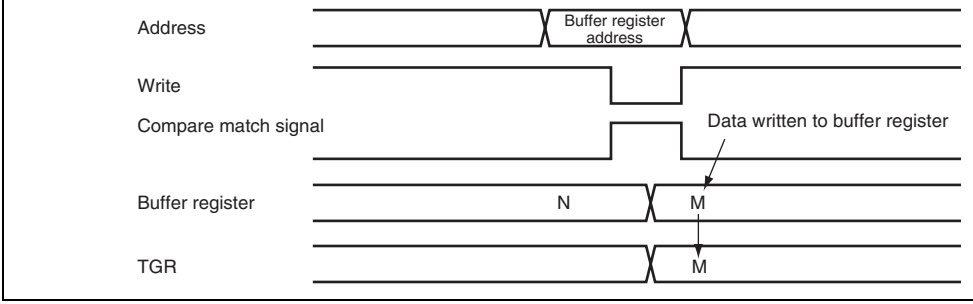


Figure 12.49 Conflict between Buffer Register Write and Compare Match

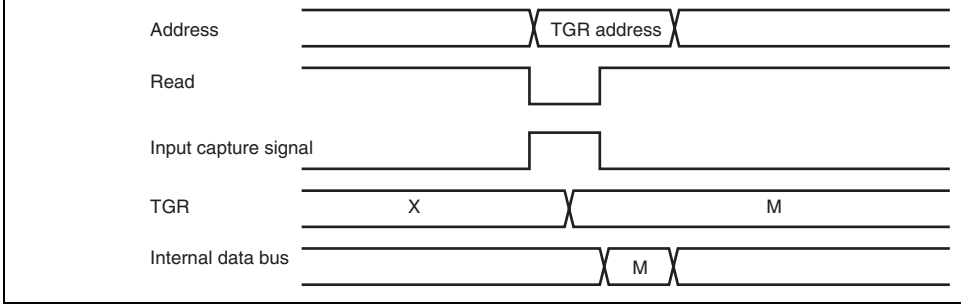


Figure 12.50 Conflict between TGR Read and Input Capture

Figure 12.51 Conflict between TGR Write and Input Capture

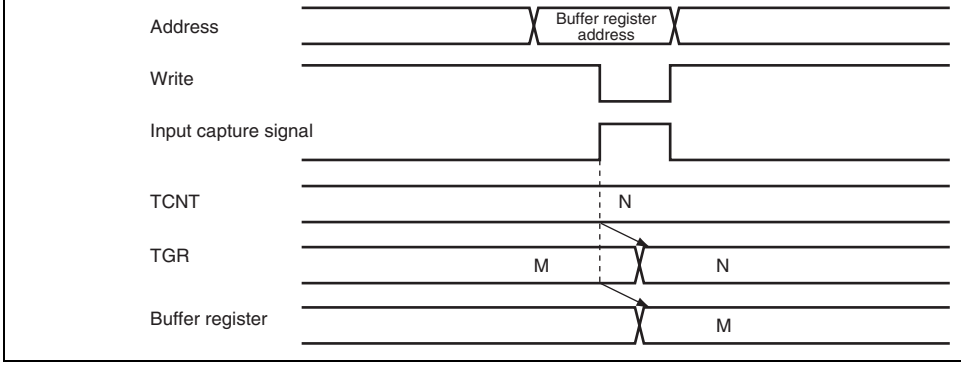


Figure 12.52 Conflict between Buffer Register Write and Input Capture

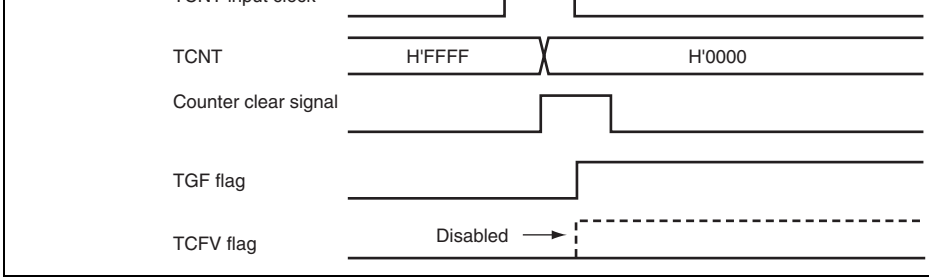


Figure 12.53 Conflict between Overflow and Counter Clearing

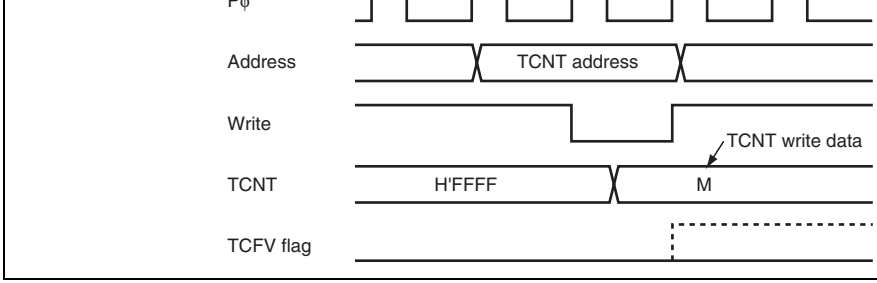


Figure 12.54 Conflict between TCNT Write and Overflow

12.10.13 Multiplexing of I/O Pins

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

12.10.14 Interrupts and Module Stop State

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

- Four output groups
- Selectable output trigger signals
- Non-overlapping mode
- Can operate together with the data transfer controller (DTC) and DMA controller (D
- Inverted output can be set
- Module stop state specifiable

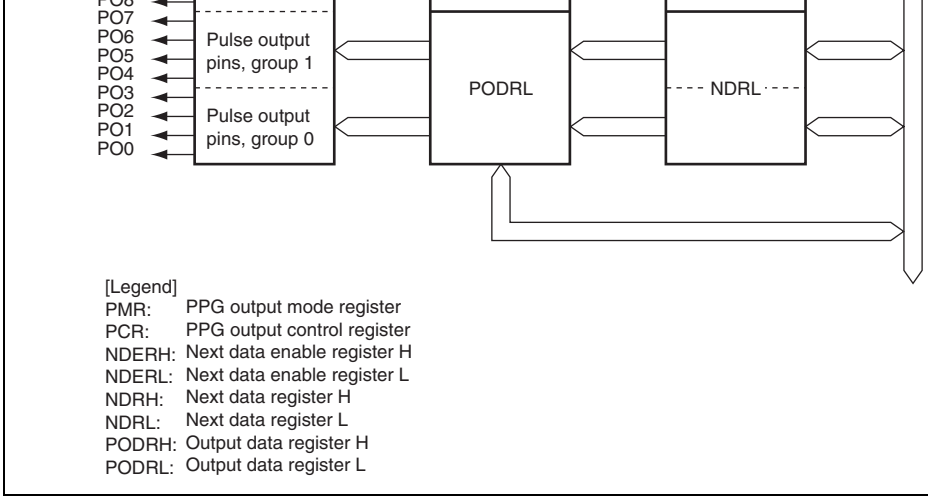


Figure 13.1 Block Diagram of PPG

PO12	Output	
PO11	Output	Group 2 pulse output
PO10	Output	
PO9	Output	
PO8	Output	
PO7	Output	Group 1 pulse output
PO6	Output	
PO5	Output	
PO4	Output	
PO3	Output	Group 0 pulse output
PO2	Output	
PO1	Output	
PO0	Output	

- PPG output control register (PCR)
- PPG output mode register (PMR)

13.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH and NDERL enable/disable pulse output on a bit-by-bit basis.

- NDERH

Bit	7	6	5	4	3	2	1	
Bit Name	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- NDERL

Bit	7	6	5	4	3	2	1	
Bit Name	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1	NDER9	0	R/W	
0	NDER8	0	R/W	

- NDERL

Bit	Bit Name	Initial Value	R/W	Description
7	NDER7	0	R/W	Next Data Enable 7 to 0
6	NDER6	0	R/W	When a bit is set to 1, the value in the correspo
5	NDER5	0	R/W	NDRL bit is transferred to the PODRL bit by the
4	NDER4	0	R/W	output trigger. Values are not transferred from N
3	NDER3	0	R/W	PODRL for cleared bits.
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

- PODRL

Bit	7	6	5	4	3	2	1	
Bit Name	POD7	POD6	POD5	POD4	POD3	POD2	POD2	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- PODRH

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 15 to 8
6	POD14	0	R/W	For bits which have been set to pulse output by I
5	POD13	0	R/W	the output trigger transfers NDRH values to this
4	POD12	0	R/W	during PPG operation. While NDERH is set to 1,
3	POD11	0	R/W	cannot write to this register. While NDERH is cle
2	POD10	0	R/W	initial output value of the pulse can be set.
1	POD9	0	R/W	
0	POD8	0	R/W	

1	POD1	0	R/W
0	POD0	0	R/W

13.3.3 Next Data Registers H, L (NDRH, NDRL)

NDRH and NDRL store the next data for pulse output. The NDR addresses differ depen
whether pulse output groups have the same output trigger or different output triggers.

- NDRH

Bit	7	6	5	4	3	2	1	
Bit Name	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- NDRL

Bit	7	6	5	4	3	2	1	
Bit Name	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

If pulse output groups 2 and 3 have different output triggers, the upper four bits and lo bits are mapped to different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 12
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be r

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be r
3	NDR11	0	R/W	Next Data Register 11 to 8
2	NDR10	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger with PCR.
1	NDR9	0	R/W	
0	NDR8	0	R/W	

3	NDR3	0	R/W	
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

If pulse output groups 0 and 1 have different output triggers, the upper four bits and bits are mapped to different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 4
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigge with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be
3	NDR3	0	R/W	Next Data Register 3 to 0
2	NDR2	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigge with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

Bit	Bit Name	Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 1 and 0
6	G3CMS0	1	R/W	These bits select output trigger of pulse output g
				00: Compare match in TPU channel 0
				01: Compare match in TPU channel 1
				10: Compare match in TPU channel 2
				11: Compare match in TPU channel 3
5	G2CMS1	1	R/W	Group 2 Compare Match Select 1 and 0
4	G2CMS0	1	R/W	These bits select output trigger of pulse output g
				00: Compare match in TPU channel 0
				01: Compare match in TPU channel 1
				10: Compare match in TPU channel 2
				11: Compare match in TPU channel 3
3	G1CMS1	1	R/W	Group 1 Compare Match Select 1 and 0
2	G1CMS0	1	R/W	These bits select output trigger of pulse output g
				00: Compare match in TPU channel 0
				01: Compare match in TPU channel 1
				10: Compare match in TPU channel 2
				11: Compare match in TPU channel 3
1	G0CMS1	1	R/W	Group 0 Compare Match Select 1 and 0
0	G0CMS0	1	R/W	These bits select output trigger of pulse output g
				00: Compare match in TPU channel 0
				01: Compare match in TPU channel 1
				10: Compare match in TPU channel 2
				11: Compare match in TPU channel 3

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

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for puls group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for puls group 2. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 1 Inversion Selects direct output or inverted output for puls group 1. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 0 Inversion Selects direct output or inverted output for puls group 0. 0: Inverted output 1: Direct output

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

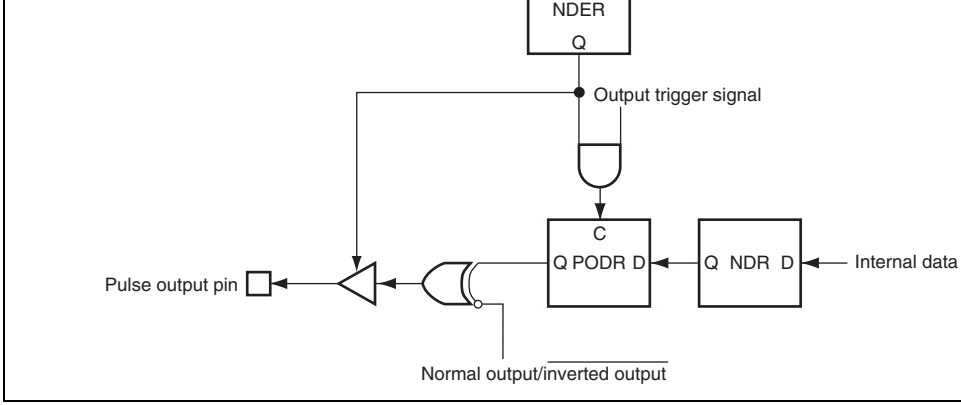


Figure 13.2 Schematic Diagram of PPG

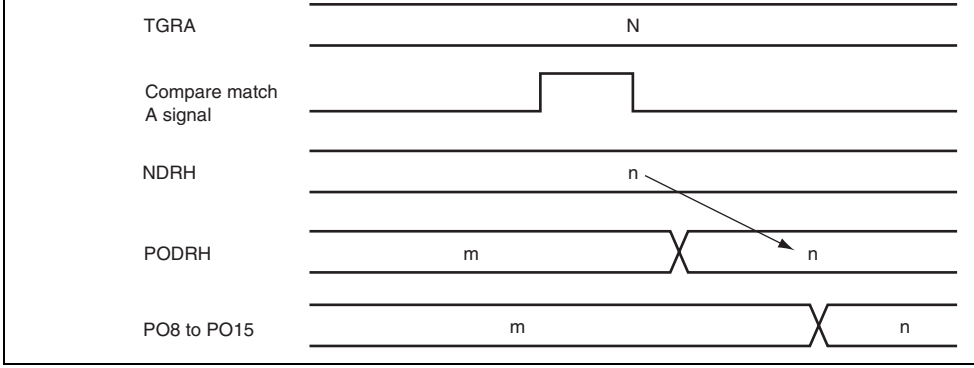
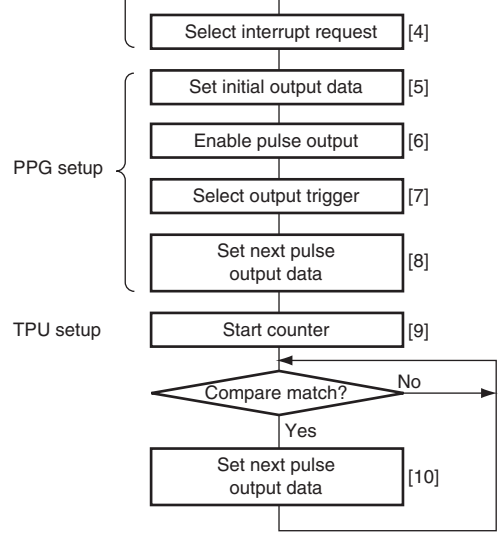


Figure 13.3 Timing of Transfer and Output of NDR Contents (Example)



[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 ND

[9] Set the CST bit in TSTR to 1 to start t TCNT counter.

[10] At each TGIA interrupt, set the next output values in NDR.

Figure 13.4 Setup Procedure for Normal Pulse Output (Example)

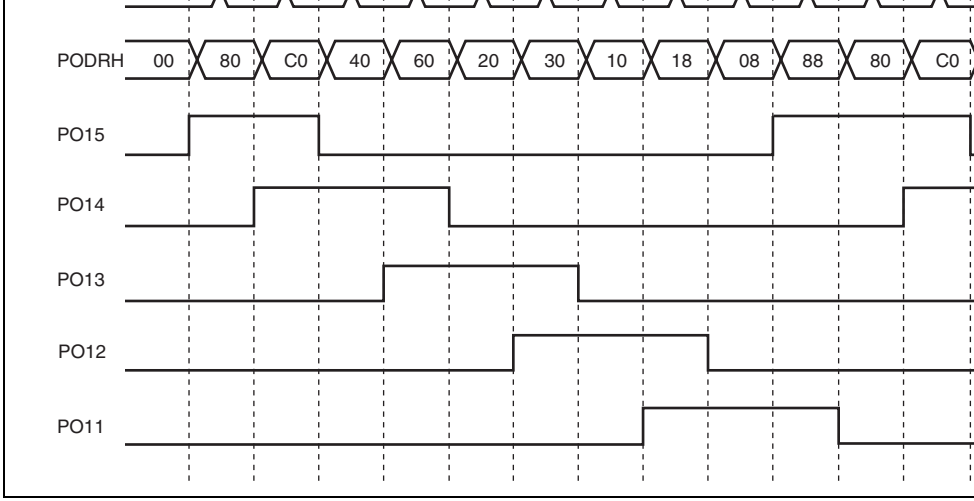


Figure 13.5 Normal Pulse Output Example (5-Phase Pulse Output)

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

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

13.4.4 Non-Overlapping Pulse Output

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

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

Figure 13.6 illustrates the non-overlapping pulse output operation.

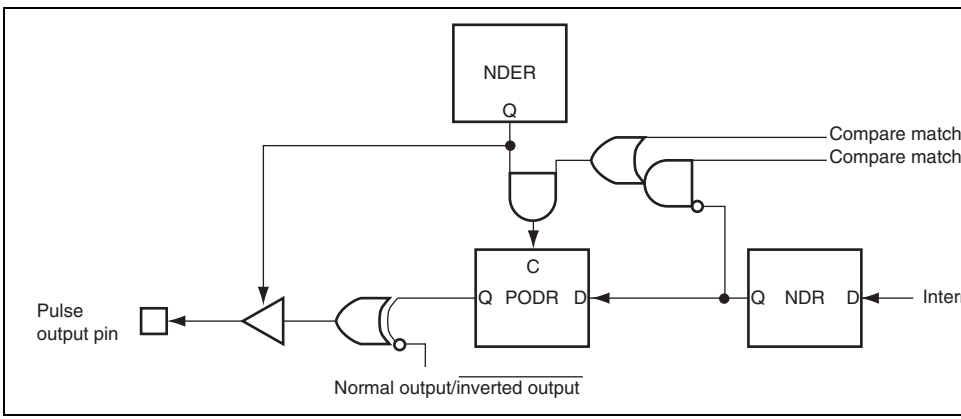


Figure 13.6 Non-Overlapping Pulse Output

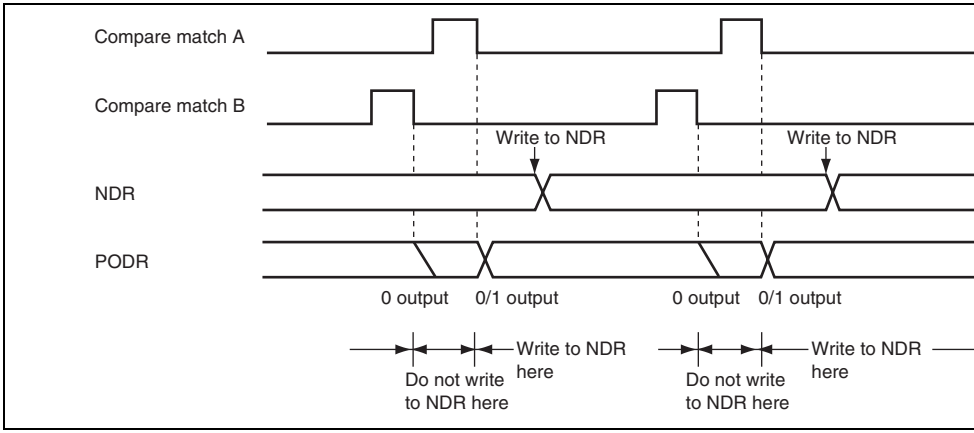
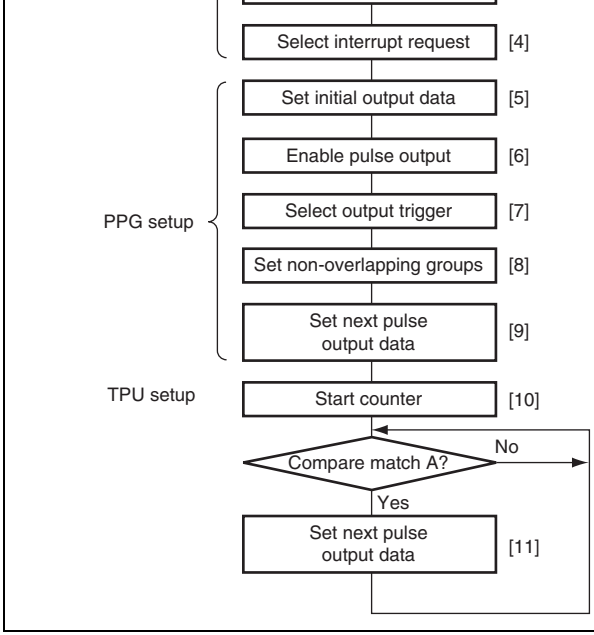


Figure 13.7 Non-Overlapping Operation and NDR Write Timing



counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. T DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the bits in NDER for the pins to b used for pulse output to 1.

[7] Select the TPU compare match even be used as the pulse output trigger in PCR.

[8] In PMR, select the groups that will operate in non-overlapping mode.

[9] Set the next pulse output values in N

[10] Set the CST bit in TSTR to 1 to start TCNT counter.

[11] At each TGIA interrupt, set the next output values in NDR.

Figure 13.8 Setup Procedure for Non-Overlapping Pulse Output (Exampl

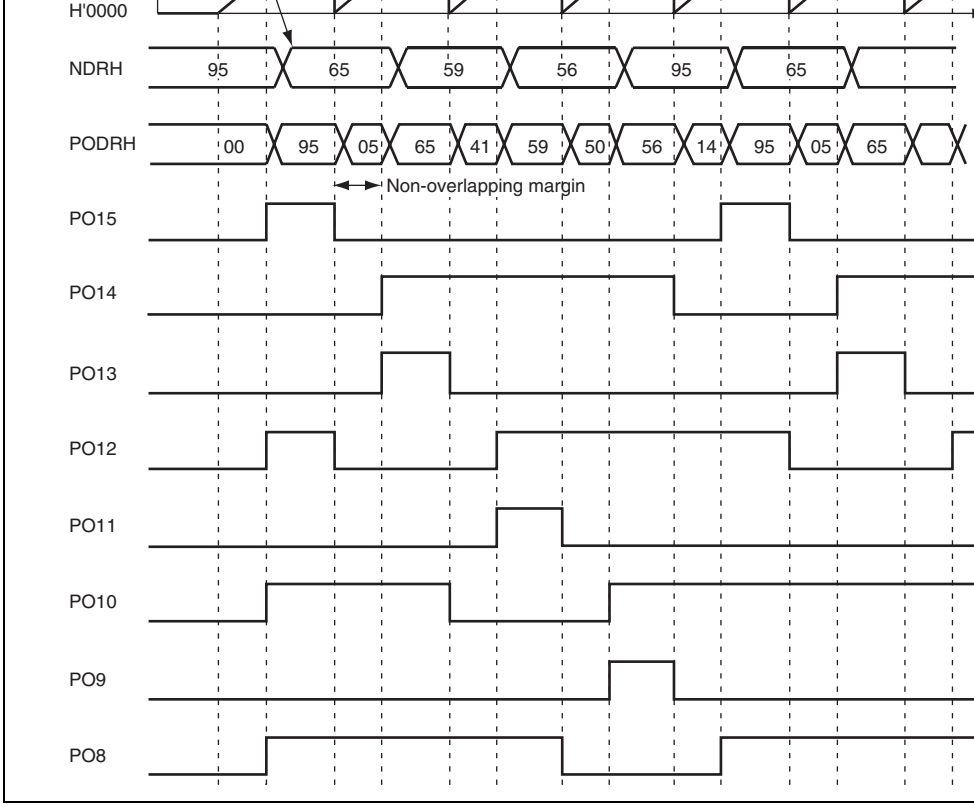


Figure 13.9 Non-Overlapping Pulse Output Example (4-Phase Complementa

to 1 (the change from 0 to 1 is delayed by the value set in TGRA).

The TGIA interrupt handling routine writes the next output data (H'65) to NDRH.

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

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

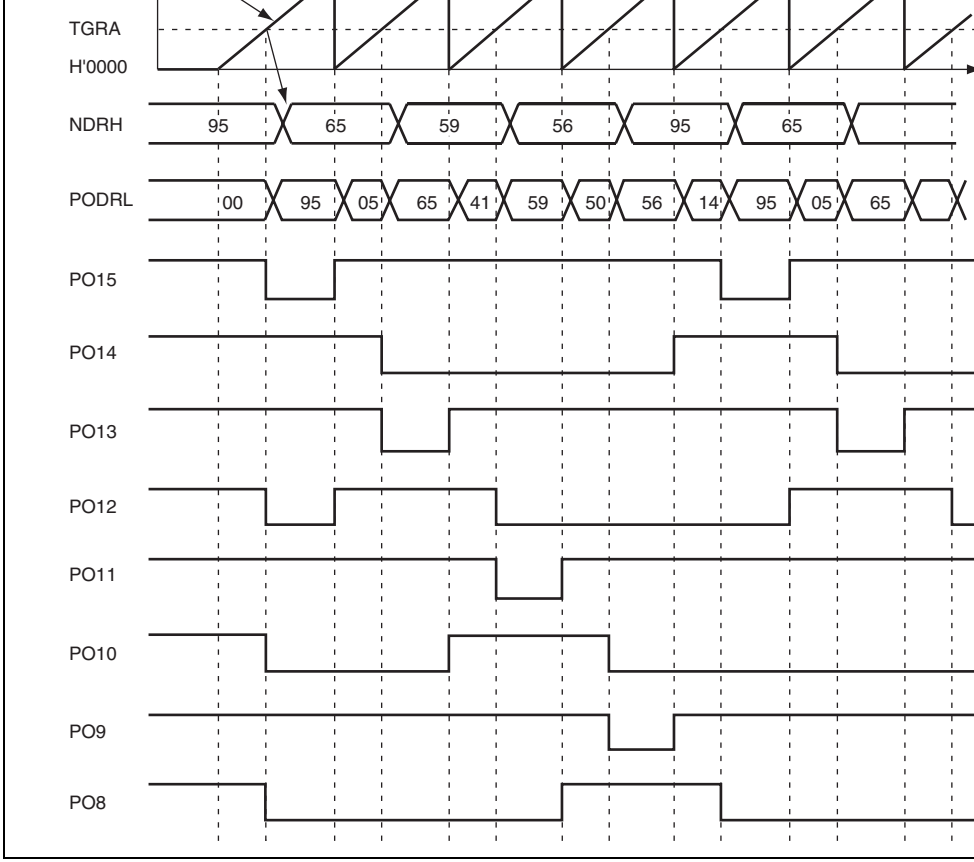


Figure 13.10 Inverted Pulse Output (Example)

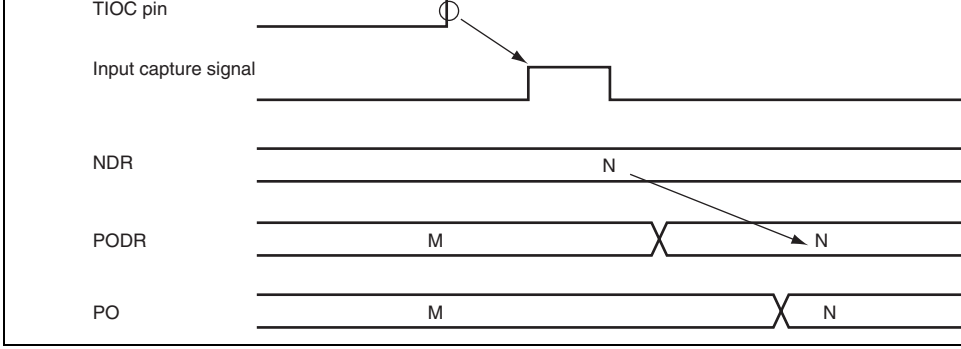


Figure 13.11 Pulse Output Triggered by Input Capture (Example)

Pins PO0 to PO15 are also used for other peripheral functions such as the TPU. When ou
another peripheral function is enabled, the corresponding pins cannot be used for pulse ou
Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of t
of the pins.

Pin functions should be changed only under conditions in which the output trigger event
occur.

have the same functions. Unit 2 and unit 3 have the same functions as unit 0 and unit 1, units 2 and 3 do not have the TMRI and TMCI pins.

14.1 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/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 indep
- Generation of trigger to start A/D converter conversion (available in unit 0 and unit
- Generation of trigger to start ΔΣ A/D converter conversion (available in unit 0 and u
- Module stop state specifiable

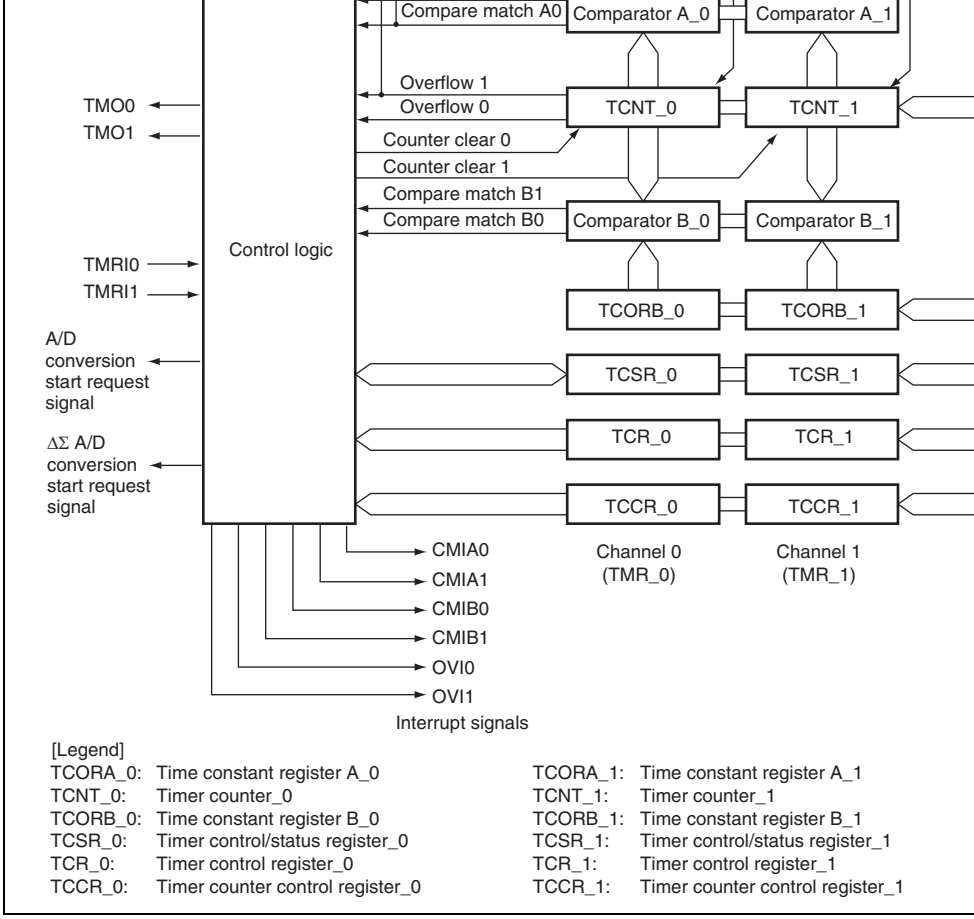


Figure 14.1 Block Diagram of 8-Bit Timer Module (Unit 0)

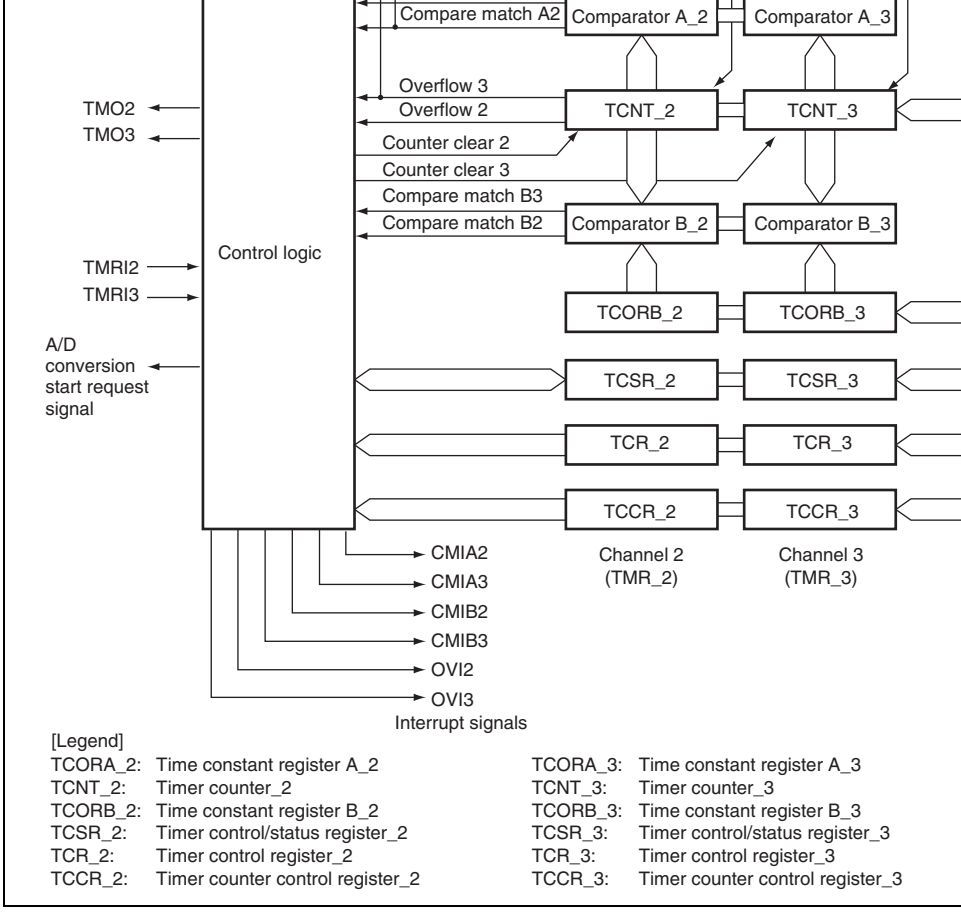


Figure 14.2 Block Diagram of 8-Bit Timer Module (Unit 1)

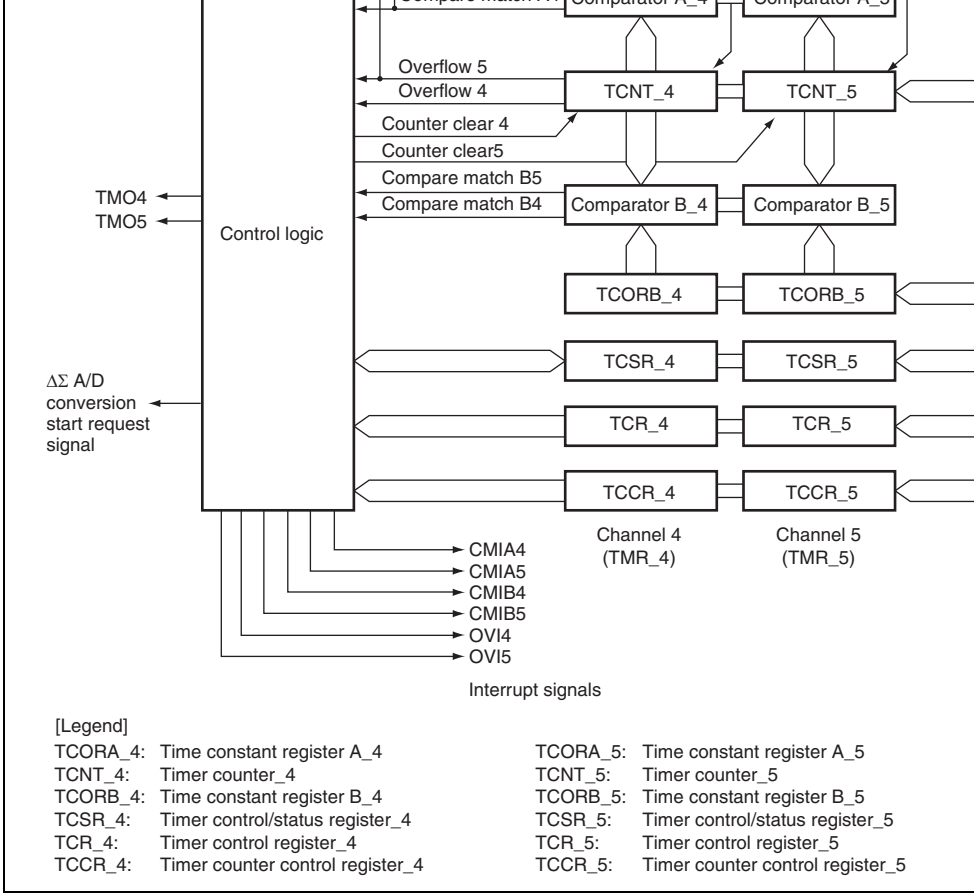
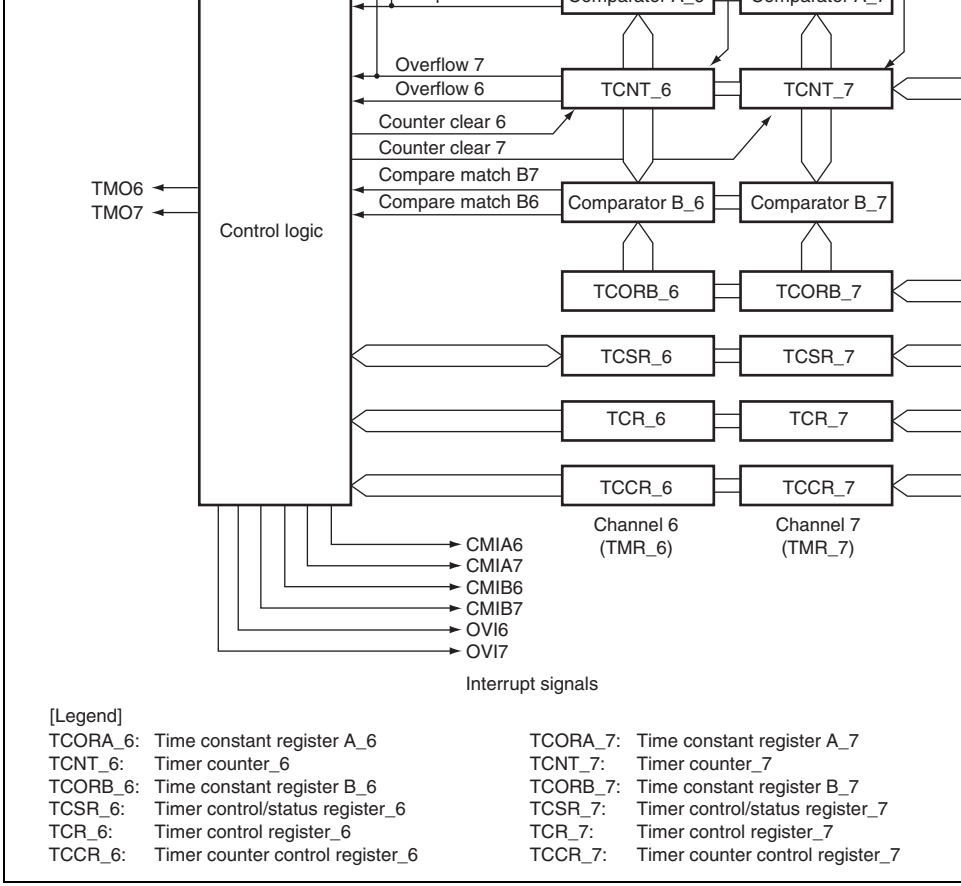


Figure 14.3 Block Diagram of 8-Bit Timer Module (Unit 2)



[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

Figure 14.4 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	Timer output pin	TMO4	Output	Outputs compare match
	5	Timer output pin	TMO5	Output	Outputs compare match
3	6	Timer output pin	TMO6	Output	Outputs compare match
	7	Timer output pin	TMO7	Output	Outputs compare match

— 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	R

14.3.2 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	R

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

14.3.4 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

				1:
				0: OVF interrupt requests (OVI) are disabled
				1: OVF interrupt requests (OVI) are enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0*[1]
3	CCLR0	0	R/W	These bits select the method by which TCNT is
				00: Clearing is disabled
				01: Cleared by compare match A
				10: Cleared by compare match B
				11: Cleared at rising edge (TMRIS in TCCR is c
				0) of the external reset input or when the ex
				reset input is high (TMRIS in TCCR is set to
2	CKS2	0	R/W	Clock Select 2 to 0*[1]
1	CKS1	0	R/W	These bits select the clock input to TCNT and c
0	CKS0	0	R/W	condition. See table 14.2.

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 11, I/O Por

2. In unit 2 and unit 3, one interrupt signal is used for CMIEB or CMIEA. For det section 14.7, 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 14.2.

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 11, I/O Ports.

				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 falling edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	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 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 falling edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	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 11, I/O Ports.

				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_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 falling 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.

				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 falling 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 falling 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

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 this flag is cleared by the CPU in the handling, be sure to read the flag after writi • When the DTC is activated by a CMIB interr the DISEL bit in MRB of the DTC is 0*[3]

				the DISEL bit in MRB in 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] When writing 0 after reading OVF = 1 (When this flag is cleared by the CPU in the inte handling, be sure to read the flag after writing 0 t
4	ADTE	0	R/W	A/D Trigger Enable*3 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 wh 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.
3. For the corresponding A/D converter channels, see section 18, A/D Converte

				while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) • When the DTC is activated by a CMIB interru the DISEL bit in MRB of the DTC is 0*3
6	CMFA	0	R/(W)*1	Compare Match Flag A [Setting condition] • When TCNT matches TCORA [Clearing conditions] • When writing 0 after reading CMFA = 1 (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) • 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 writin while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)

				11: Output is inverted when compare match B o (toggle output)
1	OS1	0	R/W	Output Select 1 and 0*2
0	OS0	0	R/W	These bits select a method of TMO pin output w compare match A of TCORA and TCNT occurs 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A 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 un 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.

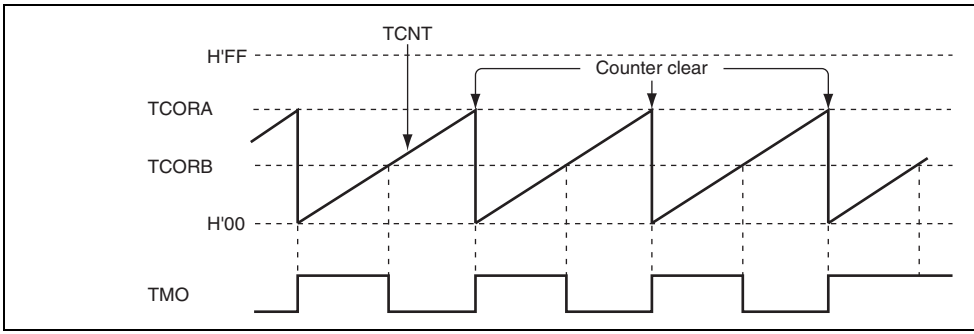


Figure 14.5 Example of Pulse Output

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

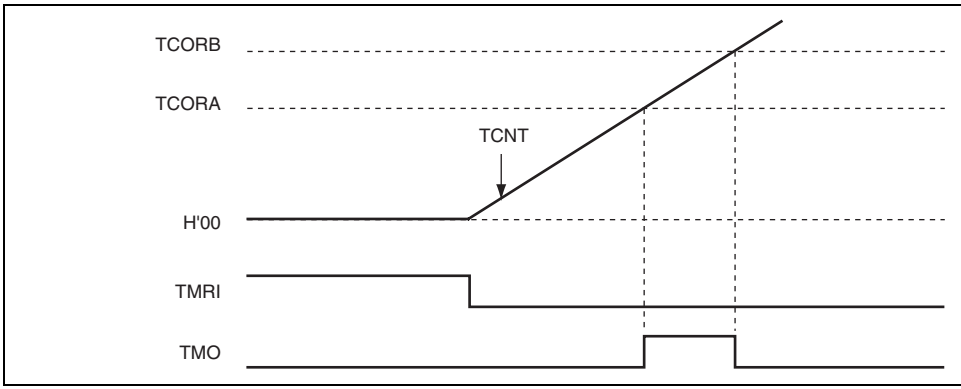


Figure 14.6 Example of Reset Input

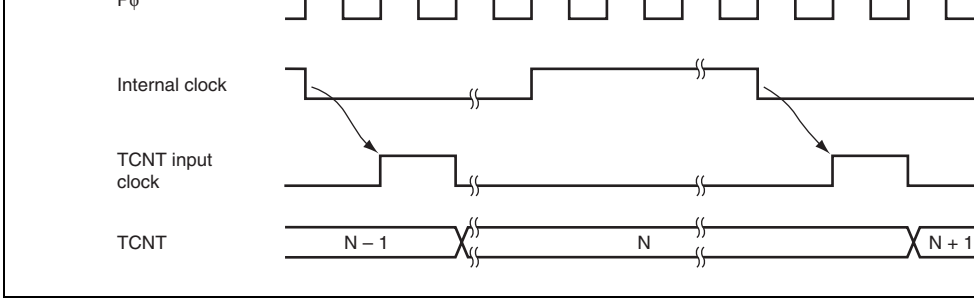


Figure 14.7 Count Timing for Internal Clock Input

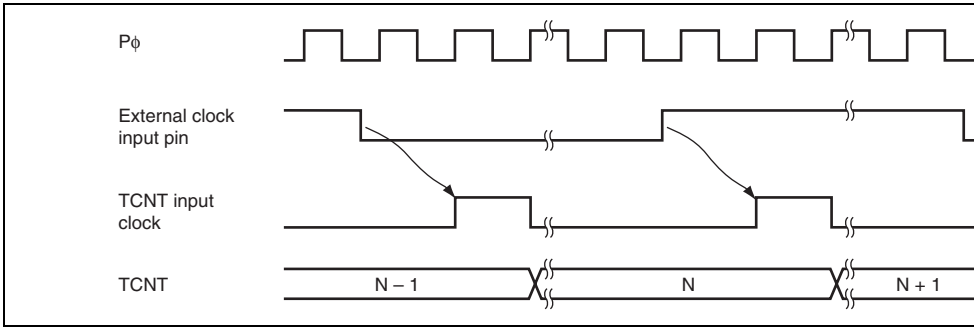


Figure 14.8 Count Timing for External Clock Input

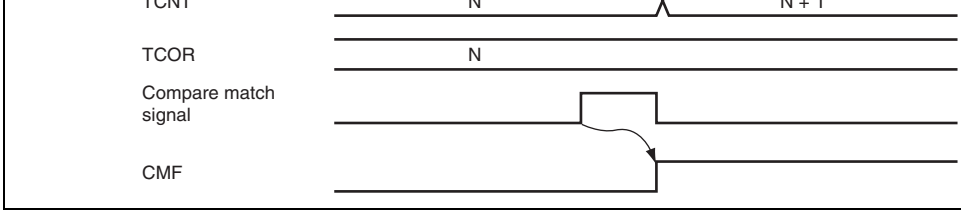


Figure 14.9 Timing of CMF Setting at Compare Match

14.5.3 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 14.10 shows the timing when the timer output is toggled by the match A signal.

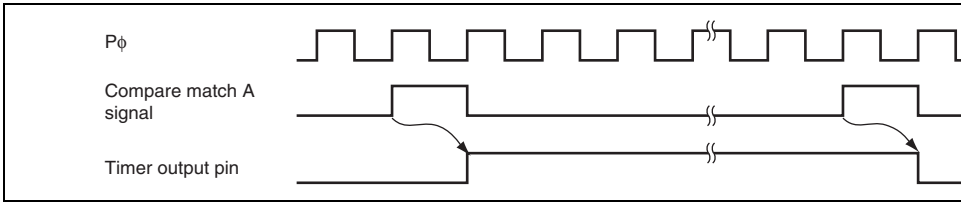


Figure 14.10 Timing of Toggled Timer Output at Compare Match A

Figure 14.11 Timing of Counter Clear by Compare Match

14.5.5 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
14.12 and Figure 14.13 shows the timing of this operation.

Note: * Clearing by an external reset is available only in units 0 and 1.

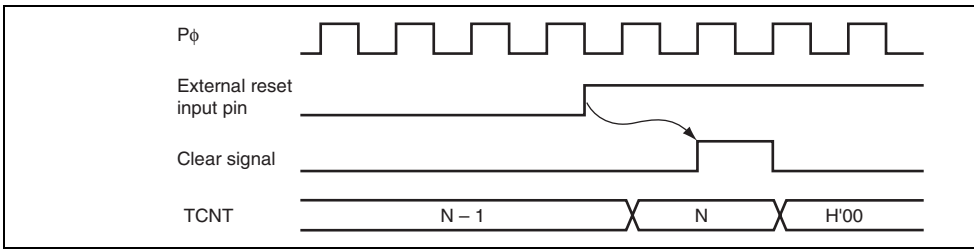


Figure 14.12 Timing of Clearance by External Reset (Rising Edge)

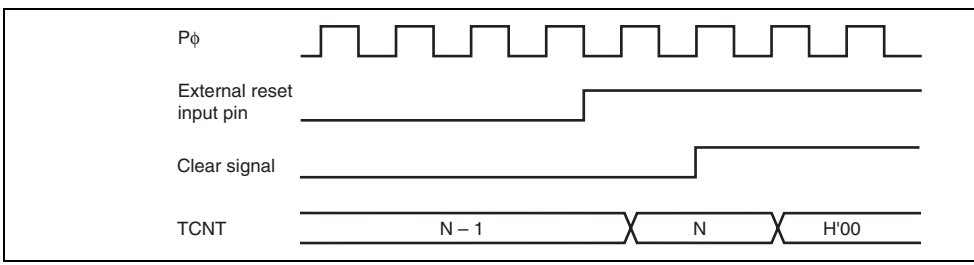


Figure 14.13 Timing of Clearance by External Reset (High Level)

Figure 14.14 Timing of OVF Setting

14.6 Operation with Cascaded Connection

If the bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers o channels are cascaded. With this configuration, a single 16-bit timer could be used (16-b mode) or compare matches of the 8-bit channel 0 could be counted by the timer of chan (compare match count mode).

14.6.1 16-Bit Counter Mode

When the bits CKS2 to CKS0 in TCR_0 are set to B'100, the timer functions as a single timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bi

(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 occur

(2) Counter Clear Specification

- If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare m occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter the TMRI0 pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits cleared independently.

flag, generation of interrupts, output from the TMO pin, and counter clear are in accordan
the settings for each channel.

14.7 Interrupt Sources

14.7.1 Interrupt Sources and DTC Activation

- Interrupt in units 0 to 3

  There are three interrupt sources for the 8-bit timers (TMR_0 to TMR_7): CMIA, CM
  OVI. Their interrupt sources and priorities are shown in table 14.6. Each interrupt sou
  enabled or disabled by the corresponding interrupt enable bit in TCR or TCSR, and
  independent interrupt requests are sent for each to the interrupt controller. It is also po
  activate the DTC by means of CMIA and CMIB interrupts.

CMIA2	CMIA2	TCORA_2 compare match	CMFA	Possible	Hig
CMIB2	CMIB2	TCORB_2 compare match	CMFB	Possible	
OVI2	OVI2	TCNT_2 overflow	OVF	Not possible	Lo
CMIA3	CMIA3	TCORA_3 compare match	CMFA	Possible	Hig
CMIB3	CMIB3	TCORB_3 compare match	CMFB	Possible	
OVI3	OVI3	TCNT_3 overflow	OVF	Not possible	Lo
CMIA4	CMIA4	TCORA_4 compare match	CMFA	Possible	Hig
CMIB4	CMIB4	TCORB_4 compare match	CMFB	Possible	
OVI4	OVI4	TCNT_4 overflow	OVF	Not possible	Lo
CMIA5	CMIA5	TCORA_5 compare match	CMFA	Possible	Hig
CMIB5	CMIB5	TCORB_5 compare match	CMFB	Possible	
OVI5	OVI5	TCNT_5 overflow	OVF	Not possible	Lo
CMIA6	CMIA6	TCORA_6 compare match	CMFA	Possible	Hig
CMIB6	CMIB6	TCORB_6 compare match	CMFB	Possible	
OVI6	OVI6	TCNT_6 overflow	OVF	Not possible	Lo
CMIA7	CMIA7	TCORA_7 compare match	CMFA	Possible	Hig
CMIB7	CMIB7	TCORB_7 compare match	CMFB	Possible	
OVI7	OVI7	TCNT_7 overflow	OVF	Not possible	Lo

f: Counter frequency
ϕ: Operating frequency
N: TCOR value

14.8.2 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 14.15.

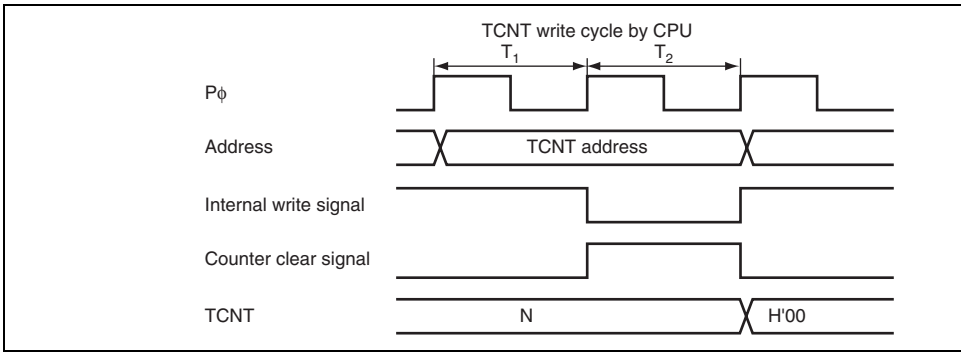


Figure 14.15 Conflict between TCNT Write and Clear

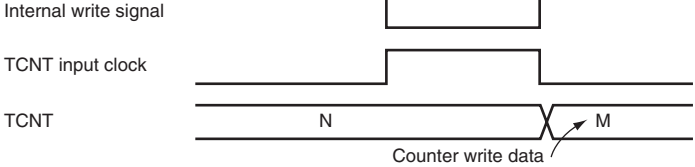


Figure 14.16 Conflict between TCNT Write and Increment

14.8.4 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 14.17.

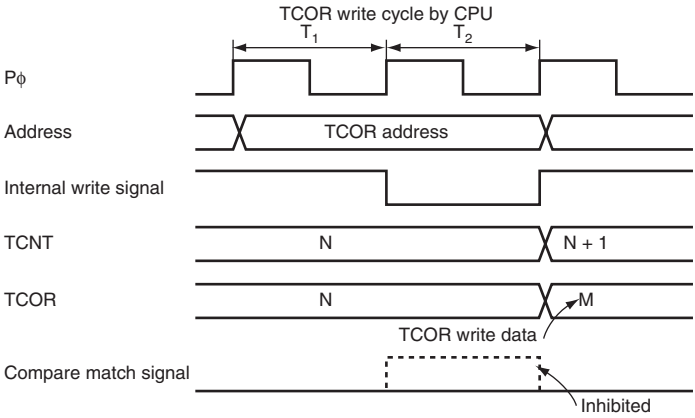


Figure 14.17 Conflict between TCOR Write and Compare Match

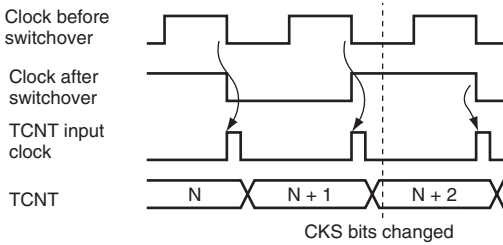
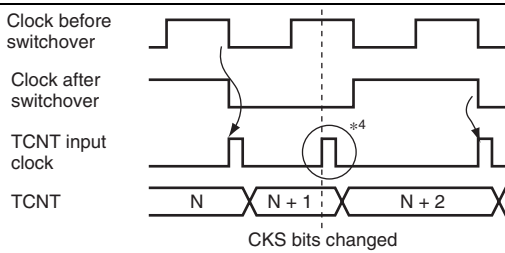
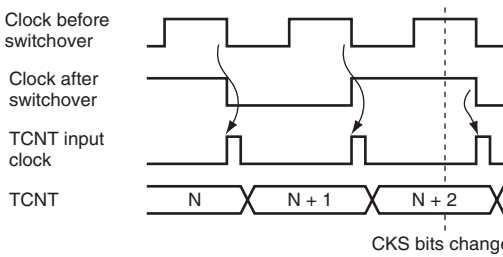
0-output	
No change	Lo

14.8.6 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switche
14.8 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 14.8 assumes that the falling edge is selected. I
signal levels of the clocks before and after switching change from high to low as shown
the change is considered as the falling edge. Therefore, a TCNT clock pulse is generated
TCNT is incremented. This is similar to when the rising edge is selected.

The erroneous 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 24, Power-Down Modes.

14.8.9 Interrupts in Module Stop State

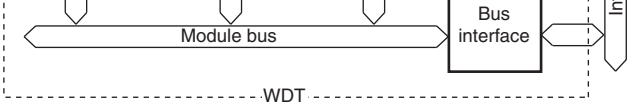
If the TMR enters the module stop state after it has requested an interrupt, the source of
to the CPU or the DTC activation source cannot be cleared. TMR interrupts should there
disabled before the TMR enters the module stop state.

15.1 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 wh
    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 15.1 Block Diagram of WDT

15.2 Input/Output Pin

Table 15.1 shows the WDT pin configuration.

Table 15.1 Pin Configuration

Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs a counter overflow signal in watchdog timer mode

15.3.1 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	

15.3.2 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/$\overline{IT}$	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 by writin while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
6	WT/$\overline{IT}$	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 $\overline{WDTOVF}$ 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	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Select the clock source to be input to TCNT. The
0	CKS0	0	R/W	cycle for Pϕ = 20 MHz is indicated in parenthese 000: Clock Pϕ/2 (cycle: 25.6 μs) 001: Clock Pϕ/64 (cycle: 819.2 μs) 010: Clock Pϕ/128 (cycle: 1.6 ms) 011: Clock Pϕ/512 (cycle: 6.6 ms) 100: Clock Pϕ/2048 (cycle: 26.2 ms) 101: Clock Pϕ/8192 (cycle: 104.9 ms) 110: Clock Pϕ/32768 (cycle: 419.4 ms) 111: Clock Pϕ/131072 (cycle: 1.68 s)

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

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 operati 0: LSI is not reset even if TCNT overflows (Tho LSI is not reset, TCNT and TCSR in WDT ar 1: LSI is reset if TCNT overflows

15.4 Operation

15.4.1 Watchdog Timer Mode

To use the WDT in watchdog timer mode, set both the WT/$\overline{IT}$ and TME bits in TCSR to

During watchdog timer operation, if TCNT overflows without being rewritten because of crash or other error, the $\overline{WDTOVF}$ signal is output. This ensures that TCNT does not ove while the system is operating normally. Software must prevent TCNT overflows by rewri TCNT value (normally H'00 is written) before overflow occurs. This $\overline{WDTOVF}$ signal ca to reset the LSI internally in watchdog timer mode.

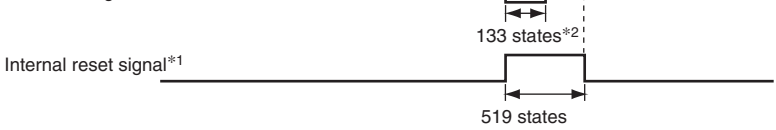
If TCNT overflows when the RSTE bit in RSTCSR is set to 1, a signal that resets this LS internally is generated at the same time as the $\overline{WDTOVF}$ signal. If a reset caused by a sig to the $\overline{RES}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{RES}$ pi has priority and the WOVF bit in RSTCSR is cleared to 0.

The $\overline{WDTOVF}$ signal is output for 133 cycles of Pϕ when RSTE = 1 in RSTCSR, and for cycles of Pϕ when RSTE = 0 in RSTCSR. The internal reset signal is output for 519 cycl

When RSTE = 1, an internal reset signal is generated. Since the system clock control regi (SCKCR) is initialized, the multiplication ratio of Pϕ becomes the initial value.

When RSTE = 0, an internal reset signal is not generated. Neither SCKCR nor the multip ratio of Pϕ is changed.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. I overflows when the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated entire LSI.



Notes: 1. If TCNT overflows when the RSTE bit is set to 1, an internal reset signal is generated.
2. 130 states when the RSTE bit is cleared to 0.

Figure 15.2 Operation in Watchdog Timer Mode

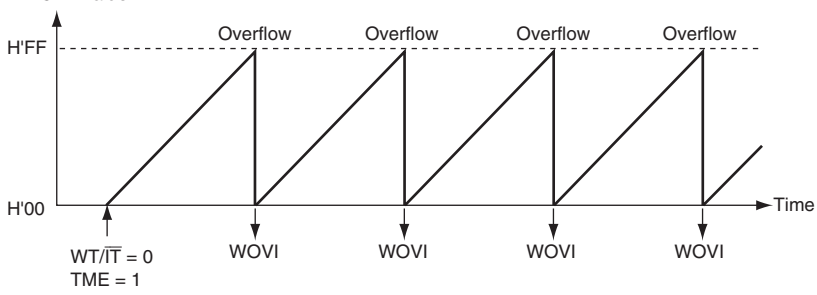


Figure 15.3 Operation in Interval Timer Mode

15.5 Interrupt Source

During interval timer mode operation, an overflow generates an interval timer interrupt (W
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 15.2 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 15.4. 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 tra
instruction cannot be used to write to RSTCSR.

The method of writing 0 to the WOVF bit in RSTCSR differs from that of writing to the
in RSTCSR. Perform data transfer as shown in figure 15.4.

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 15.4. In this case,
transfer instruction writes the value in bit 6 of the lower byte to the RSTE bit, but has no
the WOVF bit.

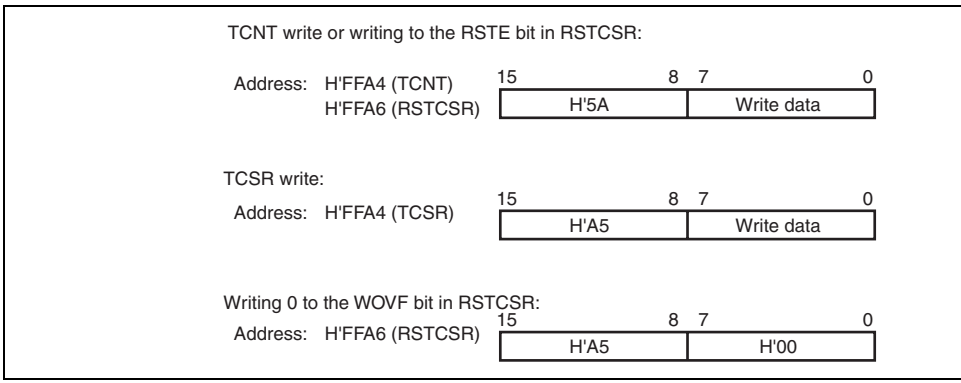


Figure 15.4 Writing to TCNT, TCSR, and RSTCSR

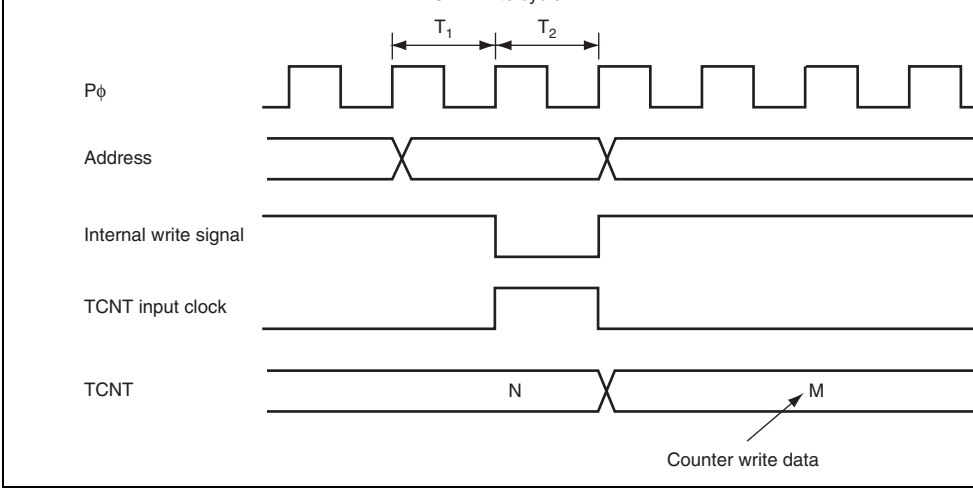


Figure 15.5 Conflict between TCNT Write and Increment

15.6.3 Changing Values of Bits CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could o the incrementation. The watchdog timer must be stopped (by clearing the TME bit to 0) b values of bits CKS2 to CKS0 are changed.

15.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the timer mode is switched from watchdog timer mode to interval timer mode while th 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 15.6.

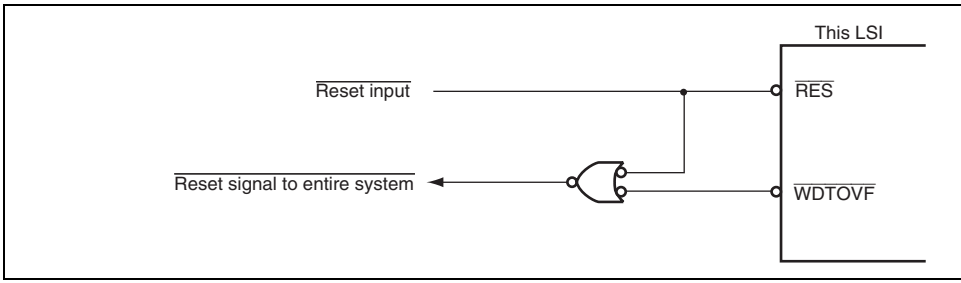


Figure 15.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

15.6.7 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.

16.1 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

Asynchronous Mode:

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

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

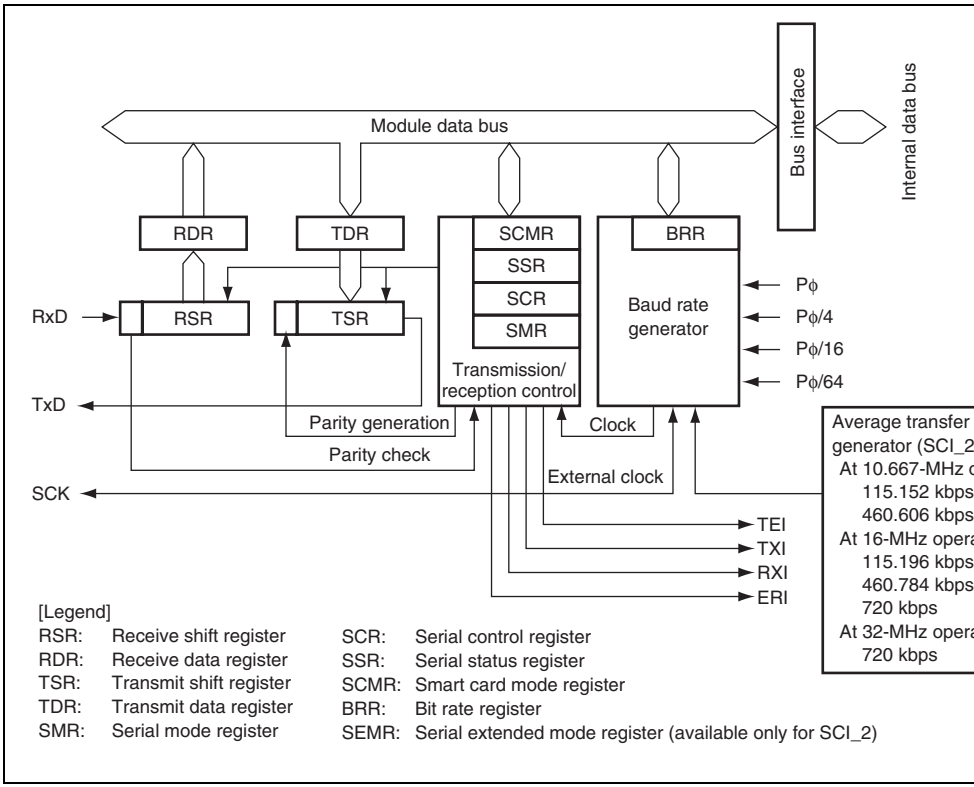


Figure 16.1 Block Diagram of SCI

	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	I/O	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output
3	SCK3	I/O	Channel 3 clock input/output
	RxD3	Input	Channel 3 receive data input
	TxD3	Output	Channel 3 transmit data output
4	SCK4	I/O	Channel 4 clock input/output
	RxD4	Input	Channel 4 receive data input
	TxD4	Output	Channel 4 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting channel designation.

- Receive data register_0 (RDR_0)
- Transmit data register_0 (TDR_0)
- Serial mode register_0 (SMR_0)
- Serial control register_0 (SCR_0)
- Serial status register_0 (SSR_0)
- Smart card mode register_0 (SCMR_0)
- Bit rate register_0 (BRR_0)

Channel 1:

- Receive shift register_1 (RSR_1)
- Transmit shift register_1 (TSR_1)
- Receive data register_1 (RDR_1)
- Transmit data register_1 (TDR_1)
- Serial mode register_1 (SMR_1)
- Serial control register_1 (SCR_1)
- Serial status register_1 (SSR_1)
- Smart card mode register_1 (SCMR_1)
- Bit rate register_1 (BRR_1)

- Bit rate register_2 (BRR_2)
- Serial extended mode register_2 (SEMR_2) (SCI_2 only)

Channel 3:

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

Channel 4:

- Receive shift register_4 (RSR_4)
- Transmit shift register_4 (TSR_4)
- Receive data register_4 (RDR_4)
- Transmit data register_4 (TDR_4)
- Serial mode register_4 (SMR_4)
- Serial control register_4 (SCR_4)
- Serial status register_4 (SSR_4)
- Smart card mode register_4 (SCMR_4)
- Bit rate register_4 (BRR_4)

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

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

16.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty
transfers the transmit data written in TDR to TSR and starts transmission. The double-bu
structures of TDR and TSR enables continuous serial transmission. If the next transmit da
already been written to TDR when one frame of data is transmitted, the SCI transfers the
data to TSR to continue transmission. Although TDR can be read from or written to by th
all times, to achieve reliable serial transmission, write transmit data to TDR for only once
confirming that the TDRE bit in SSR is set to 1.

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

- When SMIF in SCMR = 0

Bit	7	6	5	4	3	2	1
Bit Name	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1
Bit Name	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Functions in Normal Serial Communication Interface Mode (When SMIF in SC

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (valid only in asynchronous 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first and the MSB (bit 7) in TDR is not transmitt transmission. In clocked synchronous mode, a fixed data le bits is used.

				1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (valid only in asynchronous mo Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. I second stop bit is 0, it is treated as the start bit next transmit frame.
2	MP	0	R/W	Multiprocessor Mode (valid only in asynchrono When this bit is set to 1, the multiprocessor fur enabled. The PE bit and O/$\overline{E}$ bit settings are in multiprocessor mode.
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0	R/W	These bits select the clock source for the baud generator. 00: Pϕ clock (n = 0) 01: Pϕ/4 clock (n = 1) 10: Pϕ/16 clock (n = 2) 11: Pϕ/64 clock (n = 3) For the relation between the settings of these b the baud rate, see section 16.3.9, Bit Rate Reg (BRR). n is the decimal display of the value of (see section 16.3.9, Bit Rate Register (BRR)).

6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode operation. For details, see section 16.7.3, Blo Transfer Mode.
5	PE	0	R/W	Parity Enable (valid only in asynchronous mo When this bit is set to 1, the parity bit is adde transmit data before transmission, and the pa checked in reception. Set this bit to 1 in smart interface mode.
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 ca interface mode, see section 16.7.2, Data Forn (Except in Block Transfer Mode).
3	BCP1	0	R/W	Basic Clock Pulse 1,0
2	BCP0	0	R/W	These bits select the number of basic clock cy 1-bit data transfer time in smart card interface 00: 32 clock cycles (S = 32) 01: 64 clock cycles (S = 64) 10: 372 clock cycles (S = 372) 11: 256 clock cycles (S = 256) For details, see section 16.7.4, Receive Data Timing and Reception Margin. S is described 16.3.9, Bit Rate Register (BRR).

the baud rate, see section 16.3.9, Bit Rate Reg
(BRR). n is the decimal display of the value of
(see section 16.3.9, Bit Rate Register (BRR)).

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

16.3.6 Serial Control Register (SCR)

SCR is a register that enables/disables the following SCI transfer operations and interrupt
and selects the transfer clock source. For details on interrupt requests, see section 16.8, In
Sources. Some bits in SCR have different functions in normal mode and smart card interf
mode.

- When SMIF in SCMR = 0

Bit	7	6	5	4	3	2	1	
Bit Name	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1	
Bit Name	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

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

1X: External clock

A clock with a frequency 16 times the bit r
be input from the SCK pin.

- Clocked synchronous mode

0X: Internal clock

The SCK pin functions as a clock output p

1X: External clock

The SCK pin functions as a clock input pin

Clock Enable 1, 0 (for SCI_2)

These bits select the clock source and SCK p
function.

- Asynchronous mode

00: On-chip baud rate generator

The SCK pin can be used as an I/O port p

01: On-chip baud rate generator

The SCK pin outputs a clock with the sam
frequency as the bit rate.

1X: External clock or average transfer rate ge

— When using an external clock, a clock
frequency 16 times the bit rate should
from the SCK pin.

— Average transfer rate generator is use

- Clocked synchronous mode

0X: Internal clock

The SCK pin functions as a clock output p

1X: External clock

The SCK pin functions as a clock input pin

Note: X: Don't care

				When this bit is set to 1, RXI and ERI interrupt are enabled. RXI and ERI interrupt requests can be cancelle reading 1 from the RDRF, FER, PER, or OREF then clearing the flag to 0, or by clearing the R 0.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enable this condition, serial transmission is started by transmit data to TDR, and clearing the TDRE fl SSR to 0. Note that SMR should be set prior to the TE bit to 1 in order to designate the transm format. If transmission is halted by clearing this bit to 0 TDRE flag in SSR is fixed 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. this condition, serial reception is started by det the start bit in asynchronous mode or the synch clock input in clocked synchronous mode. Note SMR should be set prior to setting the RE bit to order to designate the reception format. Even if reception is halted by clearing this bit to RDRF, FER, PER, and ORER flags are not aff and the previous value is retained.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (valid only whe bit in SMR is 1 in asynchronous mode) Write 0 to this bit in smart card interface mode.

01: Clock output

1X: Reserved

- When GM in SMR = 1

00: Output fixed low

01: Clock output

10: Output fixed high

11: Clock output

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

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1	
Bit Name	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	
Initial Value	1	0	0	0	0	1	0	
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	

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

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

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

				the subsequent serial reception cannot be performed. In clocked synchronous mode transmission also cannot continue. [Clearing condition] • When 0 is written to PER after reading PE Even when the RE bit in SCR is cleared, t bit is not affected and retains its previous (When the CPU is used to clear this flag b while the corresponding interrupt is enable sure to read the flag after writing 0 to it.)
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the las transmit character [Clearing conditions] • When 0 is written to TDRE after reading T • When a TXI interrupt request is issued all DMAC or DTC to write data to TDR
1	MPB	0	R	Multiprocessor Bit Stores the multiprocessor bit value in the rece When the RE bit in SCR is cleared to 0 its pre state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Sets the multiprocessor bit value to be added transmit frame.

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

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

				ORER flag is set to 1, subsequent serial r cannot be performed. Note that, in clocke synchronous mode, serial transmission al continue. [Clearing condition] • When 0 is written to ORER after reading O Even when the RE bit in SCR is cleared, t flag is not affected and retains its previous (When the CPU is used to clear this flag b while the corresponding interrupt is enable sure to read the flag after writing 0 to it.)
4	ERS	0	R/(W)*	Error Signal Status [Setting condition] • When a low error signal is sampled [Clearing condition] • When 0 is written to ERS after reading ER

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

[Clearing condition]

- When 0 is written to PER after reading PER

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

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

				follows: When GM = 0 and BLK = 0, 2.5 etu after transmission start When GM = 0 and BLK = 1, 1.5 etu after transmission start When GM = 1 and BLK = 0, 1.0 etu after transmission start When GM = 1 and BLK = 1, 1.0 etu after transmission start [Clearing conditions] • When 0 is written to TDRE after reading T • When a TXI interrupt request is issued all DMAC or DTC to write the next data to TD
1	MPB	0	R	Multiprocessor Bit Not used in smart card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in smart card interface mode

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

7 to 4	—	All 1	R	Reserved These are read-only bits and cannot be modifie
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: Transfer with LSB-first 1: Transfer with MSB-first This bit is valid only when the 8-bit data format for transmission/reception; when the 7-bit data used, data is always transmitted/received with
2	SINV	0	R/W	Smart Card Data Invert Inverts the transmit/receive data logic level. Th does not affect the logic level of the parity bit. T the parity bit, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. R data is stored as it is in RDR. 1: TDR contents are inverted before being tran Receive data is stored in inverted form in RD
1	—	1	R	Reserved This is a read-only bit and cannot be modified.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface m selected. 0: Normal asynchronous or clocked synchrono 1: Smart card interface mode

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

[Legend]

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

$P\phi$: Operating frequency (MHz)

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

SMR Setting			SMR Setting		
CKS1	**CKS0**	**n**	**BCP1**	**BCP0**	**S**
0	0	0	0	0	3
0	1	1	0	1	6
1	0	2	1	0	3
1	1	3	1	1	2

	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

2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15
38400	0	9	0.00	—	—	—	0	11	0.00	0	12

Note: For SCI_2, the table shows the examples for the case when the ABCS bit in SEM cleared to 0. When the ABCS bit is set to 1, the bit rates are doubled.

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: For SCI_2, the table shows the examples for the case when the ABCS bit in SEM cleared to 0. When the ABCS bit is set to 1, the bit rates are doubled.

16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0
30	937500	0	0
33	1031250	0	0
35	1093750	0	0

Note: For SCI_2, the table shows the examples for the case when the ABCS bit in SEMR cleared to 0. When the ABCS bit is set to 1, the bit rates are doubled.

16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500
25	6.2500	390625
30	7.5000	468750
33	8.2500	515625
35	8.7500	546875

Note: For SCI_2, the table shows the examples for the case when the ABCS bit in SEM cleared to 0. When the ABCS bit is set to 1, the bit rates are doubled.

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

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

[Legend]

Space: Setting prohibited.

—: Can be set, but there will be error.

*: Continuous transmission or reception is not possible.

Table 16.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	Pϕ (MHz)	External Input Clock (MHz)	Maximu Rate (b
8	1.3333	1333333.3	20	3.3333	333333
10	1.6667	1666666.7	25	4.1667	416666
12	2.0000	2000000.0	30	5.0000	500000
14	2.3333	2333333.3	33	5.5000	550000
16	2.6667	2666666.7	35	5.8336	583362
18	3.0000	3000000.0			

(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	E
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6

Bit Rate	Operating Frequency Pϕ (MHz)											
	25.00			30.00			33.00			35.0		
(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	E
9600	0	3	12.49	0	3	5.01	0	4	7.59	0	4	1

Table 16.9 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372)

Pϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	Pϕ (MHz)	Maximum Bit Rate (bit/s)	n
7.1424	9600	0	0	18.00	24194	0
10.00	13441	0	0	20.00	26882	0
10.7136	14400	0	0	25.00	33602	0
13.00	17473	0	0	30.00	40323	0
14.2848	19200	0	0	33.00	44355	0
16.00	21505	0	0	35.00	47043	0

Bit	Bit Name	Value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0. The write value s always be 0.
6 to 4	—	All 0	R	Reserved These are read-only bits and cannot be modif
3	ABCS	0	R/W	Asynchronous Mode Basic Clock Select (valid asynchronous mode) Selects the basic clock. 0: The basic clock has a frequency 16 times t rate 1: The basic clock has a frequency 8 times th rate

basic clock with a frequency 16 times the
rate)

010: 460.606 kbps of average transfer rate spe
Pϕ = 10.667 MHz is selected (operated us
basic clock with a frequency 8 times the t
rate)

011: 720 kbps of average transfer rate specific
32 MHz is selected (operated using the ba
with a frequency 16 times the transfer rate

100: Setting prohibited

101: 115.196 kbps of average transfer rate spe
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the transf

110: 460.784 kbps of average transfer rate spe
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the transf

111: 720 kbps of average transfer rate specific
16 MHz is selected (operated using the ba
with a frequency 8 times the transfer rate)

The average transfer rate only supports operat
frequencies of 10.667 MHz, 16 MHz, and 32 M

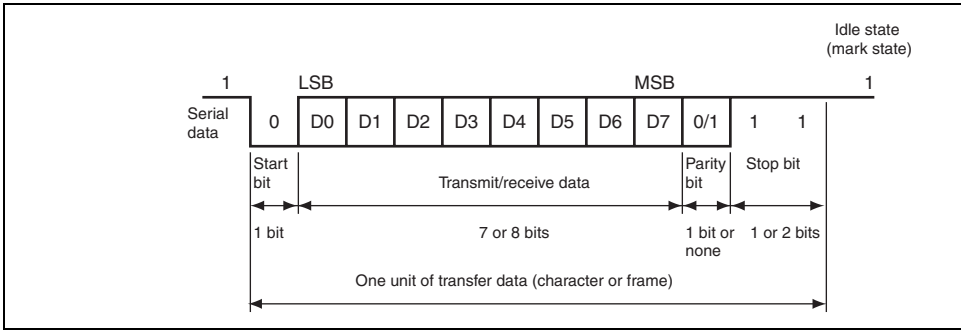
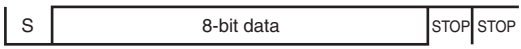
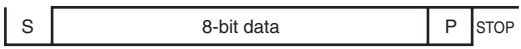
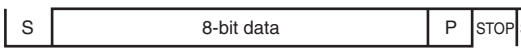
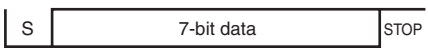
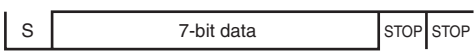
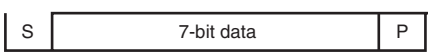
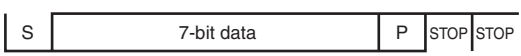
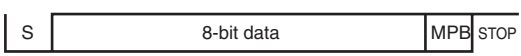
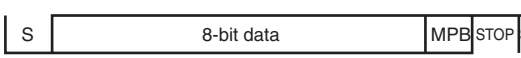
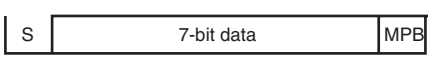
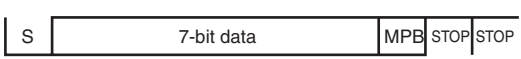


Figure 16.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

0	0	0	0	
0	0	0	1	
0	1	0	0	
0	1	0	1	
1	0	0	0	
1	0	0	1	
1	1	0	0	
1	1	0	1	
0	—	1	0	
0	—	1	1	
1	—	1	0	
1	—	1	1	

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

M: Reception margin
N: Ratio of bit rate to clock (N = 16)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$$M = \left(0.5 - \frac{1}{2 \times 16} \right) \times 100[\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

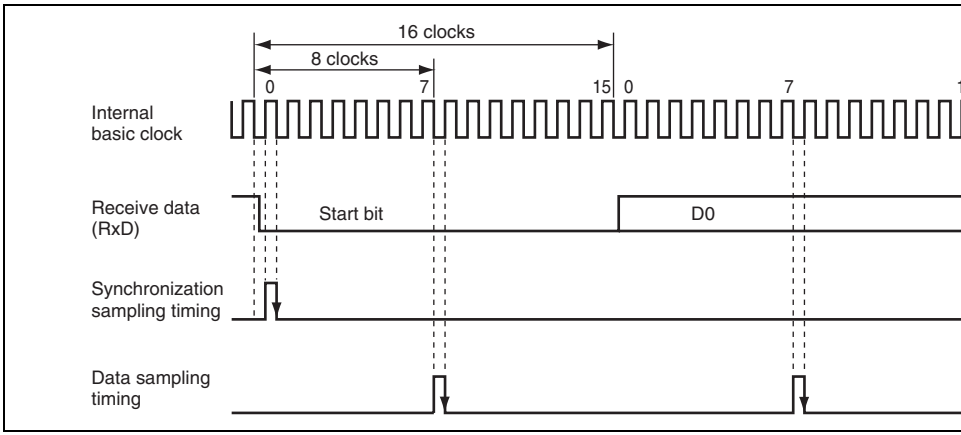


Figure 16.3 Receive Data Sampling Timing in Asynchronous Mode

When the SCI is operated on an internal clock, the clock can be output from the SCK pin
frequency of the clock output in this case is equal to the bit rate, and the phase is such tha
rising edge of the clock is in the middle of the transmit data, as shown in figure 16.4.

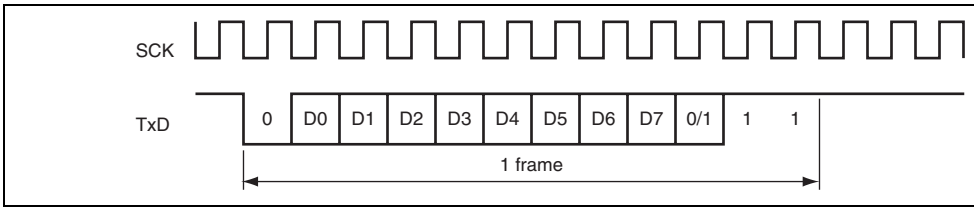
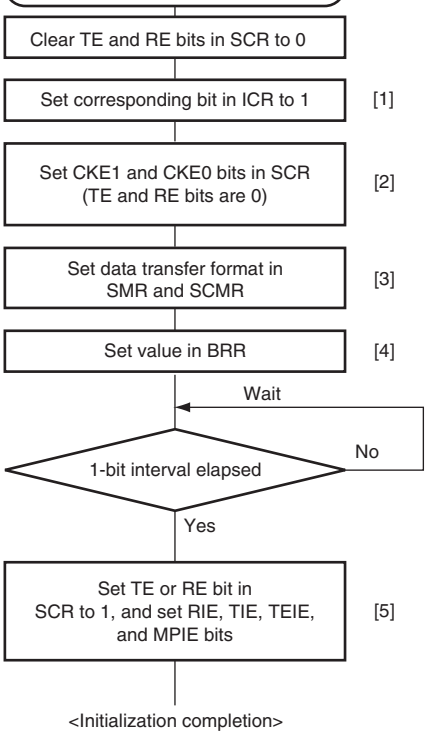


Figure 16.4 Phase Relation between Output Clock and Transmit Data (Asynchronous Mode)



[1] Set the bit in ICR for the corresponding pin when receiving data or using an external clock.

[2] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set the data transfer format in SMR and SCMR.

[4] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[5] Wait at least one bit interval, then set th TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits. Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 16.5 Sample SCI Initialization Flowchart

3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the next transmit data is transferred from TDR to TSR, the stop sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then t state is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a T interrupt request is generated.

Figure 16.7 shows a sample flowchart for transmission in asynchronous mode.

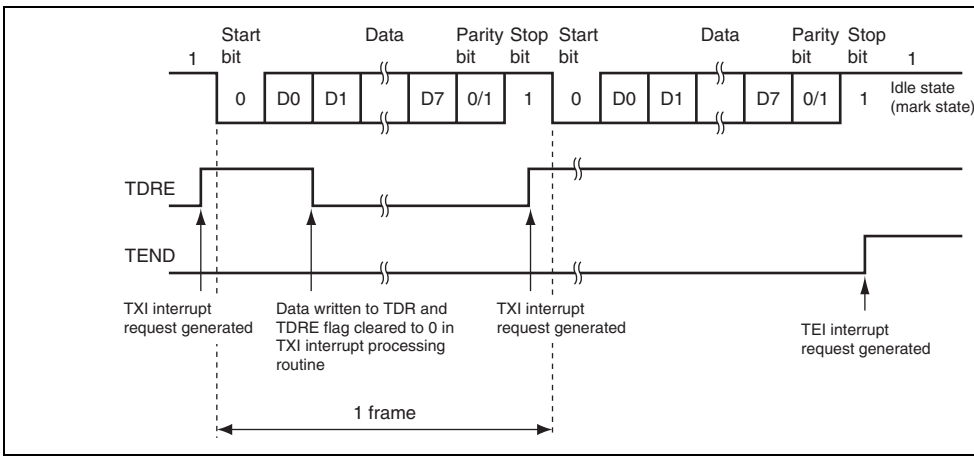
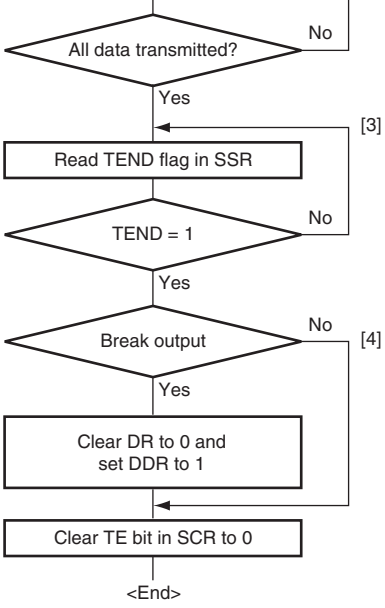


Figure 16.6 Example of Operation for Transmission in Asynchronous Mod (Example with 8-Bit Data, Parity, One Stop Bit)



To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC or DMAC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 16.7 Sample Serial Transmission Flowchart

3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferr RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generate
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 an data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interru request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt requ generated. Because the RXI interrupt processing routine reads the receive data transfe RDR before reception of the next receive data has finished, continuous reception can enabled.

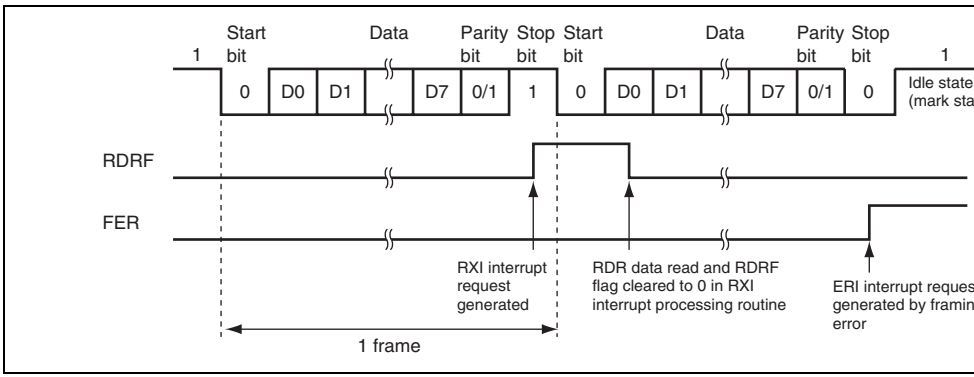
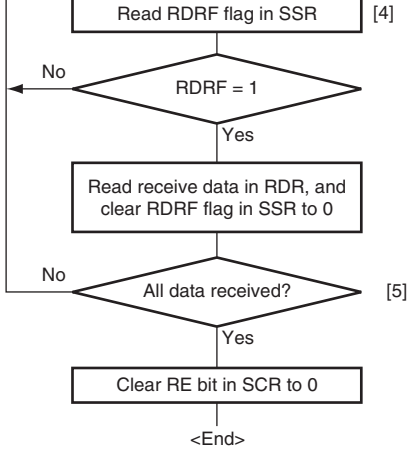


Figure 16.8 Example of SCI Operation for Reception (Example with 8-Bit Data, Parity, One Stop Bit)

0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framin
1	1	0	1	Lost	Overrun error + parity
0	0	1	1	Transferred to RDR	Framing error + parity
1	1	1	1	Lost	Overrun error + framin parity error

Note: * The RDRF flag retains the state it had before data reception.



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

[5] Serial reception continuation procedure: To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag and RDR, and clear the RDRF flag to 0. However, the RDRF flag is cleared automatically when the DTC or DMAC is initiated by an RXI interrupt and reads data from RDR.

Figure 16.9 Sample Serial Reception Flowchart (1)

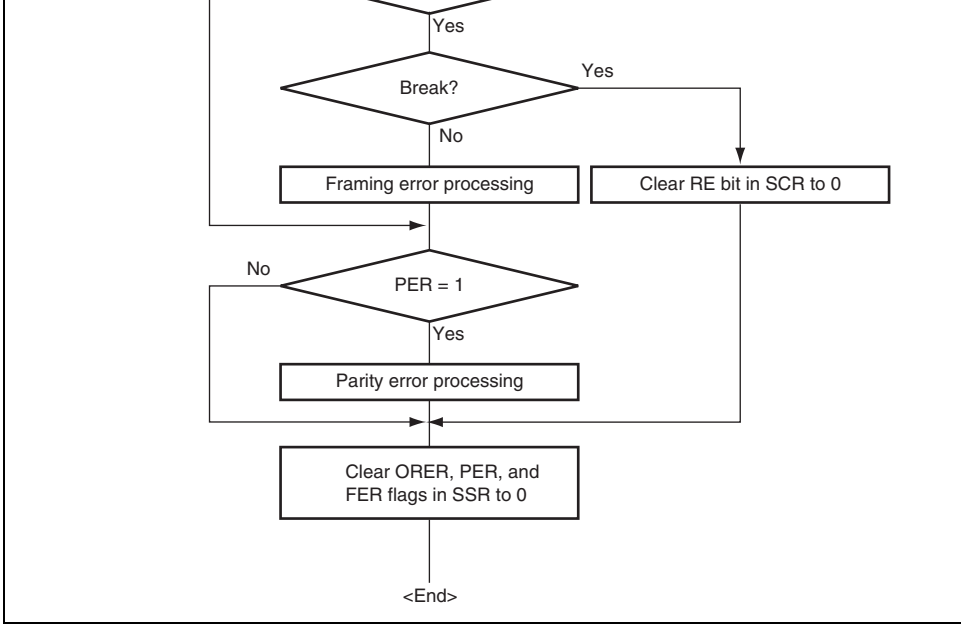


Figure 16.9 Sample Serial Reception Flowchart (2)

16.10 shows an example of inter-processor communication using the multiprocessor form
transmitting station first sends data which includes the ID code of the receiving station an
multiprocessor bit set to 1. It then transmits transmit data added with a multiprocessor bit
to 0. The receiving station skips data until data with a 1 multiprocessor bit is sent. When
a 1 multiprocessor bit is received, the receiving station compares that data with its own ID
station whose ID matches then receives the data sent next. Stations whose ID does not ma
continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status
RDRF, FER, and ORER in SSR to 1 are prohibited until data with a 1 multiprocessor bit
received. On reception of a receive character with a 1 multiprocessor bit, the MPB bit in S
set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If th
in SCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit s
are the same as those in normal asynchronous mode. The clock used for multiprocessor
communication is the same as that in normal asynchronous mode.

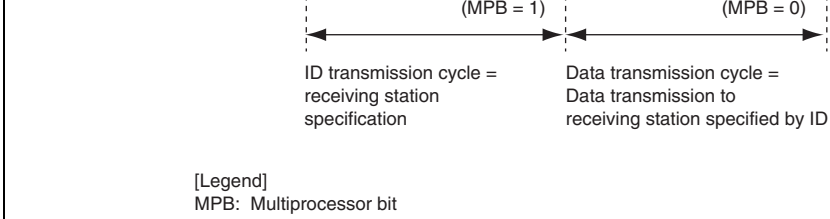
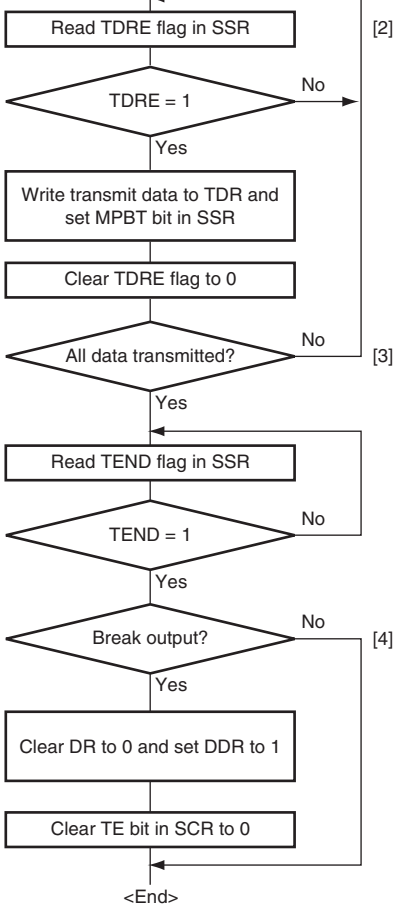


Figure 16.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



to 1, a 1 is output for one frame, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC or DMAC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Figure 16.11 Sample Multiprocessor Serial Transmission Flowchart

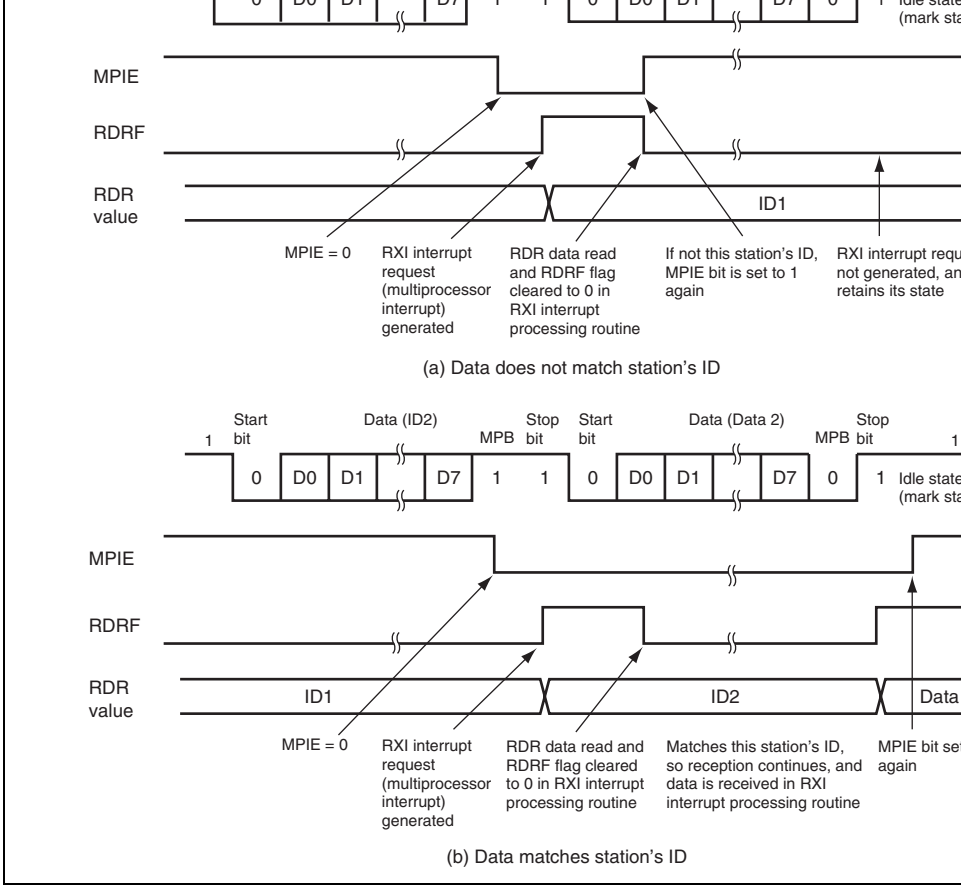
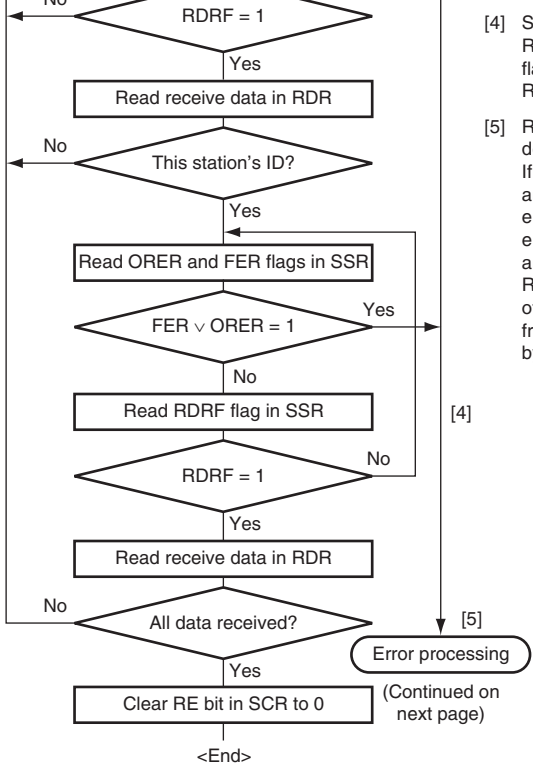


Figure 16.12 Example of SCI Operation for Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[4] SCI state check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FER flags are both cleared to 0. Reception cannot be resumed if either of these flags is set to 1. In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 16.13 Sample Multiprocessor Serial Reception Flowchart (1)

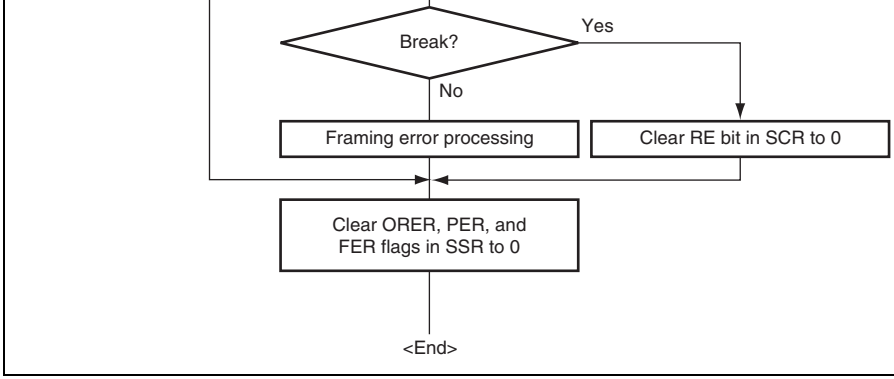
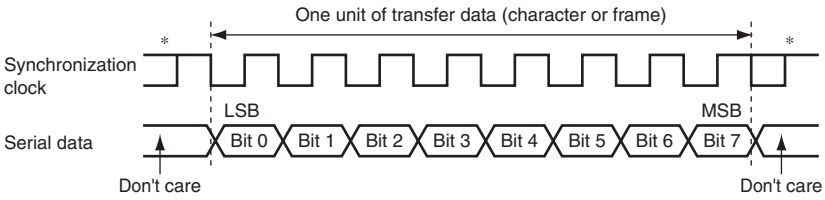


Figure 16.13 Sample Multiprocessor Serial Reception Flowchart (2)

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

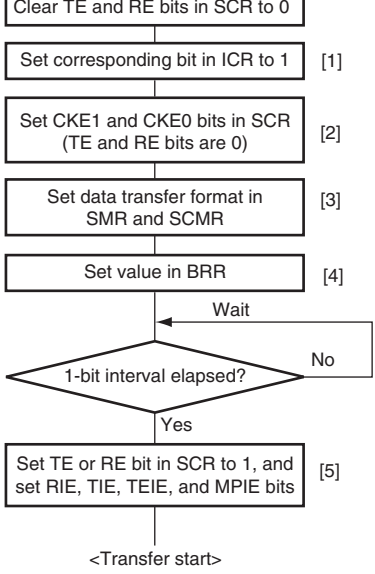


Note: * Holds a high level except during continuous transfer.

Figure 16.14 Data Format in Clocked Synchronous Communication (LSB-Fi

16.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of the and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronizatio is output from the SCK pin. Eight synchronization clock pulses are output in the transfer character, and when no transfer is performed the clock is fixed high. Note that in the case reception only, the synchronization clock is output until an overrun error occurs or until t is cleared to 0.



pin when receiving data or using an external clock.

[2] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

[3] Set the data transfer format in SMR and SCMR.

[4] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[5] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE TEIE, and MPIE bits. Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 16.15 Sample SCI Initialization Flowchart

transmission of the current transmit data has finished, continuous transmission can be

3. 8-bit data is sent from the TxD pin synchronized with the output clock when clock ou mode has been specified and synchronized with the input clock when use of an extern has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, the next transmit data is transferred from TDR to TS serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin retains output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interru is generated. The SCK pin is fixed high.

Figure 16.17 shows a sample flowchart for serial data transmission. Even if the TDRE fla cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) i Make sure to clear the receive error flags to 0 before starting transmission. Note that clea RE bit to 0 does not clear the receive error flags.

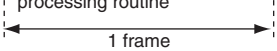
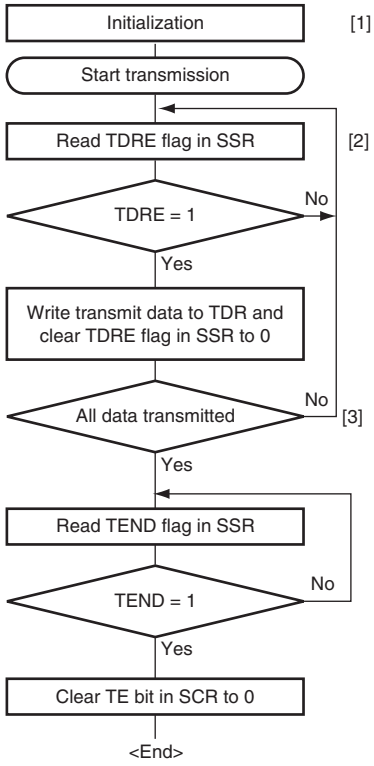


Figure 16.16 Example of Operation for Transmission in Clocked Synchronous



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

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

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC or DMAC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

Figure 16.17 Sample Serial Transmission Flowchart

3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt requ generated. Because the RXI interrupt processing routine reads the receive data transfe RDR before reception of the next receive data has finished, continuous reception can enabled.

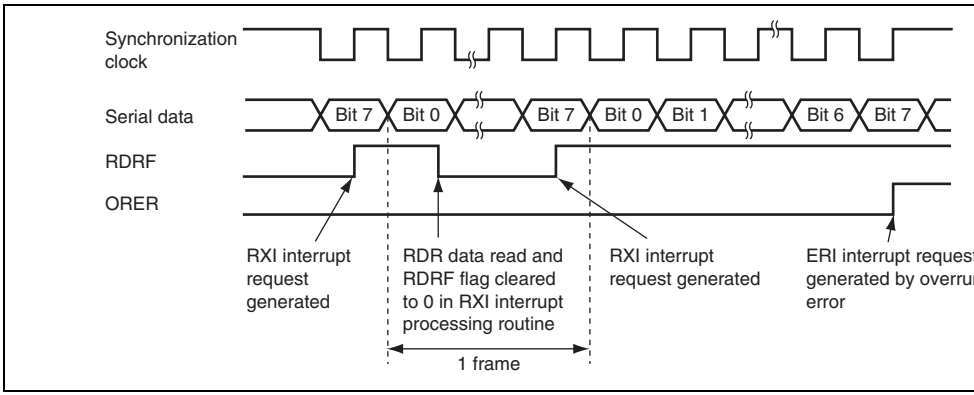
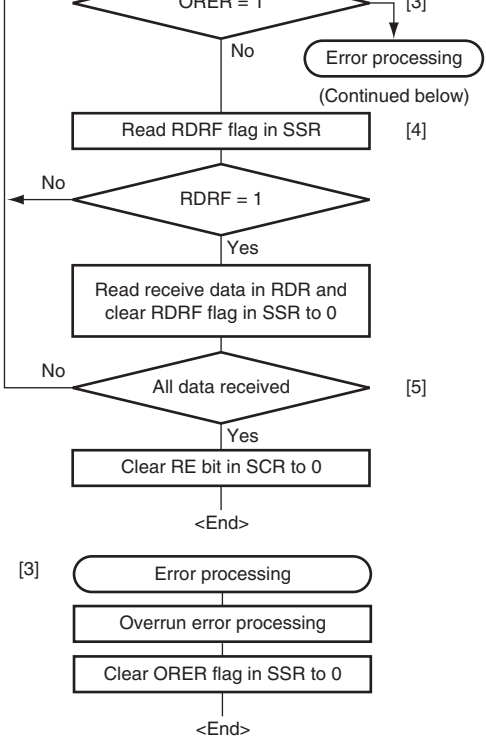


Figure 16.18 Example of Operation for Reception in Clocked Synchronous M

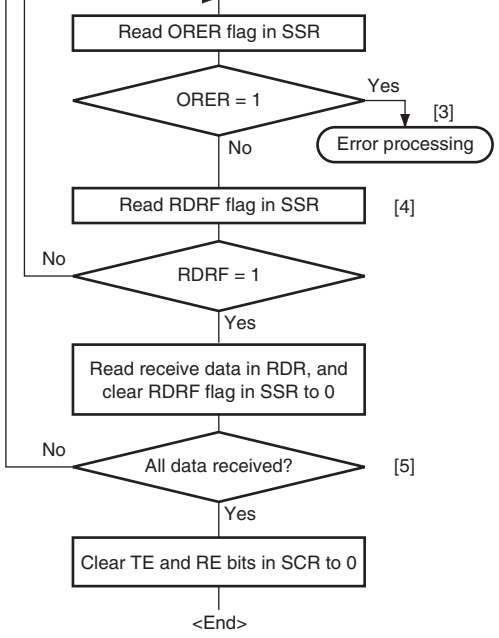


[4] SCI state check and receive data read: Read SSR and check that the RDRF flag set to 1, then read the receive data in RD and clear the RDRF flag to 0. Transition the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure: To continue serial reception, before the MSB (bit 7) of the current frame is receiv reading the RDRF flag, reading RDR, an clearing the RDRF flag to 0 should be finished. However, the RDRF flag is clea automatically when the DTC or DMAC is initiated by a receive data full interrupt (R and reads data from RDR.

Figure 16.19 Sample Serial Reception Flowchart

the RDRF bit and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneo
both the TE and RE bits to 1 with a single instruction.



Read SSR and check that the RDRF flag is
then read the receive data in RDR and clea
RDRF flag to 0. Transition of the RDRF fla
to 1 can also be identified by an RXI interru

[5] Serial transmission/reception continuation
procedure:
To continue serial transmission/ reception,
the MSB (bit 7) of the current frame is rece
finish reading the RDRF flag, reading RDR
clearing the RDRF flag to 0. Also, before th
(bit 7) of the current frame is transmitted, re
from the TDRE flag to confirm that writing is
possible. Then write data to TDR and clear
TDRE flag to 0.
However, the TDRE flag is checked and cle
automatically when the DTC or DMAC is in
by a transmit data empty interrupt (TXI) req
writes data to TDR. Similarly, the RDRF fla

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

Figure 16.20 Sample Flowchart of Simultaneous Serial Transmission and Rec

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

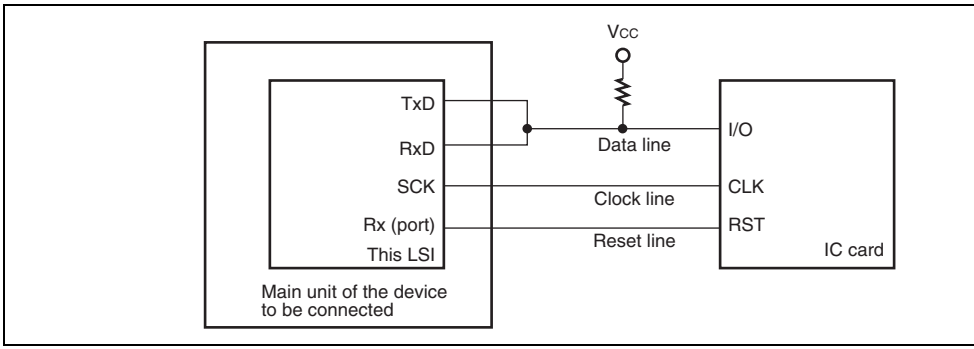


Figure 16.21 Pin Connection for Smart Card Interface

after at least 2 etu.

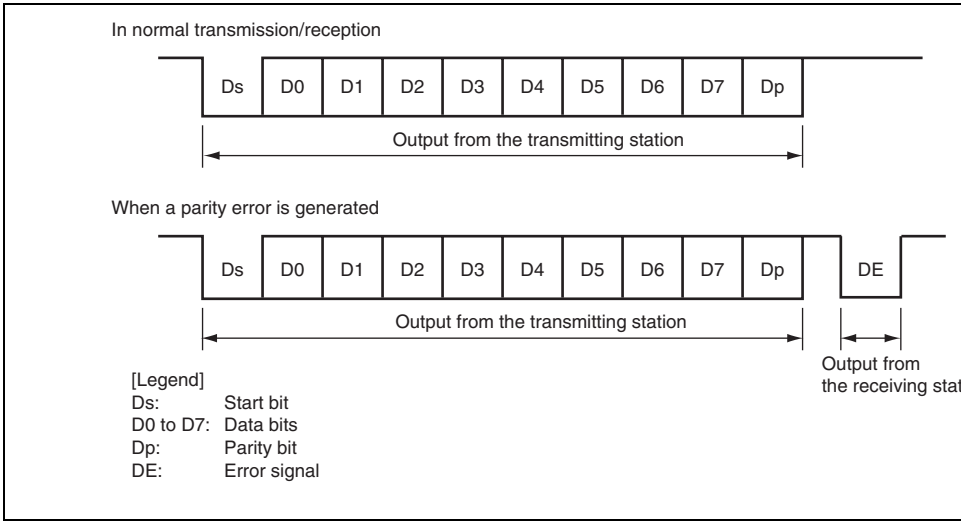


Figure 16.22 Data Formats in Normal Smart Card Interface Mode

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

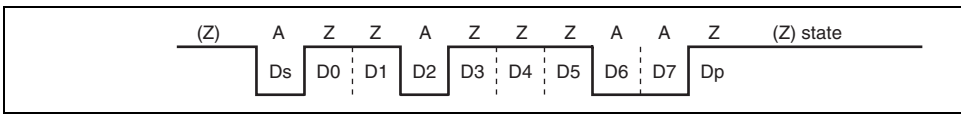


Figure 16.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respec
and data is transferred with MSB-first as the start character, as shown in figure 16.24. Th
data in the start character in the figure is H'3F. When using the inverse convention type, v
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.

16.7.3 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

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

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.

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

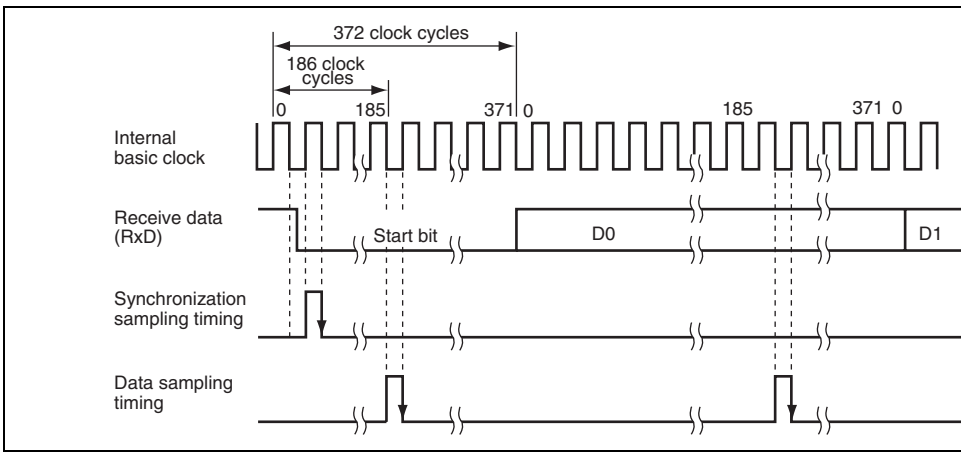


**Figure 16.25 Receive Data Sampling Timing in Smart Card Interface Mo
(When Clock Frequency is 372 Times the Bit Rate)**

5. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the DDR correspo
   the TxD pin is cleared to 0, the TxD and RxD pins are changed from port pins to SCI
   placing the pins into high impedance state.
6. Set the value corresponding to the bit rate in BRR.
7. Set the CKE1 and CKE0 bits in SCR appropriately. Clear the TIE, RIE, TE, RE, MPI
   TEIE bits to 0 simultaneously.

   When the CKE0 bit is set to 1, the SCK pin is allowed to output clock pulses.
8. Set the TIE, RIE, TE, and RE bits in SCR appropriately after waiting for at least a 1-b
   interval. Setting the TE and RE bits to 1 simultaneously is prohibited except for self d

To switch from reception to transmission, first verify that reception has completed, then i
the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. Re
completion can be verified by reading the RDRF, PER, or ORER flag. To switch from
transmission to reception, first verify that transmission has completed, then initialize the
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.

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 16.28 shows a sample flowchart for transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMAC. In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus genera TXI interrupt request if the TIE bit in SCR has been set to 1. This activates the DTC or D 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 9, DMA C (DMAC) and section 10, Data Transfer Controller (DTC).

Figure 16.26 Data Re-Transfer Operation in SCI Transmission Mode

Note that the TEND flag is set in different timings depending on the GM bit setting in SM
Figure 16.27 shows the TEND flag set timing.

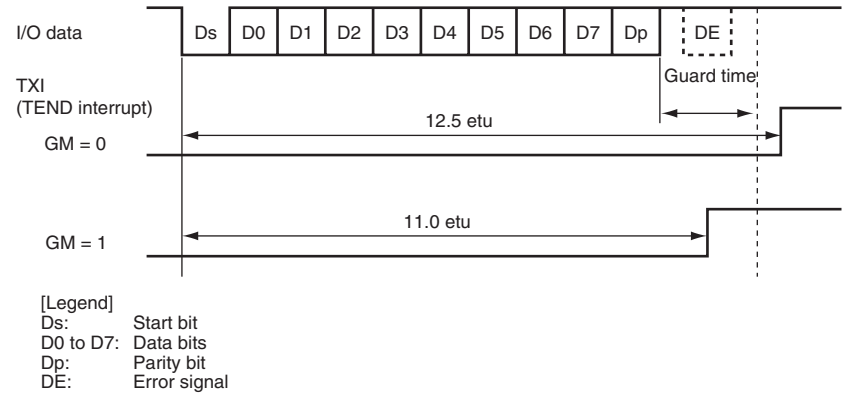


Figure 16.27 TEND Flag Set Timing during Transmission

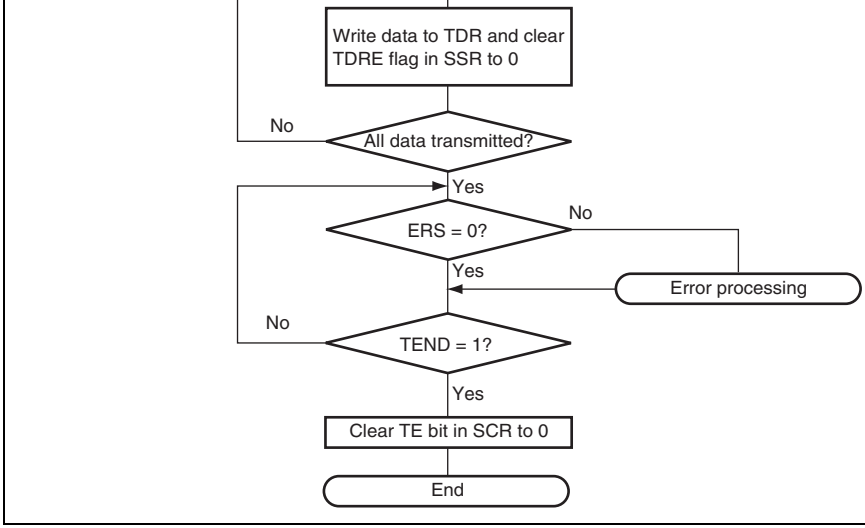


Figure 16.28 Sample Transmission Flowchart

4. In this case, data is determined to have been received successfully, and the RDRF bit set to 1. Here, an RXI interrupt request is generated if the RIE bit in SCR is set to 1.

Figure 16.30 shows a sample flowchart for reception. All the processing steps are automa performed using an RXI interrupt request to activate the DTC or DMAC. In reception, se RIE bit to 1 allows an RXI interrupt request to be generated when the RDRF flag is set to activates the DTC or DMAC by an RXI request thus allowing transfer of receive data if t interrupt request is specified as a source of DTC or DMAC activation beforehand. The R is automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs dur reception, i.e., either the ORER or PER flag is set to 1, a transmit/receive error interrupt (request is generated and the error flag must be cleared. If an error occurs, the DTC or DM not activated and receive data is skipped, therefore, the number of bytes of receive data sp in the DTC or DMAC is transferred. Even if a parity error occurs and the PER bit is set to reception, receive data is transferred to RDR, thus allowing the data to be read.

Note: For operations in block transfer mode, see section 16.4, Operation in Asynchronc

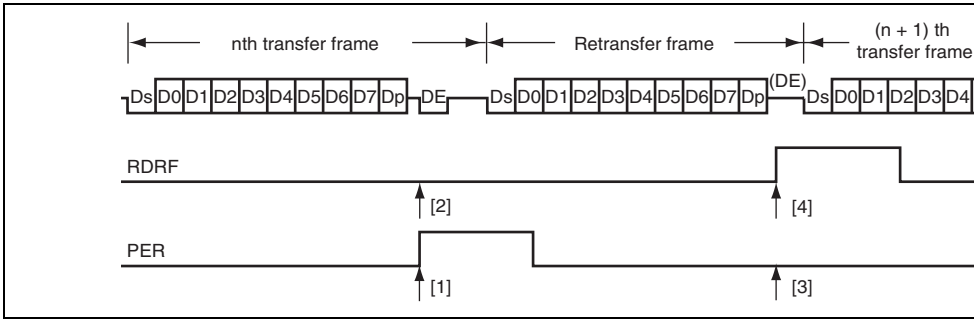


Figure 16.29 Data Re-Transfer Operation in SCI Reception Mode

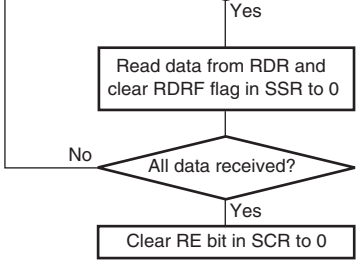


Figure 16.30 Sample Reception Flowchart

16.7.8 Clock Output Control

Clock output can be fixed using the CKE1 and CKE0 bits in SCR when the GM bit in S to 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 16.31 shows an example of clock output fixing timing when the CKE0 bit is cont with GM = 1 and CKE1 = 0.

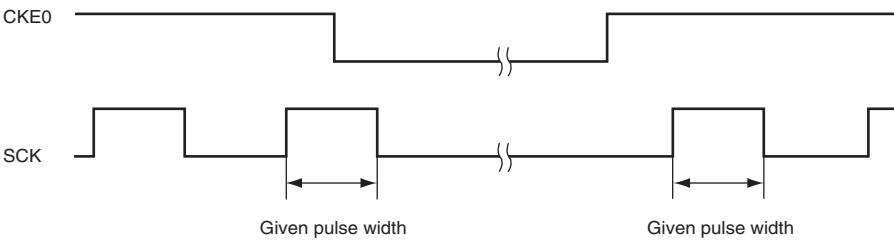


Figure 16.31 Clock Output Fixing Timing

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

- At mode switching
  - — At transition from smart card interface mode to software standby mode
    1. Set the data register (DR) and data direction register (DDR) corresponding to t pin to the values for the output fixed state in software standby mode.
    2. Write 0 to the TE and RE bits in SCR to stop transmission/reception. Simultan set the CKE1 bit to the value for the output fixed state in software standby mo
    3. Write 0 to the CKE0 bit in SCR to stop the clock.
    4. Wait for one cycle of the serial clock. In the mean time, the clock output is fix specified level with the duty cycle retained.
    5. Make the transition to software standby mode.
  - — At transition from smart card interface mode to software standby mode
    6. Clear software standby mode.
    7. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the appropriate duty cycle is then generated.

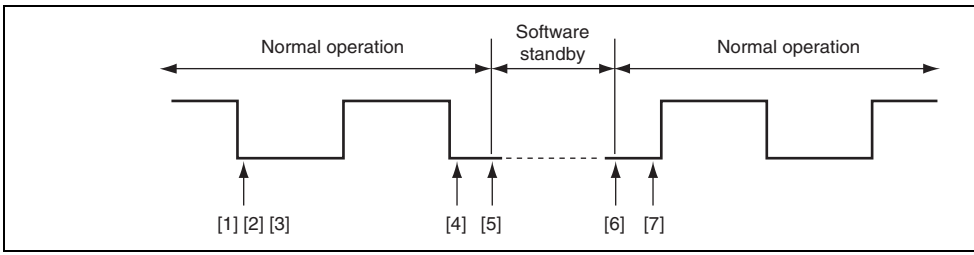


Figure 16.32 Clock Stop and Restart Procedure

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

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the
PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI inter
activate the DTC or DMAC to allow data transfer. The RDRF flag is automatically clear
data transfer by the DTC or DMAC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to
interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority
acceptance. However, note that if the TDRE and TEND flags are cleared to 0 simultaneo
the TXI interrupt processing routine, the SCI cannot branch to the TEI interrupt processi
later.

Table 16.12 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DMAC Activation	DTC Activation
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
TXI	Transmit data empty	TDRE	Possible	Possible
TEI	Transmit end	TEND	Not possible	Not possible

RXI	Receive data full	RDRF	Possible	Possible	
TXI	Transmit data empty	TDRE	Possible	Possible	L

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

When transmitting/receiving data using the DTC or DMAC, be sure to set and enable the DMAC prior to making SCI settings. For DTC or DMAC settings, see section 9, DMA C (DMAC) and section 10, Data Transfer Controller (DTC).

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1 activates the DTC or DMAC by an RXI request thus allowing transfer of receive data if t request is specified as a source of DTC or DMAC activation beforehand. The RDRF flag automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs, the R flag is not set but the error flag is set. Therefore, the DTC or DMAC is not activated and interrupt request is issued to the CPU instead; the error flag must be cleared.

When framing error detection is performed, a break can be detected by reading the RxD directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is se 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.

16.9.3 Mark State and Break Detection

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

16.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mo

Transmission cannot be started when a receive error flag (ORER, FER, or RER) is set to the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE cleared to 0.

16.9.6 Restrictions on Using DTC or DMAC

- When the external clock source is used as a synchronization clock, update TDR by the DMAC and wait for at least five Pϕ clock cycles before allowing the transmit clock to input. If the transmit clock is input within four clock cycles after TDR modification, t may malfunction (figure 16.33).
- When using the DTC or DMAC to read RDR, be sure to set the receive end interrupt the DTC or DMAC activation source.

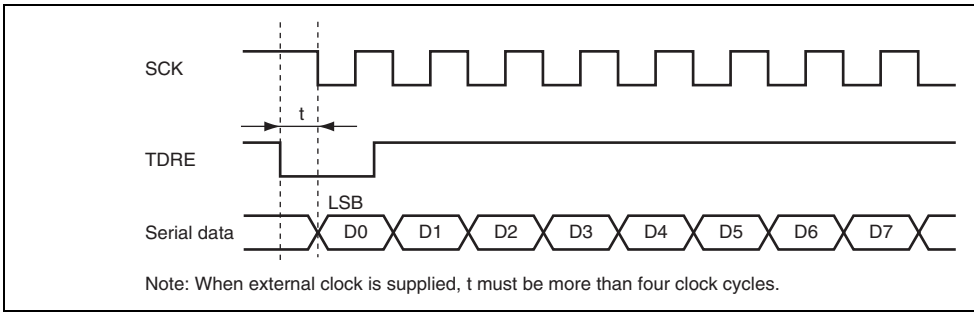


Figure 16.33 Sample Transmission using DTC in Clocked Synchronous Mo

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

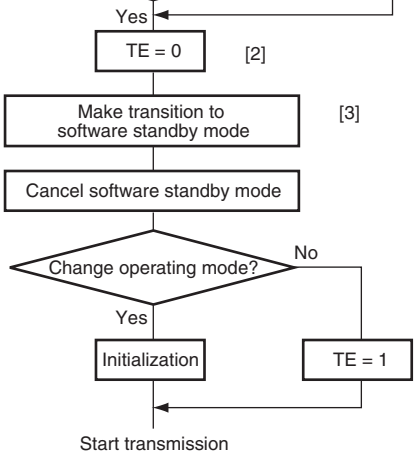
Figure 16.34 shows a sample flowchart for mode transition during transmission. Figures
16.36 show the port pin states during mode transition.

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

(2) Reception

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

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



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

[3] Transition to module stop state is included.

Figure 16.34 Sample Flowchart for Mode Transition during Transmission

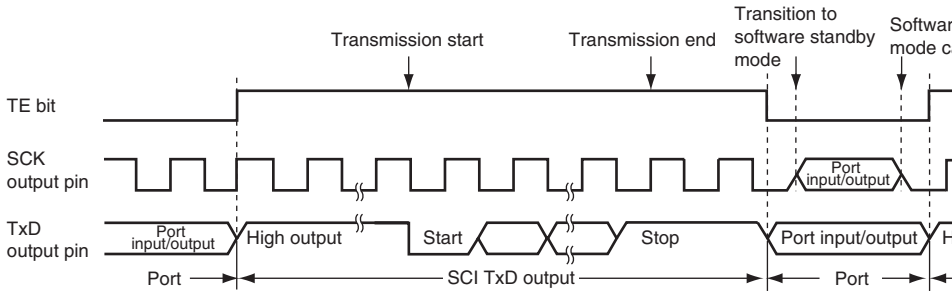


Figure 16.35 Port Pin States during Mode Transition (Internal Clock, Asynchronous Transmission)

Figure 16.36 Port Pin States during Mode Transition (Internal Clock, Clocked Synchronous Transmission)

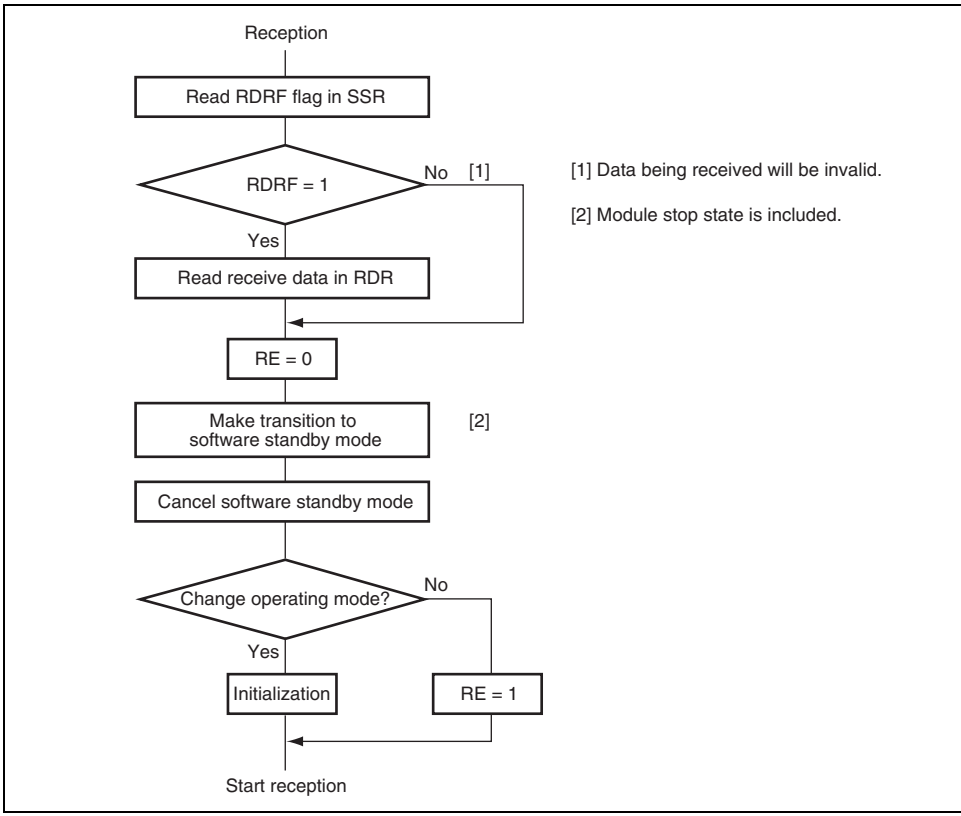


Figure 16.37 Sample Flowchart for Mode Transition during Reception

17.1 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 synchronized 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 specifiable

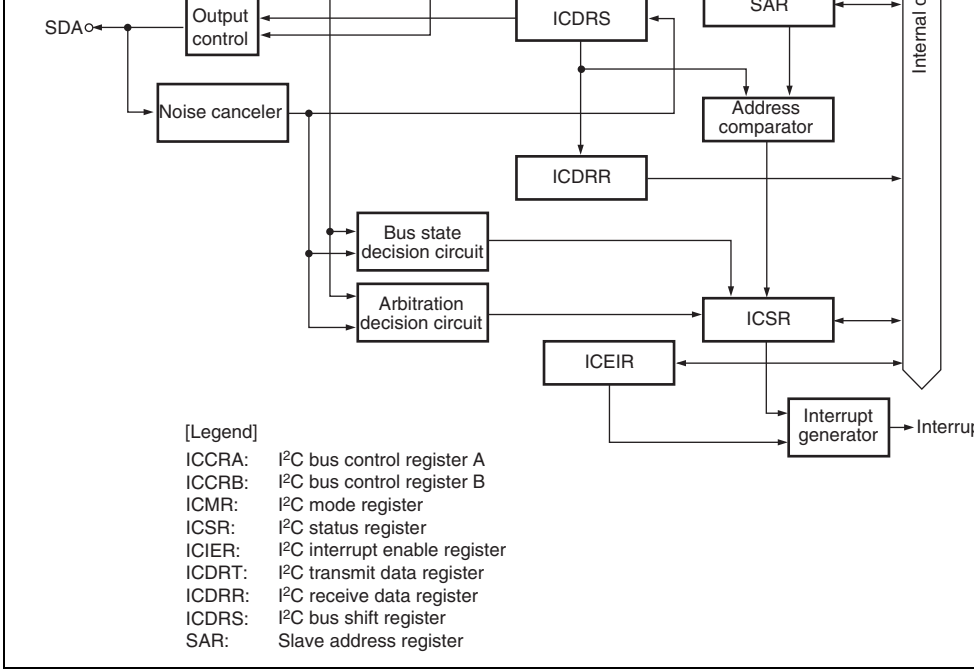


Figure 17.1 Block Diagram of I²C Bus Interface 2

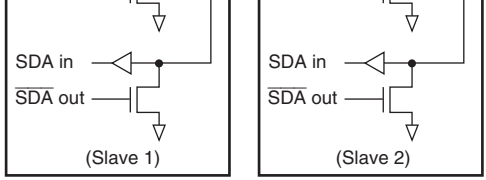


Figure 17.2 Connections to the External Circuit by the I/O Pins

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the I^2C bus interface 2.

Table 17.1 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.

- I²C bus status register_0 (ICSR_0)
- Slave address register_0 (SAR_0)
- I²C bus transmit data register_0 (ICDRT_0)
- I²C bus receive data register_0 (ICDRR_0)
- I²C bus shift register_0 (ICDRS_0)

Channel 1:

- I²C bus control register A_1 (ICCRA_1)
- I²C bus control register B_1 (ICCRB_1)
- I²C bus mode register_1 (ICMR_1)
- I²C bus interrupt enable register_1 (ICIER_1)
- I²C bus status register_1 (ICSR_1)
- Slave address register_1 (SAR_1)
- I²C bus transmit data register_1 (ICDRT_1)
- I²C bus receive data register_1 (ICDRR_1)
- I²C bus shift register_1 (ICDRS_1)

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I^2C Bus Interface Enable 0: This module is halted (SCL and SDA pins are used port function) 1: This bit is enabled for transfer operations (SCL and are bus drive state)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation when T 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, MST and TR reset by hardware, causing a transition to slave receiv Modification of the TRS bit should be made between t frames. Operating modes are described below according to M 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 These bits are valid only in master mode. Make settin according to the required transfer rate. For details on t transfer rate, see table 17.2.
2	CKS2	0	R/W	
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

17.3.2 I^2C Bus Control Register B (ICCRB)

ICCRB issues start/stop condition, manipulates the SDA pin, monitors the SCL pin, and c
reset in the I^2C control module.

Bit	7	6	5	4	3	2	1	
Bit Name	BBSY	SCP	SDAO	—	SCLO	—	IICRST	
Initial Value	0	1	1	1	1	1	0	
R/W	R/W	R/W	R	R/W	R	—	R/W	

				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 l 1: When reading the SDA pin outputs a high l
4	—	1	R/W	Reserved The write value should always be 1.
3	SCLO	1	R	This bit monitors the SCL output level. When reading and SCLO is 1, the SCL pin ou high level. When reading and SCLO is 0, the outputs a low level.
2	—	1	—	Reserved This bit is always read as 0.
1	IICRST	0	R/W	IIC Control Module Reset This bit reset the IIC control module except th registers. If hang-up occurs because of comm failure during I^2C operation, by setting this bit
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 da transfer except for the acknowledge bit. When set to 1, after the falling of the clock for the last the low period is extended for two transfer cloc When this bit is cleared to 0, data and the ackn bit are transferred consecutively with no waits i 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 cleared 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 ir

001: 2

010: 3

011: 4

100: 5

101: 6

110: 7

111: 8

I²C control module can be reset without settin
ports and initializing the registers.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, this bit or disables the transmit data empty interrupt (T request. 0: Transmit data empty interrupt (TXI) request disabled 1: Transmit data empty interrupt (TXI) request enabled
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end in (TEI) request at the rising of the ninth clock wh TDRE bit in ICSR is set to 1. The TEI request c canceled by clearing the TEND bit or the TEIE 0: Transmit end interrupt (TEI) request is disab 1: Transmit end interrupt (TEI) request is enab
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive full inte (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 t RDRF or RIE bit to 0. 0: Receive data full interrupt (RXI) request is d 1: Receive data full interrupt (RXI) request is e

				0: Stop condition detection interrupt (STPI) re disabled 1: Stop condition detection interrupt (STPI) re enabled
2	ACKE	0	R/W	Acknowledge Bit Decision Select 0: The value of the acknowledge bit is ignored continuous transfer is performed 1: If the acknowledge bit is 1, continuous tran suspended
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowle that are returned by the receive device. This b 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 conditions] • When data is transferred from ICDRT to IC and ICDRT becomes empty • When the TRS bit is set • When a start condition (that includes a retra condition) is issued • When the slave receive mode shifts to the s transmit mode [Clearing conditions] • When 0 is written to this bit after reading TI (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 data is written to ICDRT
6	TEND	0	R/W	Transmit End [Setting condition] • When the ninth clock of SCL rises while the flag is 1 [Clearing conditions] • When 0 is written to this bit after reading TI (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 data is written to ICDRT

				• When data is read from ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from th device in transmission while the ACKE bit 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 b 0 while the corresponding interrupt is enal sure to read the flag after writing 0 to it.)
3	STOP	0	R/W	Stop Condition Detection Flag [Setting condition] • In master mode, when a stop condition is after frame transfer • In slave mode, when a stop condition is d after a general call or after the slave addr came as the first byte after detection of a condition has matched the address set in [Clearing condition] • When 0 is written to this bit after reading S (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.)

				disagree at the rising of SCL in master tran mode • When the SDA pin outputs a high level in m mode while a start condition is detected [Clearing condition] • When 0 is written to this bit after reading AI (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.)
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 when frame following a start condition matches bits S SVA0 in SAR. [Setting conditions] • When the slave address is detected in slav receive mode • When the general call address is detected receive mode [Clearing condition] • When 0 is written to this bit after reading AA (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.)

sure to read the flag after writing 0 to it.)

17.3.6 Slave Address Register (SAR)

SAR is sets the slave address. In slave mode, if the upper 7 bits of SAR match the upper
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 t bus.
0	—	0	R/W	Reserved Although this bit is readable/writable, only 0 s written to.

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

17.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit read-only register that stores the receive data. When one byte of data h received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can received. ICDRR is a receive-only register; therefore, this register cannot be written to by 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	

17.3.9 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, d transferred from ICDRS to ICDRR after one by of data is received. This register cannot b from the CPU.

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

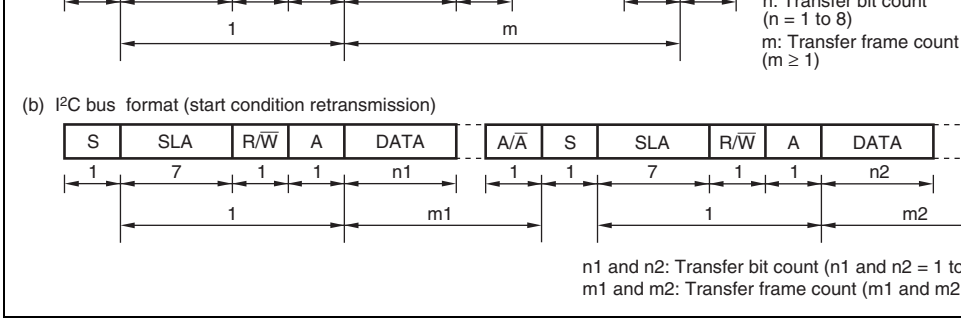


Figure 17.3 I²C Bus Formats

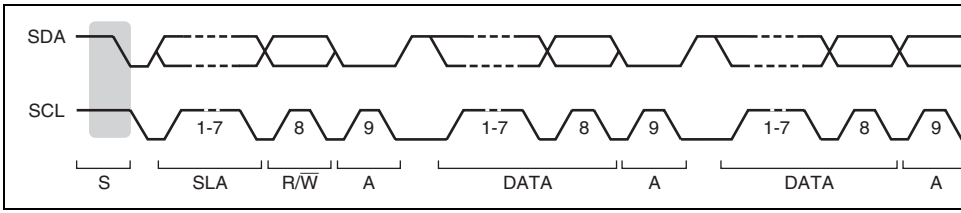


Figure 17.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is h

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer; from the slave device to the master devi
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 h

ICDRT to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using the MOV 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 by the slave address and R/W) to ICDRT. After this, when TDRE is automatically cleare 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 is at the rising of the ninth transmit clock pulse. Read the ACKBR bit in ICIER to confi the slave device has been selected. Then, write the second byte data to ICDRT. When is 1, the slave device has not been acknowledged, so issue a stop condition. To issue t condition, write 0 to BBSY and SCP using the MOV instruction. SCL is fixed to a lov 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 en byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR is 1) fr receive device while CKE in ICIER is 1. Then, issue the stop condition to clear TEND NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mod

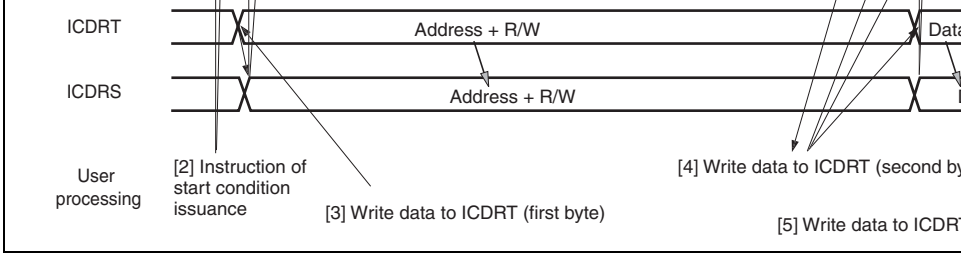


Figure 17.5 Master Transmit Mode Operation Timing 1

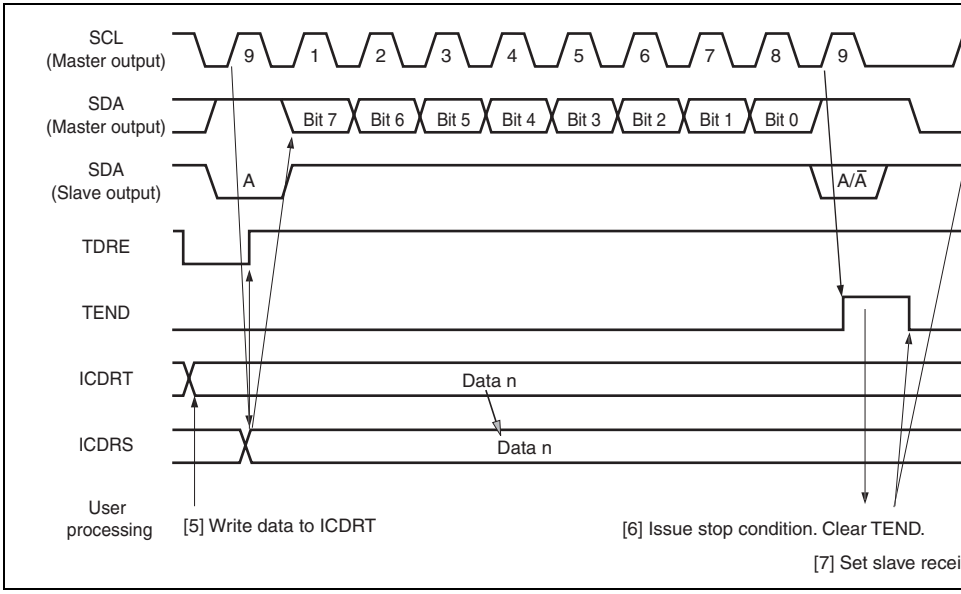


Figure 17.6 Master Transmit Mode Operation Timing 2

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 to rising of the ninth receive clock pulse. At this time, the received data is read by readin ICDRR. At the same time, RDRF is cleared.
4. The continuous reception is performed by reading ICDRR and clearing RDRF to 0 ev RDRF is set. If the eighth receive clock pulse falls after reading ICDRR by other proc 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 ICCR1 before reading IC 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 c 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.

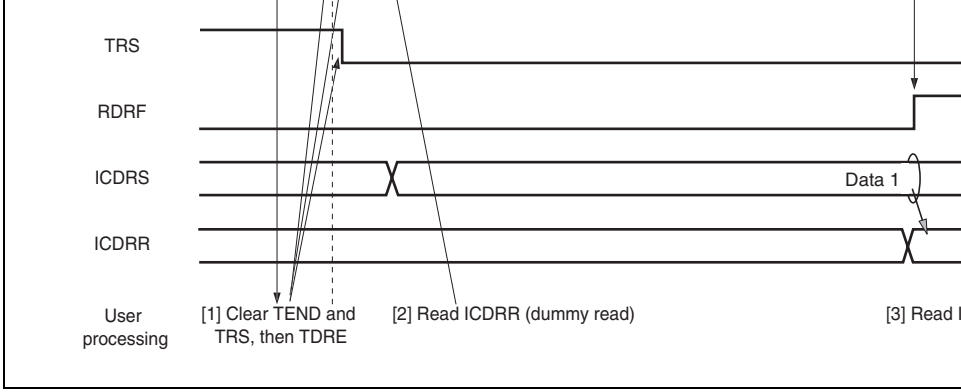


Figure 17.7 Master Receive Mode Operation Timing 1

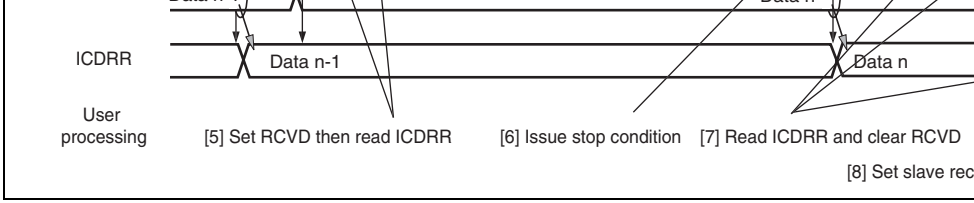


Figure 17.8 Master Receive Mode Operation Timing 2

17.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, and the master device o the receive clock pulse and returns an acknowledge signal. Figures 17.9 and 16.10 show t operation timings in slave transmit mode. The transmission procedure and operations in s 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. ACKBIT in ICIER, and perform other initial settings. Set the MST and TRS bits in IC 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, a rising of the ninth clock pulse. At this time, if the eighth bit data (R/$\overline{W}$) is 1, TRS in IC and TDRE in ICSR are set to 1, and the mode changes to slave transmit mode automa The continuous transmission is performed by writing the transmit data to ICDRT ever TDRE is set.
3. If TDRE is set after writing the last transmit data to ICDRT, wait until TEND in ICSR 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for end processing, and read ICDRR (dummy read) to free SCL.
5. Clear TDRE.

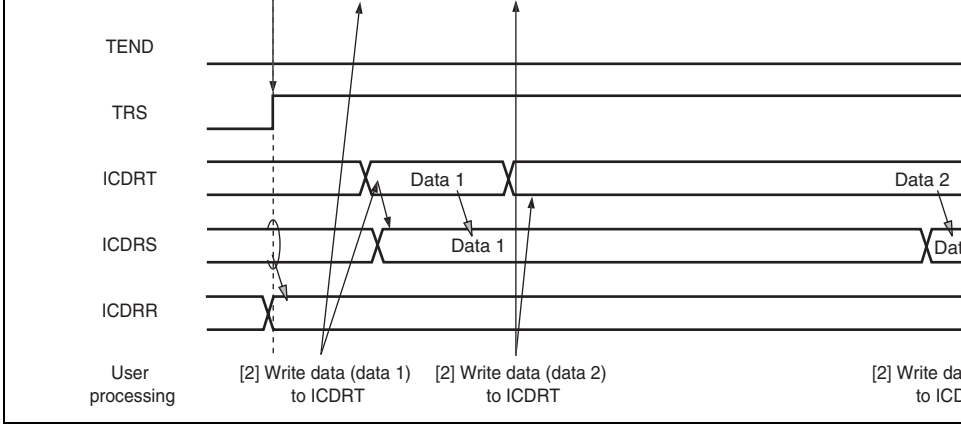


Figure 17.9 Slave Transmit Mode Operation Timing 1

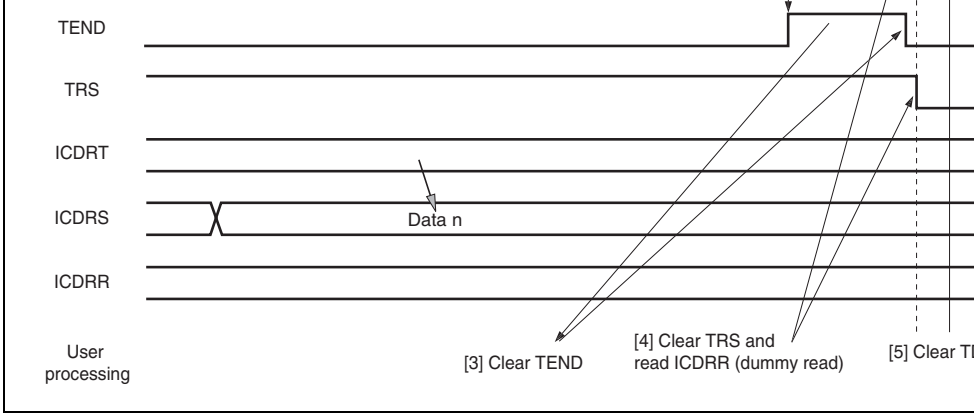


Figure 17.10 Slave Transmit Mode Operation Timing 2

2. When the slave address matches in the first frame following detection of the start co
the slave address outputs the level specified by ACKBT in ICIER to SDA, at the risi
ninth clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy re
(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
fixed to a low level until ICDRR is read. The change of the acknowledge (ACKBT)
before reading ICDRR to be returned to the master device is reflected in the next tra
frame.
4. The last byte data is read by reading ICDRR.

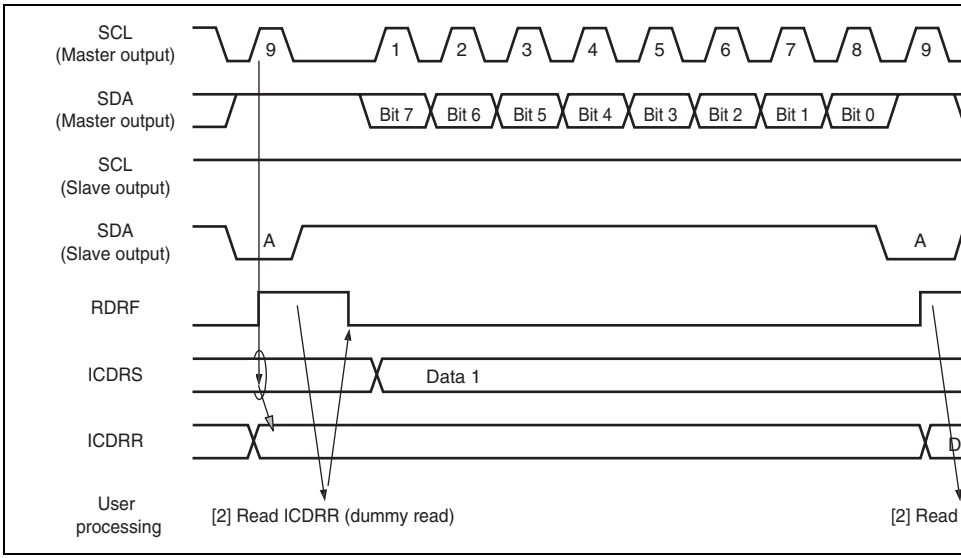


Figure 17.11 Slave Receive Mode Operation Timing 1

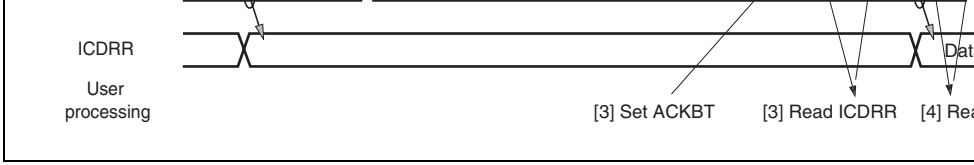


Figure 17.12 Slave Receive Mode Operation Timing 2

17.4.6 Noise Canceler

The logic levels at the SCL and SDA pins are routed through the noise cancelers before b
latched internally. Figure 17.13 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The signal inpu
(or SDA) is sampled on the system clock, but is not passed forward to the next circuit unl
outputs of both latches agree. If they do not agree, the previous value is held.

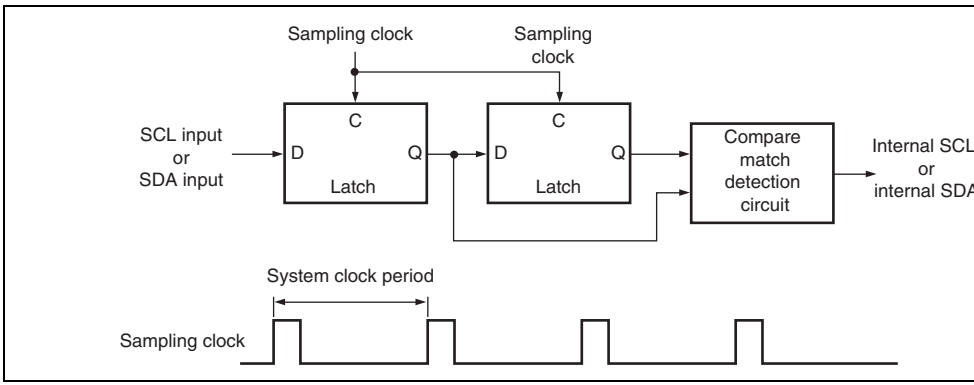


Figure 17.13 Block Diagram of Noise Canceler

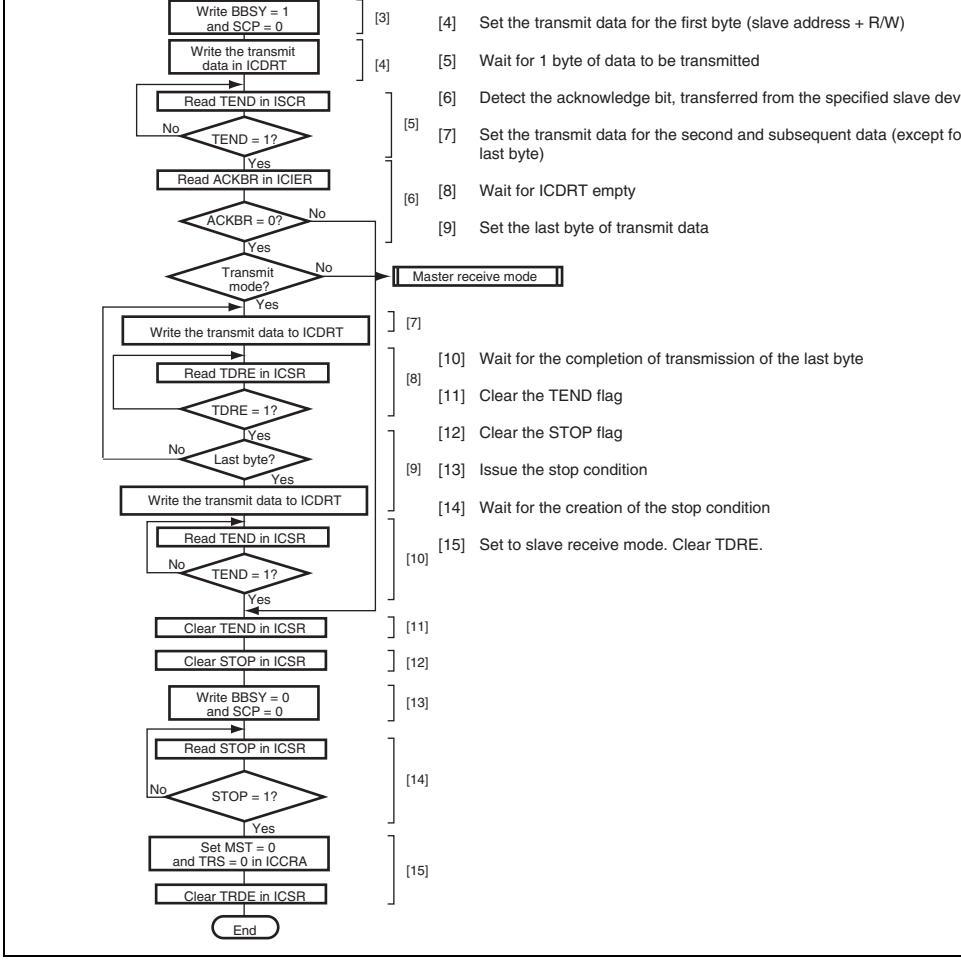
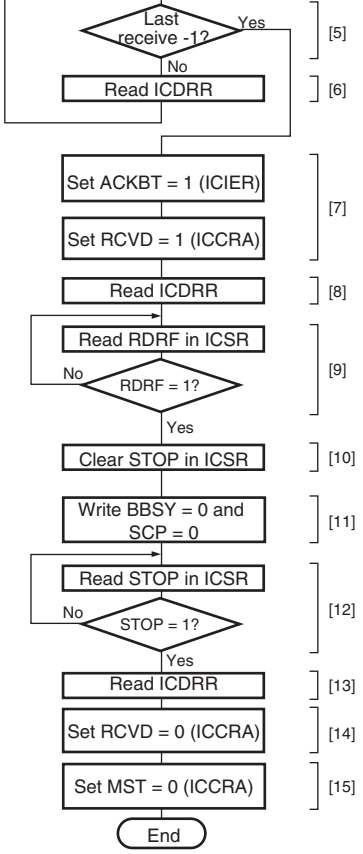
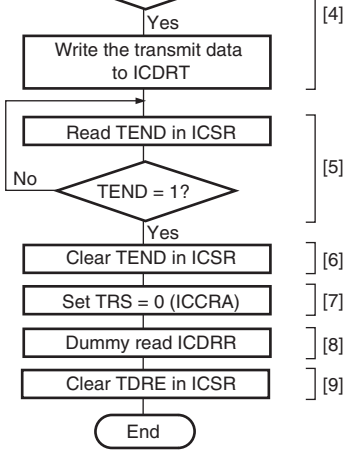


Figure 17.14 Sample Flowchart of Master Transmit Mode



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).

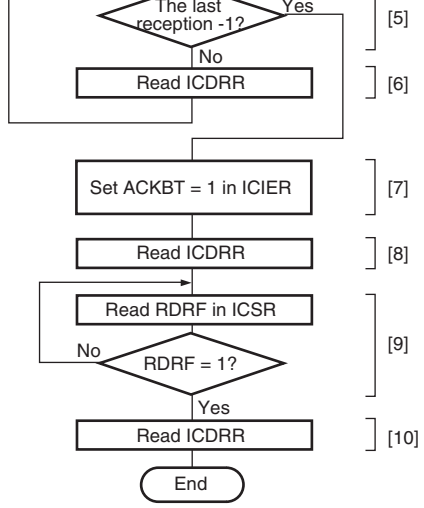
Figure 17.15 Sample Flowchart for Master Receive Mode



[8] Dummy read ICDRR to free the SCL line.

[9] Clear the TDRE flag.

Figure 17.16 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 comp

[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 17.17 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		

Figure 17.18 shows the timing of the bit synchronous circuit, and table 17.4 shows the tim
SCL output changes from low to Hi-Z and the period which SCL is monitored.

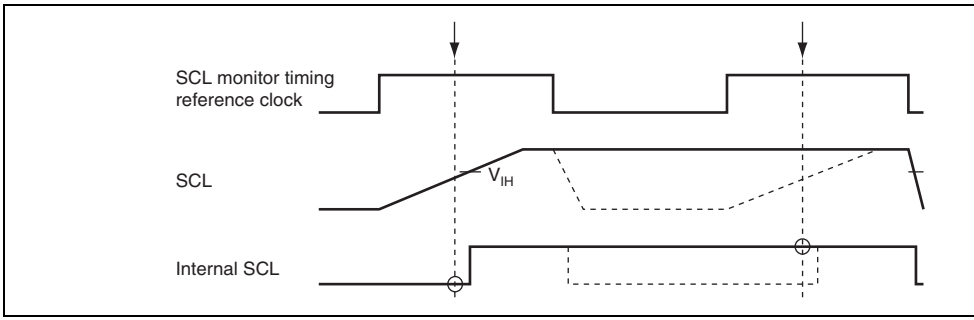


Figure 17.18 Timing of the Bit Synchronous Circuit

Table 17.4 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

Confirm the ninth falling edge of the clock before issuing a stop or a repeated start cond
ninth falling edge can be confirmed by monitoring the SCLO bit in the I²C bus control r
(ICCRB).

If a stop or a repeated start condition is issued at certain timing in either of the followin
stop or repeated start condition may be issued incorrectly.

- The rising time of the SCL signal exceeds the time given in section 17.6, Bit Synchr
  Circuit, because of the load on the SCL bus (load capacitance or pull-up resistance).
- The bit synchronous circuit is activated because a slave device holds the SCL bus lo
  the eighth clock.

this LSI to a value that is equal to or higher than 1/1.8 times the transfer rate of the fastes
For example, if the fastest rate of other master is 400 kbps, the I^2C transfer rate of this LS
be at least 223 kbps (= 400/1.8).

17.7.5 Restriction on Bit Manipulation when Setting the MST and TRS Bits in M Master Mode

If the MST and TRS bits are manipulated sequentially to select master transmit mode, a c
state (for example, the AL bit in ICSR is set to 1 in master transmit mode (MST = 1, TRS
can result depending on the timing of arbitration lost that might occur during execution o
manipulation instruction for the TRS bit. This phenomenon can be avoided by the follow
operations.

- In multi-master mode, use the MOV instruction to set the MST and TRS bits.
- If arbitration is lost, check to see whether both MST and TRS bits have been cleared t
  both bits are not clear, clear them to 0.

17.7.6 Notes on Master Receive Mode

In master receive mode, when the value of RDRF is 1 at the falling edge of the eighth clo
the SCL signal is pulled low. If ICDRR is read near the falling edge of the eighth clock p
is fixed to low only during the eighth clock cycle of the next received data and, after that,
released even if ICDRR is not read, which allows the ninth clock pulse to be output. As a
some data fails to be received. This phenomenon can be avoided by the following operati

- In master receive mode, read ICDRR before the rising edge of the eighth clock pulse.
- In master receive mode, set the RCVD bit to 1 and perform byte-wise communication

- Eight input channels
- Conversion cycles: 5.33 µs per channel (with ADCLK at 7.5 MHz operation)
- Two kinds of operating modes
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on 1 to 4 channels, or 1 to 8 channels
- Eight data registers

  A/D conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three types of conversion start

  Conversion can be started by software, a conversion start trigger from the 16-bit time unit (TPU)* or 8-bit timer (TMR)*, or an external trigger signal.
- Interrupt source

  A/D conversion end interrupt (ADI) request can be generated.
- Module stop state specifiable

Note: * Starting by a trigger from the TPU/TMR is available on the on-chip emulator available on other emulators.

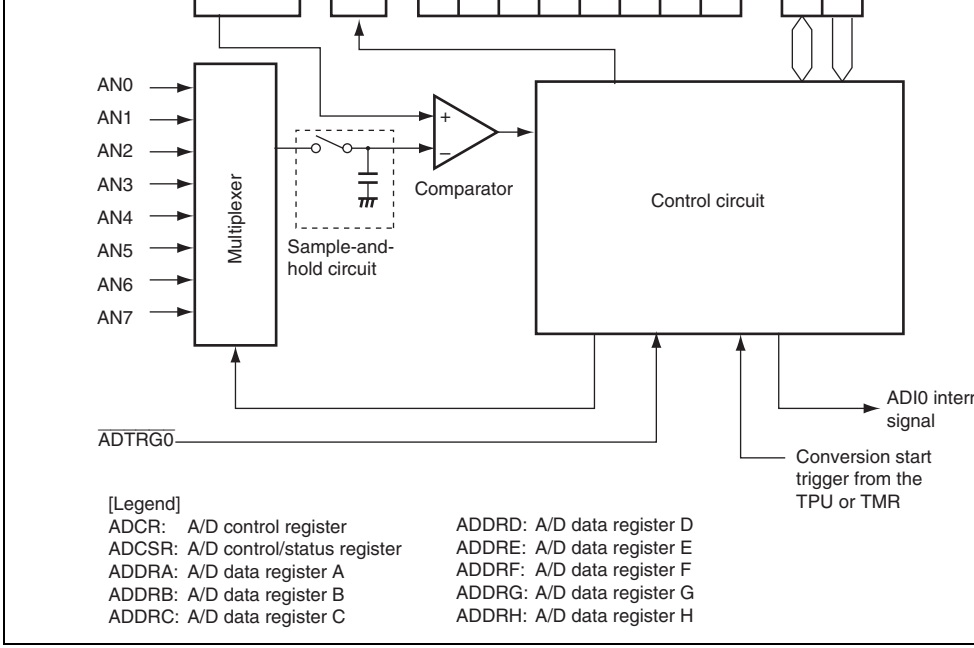


Figure 18.1 Block Diagram of A/D Converter

Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG0}}$	Input	External trigger input for starting A/D c
Analog power supply pin	AV_{CC}	Input	Analog block power supply
Analog ground pin	AV_{SS}	Input	Analog block ground
Reference voltage pin	Vref	Input	A/D conversion reference voltage

18.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D data register E (ADDRE)
- A/D data register F (ADDRF)
- A/D data register G (ADDRG)
- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
Bit Name											—	—	—	—	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 18.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register Which Stores Conversion Resul
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conv [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all specified in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When the DTC or DMAC is activated by an interrupt and ADDR is read
6	ADIE	0	R/W	A/D Interrupt Enable When this bit is set to 1, ADI interrupts by ADF enabled.
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and converter enters wait state. Setting this bit to 1 starts A/D conversion. In sin this bit is cleared to 0 automatically when A/D c on the specified channel ends. In scan mode, A conversion continues sequentially on the specif channels until this bit is cleared to 0 by softwar or hardware standby mode.

0011: AN3
0100: AN4
0101: AN5
0110: AN6
0111: AN7
1XXX: Setting prohibited

- When SCANE = 1 and SCANS = 0
  0000: AN0
  0001: AN0 and AN1
  0010: AN0 to AN2
  0011: AN0 to AN3
  0100: AN4
  0101: AN4 and AN5
  0110: AN4 to AN6
  0111: AN4 to AN7
  1XXX: Setting prohibited
- When SCANE = 1 and SCANS = 1
  0000: AN0
  0001: AN0 and AN1
  0010: AN0 to AN2
  0011: AN0 to AN3
  0100: AN0 to AN4
  0101: AN0 to AN5
  0110: AN0 to AN6
  0111: AN0 to AN7
  1XXX: Setting prohibited

[Legend]
X: Don't care
Note: * Only 0 can be written to this bit, to clear the flag.

7	TRGS1	0	R/W	Timer Trigger Select 1 and 0, Extended Trigger
6	TRGS0	0	R/W	These bits select enabling or disabling of the st
0	EXTRGS	0	R/W	conversion by a trigger signal. 000: A/D conversion start by external trigger is 010: A/D conversion start by conversion trigger is enabled. 100: A/D conversion start by conversion trigger is enabled. 110: A/D conversion start by the $\overline{\text{ADTRG0}}$ pin is enabled.* 001: External triggers are disabled 011: Setting prohibited 101: Setting prohibited 111: Setting prohibited
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	These bits select the A/D conversion operating 0X: Single mode 10: Scan mode. A/D conversion is performed continuously for channels 1 to 4. 11: Scan mode. A/D conversion is performed continuously for channels 1 to 8.

1	ADSTCLR	0	R/W	A/D Start Clear This bit sets automatic clearing of the ADST bit i mode. 0: Prohibits automatic clearing of the ADST bit in mode. 1: Performs automatic clearing in scan mode if a selected channels complete A/D conversion.

[Legend]

X: Don't care

Note: * To set A/D conversion to start by the $\overline{\text{ADTRG0}}$ pin, the DDR bit and ICR bit for corresponding pin should be set to 0 and 1, respectively. For details, see secti Ports.

In single mode, A/D conversion is to be performed only once on the analog input of the single channel.

1. A/D conversion for the selected channel is started when the ADST bit in ADCSR is software, TPU, TMR, 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 to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared A/D conversion ends. The A/D converter enters wait state. If the ADST bit is cleared during A/D conversion, A/D conversion stops and the A/D converter enters wait stat

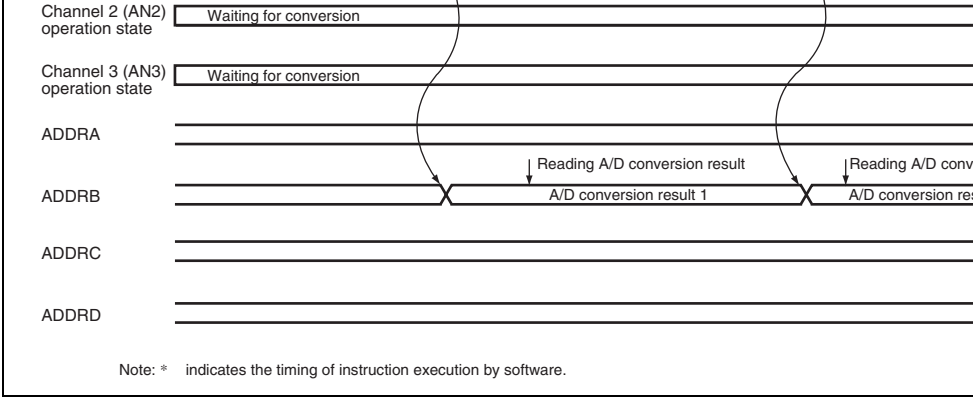
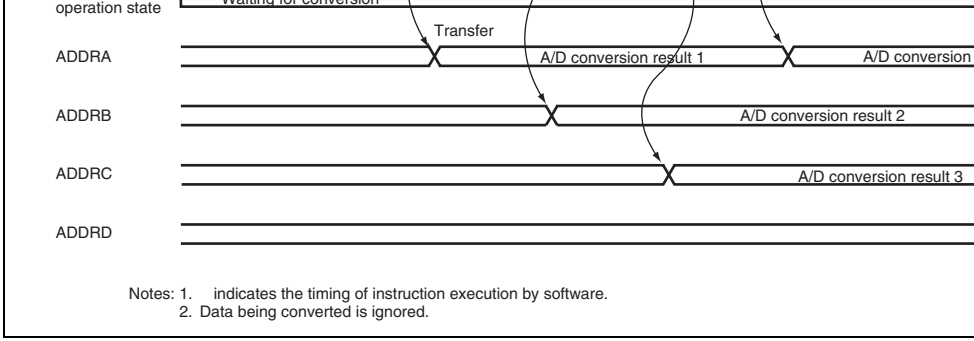


Figure 18.2 Example of A/D Converter Operation (Single Mode, Channel 1 Sele

A/D conversion on a maximum of four channels (SCANE and SCANS = B'10) or on
maximum of eight channels (SCANE and SCANS = B'11) can be selected. When co
A/D conversion is performed on four channels, A/D conversion starts on AN0 when
CH2 = B'00, whereas starts on AN4 when CH3 and CH2 = B'01. When consecutive
conversion is performed on eight channels, A/D conversion starts on AN0 when CH

2. When A/D conversion for each channel is completed, the A/D conversion result is se
transferred to the corresponding ADDR of each channel.
3. When A/D conversion of all selected channels is completed, the ADF bit in ADCSR
If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. A/D co
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 th
bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and th
converter enters wait state. If the ADST bit is later set to 1, A/D conversion starts ag
the first channel in the group.



Notes: 1. indicates the timing of instruction execution by software.
2. Data being converted is ignored.

Figure 18.3 Example of A/D Conversion (Continuous Scan Mode, Three Channels (AN0 to AN2) Selected)

(2) One-Cycle Scan Mode

1. Set the ADSTCLR bit in ADCR to 1.
2. When the ADST bit in ADCSR is set to 1 by software, TPU, TMR, or an external trig input, A/D conversion starts on the first channel in the specified channel group. Conse A/D conversion on a maximum of four channels (SCANE and SCANS = B'10) or on maximum of eight channels (SCANE and SCANS = B'11) can be selected. When con A/D conversion is performed on four channels, A/D conversion starts on AN0 when C CH2 = B'00, whereas starts on AN4 when CH3 and CH2 = B'01. When consecutive A conversion is performed on eight channels, A/D conversion starts on AN0 when CH3
3. When A/D conversion for each channel is completed, the A/D conversion result is sec transferred to the corresponding ADDR of each channel.
4. 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.

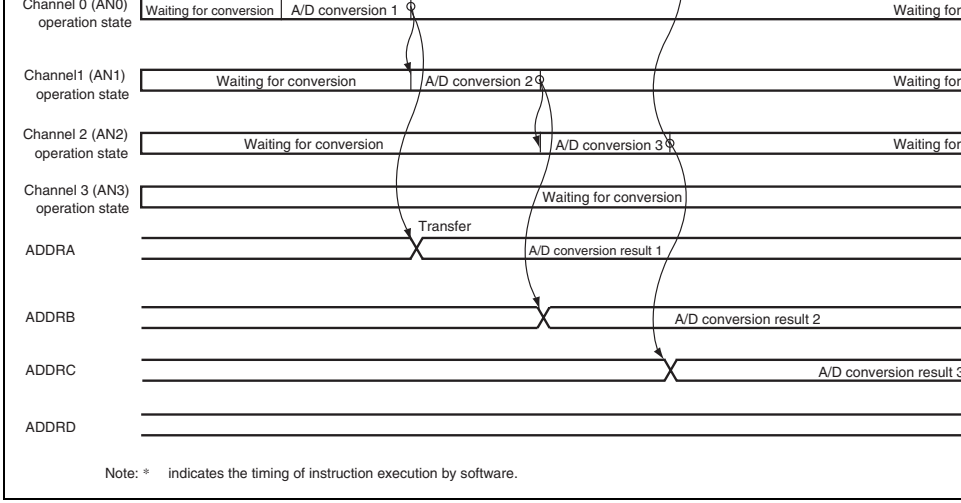
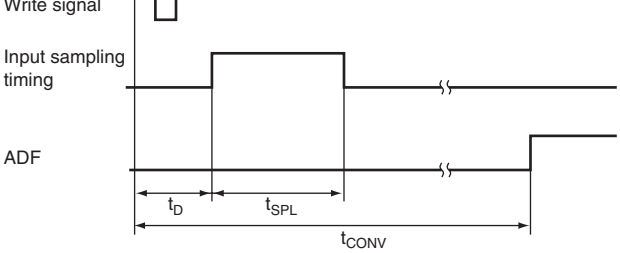


Figure 18.4 Example of A/D Conversion (One-Cycle Scan Mode, Three Channels (AN0 to AN2) Selected)

18.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples th input when the A/D conversion start delay time (t_D) passes after the ADST bit in ADCSI 1, then starts A/D conversion. Figure 18.5 shows the A/D conversion timing. Table 18.3 the A/D conversion time.

As indicated in figure 18.5, the A/D conversion time (t_{CONV}) includes t_D and the input sam (t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCSR. The conversion time therefore varies within the ranges indicated in table 18.3.



[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 18.5 A/D Conversion Timing

Table 18.3 A/D Conversion Characteristics (Single Mode)

		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	3	—	4	3	—	5	3	—	6	3	—
Input sampling time	t_{SPL}	—	15	—	—	30	—	—	60	—	—	12
A/D conversion time	t_{CONV}	45	—	46	85	—	87	165	—	168	325	—

Note: Values in the table are the number of states.

A/D conversion can be externally triggered. When the TRGS1, TRGS0, and EXTRGS b
ADCR are set to B'110, an external trigger is input from the $\overline{\text{ADTRG0}}$ pin. A/D convers
when the ADST bit in ADCSR is set to 1 on the falling edge of the $\overline{\text{ADTRG0}}$ pin. Other
operations, in both single and scan modes, are the same as when the ADST bit has been
software. Figure 18.6 shows the timing.

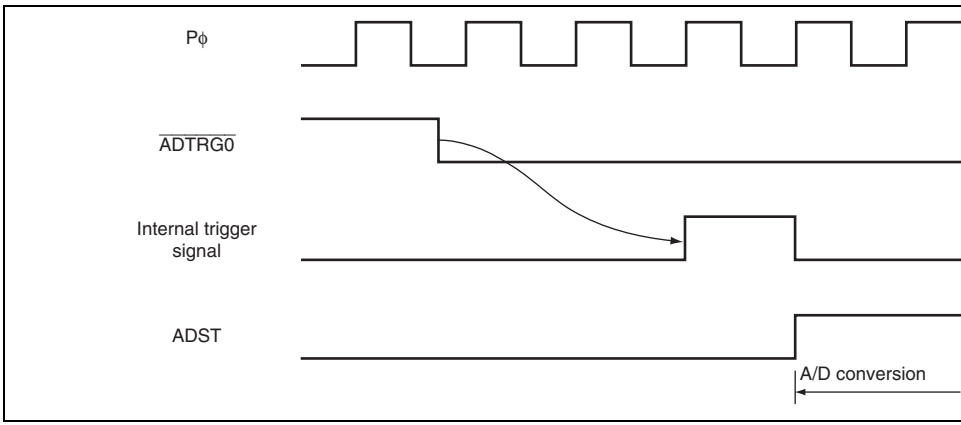


Figure 18.6 External Trigger Input Timing

ADI	A/D conversion end	ADF	Impossible	Possible

18.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes.
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 18.7).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion charac when the digital output changes from the minimum voltage value B'0000000000 (H'0 B'0000000001 (H'001) (see figure 18.8).
- 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 18.8).
- 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 18.8).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offse full-scale error, quantization error, and nonlinearity error.

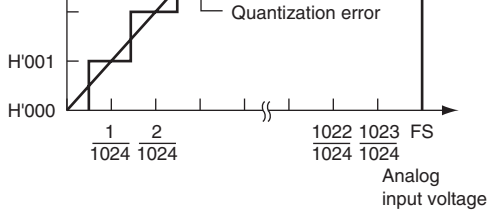


Figure 18.7 A/D Conversion Accuracy Definitions

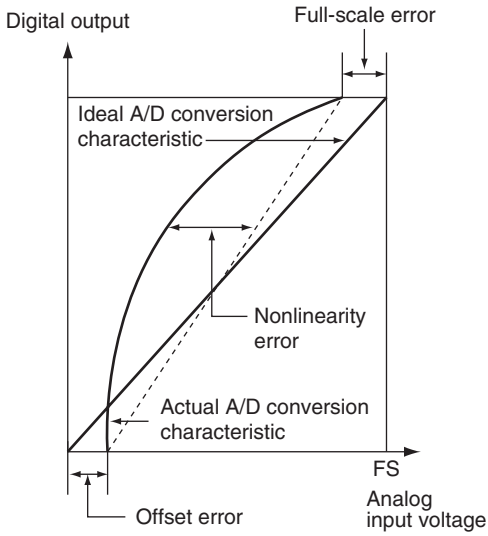


Figure 18.8 A/D Conversion Accuracy Definitions

stop control register, see section 24, Power-Down Modes.

18.7.2 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 both of the CKS
CKS0 bits to 1 and clear all of the ADST, TRGS1, TRGS0, and EXTRGS bits to 0 to dis
conversion. After that, dummy-read the ADCSR register and then enter software standby

18.7.3 Notes on A/D Conversion Start by an External Trigger

If any of actions (1 to 3 below) is performed while activation by an external trigger* is in
stopping A/D conversion may be impossible.

Note: * External trigger refers to input on the $\overline{\text{ADTRG}}$ pin or the conversion trigger fr
peripheral module (TMR or TPU).

1. When the setting for activation by an external trigger is in use, writing to change the v
the ADST bit in ADCSR from 0 to 1.
2. Changing the setting from activation by an external trigger to prohibition of A/D conv
start by an external trigger.
3. Changing the scan mode (SCANE and ADSTLCR bits; from continuous scan mode to
mode or one-cycle scan mode) while the setting for activation by an external trigger is

For details on the procedure in cases where point 2 or 3 is applicable, see figure 18.9

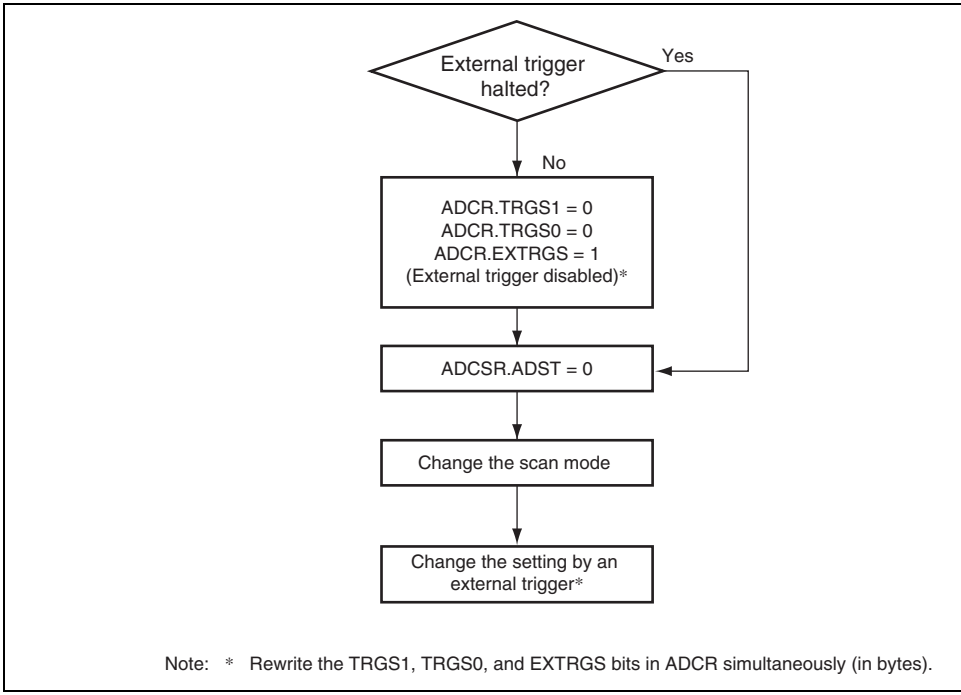


Figure 18.9 Procedure for Changing the Mode When Setting for Activation by an Trigger is in Use

(2) In Scan Mode (Continuous Scan Mode)

- When the A/D converter is activated by software

  Do not clear the ADST bit by software during A/D conversion. To stop A/D conversi
  rewrite the SCANE bit to change modes from scan mode to single mode. By rewriting
  SCANE bit, the A/D converter is stopped without clearing the ADST bit by software.

  However, after rewriting the SCANE bit, it may take up to 1.5-channel A/D conversio
  stop A/D conversion and set the A/D end flag (ADF) to 1. Moreover, the ADDR valu
  A/D conversion is completed should not be used.

  For detailed settings, see figure 18.10.

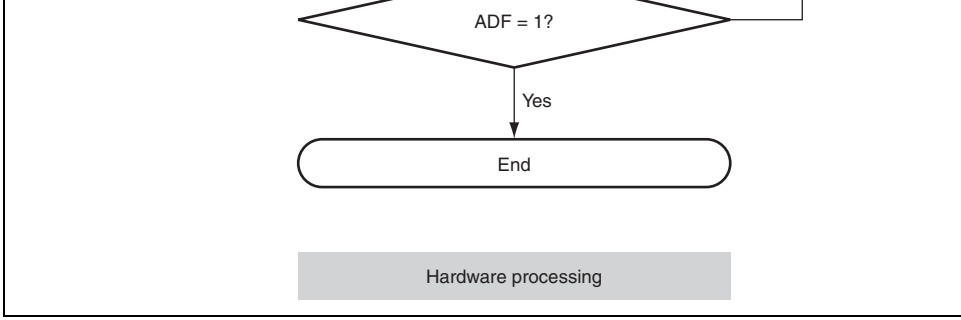


Figure 18.10 Stopping Continuous Scan Mode Activated by Software

- When the A/D converter is activated by an external trigger

  Do not clear the ADST bit by software during A/D conversion. To stop A/D convers disable external triggers and then rewrite the SCANE bit to change modes from scan single mode. 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 detailed settings, see figure 18.11.

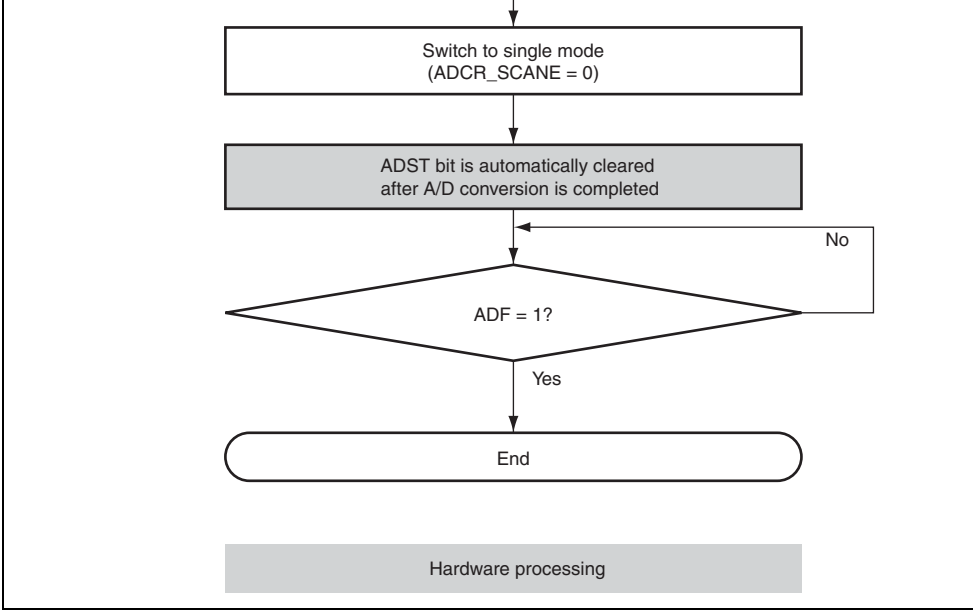


Figure 18.11 Stopping Continuous Scan Mode Activated by External Trigg

with a large differential coefficient (e.g., 5 mV/µs or greater) (see figure 18.12). When c
a high-speed analog signal or conversion in scan mode, a low-impedance buffer should b
inserted.

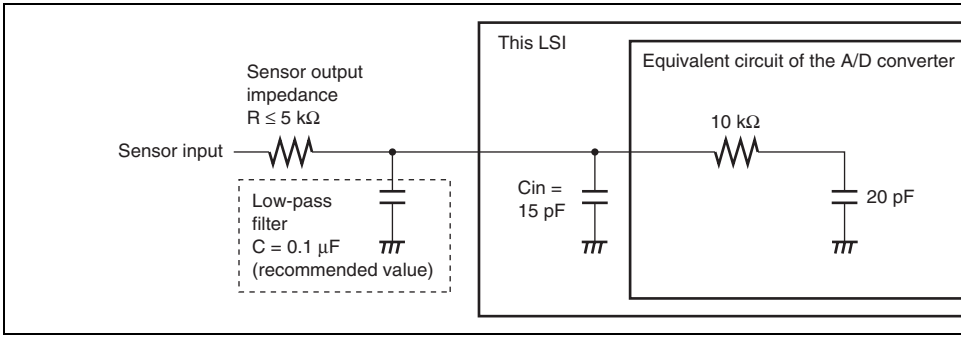


Figure 18.12 Example of Analog Input Circuit

18.7.6 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adv
affect absolute accuracy. Be sure to make the connection to an electrically stable GND s
AVss.

Care is also required to insure that filter circuits do not communicate with digital signals
mounting board, acting as antennas.

- Vref setting range

  The reference voltage at the Vref pin should be set in the range Vref ≤ AVcc.

18.7.8 Notes on Board Design

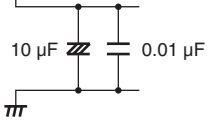
In board design, digital circuitry and analog circuitry should be as mutually isolated as po
and layout in which digital circuit signal lines and analog circuit signal lines cross or are
proximity should be avoided as far as possible. Failure to do so may result in incorrect op
of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Digital circuitry must be isolated from the analog input pins (AN0 to AN7), analog refere
power supply (Vref), and analog power supply (AVcc) by the analog ground (AVss). Als
analog ground (AVss) should be connected at one point to a stable ground (Vss) on the b

18.7.9 Notes on Countermeasure against Noise

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 between AVcc and AV
shown in figure 18.13. Also, the bypass capacitors connected to AVcc and the filter capa
connected to the AN0 to AN7 pins must be connected to AVss.

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 A
converter exceeds the current input via the input impedance (R_{in}), an error will arise in the
input pin voltage. Careful consideration is therefore required when deciding the circuit co

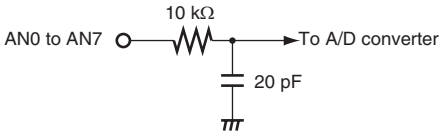


2. R_{in}: Input impedance

Figure 18.13 Example of Analog Input Protection Circuit

Table 18.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ

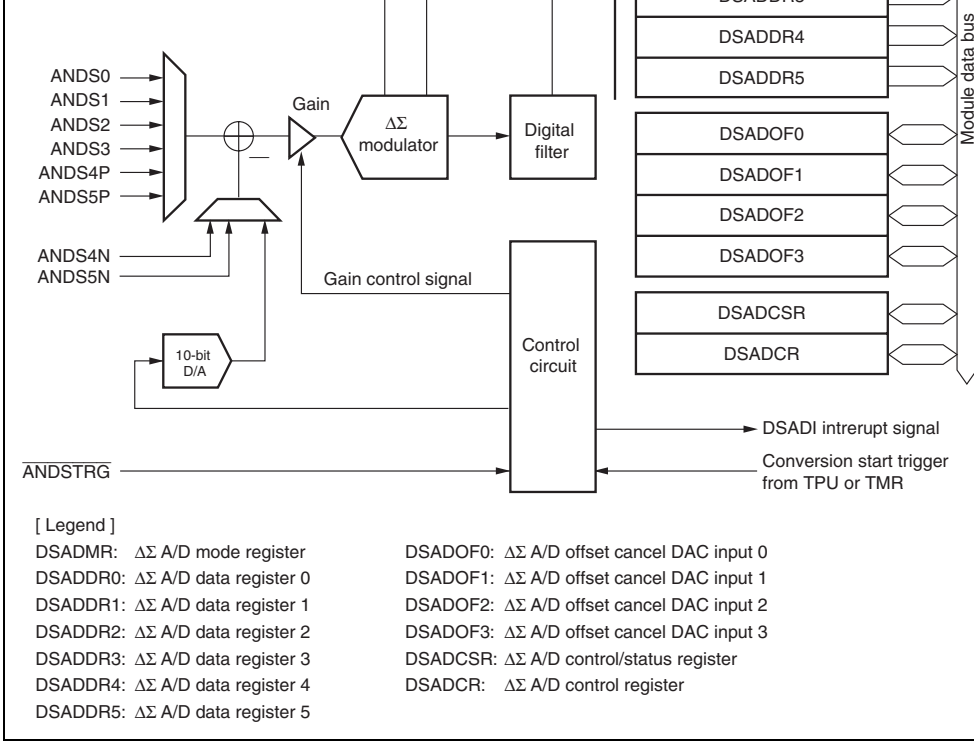


Note: Values are reference values.

Figure 18.14 Analog Input Pin Equivalent Circuit

- 16-bit resolution
- Suitable for sensor applications (cannot be applied to voice and audio applications)
- Conversion method: ΔΣ modulation-based conversion
- Six input channels (time-division multiplexing)
- Two types of input channel (four single-ended channels and two differential input ch
  Offset cancellation by 10-bit DAC on single-ended channels
- Conversion time: 91.5 μs per channel
  (for 286-"state" conversion at Aϕ = 25 MHz, where 1 state = Aϕ/8)
- Six data registers
  Results of A/D conversion are stored in 16-bit registers for the respective channels.
- Three ways of starting A/D conversion
  — Software
  — Trigger from the 16-bit timer pulse unit (TPU)* or 8-bit timer (TMR)*
  — External trigger signal
- Interrupt source
  Generates ΔΣ A/D conversion end interrupt requests (DSADI).
- Can be placed in the module stop state

Note: * Initiation of conversion by a TPU/TMR trigger is available with the on-chip
but not with other emulators.



[Legend]

DSADMR: ΔΣ A/D mode register
DSADDR0: ΔΣ A/D data register 0
DSADDR1: ΔΣ A/D data register 1
DSADDR2: ΔΣ A/D data register 2
DSADDR3: ΔΣ A/D data register 3
DSADDR4: ΔΣ A/D data register 4
DSADDR5: ΔΣ A/D data register 5
DSADOF0: ΔΣ A/D offset cancel DAC input 0
DSADOF1: ΔΣ A/D offset cancel DAC input 1
DSADOF2: ΔΣ A/D offset cancel DAC input 2
DSADOF3: ΔΣ A/D offset cancel DAC input 3
DSADCSR: ΔΣ A/D control/status register
DSADCR: ΔΣ A/D control register

Figure 19.1 Block Diagram of the ΔΣ A/D Converter

Analog input pin 3	ANDS3	Input	
Analog input pin 4-P	ANDS4P	Input	Analog input pins: Differential inp
Analog input pin 4-N	ANDS4N	Input	
Analog input pin 5-P	ANDS5P	Input	Analog input pins: Differential inp
Analog input pin 5-N	ANDS5N	Input	
External trigger input pin for ΔΣ A/D converter	$\overline{\text{ANDSTRG}}$	Input	External trigger input pin for start conversion
Analog power supply pin	AVccA*1	Input	Power supply pin for the analog s the ΔΣ A/D converter
Analog power supply pin	AVccD*1	Input	Power supply pin for the control c the ΔΣ A/D converter
Analog power supply pin	AVccP*1	Input	Power supply pin for the input pin circuit of the ΔΣ A/D converter
Analog ground pin	AVssA	Input	Ground pin for the analog section A/D converter
Analog ground pin	AVssD	Input	Ground pin for the control circuit A/D converter
Analog ground pin	AVssP	Input	Ground pin for the input pin contr the ΔΣ A/D converter
ΔΣ reference voltage (high)	AVrefT*2	Input	For connection of stabilizing capa (between $AV_{ref}B$ and $AV_{ref}T$; 10 μF
ΔΣ reference voltage (low)	AVrefB*2	Input	
Reference voltage pin	AVCM	Output	For connection of a stabilizing ca (0.1μF between AVCM and AV_{SS}
Reference current pin	REXT	Output	For connection of an external res between REXT and $AV_{SS}A$. (51 kΩ tolerance)

Notes: 1. AVccA = AVccD = AvccP must always hold.
2. AVccA = AVrefT, AVrefT > AVrefB, AVrefB = AVssA must always hold.

- ΔΣ A/D offset cancel DAC input 0 (DSADOF0)*
- ΔΣ A/D offset cancel DAC input 1 (DSADOF1)*
- ΔΣ A/D offset cancel DAC input 2 (DSADOF2)*
- ΔΣ A/D offset cancel DAC input 3 (DSADOF3)*
- ΔΣ A/D control/status register (DSADCSR)
- ΔΣ A/D control register (DSADCR)
- ΔΣ A/D mode register (DSADMR)

Note: * Offset cancellation here means canceling DC components of the signals input input pins ANDS0 to ANDS3.

Bit	Bit Name	Initial Value	R/W	Description
7	BIASE	0	R/W	Biasing Circuit Control Controls whether the biasing circuit is stopped c 0: Biasing circuit is stopped. 1: Biasing circuit runs.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
2	ACK2	0	R/W	ΔΣ A/D Converter Clock Select
1	ACK1	0	R/W	These bits select the frequency of the ΔΣ A/D c
0	ACK0	0	R/W	clock (Aϕ). The values shown below for each se frequency multipliers for the input clock. Set the that Aϕ is approximately 25 MHz. See section 2 Pulse Generator, for details. 000: × 1/6 001: × 1/5 010: × 1/4 011: × 1/3 1xx: Setting prohibited

[Legend] x: Don't care.

The A/D-converted data is stored in bit 15 to bit 0 as a signed binary number (two's comp
Bit 15 holds the MSB and bit 0 the LSB.

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

19.3.3 ΔΣ A/D Control/Status Register (DSADCSR)

DSADCSR controls A/D conversion and interrupts and selects analog input channels.

When writing to the register to change the settings of bits SCANE and CH5 to CH0, bit A
must be clear.

Bit	15	14	13	12	11	10	9	
Bit Name	ADF	ADIE	ADST	—	SCANE	—	TRGS1	TR
Initial Value:	0	0	0	0	0	0	0	
R/W:	R/(W)*	R/W	R/W	R	R/W	R	R/W	R

Bit	7	6	5	4	3	2	1	
Bit Name	—	—	CH5	CH4	CH3	CH2	CH1	C
Initial Value:	0	0	0	0	0	0	0	
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R

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

				transfer of data in DSADDRn.
14	ADIE	0	R/W	A/D Conversion Interrupt Enable Setting this bit to 1 enables generation of DSAD requests in accord with the ADF bit.
13	ADST	0	R/W	A/D Conversion Start Controls starting and stopping of A/D conversion Clearing this bit to 0 stops A/D conversion, placi converter in the wait sate. Setting this bit to 1 sta conversion. In single mode, ADST is automatically cleared a of A/D conversion on the selected channels. In s mode, ADST must be cleared by software becau cleared automatically. [Setting conditions] • Writing 1 to ADST by software • Input of an A/D conversion trigger signal whi of A/D conversion by a trigger is enabled (TR TRGS0 ≠ B'00) [Clearing conditions] • Writing 0 to ADST by software • End of A/D conversion on all selected chann SCANE = 0

				This bit is always read as 0. The write value shou be 0.
9	TRGS1	0	R/W	Timer Trigger Select 1, 0
8	TRGS0	0	R/W	These bits enable starting of A/D conversion by a signal. 00: Disables starting by trigger signals. 01: Enables starting by a trigger from the TPU. 10: Enables starting by a trigger from the TMR. 11: Enables starting by the $\overline{\text{ADSTRG}}$ pin input.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
5	CH5	0	R/W	A/D Conversion Channel Select
4	CH4	0	R/W	These bits select the analog input channels for A conversion. They are independent of each other be set as desired.
3	CH3	0	R/W	
2	CH2	0	R/W	0: Channel n is not selected.
1	CH1	0	R/W	1: Channel n is selected.
0	CH0	0	R/W	(n = 0 to 5)

Notes: 1. Only 0 can be written here, to clear the flag.

2. When selecting starting of A/D conversion by the $\overline{\text{ADSTRG}}$ signal, clear the D for the corresponding pin to 0 and set the ICR bit to 1. See section 11, I/O Port details.

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

Bit	Bit Name	Initial Value	R/W	Description
15	CKS	0	R/W	Clock Select Sets the A/D conversion time. 0: 286-state conversion 1: Setting prohibited (One “state” = Aϕ/8)
14	—	0	R	Reserved This bit is always read as 0. The write value sho be 0.
13	GAIN1	1	R/W	Gain Select
12	GAIN0	1	R/W	These bits set the gain for amplifying the analog signals. 00: ×1 01: ×2 10: ×4 11: ×8
11	—	0	R	Reserved This bit is always read as 0. The write value sho be 0.

				always be 0.
3	—	1	R/W	Reserved The write value should always be 1.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

The analog level for offset cancellation that corresponds to the register setting is calcula
using formula (1). Table 19.3 shows examples of register values and calculated analog l
offset cancellation.

$$DOF = DSADOF/2^{10} \times (AVrefT - AVrefB) \quad \ldots \text{ Formula (1)}$$

DOF: Analog level for offset cancellation (V)
DSADOF: Register value set in DSADOFn[9:0] for the corresponding channel
AVrefT: ΔΣ reference voltage (high) (V), AVrefT = AVccA
AVrefB: ΔΣ reference voltage (low), AVrefB = AvssA

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

Table 19.2 Setting Values of Gain and DSADOFn

	DSADOFn (n = 0 to 3)	
GAIN1, GAIN0	**Settable Range**	**Remarks**
B'00	H'0200	Always set to H'0200.
B'01	H'0200	Always set to H'0200.
B'10	H'0000 to H'03FE	Bit 0 must be clear (= 0)
B'11	H'0000 to H'03FF	—

H'3FF	1023/1024 × (AVrefT – AVrefB)	2.9971

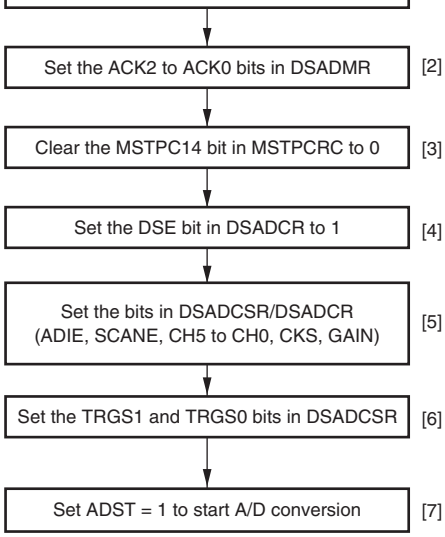
19.4 Operation

The ΔΣ A/D converter uses a ΔΣ modulator to convert analog input voltages within the ra
specified by the voltages on the AVrefT and AVrefB pins to digital values with 16-bit res
The ΔΣ A/D converter is made up of three parts: an analog block built around a ΔΣ modu
digital filter, and a control circuit.

In the analog block, the ΔΣ modulator amplifies the input signals (eight-fold when the GA
GAIN0 bits in DSADCR is set to B'11) and converts them. During this process, the DC o
the signals input from the single-ended input signal pins (ANDS0, ANDS1, ANDS2, AN
cancelled if offset values have been set in the DSADOF0 to DSADOF3 registers. Differe
input voltages on the differential input pins (ANDS4P, ANDS4N and ANDS5P, ANDS5
also be converted.

The voltage of a selected analog input signal is sampled at the Aϕ/8 clock frequency
(oversampling frequency) and converted to a series of digital values by the second-order
modulator. The result of conversion is passed through a decimation filter (digital filter) a
in the corresponding ΔΣ A/D data register as a 16-bit signed binary number (two's compl

The ΔΣ A/D converter operates in either single mode or scan mode. Multiple channels ar
specified by selecting multiple A/D conversion channel-selection bits.



details. BIASE should be set while the ΔΣ A/D con
the module stop state.

[2] Sets a frequency-divided clock signal for the ΔΣ A
converter. When changing the clock-division settin
ΔΣ A/D converter in the module stop state.

[3] Releases the ΔΣ A/D converter from the module s
On release, supply of the Aϕ clock and the Pϕ cloc
connected to the ΔΣ A/D converter start.

[4] Starts the ΔΣ modulator.
The analog circuit starts to operate, consuming a
amount of power. The ΔΣ A/D converter enters the

[5] Sets the operating mode of the ΔΣ A/D converter,
channels, etc.

[6] Sets the trigger input for starting A/D conversion.
Leave the bits at the initial value if A/D conversion
started by software.

[7] A/D conversion starts when ADST is set to 1 by so
by input of the trigger set in step [6].

Figure 19.2 Procedure for Activating the ΔΣ A/D Converter

Setting two or more CHn bits to 1 places the ΔΣ A/D converter in multi-channel mode, w
conversion of the signals on the selected channels proceeds in sequence (from channel 0 t
channel 5).

Values for canceling offsets of the single-ended input signals on channels 0 to 3 can be in
register settings. These values are set in ΔΣ A/D offset cancel DAC inputs 0 to 3 (DSADO
DSADOF3). During A/D conversion on channel n, the value set in the DSADOFn registe
looked up and input to a 10-bit D/A converter that converts it to an analog signal, which p
the level for canceling the offset on the analog input channel.

Table 19.4 shows the correspondence of the analog input channel settings.

Table 19.4 Correspondence between Settings and Analog Input Channels

No.	Analog Input Channel	A/D Conversion Channel Select Bit	Analog Input Pin	Single-Ended/ Differential Input	Offset Cancellation	O E
0	Channel 0	CH0	ANDS0	Single-ended	DSADOF0 register	
1	Channel 1	CH1	ANDS1	Single-ended	DSADOF1 register	
2	Channel 2	CH2	ANDS2	Single-ended	DSADOF2 register	
3	Channel 3	CH3	ANDS3	Single-ended	DSADOF3 register	
4	Channel 4	CH4	ANDS4P	Differential	ANDS4N pin	
5	Channel 5	CH5	ANDS5P	Differential	ANDS5N pin	

When only one channel is selected (normal single mode), A/D conversion is performed
following way.

1. A/D conversion is started for the selected channel when the ADST bit in DSADCSR
   by software or by the input of trigger signal selected by the TRGS1 and TRGS0 bits
   DSADCSR.
2. When A/D conversion is completed, the result is transferred to the ΔΣ A/D data regi
   selected channel (DSADDRn, n = 0 to 5).
3. When the result of A/D conversion is transferred to the data register and conversion
   A/D converter is complete, the ADF bit in DSADCSR is set to 1. If the ADIE bit in
   DSADCSR is set to 1 at this time, a DSADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion and is automatically cleared c
   completion of A/D conversion. When the ADST bit is again set to 1, A/D conversion
   selected channel is started again.
5. If the ADST bit is cleared to 0 during A/D conversion, the conversion is stopped and
   A/D converter enters the idle state.

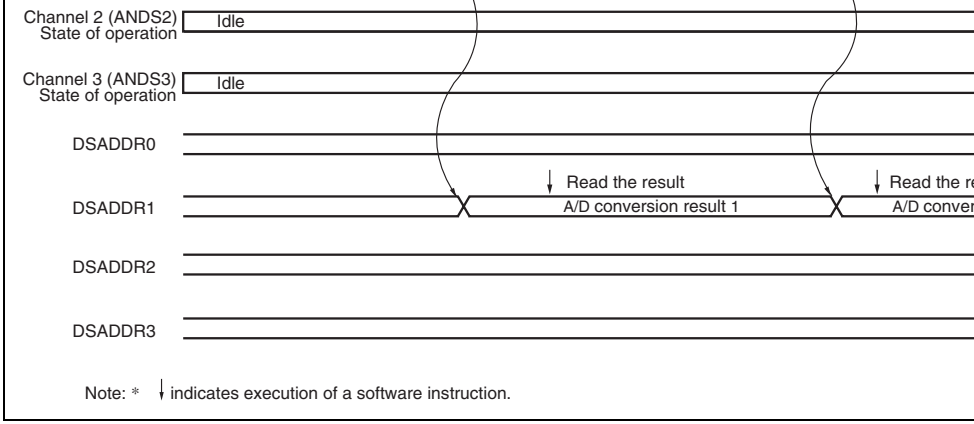


Figure 19.3 Example of ΔΣ A/D Converter Operation (Single Mode for One Cha Channel 1)

Figure 19.4 shows an example of ΔΣ A/D converter operation (in multi-channel single m channels 0 to 2 selected).

When A/D conversion is performed for two or more channels (multi-channel single mode analog input on each of the selected channels is A/D converted once in sequence from ch as described below.

1. A/D conversion is started for the selected channels when the ADST bit in DSADCSR 1 by software or by the input of a trigger signal selected by the TRGS1 and TRGS0 bi DSADCSR. Execution of A/D conversion is in order of rising channel number, so the precedence starts from channel 0.
2. When A/D conversion is completed for channel n, the result is transferred to the corre ΔΣ A/D data register (DSADDRn, n = 0 to 5).

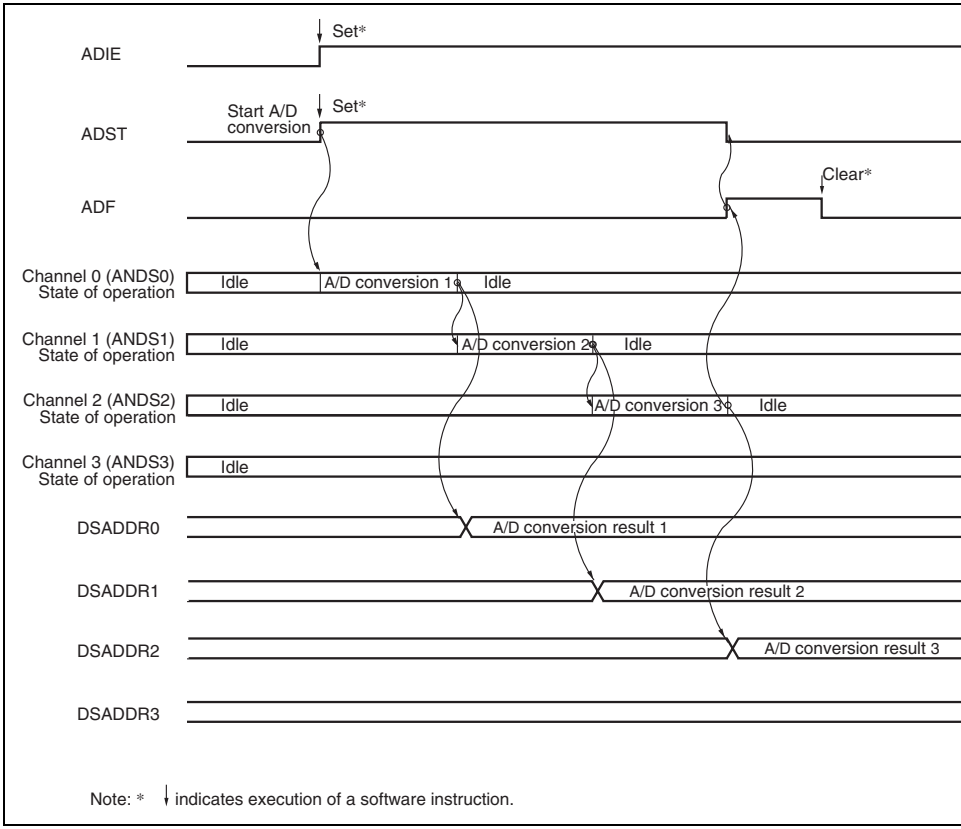
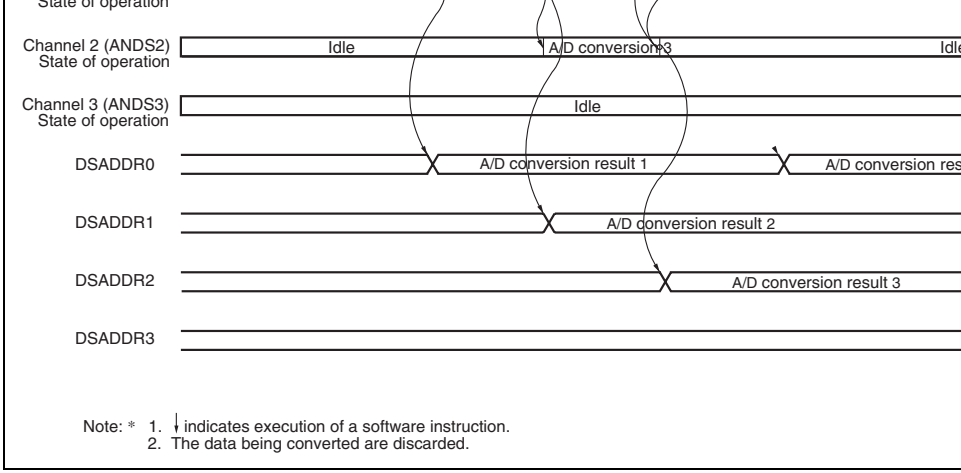


Figure 19.4 Example of ΔΣ A/D Converter Operation (Single Mode for Multiple C Channels 0 to 2)

ΔΣ A/D data register (DSADDRn, n = 0 to 5).

3. When A/D conversion for all of the selected channels has been completed, the ADF b
   DSADCSR is set to 1. If the setting of the ADIE bit in DSADCSR is 1 at this time, a
   interrupt request is also generated.
4. The ΔΣ A/D converter starts another round of A/D conversion in order of precedence
   channel 0. The ADST bit is not cleared automatically, and steps 2 to 4 are repeated as
   ADST = 1.
5. If the ADST bit is cleared to 0 during A/D conversion, the conversion is stopped and
   A/D converter enters the idle state. When the ADST bit is subsequently set to 1, A/D
   conversion for the selected channels again proceeds from channel 0.



Note: * 1. ↓ indicates execution of a software instruction.
2. The data being converted are discarded.

Figure 19.5 Example of ΔΣ A/D Converter Operation (Scan Mode with Channe Selected)

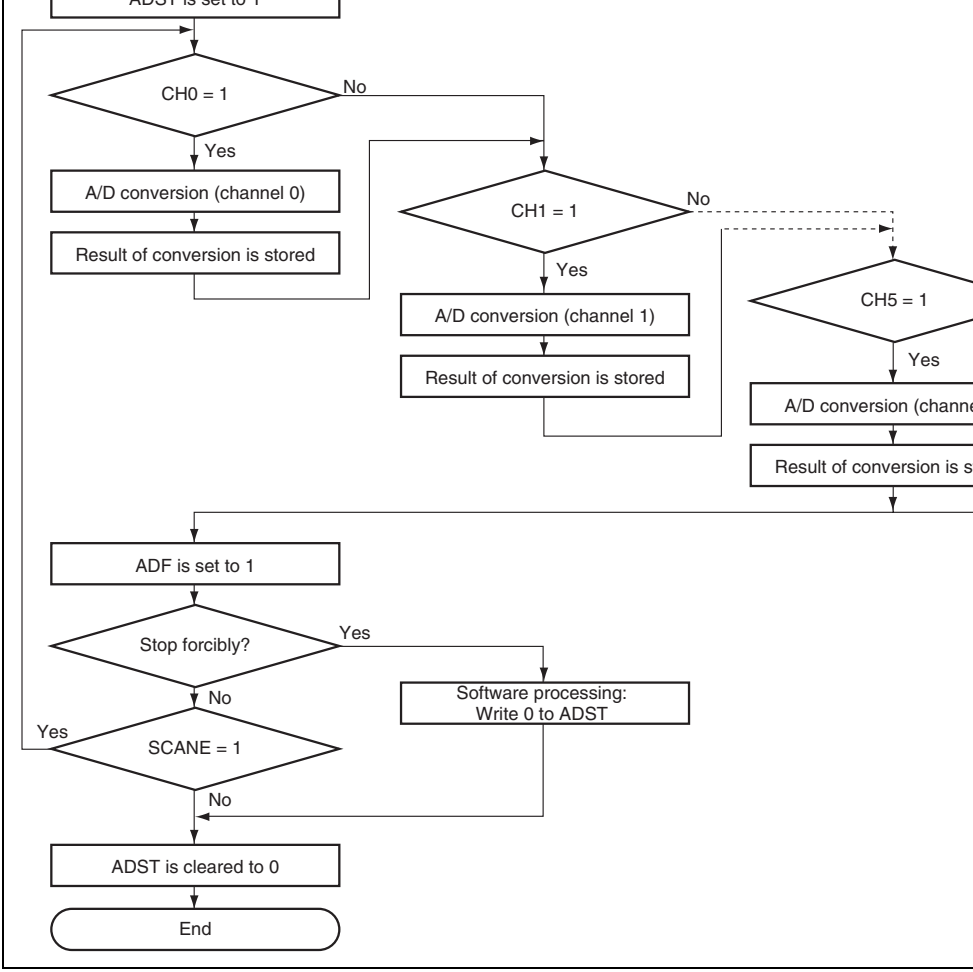


Figure 19.6 Flow of ΔΣ A/D Conversion Operation (Initiated by Software)

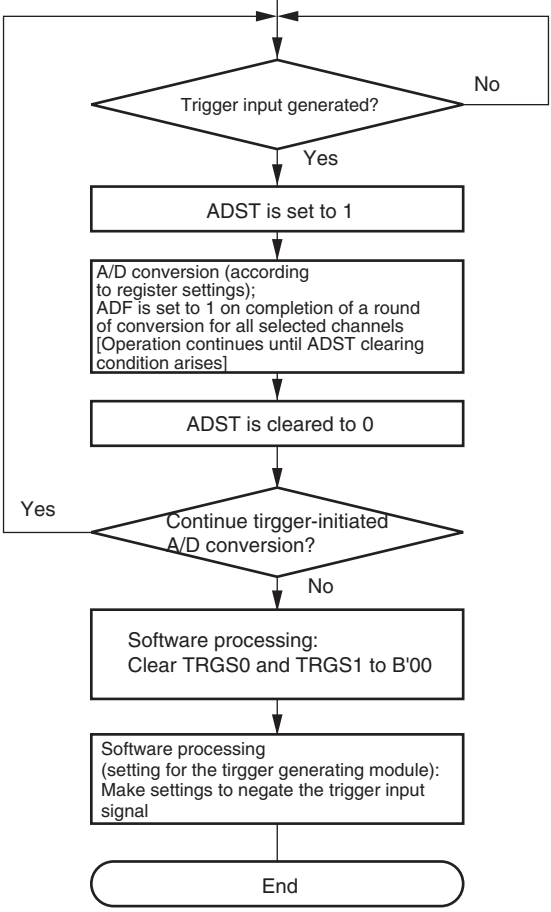


Figure 19.7 Flow of ΔΣ A/D Conversion Operation (Initiated by a Trigger In

vary because they are determined by the timing of synchronization between different cloc
and the state of control of synchronization processing at the end of the previous round of
conversion. For this reason, conversion times vary within the range shown in table 19.5.

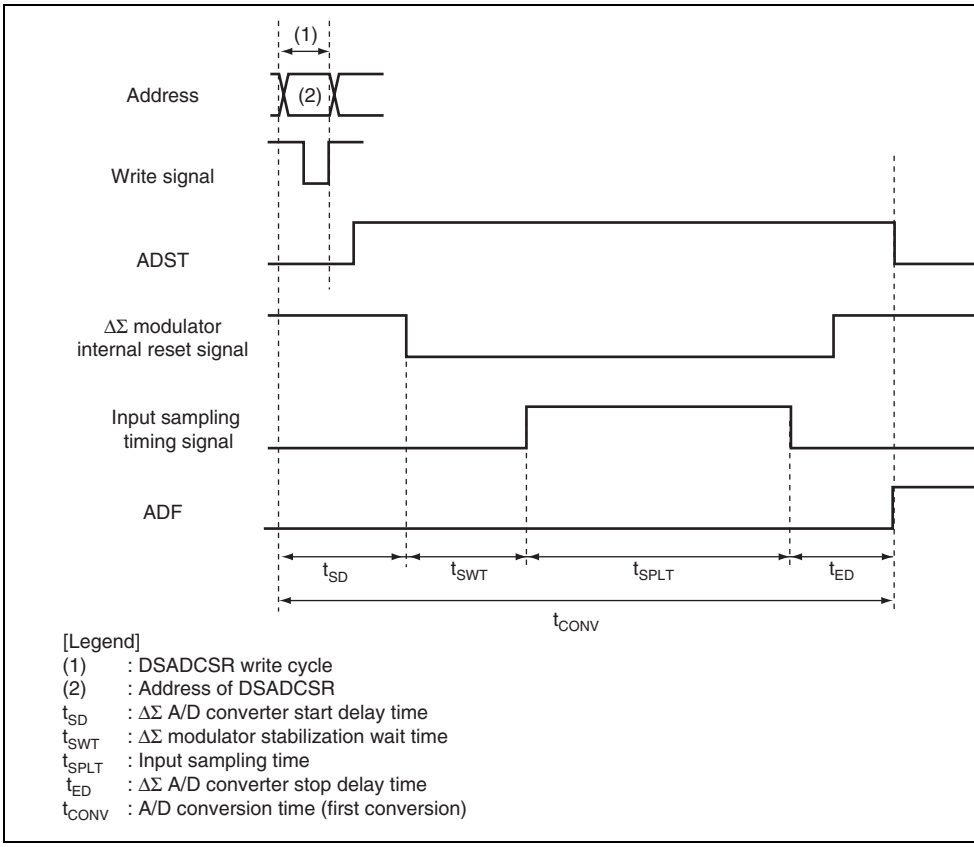
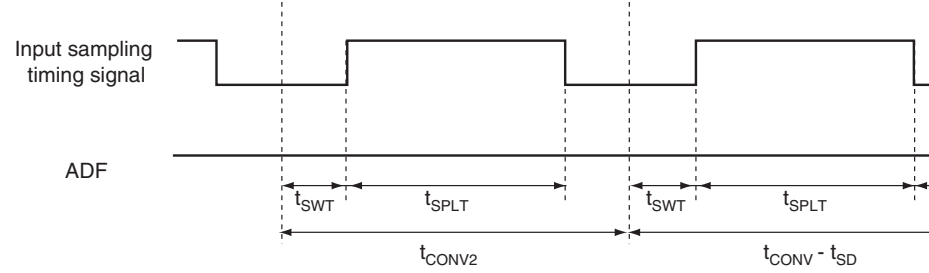


Figure 19.8 A/D Conversion Timing (Single Mode, Once, One Channel)



[Legend]

t_{SWT} : ΔΣ modulator stabilization wait time
t_{SPLT} : Input sampling time
t_{CONV2} : A/D conversion time (continuous conversion)
t_{ED} : ΔΣ A/D converter stop delay time
t_{SD} : ΔΣ A/D converter start delay time
t_{CONV} : A/D conversion time (first conversion)

Figure 19.9 Timing of Second and Subsequent Rounds of A/D Conversion (Suc Conversion)

A/D conversion time (second and subsequent rounds)	t_{CONV2}	286

Note: The unit for values in the table is the period of Aϕ/8 (“state”).

Table 19.6 A/D Conversion Time (Second and Subsequent Rounds)

CKS	Conversion Time (Aϕ/8 Periods)
0	286 (fixed)

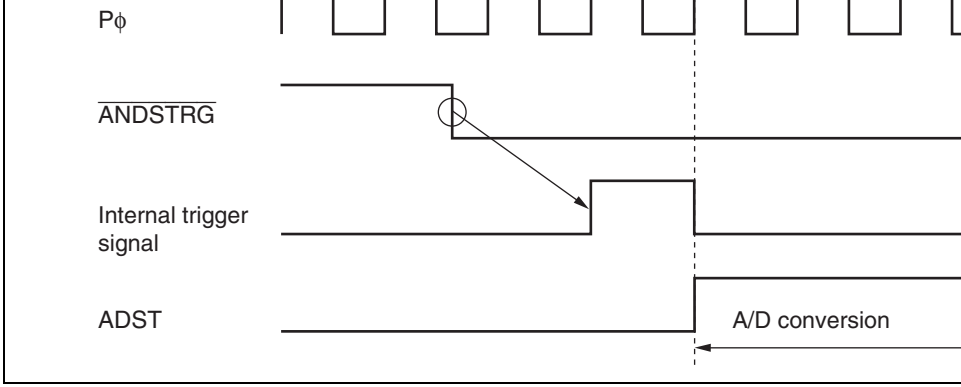


Figure 19.10 Timing of External Trigger Input

DMDR register of the DMA channel to 1 and placing the address of DSADDRn in DSAR clearing of the ADF bit when DMA transfer is executed.

Table 19.7 Interrupt Source of the ΔΣ A/D Converter

Name	Interrupt Source	Interrupt Flag	Activation of DTC	Activat DMAC
DSADI	End of A/D conversion	ADF	No	Yes

performed while the converter is in the module stop state.

To stop the ΔΣ A/D converter completely, place it in the module stop state and then stop biasing circuit by clearing the BIASE bit in DSADMR to 0.

19.6.2 Settings for the Biasing Circuit

When the BIASE bit in DSADMR is set to enable the biasing circuit before the ΔΣ A/D is used, a certain period must be secured for stabilization of the biasing circuit. If A/D co is executed without ensuring enough time for stabilization of the biasing circuit, the prec A/D conversion is not guaranteed.

When the biasing circuit is stopped by clearing the BIASE bit in DSADMR to 0 or on en hardware standby mode, the reset state, or deep software standby mode, a certain period stabilization of the biasing circuit will be required after the BIASE bit has been set to 1 a

A certain amount of biasing current flows while the biasing circuit is running. Since the in the BIASE bit is retained in software standby mode, the supply current will include th that flows through the biasing circuit if BIASE = 1. Be sure to set the BIASE bit approp before initiating software standby mode.

Ensure at least 20 ms for stabilization of the biasing circuit.

To avoid malfunctions during A/D conversion, do not change the settings of the ΔΣ A/D
registers while the ADST bit in DSADCSR is set to 1. Always write to the registers with
ADST bit cleared to 0. The exceptions are clearing of the ADST bit and clearing of the A
after reading a 1 from it.

When the TRGS1 and TRGS0 bits in DSADCSR are set to a value other than B'00, the A
may be set automatically by the trigger signal. Accordingly, before setting registers of the
converter, set the TRGS1 and TRGS0 bits to B'00 or take measures to ensure that no trigg
will be input.

19.6.5 DSE Bit

Use the ΔΣ A/D converter with the DSE bit in DSADCR set to 1.

- Module stop state specifiable

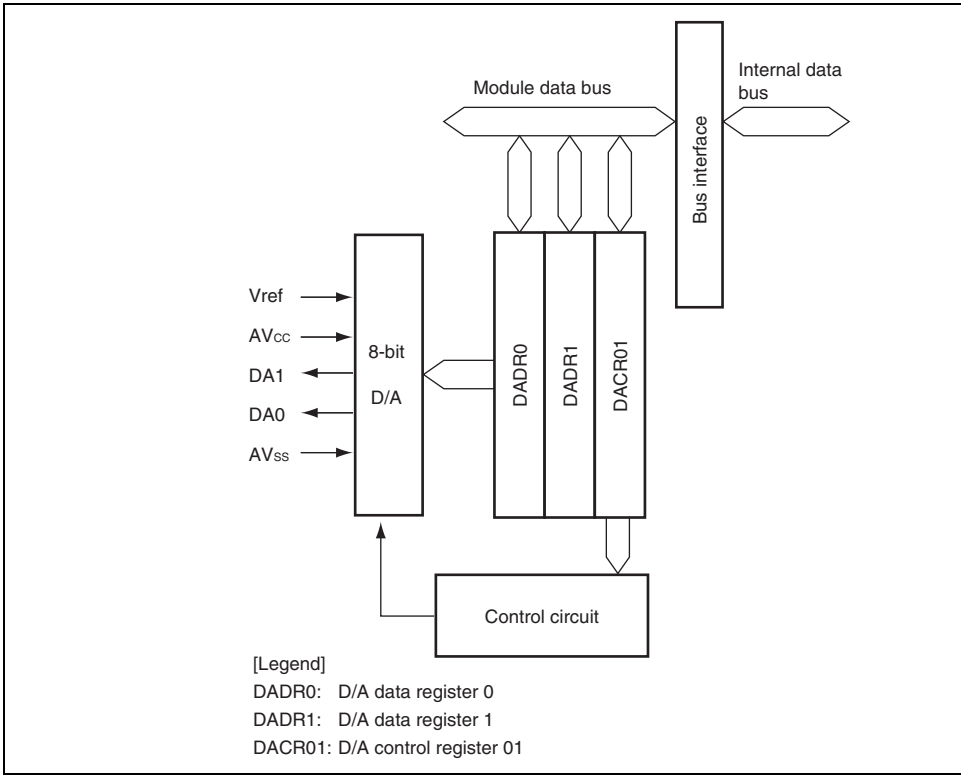


Figure 20.1 Block Diagram of D/A Converter

Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

20.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register 01 (DACR01)

20.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

DADR0 and DADR1 are 8-bit readable/writable registers that store data to which D/A co
is to be performed. Whenever an analog output is enabled, the values in DADR are conve
output to the analog output pins.

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

Bit	Bit Name	Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output. 0: Analog output of channel 1 (DA1) is disabled 1: D/A conversion of channel 1 is enabled. Ana of channel 1 (DA1) is enabled.
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output. 0: Analog output of channel 0 (DA0) is disabled 1: D/A conversion of channel 0 is enabled. Ana of channel 0 (DA0) is enabled.
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits D/A conversion. When this bit is cleared to 0, D conversion is controlled independently for chan 1. When this bit is set to 1, D/A conversion for c and 1 is controlled together. Output of conversion results is always controlle DAOE0 and DAOE1 bits. For details, see table Control of D/A Conversion.
4 to 0	—	All 1	R	Reserved These are read-only bits and cannot be modifie

			Analog output of channel 0 (DA0) is disabled and an output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is enabled.
1	0	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is disabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is enabled and an output of channel 1 (DA1) is disabled.
	1	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is disabled and an output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is enabled.

from the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The con
result continues to be output until DADR0 is written to again or the DAOE0 bit is cl
The output value is expressed by the following formula:

$$\text{Contents of DADR}/256 \times V_{ref}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion
output after the conversion time t_{DCONV} has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

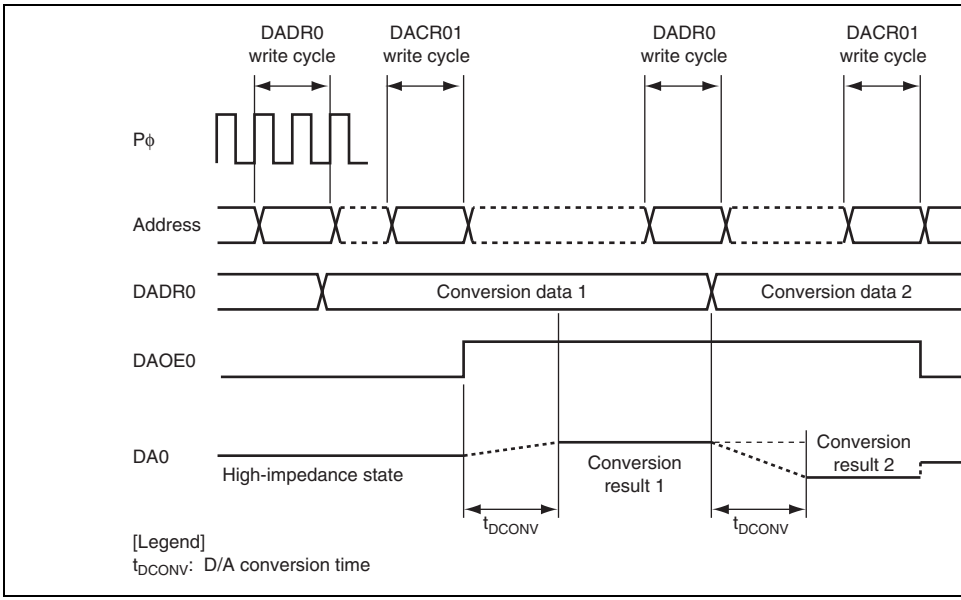


Figure 20.2 Example of D/A Converter Operation

When this LSI enters software standby mode with D/A conversion enabled, the D/A outp
retained, and the analog power supply current is equal to as during D/A conversion. If the
power supply current needs to be reduced in software standby mode, clear the DAOE0, D
and DAE bits all to 0 to disable the D/A outputs.

- 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: 256 Kbytes

  — User boot MAT is initiated at a 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

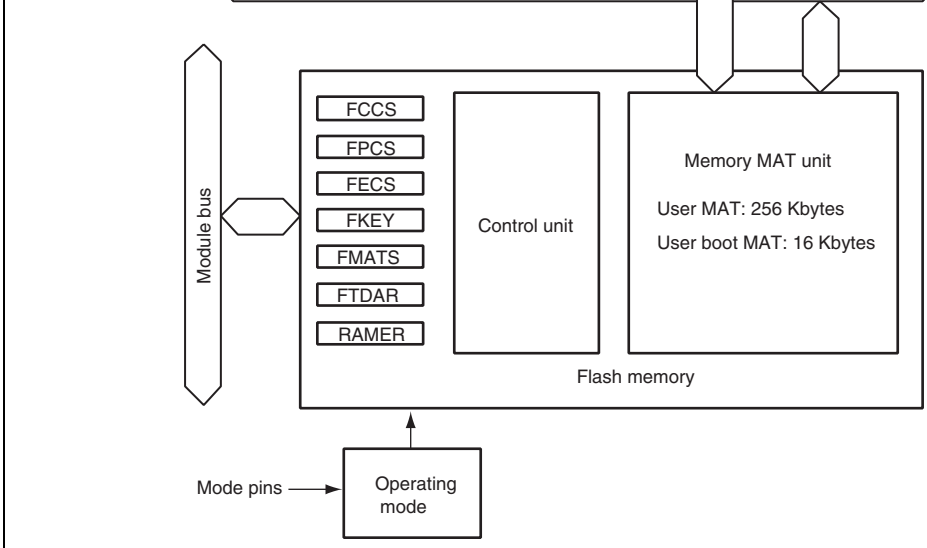
  The number of programming can be up to 100 times at the minimum. (1 to 100 time
  guaranteed.)
- Three on-board programming modes

  Boot mode: Using the on-chip SCI_4, the user MAT and user boot MAT can be
  programmed/erased. In boot mode, the bit rate between the host and this LSI can be
  automatically.

  User program mode: Using a desired interface, the user MAT can be programmed/er

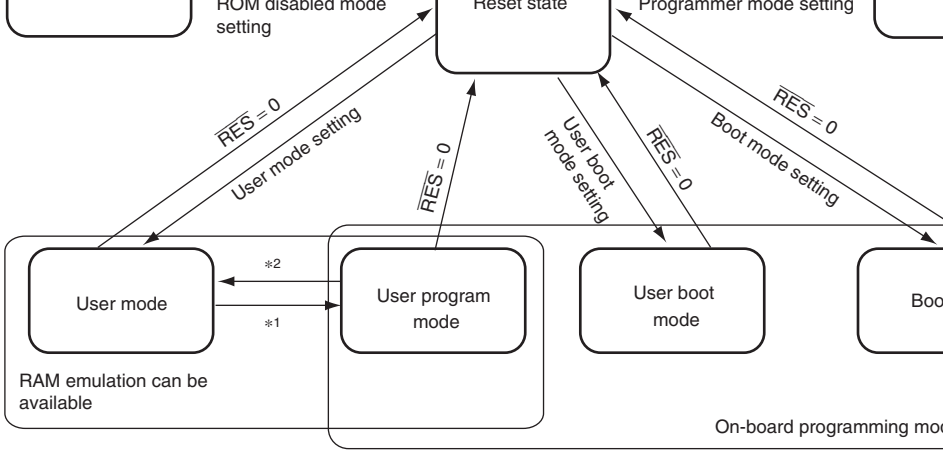
  User boot mode: Using a desired interface, the user boot program can be made and t
  MAT can be programmed/erased.
- Off-board programming mode

  Programmer mode: Using a PROM programmer, the user MAT and user boot MAT
  programmed/erased.



[Legend]
FCCS: Flash code control/status register
FPCS: Flash program code select register
FECS: Flash erase code select register
FKEY: Flash key code register
FMATS: Flash MAT select register
FTDAR: Flash transfer destination address register
RAMER: RAM emulation register

Figure 22.1 Block Diagram of Flash Memory



Notes: * In this LSI, the user program mode is defined as the period from the timing when a prog concerning programming and erasure is started in user mode to the timing when the pro completed.

1. Programming and erasure is started.
2. Programming and erasure is completed.

Figure 22.2 Mode Transition of Flash Memory

Programming/ erasing control	Command	Programming/ erasing interface	Programming/ erasing interface	Command
All erasure	O (Automatic)	O	O	O (Automa
Block division erasure	O*[1]	O	O	×
Program data transfer	From host via SCI	From desired device via RAM	From desired device via RAM	Via progra
RAM emulation	×	O	O	×
Reset initiation MAT	Embedded program storage area	User MAT	User boot MAT*[2]	—
Transition to user mode	Changing mode and reset	Completing Programming/ erasure*[3]	Changing mode and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

2. First, the reset vector is fetched from the embedded program storage area. Aft flash memory related registers are checked, the reset vector is fetched from th boot MAT.

3. In this LSI, the user programming mode is defined as the period from the timin program concerning programming and erasure is started to the timing when th program is completed. For details on a program concerning programming and see section 22.8.2, User Program Mode.

The size of the user MAT is different from that of the user boot MAT. Addresses which
the size of the 16-Kbyte user boot MAT should not be accessed. If an attempt is made, d
as an undefined value.

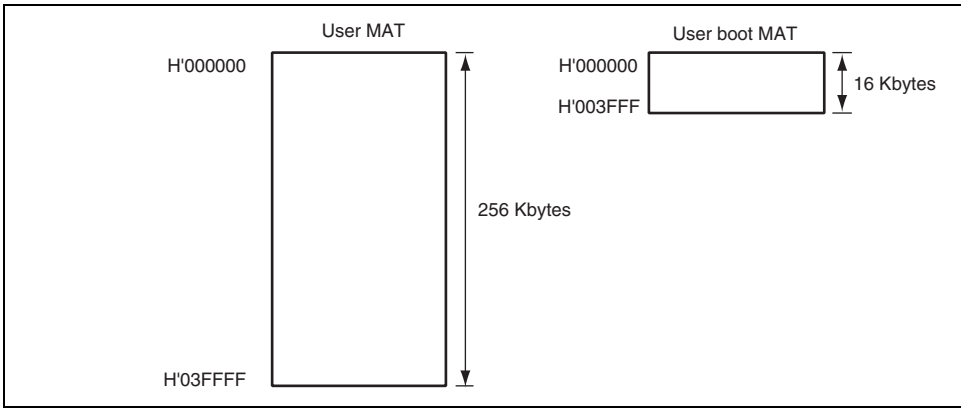


Figure 22.3 Memory MAT Configuration

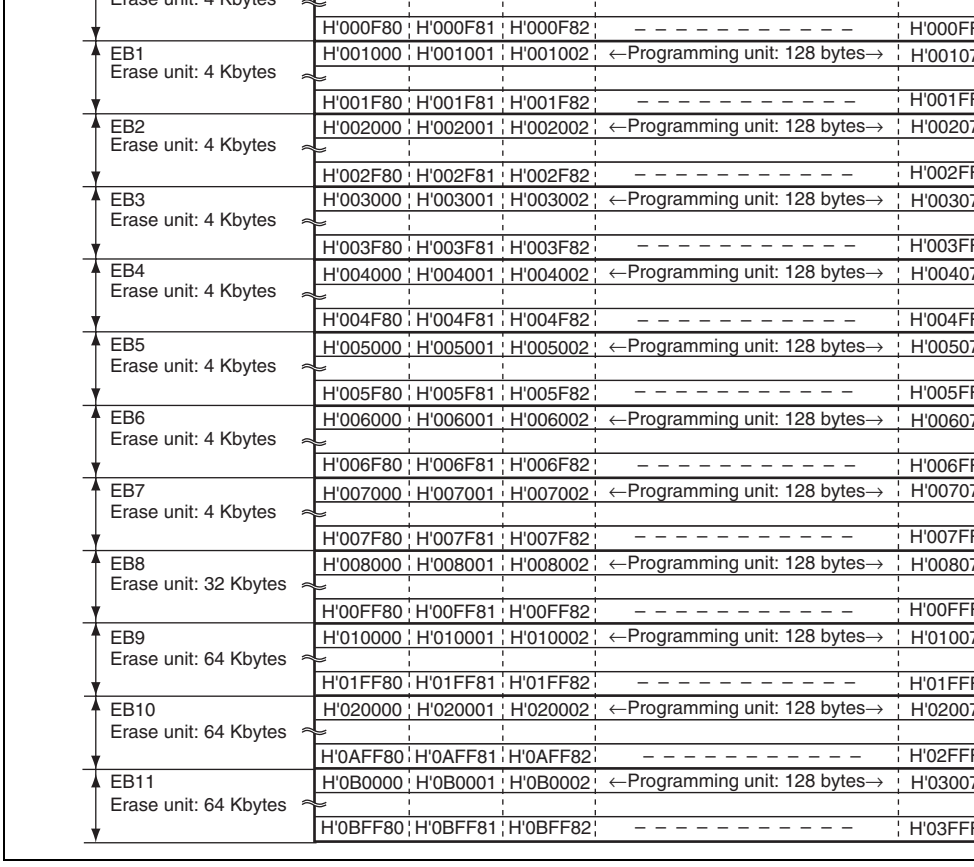


Figure 22.4 User MAT Block Structure

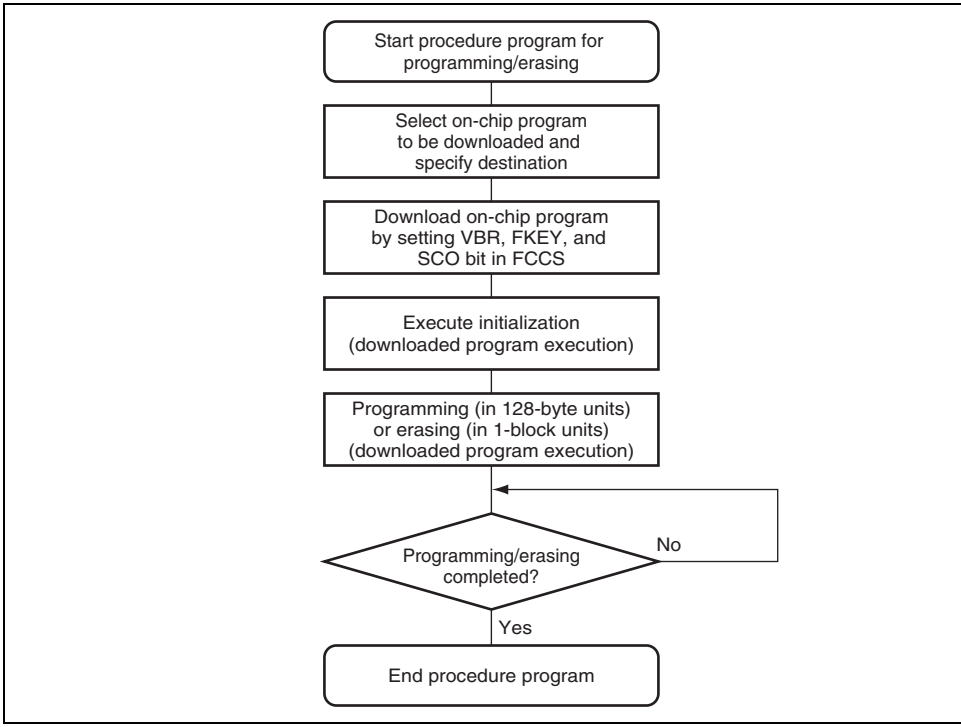


Figure 22.5 Procedure for Creating Procedure Program

(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 12
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 are
required when the same processing is executed consecutively.

TxD4	Output	Serial transmit data output (used in boot mo
RxD4	Input	Serial receive data input (used in boot mode

22.7 Register Descriptions

The flash memory has the following registers.

Programming/Erasing Interface Registers:

- Flash code control/status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)

Programming/Erasing Interface Parameters:

- Download pass and fail result parameter (DPFR)
- Flash pass and fail result parameter (FPFR)
- Flash program/erase frequency parameter (FPEFEQ)
- Flash multipurpose address area parameter (FMPAR)
- Flash multipurpose data destination area parameter (FMPDR)
- Flash erase block select parameter (FEBS)

- RAM emulation register (RAMER)

	FKEY	O	—	O	O	—	—
	FMATS	—	—	O*1	O*1	O*2	—
	FTDAR	O	—	—	—	—	—
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 bo
2. The setting may be required according to the combination of initiation mode ar target memory MAT.

22.7.1 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 on program to be downloaded to the on-chip RAM.

4	FLER	0	R	Flash Memory Error Indicates that an error has occurred during pro or erasing the flash memory. When this bit is s the flash memory enters the error protection s When this bit is set to 1, high voltage is applie internal flash memory. To reduce the damage flash memory, the reset must be released afte input period (period of $\overline{RES}$ = 0) of at least 10 0: Flash memory operates normally (Error pro invalid) [Clearing condition] • At a reset 1: An error occurs during programming/erasin 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 r an instruction fetch). • When the SLEEP instruction is executed d programming/erasure (including software s mode). • When a bus master other than the CPU, s DMAC and DTC, obtains bus mastership d programming/erasure.

must be canceled, H'A5 must be written to FKE
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 interru
be disabled during download. This bit is cleared
when download is completed.

During program download initiated with this bit,
particular processing which accompanies bank
switching of the program storage area is execu
Before a download request, initialize the VBR c
to H'00000000. After download is completed, th
contents can be changed.

0: Download of the programming/erasing progr
requested.

[Clearing condition]

- When download is completed

1: Download of the programming/erasing progr
requested.

[Setting conditions] (When all of the following c
are satisfied)

- Not in RAM emulation mode (the RAMS bit
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.

7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifi
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program to be down 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	—	—	—	—	—	—	—	
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 1	—	All 0	R	Reserved These are read-only bits and cannot be modifi
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program to be downloade 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 SC
5	K5	0	R/W	FCCS is enabled. When a value other than H'A written, the SCO bit cannot be set to 1. Therefo
4	K4	0	R/W	on-chip program cannot be downloaded to the
3	K3	0	R/W	RAM.
2	K2	0	R/W	Only when H'5A is written can programming/era
1	K1	0	R/W	the flash memory be executed. When a value o
0	K0	0	R/W	H'5A is written, even if the programming/erasin program is executed, programming/erasure car performed.
				H'A5: Writing to the SCO bit is enabled. (The S cannot be set to 1 when FKEY is a value than H'A5.)
				H'5A: Programming/erasure of the flash memor enabled. (When FKEY is a value other th 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	
4	MS4	0	R/W	When H'AA is written to FMATS, the user boo selected. When a value other than H'AA is wri user MAT is selected. Switch the MATs follow memory MAT switching procedure in section 2 Switching between User MAT and User Boot M user boot MAT cannot be selected by FMATS programming mode. The user boot MAT can b selected in boot mode or programmer mode.
3	MS3	0/1*	R/W	
2	MS2	0	R/W	
1	MS1	0/1*	R/W	
0	MS0	0	R/W	
				H'AA: The user boot MAT is selected. (The us selected when FMATS is a value other t H'AA.) (Initial value when initiated in user boot
				H'00: The user MAT is selected. (Initial value when initiated in a mode ex user boot mode.)

Note: * This bit is set to 1 in user boot mode, otherwise cleared to 0.

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 i the start address specified by bits TDA6 to TDA A start address error is determined by whether set in bits TDA6 to TDA0 is within the range of H'02 when download is executed by setting the in FCCS to 1. Make sure that this bit is cleared before setting the SCO bit to 1 and the value sp by bits TDA6 to TDA0 should be within the rang H'00 to H'02. 0: The value specified by bits TDA6 to TDA0 is the range. 1: The value H'03 to H'FF, specified by bits TDA TDA6 to TDA0, stops download.
6	TDA6	0	R/W	Transfer Destination Address Specifies the on-chip RAM start address of the download destination. By the value H'00 to H'02 H'20, up to 4 Kbytes can be specified as the sta address of the on-chip RAM. H'00: H'FF9000 is specified as the start addres H'01: H'FFA000 is specified as the start addres H'02: H'FFB000 is specified as the start addres H'03 to H'7F: Setting prohibited. (Specifying a value from H'03 to H the TDER bit to 1 and stops down the on-chip program.)
5	TDA5	0	R/W	
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	0	R/W	
0	TDA0	0	R/W	

processing result is written in R0L. The programming/erasing interface parameters are u
download control, initialization before programming or erasing, programming, and erasi
22.4 shows the usable parameters and target modes. The meaning of the bits in the flash
fail result parameter (FPFR) varies in initialization, programming, and erasure.

Table 22.4 Parameters and Target Modes

Parameter	Download	Initialization	Programming	Erasure	R/W	Initial Value	All
DPFR	O	—	—	—	R/W	Undefined	On
FPFR	O	O	O	O	R/W	Undefined	R0
FPEFEQ	—	O	—	—	R/W	Undefined	ER
FMPAR	—	—	O	—	R/W	Undefined	ER
FMPDR	—	—	O	—	R/W	Undefined	ER
FEBS	—	—	—	O	R/W	Undefined	ER

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 S
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/erasin
registers, and the download pass and fail result parameter (DPFR) indicates the return va

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 12
128-byte program data is prepared by filling the dummy code (H'FF). The boundary of th
address of the programming destination on the user MAT is aligned at an address where t
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 se
general register ER0. This parameter is called the flash multipurpose data destination area
parameter (FMPDR).

For details on the programming procedure, see section 22.8.2, User Program Mode.

Erasure: When the flash memory is erased, the erase block number on the user MAT mu
passed to the erasing program which is downloaded.

The erase block number on the user MAT must be set in general register ER0. This paran
called the flash erase block select parameter (FEBS).

One block is selected from the block numbers of 0 to 11 as the erase block number.

For details on the erasing procedure, see section 22.8.2, User Program Mode.

7 to 3	—	—	—	Unused These bits return 0.
2	SS	—	R/W	Source Select Error Detect Only one type can be specified for the on-chip which can be downloaded. When the program downloaded is not selected, more than two typ programs are selected, or a program which is 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 th 0: FKEY setting is normal (H'A5) 1: FKEY setting is abnormal (value other than
0	SF	—	R/W	Success/Fail Returns the download result. Reads back the downloaded to the on-chip RAM and determin whether it has been transferred to the on-chip 0: Download of the program has ended norma error) 1: Download of the program has ended abnor (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 frequer 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 occ

6	MD	—	R/W	Programming Mode Related Setting Error Det Detects the error protection state and returns When the error protection state is entered, thi to 1. Whether the error protection state is ente can be confirmed with the FLER bit in FCCS. conditions to enter the error protection state, s 22.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming ca performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect Writes 1 to this bit when the specified data cou written because the user MAT was not erased is set to 1, there is a high possibility that the u has been written to partially. In this case, after the error factor, erase the user MAT. If FMATS H'AA and the user boot MAT is selected, an e when programming is performed. In this case, user MAT and user boot MAT have not been v Programming the user boot MAT should be pe boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally (progr result is not guaranteed)

				When an address not in the flash memory area specified as the start address of the storage de for the program data, an error occurs. 0: Setting of the start address of the storage de for the program data is normal 1: Setting of the start address of the storage de for the program data is abnormal
1	WA	—	R/W	Write Address Error Detect When the following items are specified as the s address of the programming destination, an err occurs. • An area other than flash memory • The specified address is not aligned with th byte boundary (lower eight bits of the addre other than H'00 and H'80) 0: Setting of the start address of the programmi destination is normal 1: Setting of the start address of the programmi 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 o

6	MD	—	R/W	Erasure Mode Related Setting Error Detect Detects the error protection state and returns When the error protection state is entered, thi to 1. Whether the error protection state is ente can be confirmed with the FLER bit in FCCS. conditions to enter the error protection state, s 22.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming ca performed (FLER = 1)
5	EE	—	R/W	Erasure Execution Error Detect Returns 1 when the user MAT could not be er when the flash memory related register setting partially changed. If this bit is set to 1, there is possibility that the user MAT has been erased In this case, after removing the error factor, er user MAT. If FMATS is set to H'AA and the us MAT is selected, an error occurs when erasur performed. In this case, both the user MAT an boot MAT have not been erased. Erasing of th boot MAT should be performed in boot mode 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 occurs)

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	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused These bits should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set These bits set the operating frequency of the When the PLL multiplication function is used, multiplied frequency. The setting value must b calculated as follows: 1. The operating frequency shown in MHz uni rounded in a number of three decimal place shown in a number of two decimal places. 2. The value multiplied by 100 is converted to digit and is written to FPEFEQ (general reg For example, when the operating frequency o 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 binary digit and B'0000 1101 1010 1100 (H' set to ER0.

Bit	23	22	21	20	19	18	17	
Bit Name	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MO

Bit	15	14	13	12	11	10	9	
Bit Name	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	M

Bit	7	6	5	4	3	2	1	
Bit Name	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	M

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	These bits store the start address of the progra destination on the user MAT. Consecutive 128- programming is executed starting from the spec start address of the user MAT. Therefore, the s start address of the programming destination b 128-byte boundary, and MOA6 to MOA0 are al cleared to 0.

Bit	23	22	21	20	19	18	17	
Bit Name	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	M

Bit	15	14	13	12	11	10	9	
Bit Name	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	

Bit	7	6	5	4	3	2	1	
Bit Name	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	

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 stores the program data for the user MAT. Co 128-byte data is programmed to the user MAT from the specified start address.

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	R

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	R

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 modifi
3	RAMS	0	R/W	RAM Select Selects the function which emulates the flash using the on-chip RAM. 0: Disables RAM emulation function 1: Enables RAM emulation function (all blocks user MAT are protected against programmi erasing)
2	RAM2	0	R/W	Flash Memory Area Select
1	RAM1	0	R/W	These bits select the user MAT area overlaid on-chip RAM when RAMS = 1. The following a correspond to the 4-Kbyte erase blocks.
0	RAM0	0	R/W	000: H'000000 to H'000FFF (EB0) 001: H'001000 to H'001FFF (EB1) 010: H'002000 to H'002FFF (EB2) 011: H'003000 to H'003FFF (EB3) 100: H'004000 to H'004FFF (EB4) 101: H'005000 to H'005FFF (EB5) 110: H'006000 to H'006FFF (EB6) 111: H'007000 to H'007FFF (EB7)

Table 22.5 On-Board Programming Mode Setting

Mode Setting	EMLE	MD2	MD1	MD0
User boot mode	0	0	0	1
Boot mode	0	0	1	0
User program mode	0	1	1	0
	0	1	1	1

22.8.1 Boot Mode

Boot mode executes programming/erasure of the user MAT or user boot MAT by means
control command and program data transmitted from the externally connected host via th
SCI_4.

In boot mode, the tool for transmitting the control command and program data, and the pr
data must be prepared in the host. The serial communication mode is set to asynchronous
The system configuration in boot mode is shown in figure 22.6. Interrupts are ignored in
mode. Configure the user system so that interrupts do not occur.

(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 boot mode is made, the boot program embedded in this LSI is initia

When the boot program is initiated, this LSI measures the low period of asynchronous s
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 t
adjustment end sign. When the host receives this bit adjustment end sign normally, it tra
byte of H'55 to this LSI. When reception is not executed normally, initiate boot mode ag
bit rate may not be adjusted within the allowable range depending on the combination o
rate of the host and the system clock frequency of this LSI. Therefore, the transfer bit ra
host and the system clock frequency of this LSI must be as shown in table 22.6.

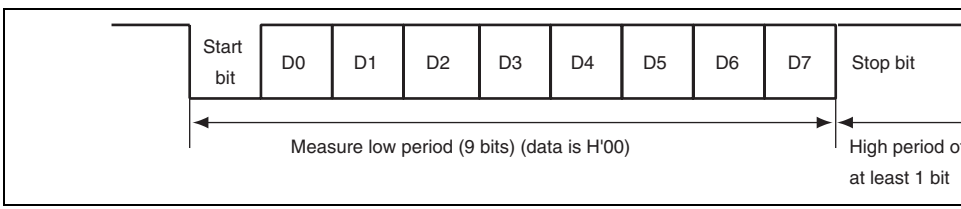


Figure 22.7 Automatic-Bit-Rate Adjustment Operation

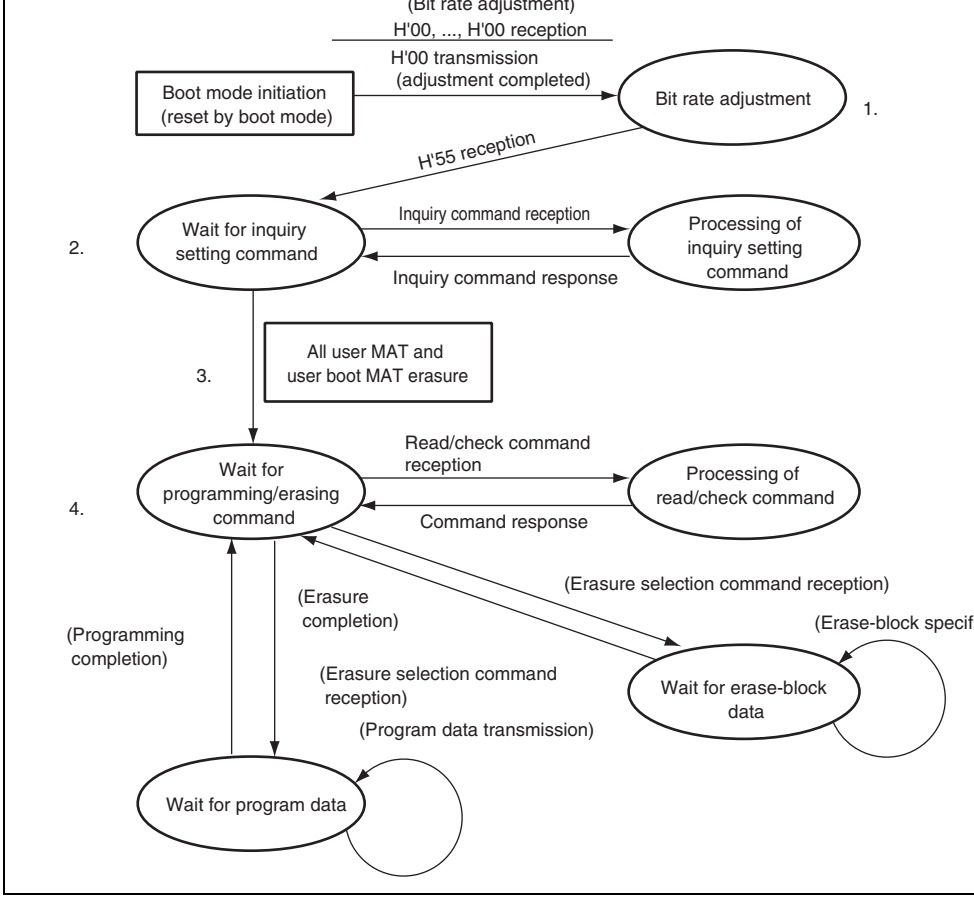
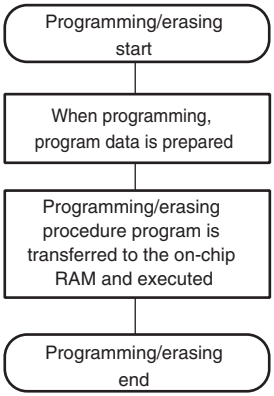


Figure 22.8 Boot Mode State Transition Diagram

waiting for erase block data is entered. The erase block number must be transmitted
erasing command is transmitted. When the erasure is finished, the erase block numb
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
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 era
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), an
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
MAT/user boot MAT has automatically been erased. No other data can be read.



1. Exit RAM emulation mode beforehand. Download is not allowed 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 memory by the hardware protection.

Figure 22.9 Programming/Erasing Flow

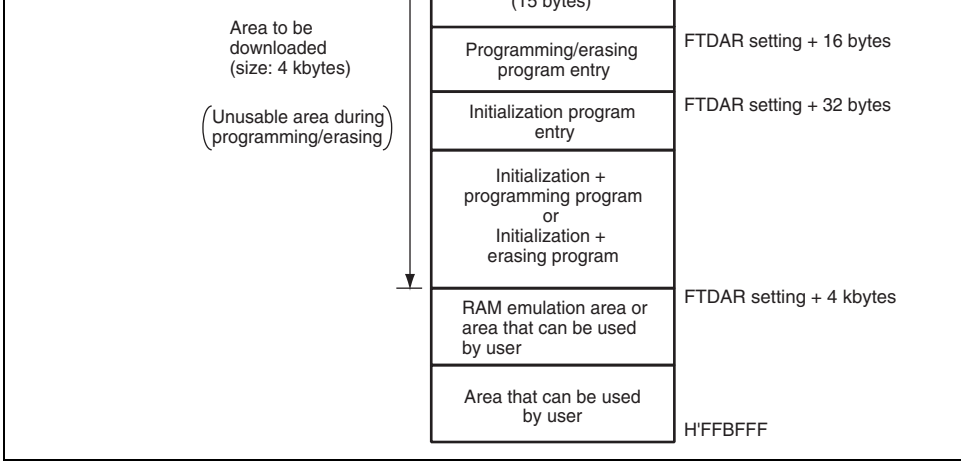


Figure 22.10 RAM Map when Programming/Erasure is Executed

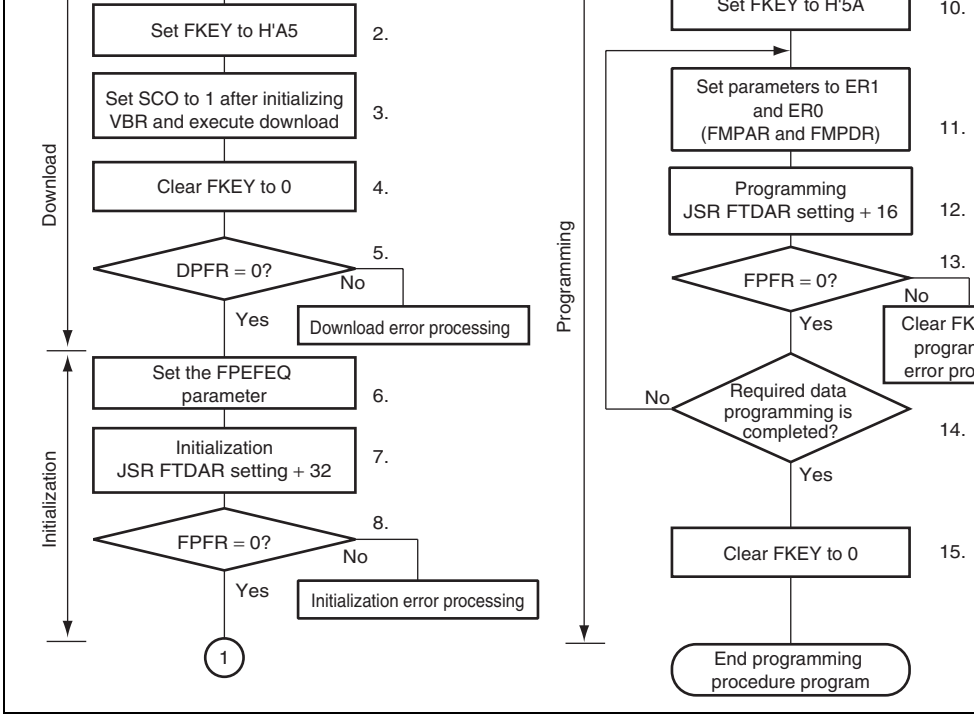


Figure 22.11 Programming Procedure in User Program Mode

H'FF, the program processing time can be shortened.

1. Select the on-chip program to be downloaded and the download destination. When t
   bit in FPCS 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.
2. Write H'A5 in FKEY. If H'A5 is not written to FKEY, the SCO bit in FCCS cannot b
   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
   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
   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 byt
   on-chip RAM start address specified by FTDAR, which becomes the DPFR paramet
   value other than the return value (e.g. H'FF). Since particular processing that is acco
   by bank switching as described below is performed when download is executed, init
   VBR contents to H'00000000. Dummy read of FCCS must be performed twice imm
   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 specifie
     FTDAR are checked, they are transferred to the on-chip RAM.
   — FPCS, FECS, and the SCO bit in FCCS are cleared to 0.

— If access to the flash memory is requested by the DMAC or DTC during downloa operation cannot be guaranteed. Make sure that an access request by the DMAC o 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 t of the DPFR parameter (one byte of start address of the download destination specifie 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 by description below.
   — If the value of the DPFR parameter is the same as that before downloading, the se the start address of the download destination in FTDAR may be abnormal. In this confirm the setting of the TDER bit in FTDAR.
   — If the value of the DPFR parameter is different from that before downloading, che bit or FK bit in the DPFR parameter to confirm the download program selection a setting, respectively.
6. The operating frequency of the CPU is set in the FPEFEQ parameter for initialization settable operating frequency of the FPEFEQ parameter ranges from 8 to 50 MHz. Wh frequency is set otherwise, an error is returned to the FPFR parameter of the initializa program and initialization is not performed. For details on setting the frequency, see s 22.7.2 (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 byte maximum must be allocated in RAM.

— Interrupts can be accepted during execution of the initialization program. Make s program storage area and stack area in the on-chip RAM and register values are n overwritten.

8. The return value in the initialization program, the FPFR parameter is determined.
9. All interrupts and the use of a bus master other than the CPU are disabled during programming/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 t CPU during programming/erasure, causing a voltage exceeding the specifications to applied, the flash memory may be damaged. Therefore, interrupts are disabled by set (I bit) in the condition code register (CCR) to B'1 in interrupt control mode 0 and by bits 2 to 0 (I2 to I0 bits) in the extend register (EXR) to B'111 in interrupt control mo Accordingly, interrupts other than NMI are held and not executed. Configure the use so that NMI interrupts do not occur. The interrupts that are held must be executed af programming completes. When the bus mastership is moved to other than the CPU, the DMAC or DTC, the error protection state is entered. Therefore, make sure the D not acquire the bus.
10. FKEY must be set to H'5A and the user MAT must be prepared for programming.

executed and an error is returned to the FPFR parameter. In this case, the program must be transferred to the on-chip RAM and then programming must be executed.

12. Programming is executed. The entry point of the programming program is at the addr is 16 bytes after #DLTOP (start address of the download destination specified by FTD Call the subroutine to execute programming by using the following steps.

```
MOV.L    #DLTOP+16,ER2   ; Set entry address to ER2
JSR      @ER2            ; Call programming routine
NOP
```

— The general registers other than ER0 and ER1 are held in the programming progra
— R0L is a return value of the FPFR parameter.
— Since the stack area is used in the programming program, a stack area of 128 byte 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 128 data are to be programmed, update the FMPAR and FMPDR parameters in 128-byte u repeat steps 11 to 14. Increment the programming destination address by 128 bytes an the programming data pointer correctly. If an address which has already been progran written to again, not only will a programming error occur, but also flash memory will 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 inpu (period of $\overline{RES} = 0$) of at least 100 µs.

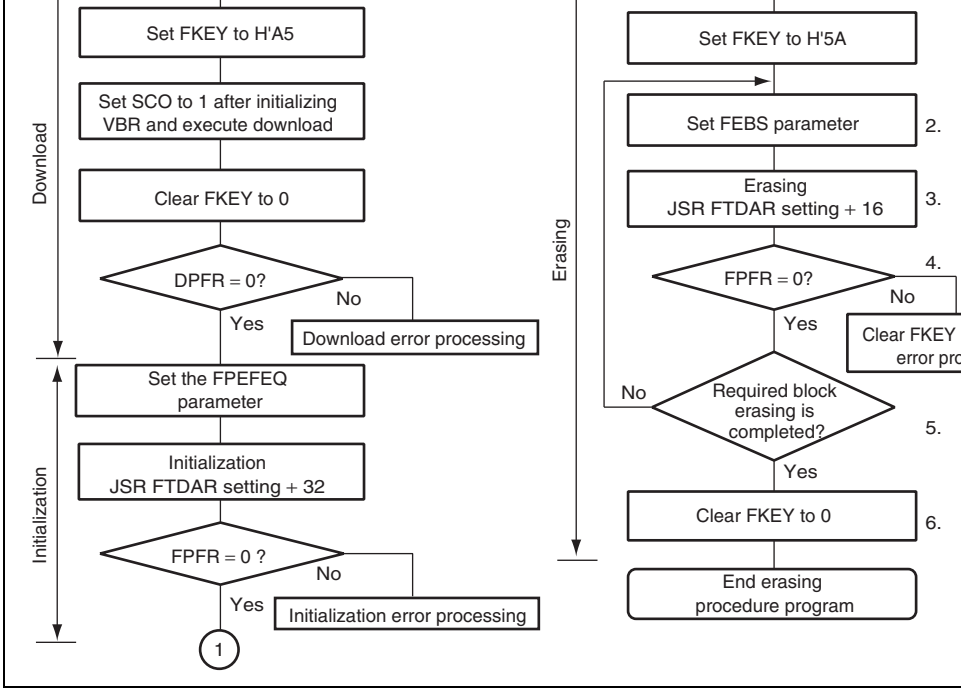


Figure 22.12 Erasing Procedure in User Program Mode

bit in FPCS is set to 1, the programming program is selected. Several programming/er
programs cannot be selected at one time. If several programs are selected, a download
returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the d
destination is specified by FTDAR.

For the procedures to be carried out after setting FKEY, see section 22.8.2 (2), Progra
Procedure in User Program Mode.

2. Set the FEBS parameter necessary for erasure. Set the erase block number (FEBS para
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 progr
the address which is 16 bytes after #DLTOP (start address of the download destinatio
specified by FTDAR). Call the subroutine to execute erasure by using the following s

```
MOV.L #DLTOP+16, ER2     ; Set entry address to ER2
JSR   @ER2               ; Call erasing routine
NOP
```

   - The general registers other than ER0 and ER1 are held in the erasing program.
   - R0L is a return value of the FPFR parameter.
   - Since the stack area is used in the erasing program, a stack area of 128 bytes at the
     maximum must be allocated in RAM.
4. The return value in the erasing program, the FPFR parameter is determined.
5. Determine whether erasure of the necessary blocks has finished. If more than one bloc
be erased, update the FEBS parameter and repeat steps 2 to 5.
6. After erasure completes, clear FKEY and specify software protection. If this LSI is re
a reset immediately after erasure has finished, secure the reset input period (period of
of at least 100 μs.

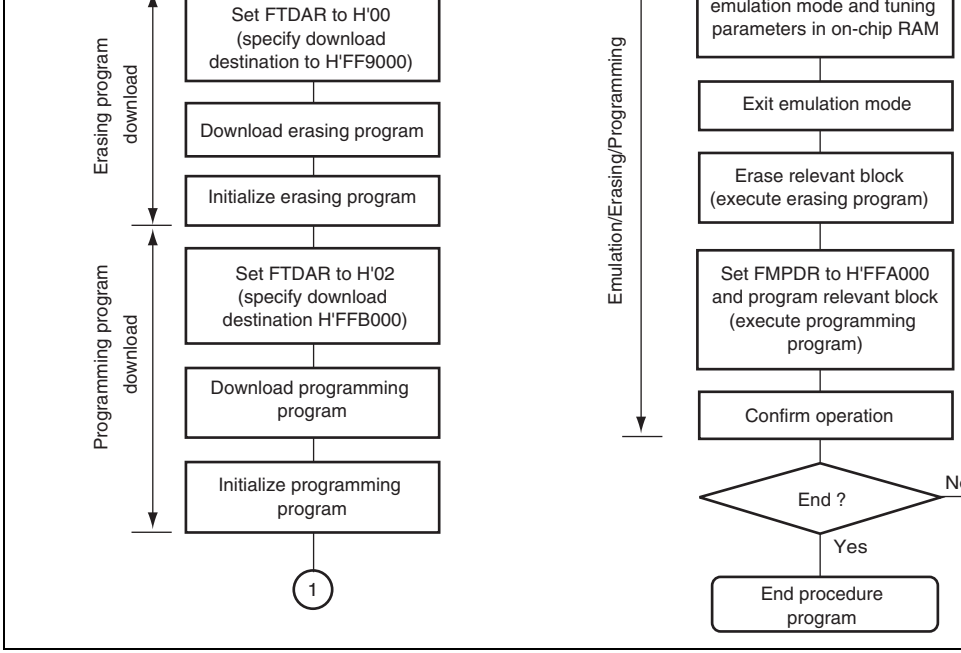


Figure 22.13 Repeating Procedure of Erasing, Programming, and RAM Emulation in User Program Mode

Initialization must be executed for both entry addresses: #DLTOP (start address of do
destination for erasing program) + 32 bytes, and #DLTOP (start address of download
destination for programming program) + 32 bytes.

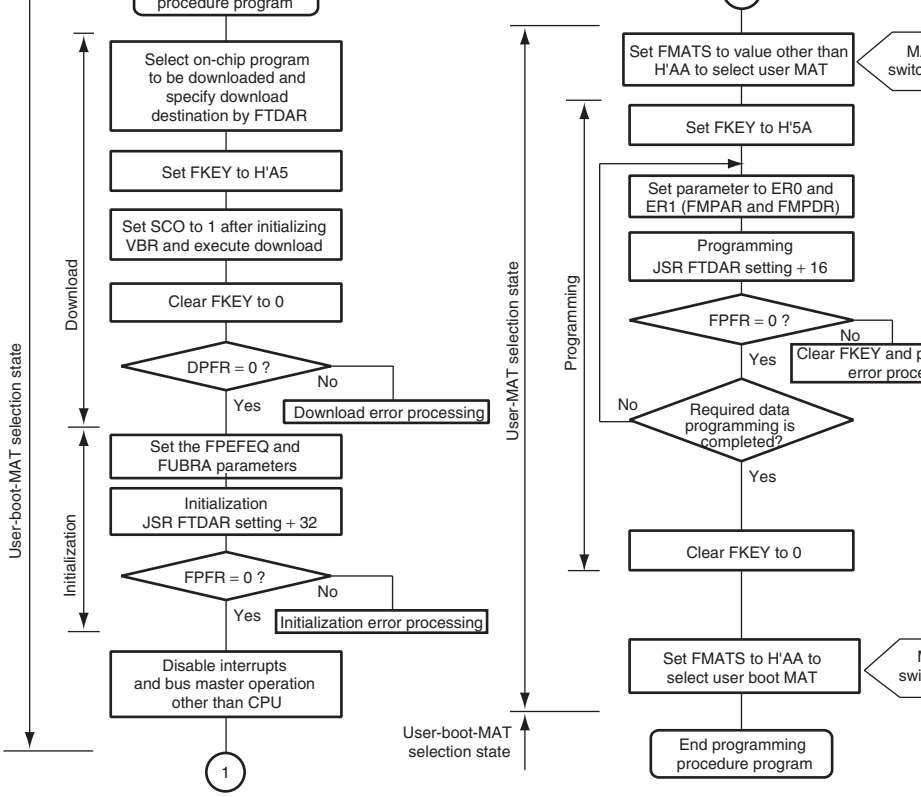
22.8.3 User Boot Mode

Branching to a programming/erasing program prepared by the user enables user boot mo
is a user-arbitrary 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 routin
running, NMI and all other interrupts cannot be accepted. Next, processing starts from th
execution start address of the reset vector in the user boot MAT. At this point, the user bo
is selected (FMATS = H'AA) as the execution memory MAT.



Note: The MAT must be switched by FMATS to perform the programming error processing in the user boot MA

Figure 22.14 Procedure for Programming User MAT in User Boot Mode

description in section 22.11, Switching between User MAT and User Boot MAT.

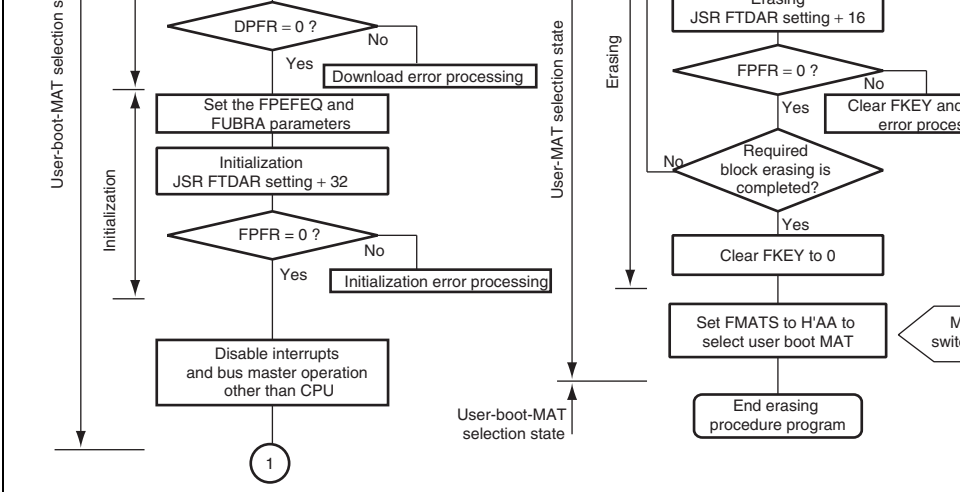
Except for memory MAT switching, the programming procedure is the same as that in us
program mode.

The area that can be executed in the steps of the procedure program (on-chip RAM, user
and external space) is shown in section 22.8.4, On-Chip Program and Storable Area for P
Data.

(3) User MAT Erasing in User Boot Mode

Figure 22.15 shows the procedure for erasing the user MAT in user boot mode.

The difference between the erasing procedures in user program mode and user boot mode
memory MAT switching as shown in figure 22.15. For erasing the user MAT in user boo
additional processing made by setting FMATS is required: switching from the user boot M
the user MAT, and switching back to the user boot MAT after erasing completes.



Note: The MAT must be switched by FMATS to perform the erasing error processing in the user boot MAT.

Figure 22.15 Procedure for Erasing User MAT in User Boot Mode

Memory MAT switching is enabled by setting FMATS. However note that access to a m
MAT is not allowed until memory MAT switching is completed. During memory MAT
the LSI is in an unstable state, e.g. if an interrupt occurs, from which memory MAT the
vector is read is undetermined. Perform memory MAT switching in accordance with the
description in section 22.11, Switching between User MAT and User Boot MAT.

Except for memory MAT switching, the erasing procedure is the same as that in user pr
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 22.8.4, On-Chip Program and Storable Area for
Data.

RAM because it will require switching of the memory MATs.

- In an operating mode in which the external address space is not accessible, such as sin mode, the required procedure programs, NMI handling vector table, and NMI handlin should be transferred to the on-chip RAM before programming/erasure starts (downlo is determined).
- The flash memory is not accessible during programming/erasure. Programming/erasu executed by the program downloaded to the on-chip RAM. Therefore, the procedure p that initiates operation, the NMI handling vector table, and the NMI handling routine stored in the on-chip RAM other than the flash memory.
- After programming/erasure starts, access to the flash memory should be inhibited unti is cleared. The reset input state (period of $\overline{RES}$ = 0) must be set to at least 100 µs whe operating mode is changed and the reset start executed on completion of programmin Transitions to the reset state are inhibited during programming/erasure. When the rese is input, a reset input state (period of $\overline{RES}$ = 0) of at least 100 µs is needed before the signal is released.
- Switching of the memory MATs by FMATS should be needed when programming/er the user MAT is operated in user boot mode. The program which switches the memor should be executed from the on-chip RAM. For details, see section 22.11, Switching b User MAT and User Boot MAT. Make sure you know which memory MAT is curren selected when switching them.
- When the program data storage area is within the flash memory area, an error will occ when the data stored is normal program data. Therefore, the data should be transferred on-chip RAM to place the address that the FMPDR parameter indicates in an area oth the flash memory.

FCCS (download)			
Operation for clearing FKEY	O	O	O
Decision of download result	O	O	O
Operation for download error	O	O	O
Operation for setting initialization parameter	O	O	O
Execution of initialization	O	×	O
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
NMI handling routine	O	×	O
Operation for disabling interrupts	O	O	O
Operation for writing H'5A to FKEY	O	O	O
Operation for setting programming parameter	O	×	O
Execution of programming	O	×	O
Decision of programming result	O	×	O
Operation for programming error	O	×	O
Operation for clearing FKEY	O	×	O

Note: * Transferring the program data to the on-chip RAM beforehand enables this are used.

Operation for clearing FKEY	O	O	O
Decision of download result	O	O	O
Operation for download error	O	O	O
Operation for setting initialization parameter	O	O	O
Execution of initialization	O	×	O
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
NMI handling routine	O	×	O
Operation for disabling interrupts	O	O	O
Operation for writing H'5A to FKEY	O	O	O
Operation for setting erasure parameter	O	×	O
Execution of erasure	O	×	O
Decision of erasure result	O	×	O
Operation for erasure error	O	×	O
Operation for clearing FKEY	O	×	O

FCCS (download)				
Operation for clearing FKEY	O	O		O
Decision of download result	O	O		O
Operation for download error	O	O		O
Operation for setting initialization parameter	O	O		O
Execution of initialization	O	×		O
Decision of initialization result	O	O		O
Operation for initialization error	O	O		O
NMI handling routine	O	×		O
Operation for disabling interrupts	O	O		O
Switching memory MATs by FMATS	O	×	O	
Operation for writing H'5A to FKEY	O	×	O	
Operation for setting programming parameter	O	×	O	
Execution of programming	O	×	O	
Decision of programming result	O	×	O	
Operation for programming error	O	×*2	O	
Operation for clearing FKEY	O	×	O	
Switching memory MATs by FMATS	O	×		O

Notes: 1. Transferring the program data to the on-chip RAM beforehand enables this are used.

2. Switching memory MATs by FMATS by a program in the on-chip RAM enables area to be used.

Operation for clearing FKEY	O	O		O
Decision of download result	O	O		O
Operation for download error	O	O		O
Operation for setting initialization parameter	O	O		O
Execution of initialization	O	×		O
Decision of initialization result	O	O		O
Operation for initialization error	O	O		O
NMI handling routine	O	×		O
Operation for disabling interrupts	O	O		O
Switching memory MATs by FMATS	O	×	O	
Operation for writing H'5A to FKEY	O	×	O	
Operation for setting erasure parameter	O	×	O	
Execution of erasure	O	×	O	
Decision of erasure result	O	×	O	
Operation for erasure error	O	×*	O	
Operation for clearing FKEY	O	×	O	
Switching memory MATs by FMATS	O	×	O	

Note: Switching memory MATs by FMATS by a program in the on-chip RAM enables th
be used.

program is initiated, and the error in programming/erasure is indicated by the FPFR paral

Table 22.12 Hardware Protection

		Function to be Pro	
Item	**Description**	**Download**	**Progra Erasin**
Reset protection	• The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and the programming/erasing protection state is entered. • The reset state will not be entered by a reset using the $\overline{\text{RES}}$ pin unless the $\overline{\text{RES}}$ pin is held low until oscillation has settled after a power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width given in the AC characteristics. If a reset is input during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	O	O

by SCO bit	entered when the SCO bit in FCCS is cleared to 0 to disable download of the programming/erasing programs.		
Protection by FKEY	The programming/erasing protection state is entered because download and programming/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

22.9.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when a CPU runav occurs or operations not according to the programming/erasing procedures are detected programming/erasure of the flash memory. Aborting programming or erasure in such ca prevents damage to the flash memory due to excessive programming or erasing.

If an error occurs during programming/erasure of the flash memory, the FLER bit in FC to 1 and the error protection state is entered.

- When an interrupt request, such as NMI, occurs during programming/erasure.
- When the flash memory is read from during programming/erasure (including a vecto 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 m during programming/erasure.

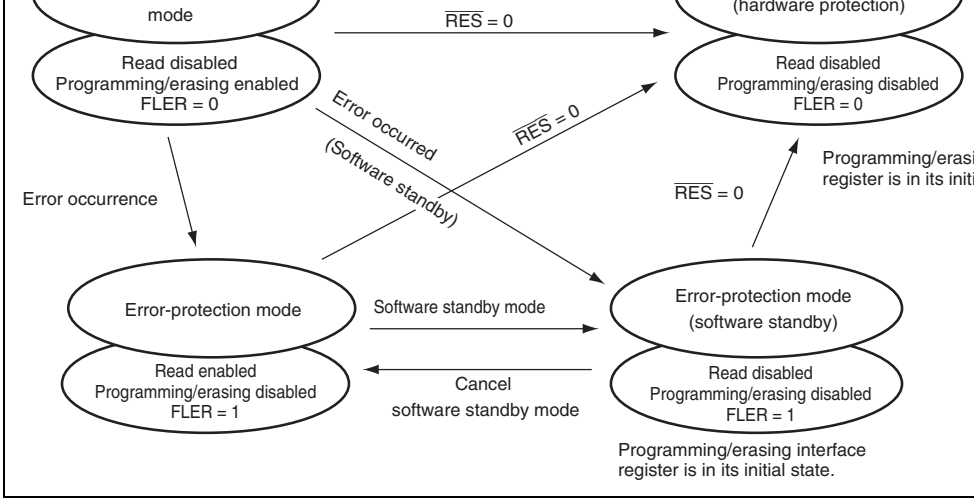


Figure 22.16 Transitions to Error Protection State

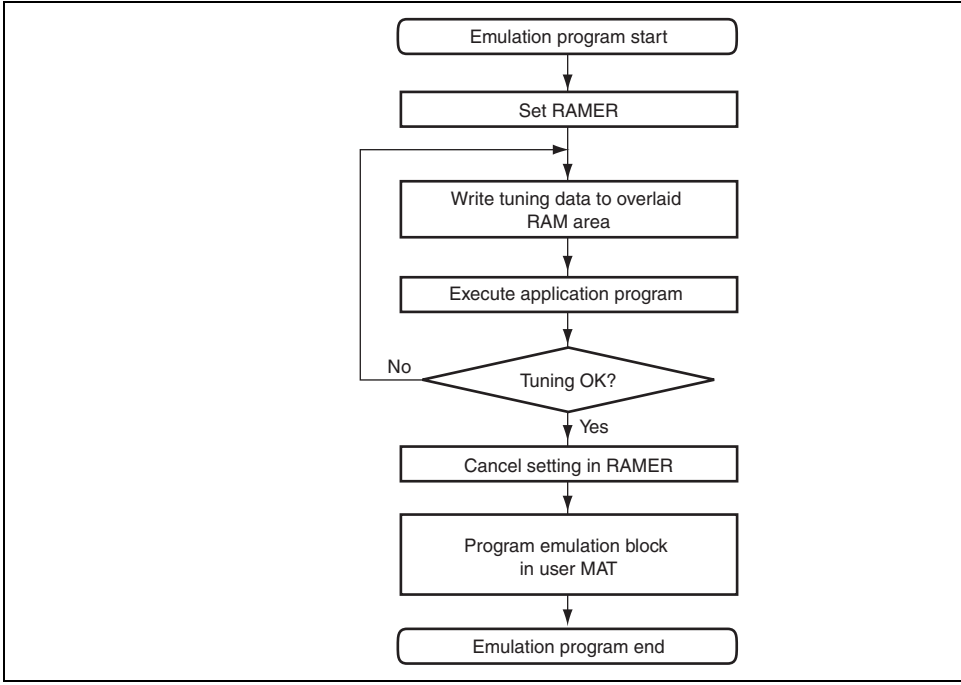


Figure 22.17 RAM Emulation Flow

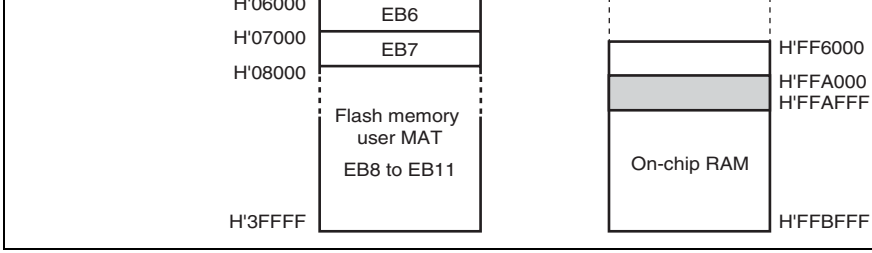


Figure 22.18 Address Map of Overlaid RAM Area

The flash memory area that can be emulated is the one area selected by bits RAM2 to RA
RAMER from among the eight blocks, EB0 to EB7, of the user MAT.

To overlay a part of the on-chip RAM with block EB0 for realtime emulation, set the RA
RAMER to 1 and bits RAM2 to RAM0 to B'000.

For programming/erasing the user MAT, the procedure programs including a download p
of the on-chip program must be executed. At this time, the download area should be spec
that the overlaid RAM area is not overwritten by downloading the on-chip program. Sinc
in which the tuned data is stored is overlaid with the download area when FTDAR = H'01
tuned data must be saved in an unused area beforehand.

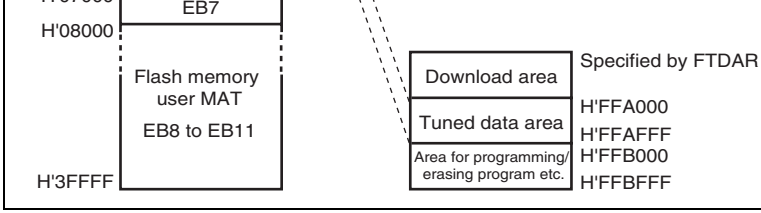


Figure 22.19 Programming Tuned Data

1. After tuning program data is completed, clear the RAMS bit in RAMER to 0 to canc overlaid RAM.
2. Transfer the user-created procedure program to the on-chip RAM.
3. Start the procedure program and download the on-chip program to the on-chip RAM address of the download destination should be specified by FTDAR so that the tuned does not overlay the download area.
4. When block EB0 of the user MAT has not been erased, the programming program m downloaded after block EB0 is erased. Specify the tuned data saved in the FMPAR a FMPDR parameters and then execute programming.

Note: Setting the RAMS bit to 1 makes all the blocks of the user MAT enter the programming/erasing protection state (emulation protection state) regardless of of the RAM2 to RAM0 bits. Under this condition, the on-chip program cannot b downloaded. When data is to be actually programmed and erased, clear the RAM to 0.

for eight times (this prevents access to the flash memory during memory MAT switch

3. If an interrupt request has occurred during memory MAT switching, there is no guara which memory MAT is accessed. Always mask the maskable interrupts before switch memory MATs. In addition, configure the system so that NMI interrupts do not occur memory MAT switching.
4. After the memory MATs have been switched, take care because the interrupt vector ta also have been switched. If interrupt processing is to be the same before and after mer MAT switching, transfer the interrupt processing routines to the on-chip RAM and sp VBR to place the interrupt vector table in the on-chip RAM.
5. The size of the user MAT is different from that of the user boot MAT. Addresses whi the size of the 16-Kbyte user boot MAT should not be accessed. If an attempt is made read as an undefined value.

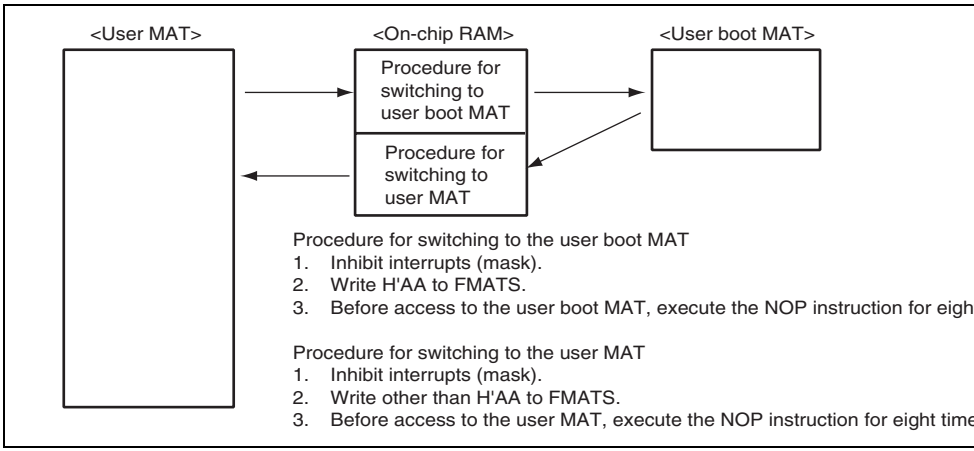


Figure 22.20 Switching between User MAT and User Boot MAT

22.13 Standard Serial Communication Interface Specifications for Mode

The boot program initiated in boot mode performs serial communication using the host a chip SCI_4. The serial communication interface specifications are shown below.

The boot program has three states.

1. Bit-rate-adjustment state

   In this state, the boot program adjusts the bit rate to achieve serial communication w host. Initiating boot mode enables starting of the boot program and entry to the bit-ra adjustment state. The program receives the command from the host to adjust the bit r adjusting the bit rate, the program enters the inquiry/selection state.

2. Inquiry/selection state

   In this state, the boot program responds to inquiry commands from the host. The dev clock mode, and bit rate are selected. After selection of these settings, the program is enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure t chip RAM and erases the user MATs and user boot MATs before the transition.

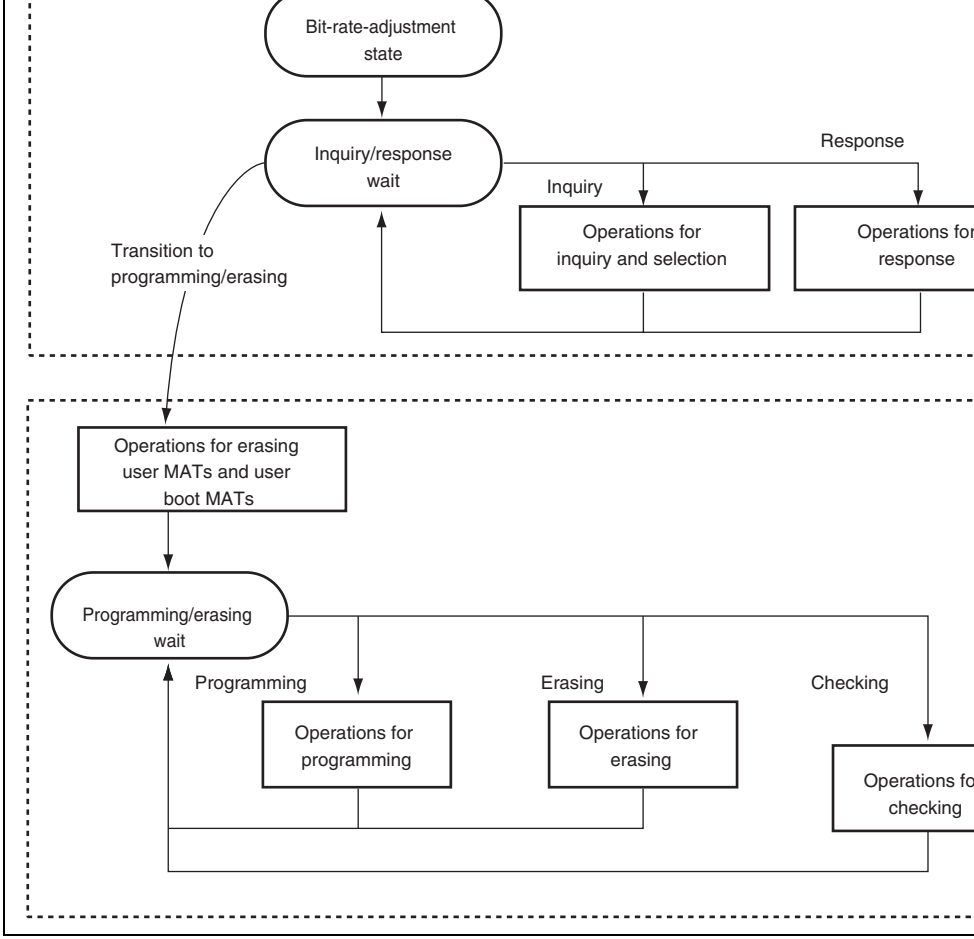


Figure 22.21 Boot Program States

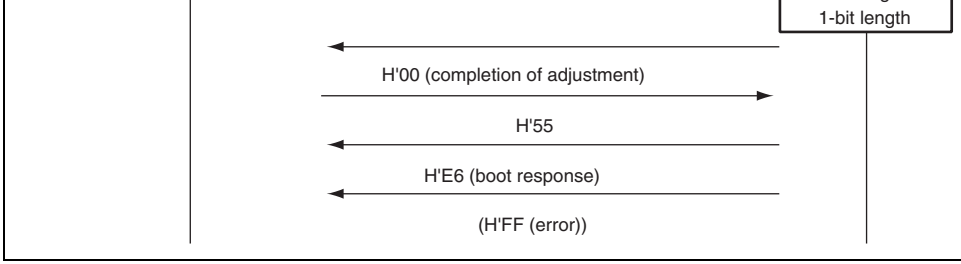


Figure 22.22 Bit-Rate-Adjustment Sequence

(2) Communications Protocol

After adjustment of the bit rate, the protocol for serial communications between the host boot program is as shown below.

1. One-byte commands and one-byte responses

   These one-byte commands and one-byte responses consist of the inquiries and the A successful completion.
2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selectio responses to inquiries.

   The program data size is not included under this heading because it is determined in command.
3. Error response

   The error response is a response to inquiries. It consists of an error response and an e and comes two bytes.
4. Programming of 128 bytes

   The size is not specified in commands. The size of n is indicated in response to the programming unit inquiry.

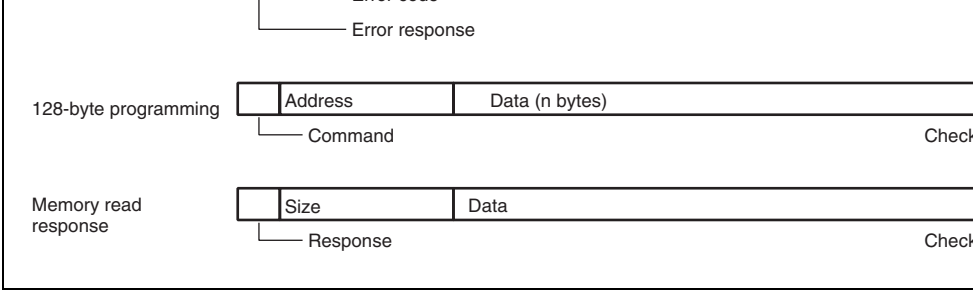


Figure 22.23 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasin checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, amoun and checksum
- Checksum (one byte): The checksum is calculated so that the total of all values from t command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error response (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

H'10	Device selection	Selection of device code
H'21	Clock mode inquiry	Inquiry regarding numbers of clock and values of each mode
H'11	Clock mode selection	Indication of the selected clock mod
H'22	Multiplication ratio inquiry	Inquiry regarding the number of freq multiplied clock types, the number o multiplication ratios, and the values multiple
H'23	Operating clock frequency inquiry	Inquiry regarding the maximum and values of the main clock and periph
H'24	User boot MAT information inquiry	Inquiry regarding the number of use MATs and the start and last addres each MAT
H'25	User MAT information inquiry	Inquiry regarding the a number of u and the start and last addresses of
H'26	Block for erasing information Inquiry	Inquiry regarding the number of blo the start and last addresses of each
H'27	Programming unit inquiry	Inquiry regarding the unit of progra
H'3F	New bit rate selection	Selection of new bit rate
H'40	Transition to programming/erasing state	Erasing of user MAT and user boot entry to programming/erasing state
H'4F	Boot program status inquiry	Inquiry into the operated status of th program

response to the supported device inquiry.

Command

H'20

- Command, H'20, (one byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
...			
SUM			

- Response, H'30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transmitted, excluding the command, size, and checksum, that is, the amount of data contributes by the number of devices, characters codes and product names
- Number of devices (one byte): The number of device types supported by the boot pro
- Number of characters (one byte): The number of characters in the device codes and bo program's name
- Device code (four bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one byte): Checksum

  The checksum is calculated so that the total number of all values from the command b the SUM byte becomes H'00.

- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response	H'90	ERROR

- Error response, H'90, (one byte): Error response to the device selection command
  ERROR : (one byte): Error code
  H'11: Sum check error
  H'21: Device code error, that is, the device code does not match

(c) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode i

Command	H'21

- Command, H'21, (one byte): Inquiry regarding clock mode

Response	H'31	Size	Mode	···	SUM

- Response, H'31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the modes
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode
- SUM (one byte): Checksum

- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response

H'91	ERROR

- Error response, H'91, (one byte) : Error response to the clock mode selection comman
- ERROR : (one byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock m
be selected using these respective values.

...				
SUM				

- Response, H'32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of multipliable opera clocks and multiplication ratios and the multiplication ratios
- Number of multipliable operating clocks (one byte): The number of clocks that can b for multiplication (e.g. if the main and peripheral clock frequencies can be multiplie number of multipliable operating clocks will be H'02.)
- Number of multiplication ratios (one byte): The number of multiplication ratios for e (e.g. the number of multiplication ratios to which the main clock can be set and the p clock can be set.)
- Multiplication ratio (one byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-freque multiplier is four, the value of multiplication ratio will be H'04.)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when th divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

  The number of multiplication ratios returned is the same as the number of multiplica and as many groups of data are returned as there are multipliable operating clocks.
- SUM (one byte): Checksum

operating clock frequency	frequency
...	
SUM	

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operatin frequency types
  (e.g. when there are two operating clock frequency types, which are the main and peri clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of the multiplied or divided clock frequency.

  The minimum and maximum values of the operating clock frequency represent the va MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the va 20.00 MHz, it will be 2000, which is H'07D0.)
- Maximum value (two bytes): Maximum value among the multiplied or divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating cloc frequencies.
- SUM (one byte): Checksum

- Response, H'34, (one byte): Response to user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start a and area-last address
- Number of Areas (one byte): The number of consecutive user boot MAT areas
  When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Area-start address (four byte): Start address of the area
- Area-last address (four byte): Last address of the area
  There are as many groups of data representing the start and last addresses as there ar
- SUM (one byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command

H'25

- Command, H'25, (one byte): Inquiry regarding user MAT information

Response

H'35	Size	Number of areas	
Start address area			Last address area
...			
SUM			

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start a and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area

Response	H'36	Size	Number of blocks	
	Block start address			Block last address
	...			
	SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-st addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block
- Block last Address (four bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command	H'27

- Command, H'27, (one byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM

- Response, H'37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicate the programming unit, which is fix
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (one byte): Checksum

- Command, H'3F, (one byte): Selection of new bit rate
- Size (one byte): The number of bytes that represents the bit rate, input frequency, nu multipliable operating clocks, and multiplication ratios
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which i
- Input frequency (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 1 when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Number of multipliable operating clocks (one byte): The number of operating clocks device that can be selected for multiplication.
- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for th operating frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the c frequency is multiplied by four, the multiplication ratio will be H'04.)
  Division ratio: The inverse of the division ratio, as a negative number (e.g. when the frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for th peripheral frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the c frequency is multiplied by four, the multiplication ratio will be H'04.)
  (Division ratio: The inverse of the division ratio, as a negative number (E.g. when th divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- SUM (one byte): Checksum

Response | H'06 |

- Response, H'06, (one byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

H'27: Operating frequency error

The frequency is not within the specified range.

(4) Receive Data Check

The methods for checking of receive data are listed below.

1. Input frequency

   The received value of the input frequency is checked to ensure that it is within the ran
   minimum to maximum frequencies which matches the clock modes of the specified d
   When the value is out of this range, an input-frequency error is generated.

2. Multiplication ratio

   The received value of the multiplication ratio or division ratio is checked to ensure tha
   matches the clock modes of the specified device. When the value is out of this range,
   frequency error is generated.

3. Operating frequency error

   Operating frequency is calculated from the received value of the input frequency and
   multiplication or division ratio. The input frequency is input to the LSI and the LSI is
   at the operating frequency. The expression is given below.

   Operating frequency = Input frequency × Multiplication ratio, or
   Operating frequency = Input frequency ÷ Division ratio

   The calculated operating frequency should be checked to ensure that it is within the ra
   minimum to maximum frequencies which are available with the clock modes of the sp
   device. When it is out of this range, an operating frequency error is generated.

When the new bit rate is selectable, the rate will be set in the register after sending ACK response. The host will send an ACK with the new bit rate for confirmation and the boot will response with that rate.

Confirmation	H'06

- Confirmation, H'06, (one byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06, (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 22.24.

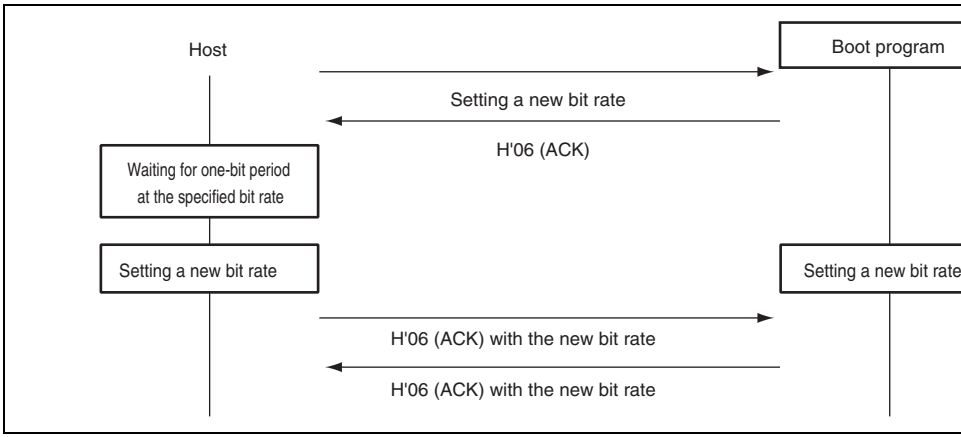


Figure 22.24 New Bit-Rate Selection Sequence

Command	H'40

- Command, H'40, (one byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (one byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MAT and user boot MAT have been
  by the transferred erasing program.

Error Response	H'C0	H'51

- Error response, H'C0, (one byte): Error response for user boot MAT blank check
- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

(6) Command Error

A command error will occur when a command is undefined, the order of commands is inc
or a command is unacceptable. Issuing a clock-mode selection command before a device
or an inquiry command after the transition to programming/erasing state command, are ex

Error Response	H'80	H'xx

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

be made, such as the multiplication-ratio inquiry (H'22) or operating frequency inqui
which are needed for a new bit-rate selection.

6. A new bit rate should be selected with the new bit-rate selection (H'3F) command, ac
to the returned information on multiplication ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user boot MAT
MAT should be made to inquire about the user boot MATs information inquiry (H'2
MATs information inquiry (H'25), erased block information inquiry (H'26), and prog
unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to
programming/erasing state command (H'40). The boot program will then enter the
programming/erasing state.

H'43	User MAT programming selection	Transfers the user MAT programmi program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user b
H'4B	User MAT sum check	Checks the checksum of the user M
H'4C	User boot MAT blank check	Checks the blank data of the user b
H'4D	User MAT blank check	Checks the blank data of the user M
H'4F	Boot program status inquiry	Inquires into the boot program's sta

command represents the data programmed according to the method specified by the command. When more than 128-byte data is programmed, 128-byte commands shou repeatedly be executed. Sending a 128-byte programming command with H'FFFFFF address will stop the programming. On completion of programming, the boot progra wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programm another method or of another MAT, the procedure must be repeated from the progra selection command.

The sequence for the programming selection and 128-byte programming commands in figure 22.25.

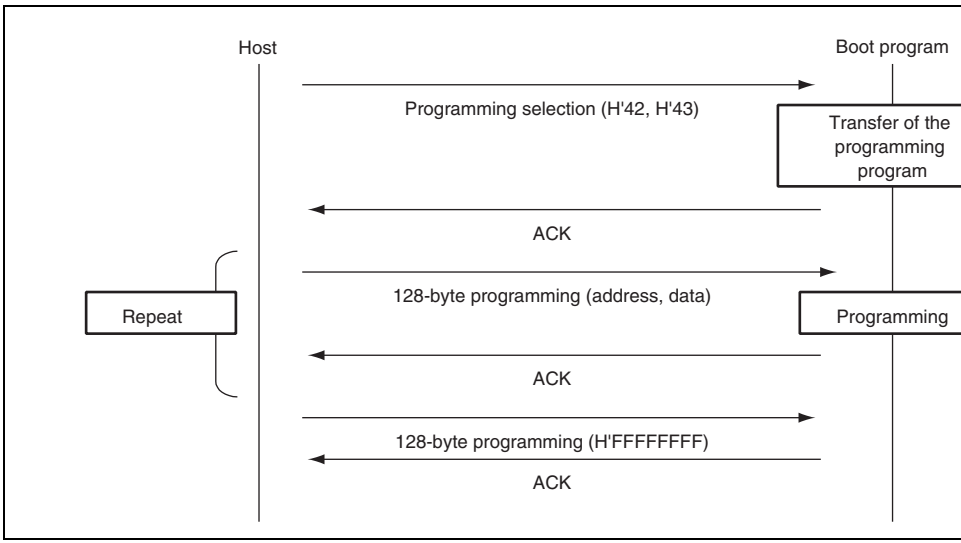


Figure 22.25 Programming Sequence

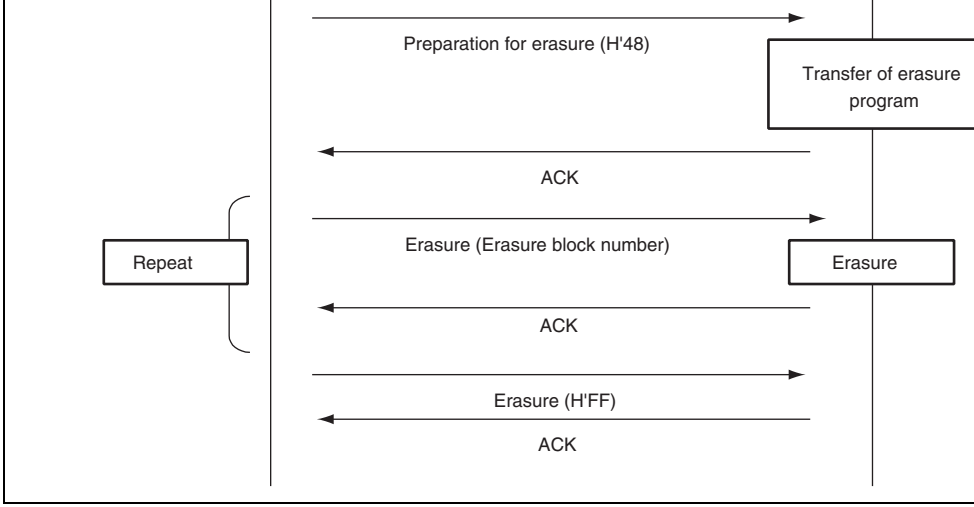


Figure 22.26 Erasure Sequence

Error Response	H'C2	ERROR

- Error response : H'C2 (1 byte): Error response to user boot MAT programming selec
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not complet

(b) User MAT Programming Selection

The boot program will transfer a program for user MAT programming selection. The da
programmed to the user MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (one byte): User-program programming selection

Response	H'06

- Response, H'06, (one byte): Response to user-program programming selection
  When the programming program has been transferred, the boot program will return A

Error Response	H'C3	ERROR

- Error response : H'C3 (1 byte): Error response to user boot MAT programming selec
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not complet

- Programming Address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00 : H'01000000)
- Program data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum Error

  H'2A: Address error

  The address is not in the specified MAT.

  H'53: Programming error

  A programming error has occurred and programming cannot be continu

The specified address should match the unit for programming of data. For example, when programming is in 128-byte units, the lower eight bits of the address should be H'00 or H When there are less than 128 bytes of data to be programmed, the host should fill the rest H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop programming operation. The boot program will interpret this as the end of the programmi wait for selection of programming or erasing.

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum error

  H'53: Programming error

  An error has occurred in programming and programming cannot be co

(d) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the tra
erasure program.

Command	H'48

- Command, H'48, (one byte): Erasure selection

Response	H'06

- Response, H'06, (one byte): Response for erasure selection
  After the erasure program has been transferred, the boot program will return ACK.

Error Response	H'C8	ERROR

- Error Response, H'C8, (one byte): Error response to erasure selection
- ERROR: (one byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complet

Response	H'06

- Response, H'06, (one byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error Response, H'D8, (one byte): Response to Erasure
- ERROR (one byte): Error code

  H'11: Sum check error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasure error
  An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a select command.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H'FF
  Stop code for erasure
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the proce should be executed from the erasure selection command.

H'2A: Address error

An address error occurs when the area setting is incorrect.

- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response

H'52	Read size						
Data	···						
SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Error Response

H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

H'11: Sum check error

H'2A: Address error

The read address is not in the MAT.

H'2B: Size error

The read size exceeds the MAT.

This is fixed to 4.

- Checksum of user boot program (four bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(h) User-Program Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the use program.

Command	H'4B

- Command, H'4B, (one byte): Sum check for user program

Response	H'5B	Size	Checksum of user program	SUM

- Response, H'5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

Error Response	H'CC	H'52

- Error Response, H'CC, (one byte): Response to blank check for user boot MAT
- Error Code, H'52, (one byte): Erasure has not been completed.

(j) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the resu

Command	H'4D

- Command, H'4D, (one byte): Blank check for user MATs

Response	H'06

- Response, H'06, (one byte): Response to the blank check for user MATs
  If the contents of all user MATs are blank (H'FF), the boot program will return ACK

Error Response	H'CD	H'52

- Error Response, H'CD, (one byte): Error response to the blank check of user MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

- Status (one byte): State of the boot program
- ERROR (one byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (one byte): Sum check

Table 22.17 Status Code

Code	Description
H'11	Device selection wait
H'12	Clock mode selection wait
H'13	Bit rate selection wait
H'1F	Programming/erasing state transition wait (bit rate selection is completed)
H'31	Programming state for erasure
H'3F	Programming/erasing selection wait (erasure is completed)
H'4F	Program data receive wait
H'5F	Erase block specification wait (erasure is completed)

H'26	Multiplication ratio error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data length error
H'51	Erasure error
H'52	Erasure incomplete error
H'53	Programming error
H'54	Selection processing error
H'80	Command error
H'FF	Bit-rate-adjustment confirmation error

3.3-V programming voltage. Use only the specified socket adapter.

5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasure in which a high voltage is applied to the flash memory. Doing s damage the flash memory permanently. If a reset is input accidentally, the reset must released after the reset input period of at least 100µs.
6. The flash memory is not accessible until FKEY is cleared after programming/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) o 100µs. Transition to the reset state during programming/erasure is inhibited. If a reset accidentally, the reset must be released after the reset input period of at least 100µs.
7. At powering on or off the Vcc power supply, fix the $\overline{RES}$ pin to low and set the flash to hardware protection state. This power on/off timing must also be satisfied at a pow power-on caused by a power failure and other factors.
8. In on-board programming mode or programmer mode, programming of the 128-byte programming-unit block must be performed only once. Perform programming in the s where the programming-unit block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of progra erasure in on-board programming mode, it is recommended that automatic programm performed after execution of automatic erasure.
10. To program the flash memory, the program data and program must be allocated to ad which are higher than those of the external interrupt vector table and H'FF must be wr all the system reserved areas in the exception handling vector table.
11. The programming program that includes the initialization routine and the erasing prog includes the initialization routine are each 4 Kbytes or less. Accordingly, when the CP frequency is 35 MHz, the download for each program takes approximately 60 µs at th maximum.

Immediately after executing the instruction to set the SCO bit to 1, dummy read of th
must be executed twice.

15. The contents of general registers ER0 and ER1 are not saved during download of an
program, initialization, programming, end of the programming, or erasure. When nec
the general registers before a download request or before execution of initialization,
programming, or erasure using the procedure program.

by the frequency dividers, PLL circuit, and selectors. The frequencies of the system clo
peripheral module clock and external bus clock are changed the by setting the system cl
control register (SCKCR) by software. The ΔΣ A/D converter clock is generated from th
oscillator output multiplied by 8, the frequency of which can be changed by setting the Δ
mode register (DSADMR) by software.

Frequencies of the peripheral module clock, the external bus clock, and the system clock
set independently, although the peripheral module clock and the external bus clock only
frequencies lower than the system clock frequency. Since the ΔΣ A/D converter has bee
to deliver the maximum precision at approximately 25 MHz, the division ratio for the ΔΣ
converter clock should be set in DSADMR so as to make the frequency near 25 MHz.

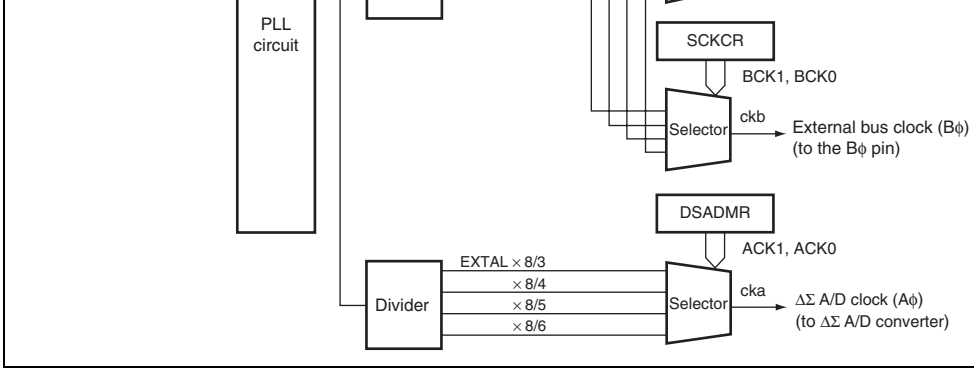


Figure 23.1 Block Diagram of Clock Pulse Generator

Table 23.1 Selection for Clock Pulse Generator

EXTAL Input Clock Frequency	Iϕ/Pϕ/Bϕ	Aϕ (ΔΣ A/D Converter)
8 MHz to 18 MHz	EXTAL ×4, ×2, ×1, ×1/2	[EXTAL ×8] ×1/3, ×1/4, ×1/5 (Frequency near 25 MHz is recommended)

bus clocks.

Bit	15	14	13	12	11	10	9
Bit Name	PSTOP1	—	POSEL1	—	—	ICK2	ICK1
Initial Value	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	—	PCK2	PCK1	PCK0	—	BCK2	BCK1
Initial Value	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PSTOP1	0	R/W	Bϕ Output Enable Enables the Bϕ output on PA7. • Normal operation 0: Bϕ output 1: Fixed high
14	—	0	R/W	Reserved This bit enables read/write operations, but the v should always be 0.
13	POSEL1	0	R/W	Clock Output Select 1 Selects the clock signal to be output from PA7. 0: External bus clock (Bϕ) 1: Setting prohibited

				001: × 2 010: × 1 011: × 1/2 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited The frequencies of the peripheral module clock a external bus clock change to the same frequenc system clock if the frequency of the system clock than that of the two clocks.
7	—	0	R/W	Reserved This bit enables read/write operations, but the w should always be 0.

				101: Setting prohibited 110: Setting prohibited 111: Setting prohibited The frequency of the peripheral module clock s lower than that of the system clock. Though the can be set so as to make the frequency of the p module clock higher than that of the system clo clocks will have the same frequency in reality.
3	—	0	R/W	Reserved This bit enables read/write operations, but the v should always be 0.
2	BCK2	0	R/W	External Bus Clock (Bϕ) Select
1	BCK1	1	R/W	These bits select the frequency of the external
0	BCK0	0	R/W	The ratio to the input clock is as follows: 000: × 4 001: × 2 010: × 1 011: × 1/2 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited The frequency of the external bus clock should than that of the system clock. Though these bits set so as to make the frequency of the external higher than that of the system clock, the clocks the same frequency in reality.

[Legend]
X: Don't care

Bit	Bit Name	Initial Value	R/W	Description
7	BIASE	0	R/W	Bias Circuit Control Sets whether the bias circuit is to be operated or stopped. 0: Stops the bias circuit. 1: Operates the bias circuit.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
2	ACK2	0	R/W	ΔΣ A/D Converter Frequency Division Clock Sele
1	ACK1	0	R/W	These bits select the frequency of the ΔΣ A/D clo
0	ACK0	0	R/W	The ratio to the input clock is as follows. In settin value of Aϕ should be in the neighborhood of 25 000: × 1/6 001: × 1/5 010: × 1/4 011: × 1/3 1xx: Setting prohibited

[Legend]
X: Don't care

frequency of 8 to 18 MHz should be connected.

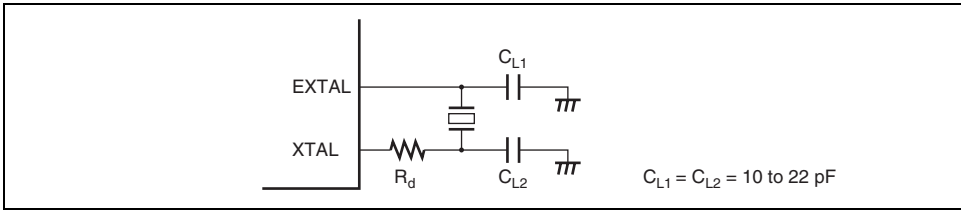


Figure 23.2 Connection of Crystal Resonator (Example)

Table 23.1 Damping Resistance Value

Frequency (MHz)	8	12	16	18
R_d (Ω)	200	0	0	0

Figure 23.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator t
the characteristics shown in table 23.2.

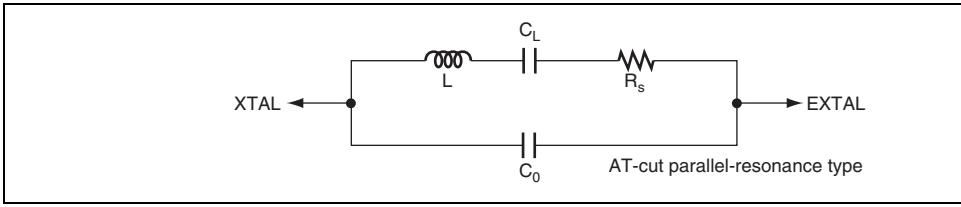


Figure 23.3 Crystal Resonator Equivalent Circuit

input to the XTAL pin, make sure that the external clock is held high in standby mode.

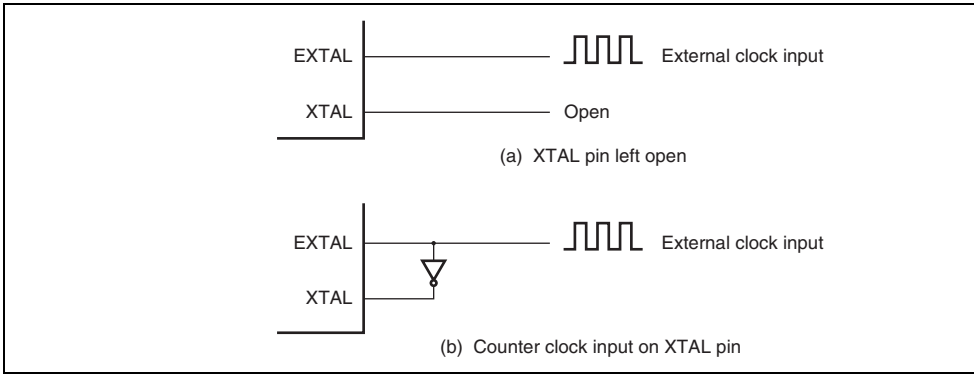


Figure 23.4 External Clock Input (Examples)

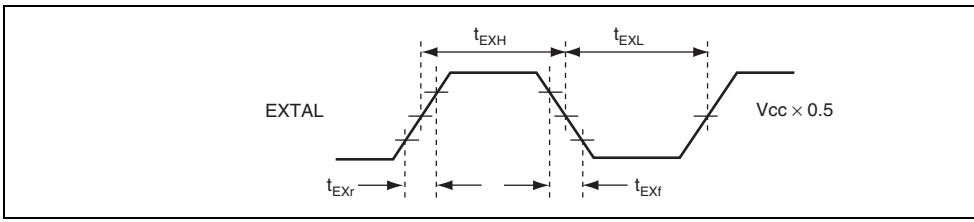


Figure 23.5 External Clock Input Timing

The frequency divider divides the PLL clock to generate a 1/2, 1/4, or 1/8 clock. After b
to ICK0, PCK 2 to PCK0, and BCK2 to BCK0 are modified, this LSI operates at the mo
frequency.

23.4.2 Aϕ Frequency Divider

The frequency divider divides the frequency of the PLL clock to create 1/3, 1/4, 1/5, and
clocks. After the ACK2, ACK1, and ACK0 bits are rewritten, the ΔΣ A/D converter ope
according to the frequency available after change. Before rewriting these bits, you need
ΔΣ A/D converter's module stop bit to 1 so that the ΔΣ A/D converter is stopped. Setting
frequency is recommended because of the characteristics of the ΔΣ A/D converter: that i
designed to produce maximum accuracy in the neighborhood of 25 MHz.

≤ Pϕ ≤ 35 MHz, and 8 MHz ≤ Bϕ ≤ 50 MHz.

2. All the on-chip peripheral modules (except for the DMAC and DTC) operate on the P
   Therefore, note that the time processing of modules such as a timer and SCI differs be
   after changing the clock division ratio.

   In addition, wait time for clearing software standby mode differs by changing the cloc
   division ratio. For details, see section 24.7.3, Setting Oscillation Settling Time after E
   Software Standby Mode.

3. The relationship among the system clock, peripheral module clock, and external bus c
   ≥ Pϕ and Iϕ ≥ Bϕ. In addition, the system clock setting has the highest priority. Accor
   Pϕ or Bϕ may have the frequency set by bits ICK2 to ICK0 regardless of the settings
   PCK2 to PCK0 or BCK2 to BCK0.

4. Figure 23.6 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, eac
   frequency will be modified within one cycle (worst case) of the external input clock ϕ

Figure 23.6 Clock Modification Timing

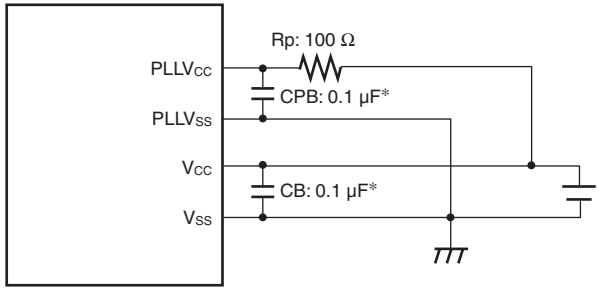
23.5.2 Notes on Resonator

Since various characteristics related to the resonator are closely linked to the user's boar thorough evaluation is necessary on the user's part, using the resonator connection exam shown in this section as a reference. As the parameters for the resonator will depend on floating capacitance of the resonator and the mounting circuit, the parameters should be determined in consultation with the resonator manufacturer. The design must ensure tha exceeding the maximum rating is not applied to the resonator pin.

23.5.3 Notes on Board Design

When using the crystal resonator, place the crystal resonator and its load capacitors as cl possible to the XTAL and EXTAL pins. Other signal lines should be routed away from t oscillation circuit as shown in figure 23.7 to prevent induction from interfering with cor oscillation.

PLLVss from the other Vcc and Vss lines at the board power supply source, and be sure t
bypass capacitors CPB and CB close to the pins.



Note: * CB and CPB are laminated ceramic capacitors.

Figure 23.8 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

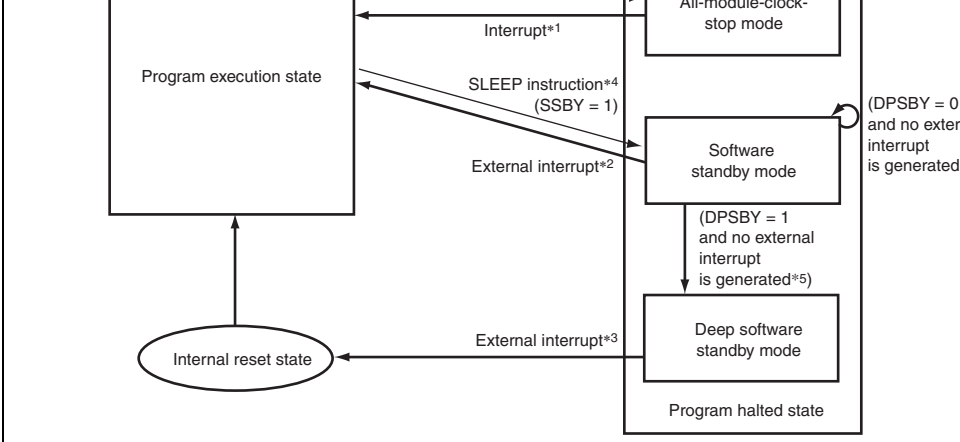
Table 24.1 shows conditions to shift to a power-down mode, states of the CPU and perip
modules, and clearing method for each mode. After the reset state, since this LSI operate
normal program execution state, the modules, other than the DMAC and DTC, are stopp

Oscillator	Operating	Operating	Halted	Halted	Halte
CPU	Halted (retained)	Halted (retained)	Halted (retained)	Halted (undefined)	Halte (unde
On-chip RAM	Operating (retained)	Halted (retained)	Halted (retained)	Halted (retained/ undefined)*5	Halte (unde
Watchdog timer	Operating	Operating	Halted (retained)	Halted (undefined)	Halte (unde
8-bit timer (unit 0/1)	Operating	Operating*4	Halted (retained)	Halted (undefined)	Halte (unde
Other peripheral modules	Operating	Halted*1	Halted*1	Halted*7 (undefined)	Halte (unde
I/O ports	Operating	Retained	Retained*6	Halted*6	Hi-Z

Notes: "Halted (retained)" in the table means that the internal values are retained and operations are suspended.

"Halted (undefined)" in the table means that the internal values are undefined a power supply for internal operations is turned off.

1. SCI and ΣΔ A/D converter enters the reset state, and other peripheral modules their states.
2. External interrupt and some internal interrupts (8-bit timer and watchdog timer)
3. All peripheral modules enter the reset state.
4. "Functioning" or "Halted" is selectable through the setting of bits MSTPA11 to in MSTPCRA.
5. "Retained" or "undefined" of the contents of RAM is selected by the setting of t RAMCUT2 to RAMCUT0 in DPSBYCR.
6. Retention or high-impedance for the address bus and bus-control signals ($\overline{CS0}$ $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$) is selected by the setting of the OPE bit in SBYCR.
7. Some peripheral modules enter a state where the register values are retained.



[Legend] ——→ Transition after exception handling

Notes: 1. NMI, IRQ0 to IRQ15, 8-bit timer interrupts, and watchdog timer interrupts.
Note that the 8-bit timer interrupt is valid when the MSTPCRA11 to MSTPCRA8 bit is cleared to 0.
2. NMI, and IRQ0 to IRQ15. Note that IRQ is valid only when the corresponding bit in SSIER is set to 1.
3. NMI, and IRQ0-A to IRQ3-A. Note that IRQ is valid only when the corresponding bit in DPSIER is set to 1.
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 through execution of interrupt exception handling. In this case, a transition to deep software standby mode made. For details, refer to section 24.12, Usage Notes.

From any state, a transition to hardware standby mode occurs when STBY is driven low.
From any state except hardware standby mode, a transition to the reset state occurs when RES is driven low.

Figure 24.1 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 (DPSBKRn)

24.2.1 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	

				operation. For clearing, write 0 to this bit. When is used in watchdog timer mode, the setting of t disabled. In this case, a transition is always ma sleep mode or all-module-clock-stop mode afte SLEEP instruction is executed. When the SLPI 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 (CS0 to CS7, AS, RD, HWR, an retained or these lines are set to the high-Z stat software standby mode or deep software stand 0: In software standby mode or deep software s mode, address bus and bus control signal lin high-impedance. 1: In software standby mode or deep software s mode, output states of address bus and bus signals are retained.
13	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.

	While oscillation is being settled, the timer is coun 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 valu always be 0.

Notes: 1. x: Don't care
2. With the F-ZTAT version, the flash memory settling time must be reserved.

24.2.2 Module Stop Control Registers A and B (MSTPCRA and MSTPCRB)

MSTPCRA and MSTPCRB control module stop state. Setting a bit to 1 makes the corre 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	
Initial value:	0	0	0	0	1	1	1	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit name	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	
Initial value:	1	1	1	1	1	1	1	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- MSTPCRA

Bit	Bit Name	Initial Value	R/W	Module
15	ACSE	0	R/W	All-Module-Clock-Stop Mode Enable Enables/disables all-module-clock-stop state for current consumption by stopping the bus control I/O ports operations when the CPU executes the instruction after module stop mode has been set the on-chip peripheral modules controlled by MS 0: All-module-clock-stop mode disabled 1: All-module-clock-stop mode enabled
14	MSTPA14	0	R/W	Reserved
13	MSTPA13	0	R/W	DMA controller (DMAC)
12	MSTPA12	0	R/W	Data transfer controller (DTC)
11	MSTPA11	1	R/W	8-bit timer (TMR_7 and TMR_6)
10	MSTPA10	1	R/W	8-bit timer (TMR_5 and TMR_4)
9	MSTPA9	1	R/W	8-bit timer (TMR_3 and TMR_2)
8	MSTPA8	1	R/W	8-bit timer (TMR_1 and TMR_0)
7	MSTPA7	1	R/W	Reserved
6	MSTPA6	1	R/W	These bits are always read as 1. The write value always be 1.

1	MSTPA1	1	R/W	These bits are always read as 1. The write valu always be 1.
0	MSTPA0	1	R/W	16-bit timer pulse unit (TPU channels 5 to 0)

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Module
15	MSTPB15	1	R/W	Programmable pulse generator (PPG)
14	MSTPB14	1	R/W	Reserved
13	MSTPB13	1	R/W	These bits are always read as 1. The write valu always be 1.
12	MSTPB12	1	R/W	Serial communication interface_4 (SCI_4)
11	MSTPB11	1	R/W	Serial communication interface_3 (SCI_3)
10	MSTPB10	1	R/W	Serial communication interface_2 (SCI_2)
9	MSTPB9	1	R/W	Serial communication interface_1 (SCI_1)
8	MSTPB8	1	R/W	Serial communication interface_0 (SCI_0)
7	MSTPB7	1	R/W	I^2C bus Interface 1 (IIC_1)
6	MSTPB6	1	R/W	I^2C bus Interface 0 (IIC_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 valu always be 1.
2	MSTPB2	1	R/W	
1	MSTPB1	1	R/W	
0	MSTPB0	1	R/W	

Bit							
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	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	M
Initial value:	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Module
15	MSTPC15	1	R/W	Reserved This bit is always read as 1. The write value should a 1.
14	MSTPC14	1	R/W	ΔΣ A/D converter
13	MSTPC13	1	R/W	A/D converter
12	MSTPC12	1	R/W	Reserved
11	MSTPC11	1	R/W	These bits are always read as 1. The write value sho always be 1.
10	MSTPC10	1	R/W	
9	MSTPC9	1	R/W	
8	MSTPC8	1	R/W	

2	MSTPC2	0	R/W	On-chip RAM_2 (H'FF6000 to H'FF7FFF) Always set the MSTPC2 and MSTPC5 bits to the sa
1	MSTPC1	0	R/W	On-chip RAM_1, 0 (H'FF8000 to H'FFBFFF)
0	MSTPC0	0	R/W	Always set the MSTPC1 and MSTPC0 bits to the sa

24.2.4 Deep Standby Control Register (DPSBYCR)

DPSBYCR controls deep software standby mode.

DPSBYCR is initialized by input of the reset signal on the $\overline{\text{RES}}$ pin, but is not initialized internal reset signal upon exit from deep software standby mode.

Bit	7	6	5	4	3	2	1	
Bit name	DPSBY	IOKEEP	RAMCUT2	RAMCUT1	—	—	—	R
Initial value:	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1	0	Enters software standby after execution of a SLEI instruction.
1	1	Enters deep software sta mode after execution of a instruction.

When deep software standby mode is canceled
external interrupt, this bit remains at 1. Write a 0
clear it. Setting of this bit has no effect when the
used in watchdog timer mode. In this case, exec
SLEEP instruction always initiates entry to sleep
all-module-clock-stop mode. Be sure to clear this
when setting the SLPIE bit to 1.

				simultaneously with exit from dee software standby mode.
				1 The retained port states are relea when a 0 is written to this bit follo from deep software standby mod

In operation in external extended mode, howev address bus, bus control signals ($\overline{CS0}$, $\overline{AS}$, $\overline{RD}$ and $\overline{LWR}$), and data bus are set to the initial sta exit from deep software standby mode.

5	RAMCUT2	0	R/W	On-chip RAM Power Off 2 Controls the internal power supply to the on-chi deep software standby mode. For details, see descriptions of the RAMCUT0 bit.
4	RAMCUT1	0	R/W	On-chip RAM Power Off 1 Controls the internal power supply to the on-chi deep software standby mode. For details, see descriptions of the RAMCUT0 bit.
3 to 1	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
0	RAMCUT0	1	R/W	On-chip RAM Power Off 0 Controls the internal power supply to the on-chi deep software standby mode, in combination w RAMCUT2 and RAMCUT 1. 000: Power is supplied to the on-chip RAM. 111: Power is not supplied to the on-chip RAM. 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 value always be 0.

During the oscillation settling period, counting is performed with the clock frequency input to the

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

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	—	0	R/W	Reserved This bit is always read as 0. The write value should be 0.
6 to 4	—	All 0	R/W	Reserved These bits are always read as 0. The write value sh always be 0.
3	DIRQ3E	0	R/W	IRQ3 Interrupt Enable Enables or disables exit from deep software standb by IRQ3. 0: Disables exit from deep software standby mode b 1: Enables exit from deep software standby mode b
2	DIRQ2E	0	R/W	IRQ2 Interrupt Enable Enables or disables exit from deep software standb by IRQ2. 0: Disables exit from deep software standby mode b 1: Enables exit from deep software standby mode b

0: Disables exit from deep software standby mode
1: Enables exit from deep software standby mode

24.2.7 Deep Standby Interrupt Flag Register (DPSIFR)

DPSIFR is used to request an exit from deep software standby mode. When the interrup
in DPSIEGR is generated, the applicable bit in DPSIFR is set to 1. The bit is set to 1 eve
interrupt is generated in the modes other than deep software standby. Therefore, a transi
deep software standby should be made after this register bits are cleared to 0.

DPSIFR is initialized by input of the reset signal on the $\overline{\text{RES}}$ pin, but is not initialized by
internal reset signal upon exit from deep software standby mode.

Bit	7	6	5	4	3	2	1	
Bit name	DNMIF	—	—	—	DIRQ3F	DIRQ2F	DIRQ1F	
Initial value:	0	0	0	0	0	0	0	
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	

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

3	DIRQ3F	0	R/(W)*	IRQ3 Interrupt Flag [Setting condition] IRQ3 input specified in DPSIEGR is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.
2	DIRQ2F	0	R/(W)*	IRQ2 Interrupt Flag [Setting condition] IRQ2 input specified in DPSIEGR is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.
1	DIRQ1F	0	R/(W)*	IRQ1 Interrupt Flag [Setting condition]* IRQ1 input specified in DPSIEGR is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.
0	DIRQ0F	0	R/(W)*	IRQ0 Interrupt Flag [Setting condition]* IRQ0 input specified in DPSIEGR is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.

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

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 rising
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 IRQ3 pin input. 0: The interrupt request is generated by a fallin 1: The interrupt request is generated by a rising
2	DIRQ2EG	0	R/W	IRQ2 Interrupt Edge Select Selects the active edge for IRQ2 pin input. 0: The interrupt request is generated by a fallin 1: The interrupt request is generated by a rising
1	DIRQ1EG	0	R/W	IRQ1 Interrupt Edge Select Selects the active edge for IRQ1 pin input. 0: The interrupt request is generated by a fallin 1: The interrupt request is generated by a rising

The DPSRSTF bit in RSTSR indicates that deep software standby mode has been cancel
interrupt.

RSTSR is initialized by input of the reset signal on the $\overline{\text{RES}}$ pin, but is not initialized by t
internal reset signal upon exit from deep software standby mode.

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

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

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 external interrupt source specifie DPSIER or DPSIEGR and an internal reset is ge [Setting condition] Deep software standby mode is canceled by an interrupt source. [Clearing condition] Writing a 0 to this bit after reading it as 1.
6 to 0	—	0	R/W	Reserved These bits are always read as 0. The write value always be 0.

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

24.3 Multi-Clock Function

When bits ICK2 to ICK0, PCK2 to PCK0, and BCK2 to BCK0 in SCKCR are set, the cl
frequency is changed at the end of the bus cycle. The CPU and bus masters operate on th
operating clock specified by bits ICK2 to ICK0. The peripheral modules operate on the
clock specified by bits PCK2 to PCK0. The external bus operates on the operating clock
by bits BCK2 to BCK0.

Even if the frequencies specified by bits PCK2 to PCK0 and BCK2 to BCK0 are higher
frequency specified by bits ICK2 to ICK0, the specified values are not reflected in the p
module and external bus clocks. The peripheral module and external bus clocks are restr
the operating clock specified by bits ICK2 to ICK0.

24.4 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 r
other than the SCI are retained.

After the reset state is cleared, all modules other than the DMAC, DTC, and on-chip RA
placed in a module stop state.

The registers of the module for which the module stop state is selected cannot be read fr
written to.

Sleep mode is exited by any interrupt, signals on the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin, and a reset cause
watchdog timer overflow.

- Exit from sleep mode by interrupt

  When an interrupt occurs, sleep mode is exited and interrupt exception processing sta
  mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked
  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 du
  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 mod
- 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.

All-module-clock-stop mode is cleared by an external interrupt (NMI or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$ $\overline{\text{RES}}$ pin input, or an internal interrupt (8-bit timer* or watchdog timer), and the CPU re normal program execution state via the exception handling state. All-module-clock-stop not cleared if interrupts are disabled, if interrupts other than NMI are masked on the CP if the relevant interrupt is designated as a DTC activation source.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Note: * Operation or halting of the 8-bit timer can be selected by bits MSTPA11 to M in MSTPCRA.

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 softw
standby mode. The WDT should be stopped before the SLEEP instruction execution.

24.7.2 Exit from Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$*) or l
of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

1. Exit from software standby mode by interrupt

   When an NMI, or IRQ0 to IRQ15* interrupt request signal is input, clock oscillation s
   and after the elapse of the time set in bits STS4 to STS0 in SBYCR, stable clocks are
   to the entire LSI, software standby mode is cleared, and interrupt exception handling i

   When clearing software standby mode with an IRQ0 to IRQ11* interrupt, set the
   corresponding enable bit to 1 and ensure that no interrupt with a higher priority than i
   IRQ0 to IRQ11* is generated. Software standby mode cannot be cleared if the interru
   been masked on the CPU side or has been designated as a DTC activation source.

   Note: * By setting the SSIn bit in SSIER to 1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$ can be used as a soft
   standby mode clearing source.

2. Exit from software standby mode by $\overline{\text{RES}}$ pin

   When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as cloc
   oscillation starts, clocks are supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be
   until clock oscillation settles. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exce
   handling.

3. Exit from software standby mode by $\overline{\text{STBY}}$ pin

   When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

			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	128.0
	1	0	0	0	2048	58.5	81.9	102.4	157.5	204.8	256.0
				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	16.38
			1	0	262144	7.49	10.49	13.11	20.16	26.21	32.77
				1	524288	14.98	20.97	26.21	40.33	52.43	65.54
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 s
time, which includes a period where the oscillation by an oscillator is not stable
depends on the resonator characteristics. The above figures are for reference.

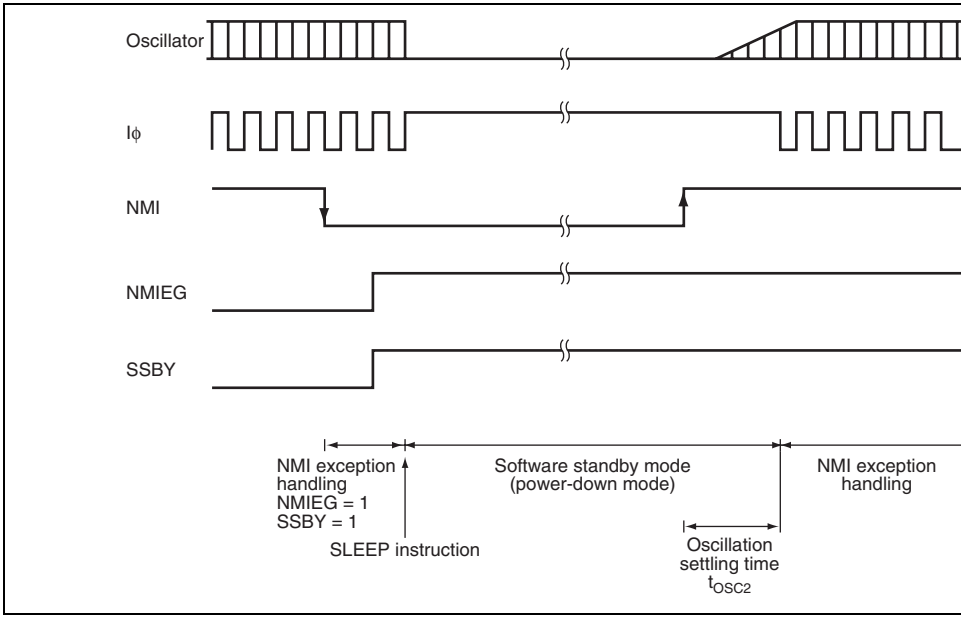


Figure 24.2 Software Standby Mode Application Example

DPSBY bit setting, and the interrupt exception handling starts after the oscillation setting
software standby mode specified by the bits STS4 to STS0 in SBYCR has elapsed.

When both of the SSBY bit in SBYCR and the DPSBY bit in DPSBYCR are set to 1 and
software standby mode clearing source event occurs, a transition to deep software standb
will be made immediately after software standby mode is entered.

In deep software standby mode, the CPU, on-chip peripheral functions, on-chip RAM, an
oscillator functionality are all halted. In addition, the internal power supply to these modu
resulting in a significant reduction in power consumption. At this time, the contents of all
registers of the CPU, on-chip peripheral functions, and on-chip RAM become undefined.

Contents of the on-chip RAM can be retained when all the bits RAMCUT2 to RAMCUT
DPSBYCR have been cleared to 0. If these bits are set to all 1, the internal power supply
chip RAM stops and the power consumption is further reduced. At this time, the contents
on-chip RAM become undefined.

The I/O ports can be retained in the same state as in software standby mode.

with DPSIECR.

When deep software standby mode clearing source is generated, internal power supp 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 cle Deep software standby mode is canceled on clearing of the internal reset, and then th exception handling starts.

When deep software standby mode is canceled by an external interrupt pin, the DPS in RSTSR is set to 1.

2. Exit from deep software standby mode by the signal on the $\overline{\text{RES}}$ pin

   Clock oscillation and internal power supply start as soon as the signal on the $\overline{\text{RES}}$ pi low. At the same time, clock signals are supplied to the LSI. In this case, the $\overline{\text{RES}}$ pi held low until the clock oscillation has become stable. Once the signal on the $\overline{\text{RES}}$ pi driven high, the CPU starts reset exception handling.

3. Exit from deep software standby mode by the signal on the $\overline{\text{STBY}}$ pin

   When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

(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 standb
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 o
  internal state though the internal of the LSI is initialized by an internal reset caused by
  software standby mode. At this time, the port states that were held in software standby
  are retained even if settings of I/O ports or peripheral modules are set. Subsequently, t
  retained port states are released when the IOKEEP bit is cleared to 0 and operation is
  performed according to the internal settings.

24.8.4 Bϕ Operation after Exit from Deep Software Standby Mode

When the IOKEEP bit is 0, Bϕ output is undefined for a maximum of one cycle immediat
exit from deep software standby mode. At this time, the output state cannot be guaranteed
when the IOKEEP bit is set to 1, Bϕ output is undefined for a maximum of one cycle imm
after the IOKEEP bit is cleared to 0 after deep software standby mode was canceled, and
output state cannot be guaranteed. (See figure 24.3)

However, clock can be normally output by canceling deep software standby mode with th
IOKEEP bit set to 1 and then controlling the Bϕ output with the IOKEEP and PSTOP1 bi
the following procedure.

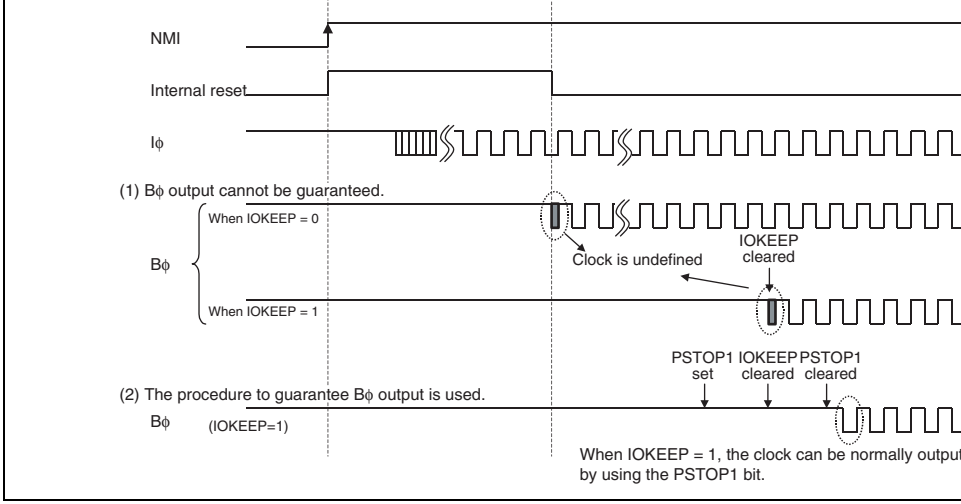


Figure 24.3 Bϕ Operation after Exit from Deep Software Standby Mode

24.8.5 Setting Oscillation Settling Time after Exit from Deep Software Standby

The WTSTS5 to WTSTS0 bits in DPSWCR should be set as follows:

1. Using a crystal resonator

   Specify the WTSTS5 to WTSTS0 bits so that the standby time is at least equal to the oscillation settling time. Table 24.3 shows EXTAL input clock frequencies and the s time according to WTSTS5 to WTSTS0 settings.
2. Using an external clock

   The PLL circuit settling time should be considered. See table 24.3 to set the standby

				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]

: Recommended setting when external clock is in use

: 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.

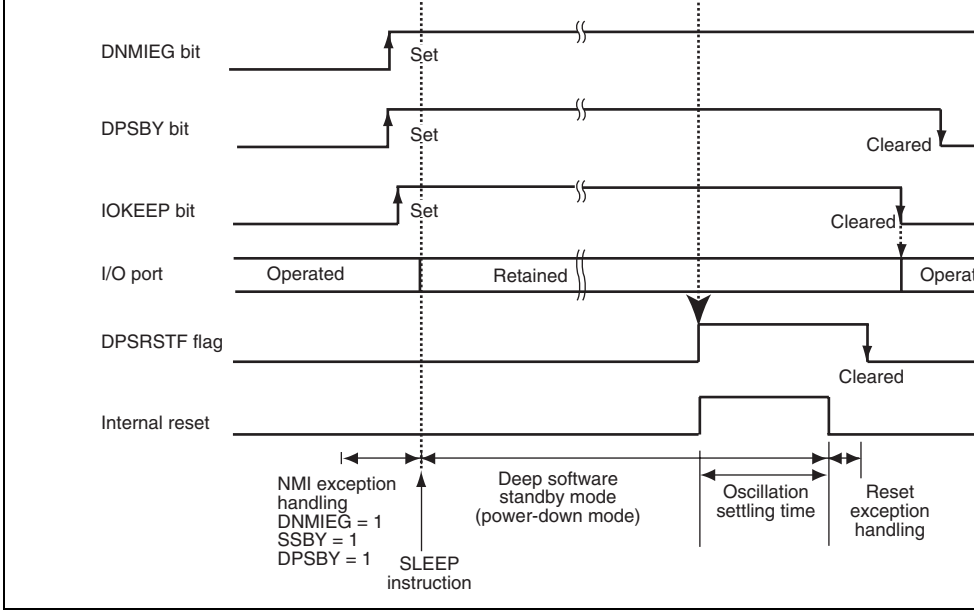


Figure 24.4 Deep Software Standby Mode Application Example (IOKEEP =

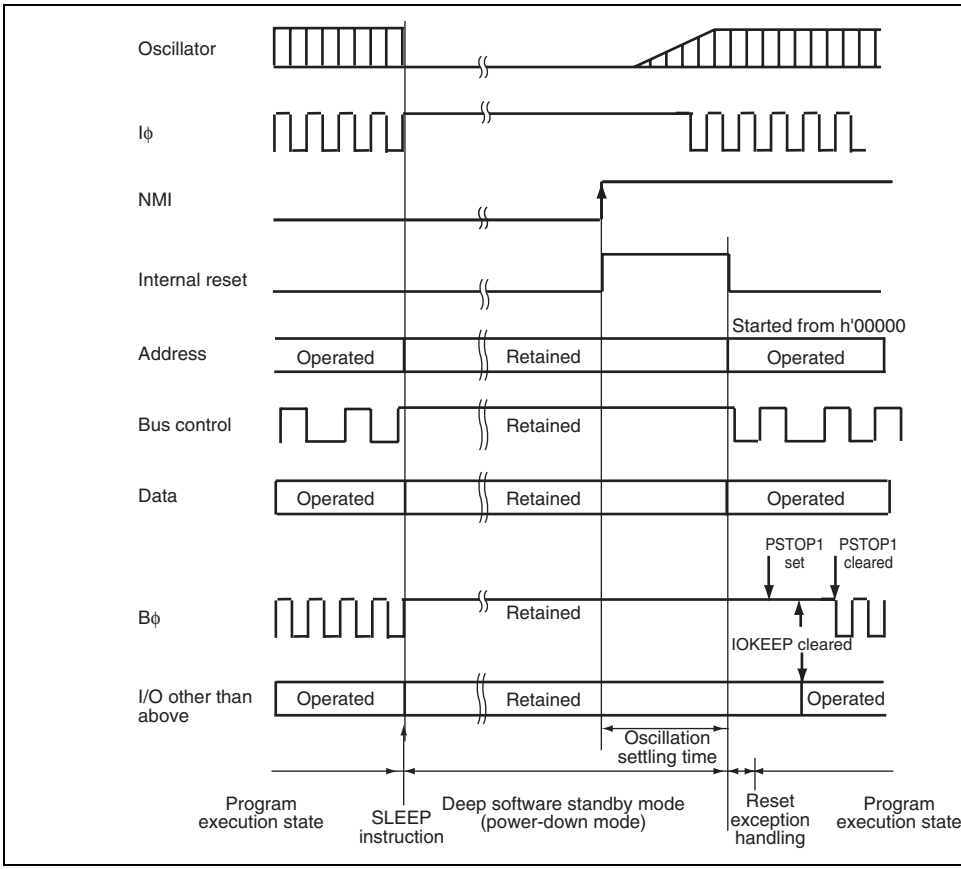


Figure 24.5 Example of Deep Software Standby Mode Operation in External Extended Mode (IOKEEP = OPE = 1)

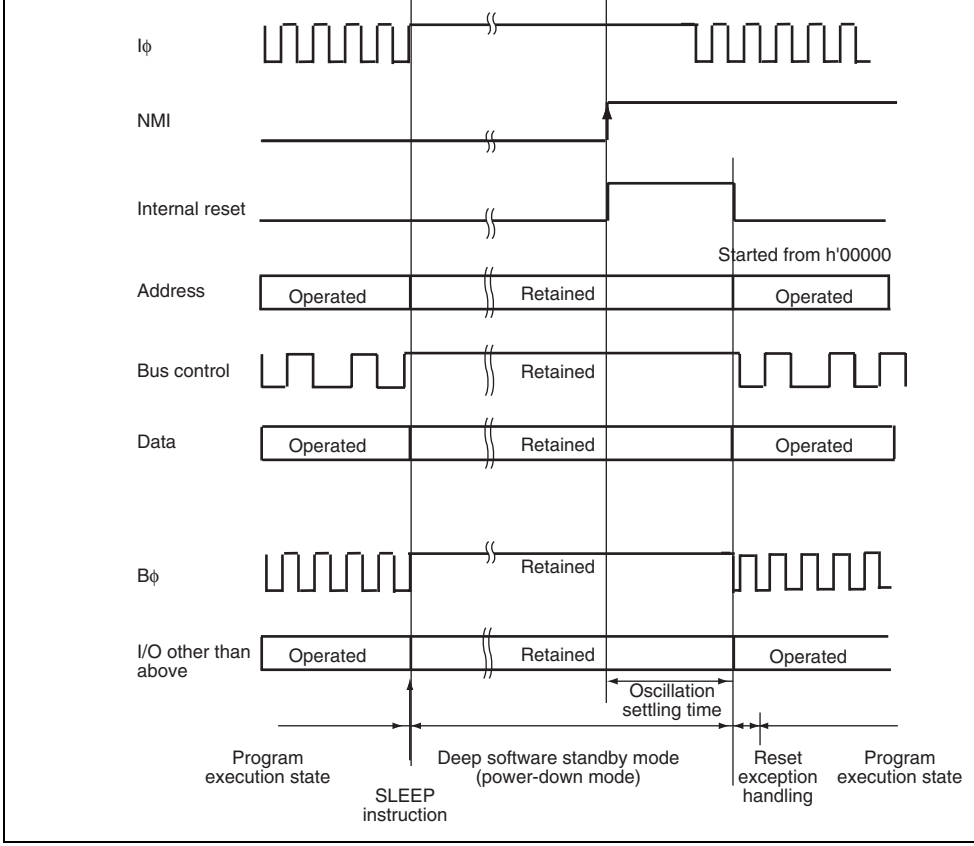


Figure 24.6 Example of Deep Software Standby Mode Operation in External Extended Mode (IOKEEP = 0, OPE = 1)

in Bφ output is also set.

In this flowchart, an interrupt source is checked by reading DPSIFR before the I/O ports
DPSIFR is read after the I/O ports setting, a source flag may be set without intention by
ports setting.

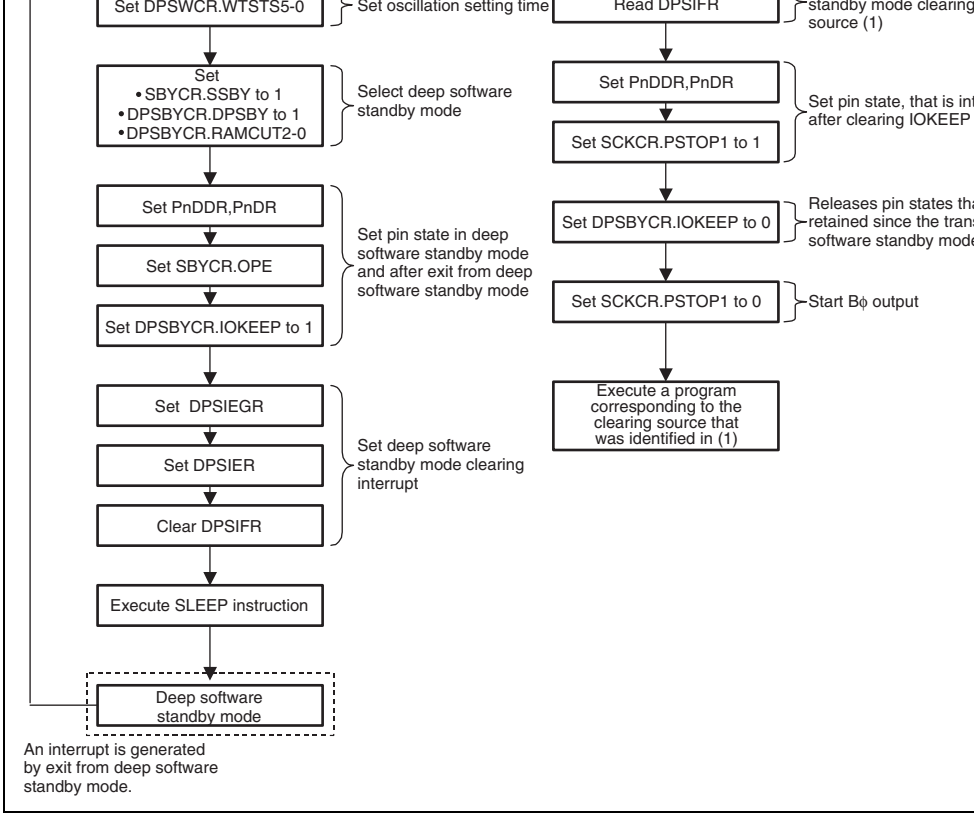


Figure 24.7 Flowchart of Deep Software Standby Mode Operation

Do not change the states of mode pins (MD2 to MD0) while this LSI is in hardware stan

24.9.2 Clearing Hardware Standby Mode

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When th
pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is entered and clock oscillatio
started. Ensure that the $\overline{\text{RES}}$ pin is held low until clock oscillation settles (for details on
oscillation settling time, refer to table 24.2). When the $\overline{\text{RES}}$ pin is subsequently driven h
transition is made to the program execution state via the reset exception handling state.

24.9.3 Hardware Standby Mode Timing

Figure 24.8 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is
hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin hi
waiting for the oscillation settling time, then changing the $\overline{\text{RES}}$ pin from low to high.

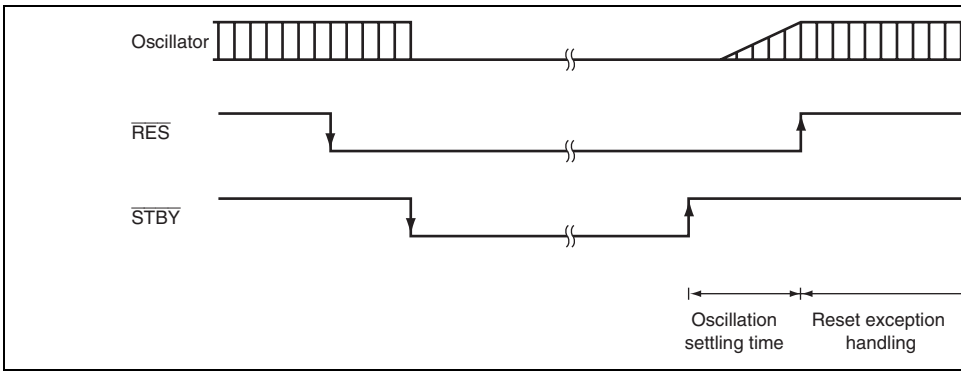


Figure 24.8 Hardware Standby Mode Timing

Timing.

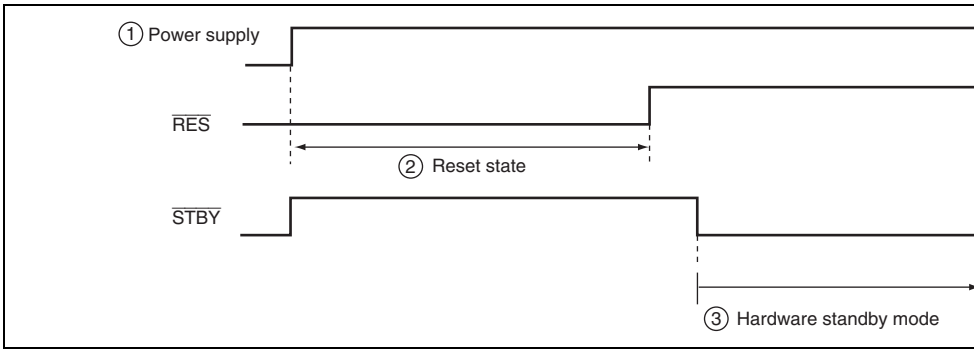


Figure 24.9 Timing Sequence at Power-On

instruction. Transitions to the power-down state are inhibited when sleep instruction exc
handling is initiated, and the CPU immediately starts sleep instruction exception handlin

When a SLEEP instruction is executed while the SLPIE bit is cleared to 0, a transition is
the power-down state. The power-down state is canceled by a canceling factor interrupt
24.10).

When a canceling factor interrupt is generated immediately before the execution of a SL
instruction, exception handling for the interrupt starts. When execution returns from the
service routine, the SLEEP instruction is executed to enter the power-down state. In this
power-down state is not canceled until the next canceling factor interrupt is generated (s
24.11).

When the SLPIE bit is set to 1 in the service routine for a canceling factor interrupt so th
execution of a SLEEP instruction will produce sleep instruction exception handling, the
of the system is as shown in figure 24.12. Even if a canceling factor interrupt is generate
immediately before the SLEEP instruction is executed, sleep instruction exception hand
initiated by execution of the SLEEP instruction. Therefore, the CPU executes the instruc
follows the SLEEP instruction after sleep instruction exception and exception service ro
without shifting to the power-down state.

When the SLPIE bit is set to 1 to start sleep exception handling, clear the SSBY bit in S
0.

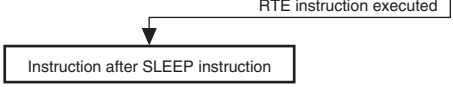


Figure 24.10 When Canceling Factor Interrupt is Generated after SLEEP Instruction Execution

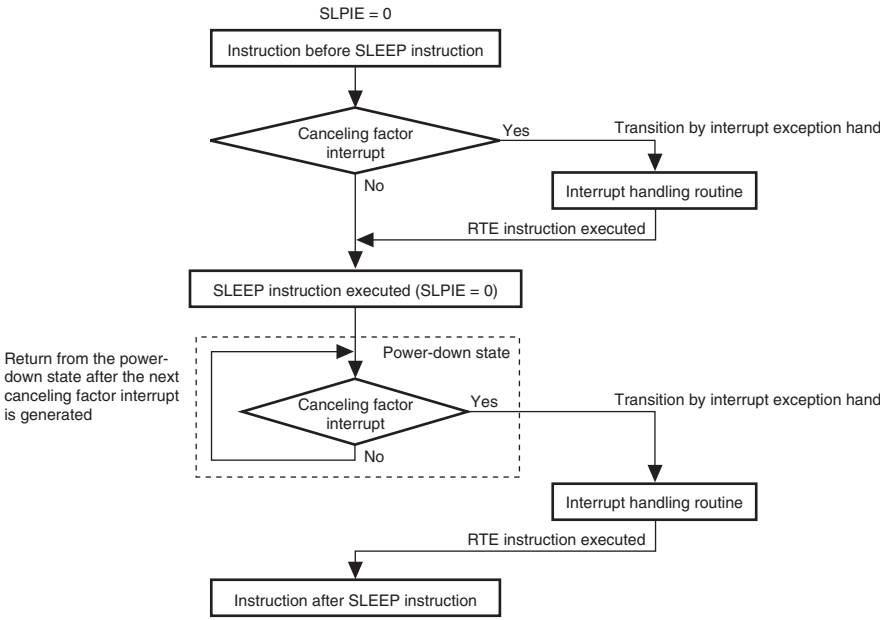


Figure 24.11 When Canceling Factor Interrupt is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Not Initia

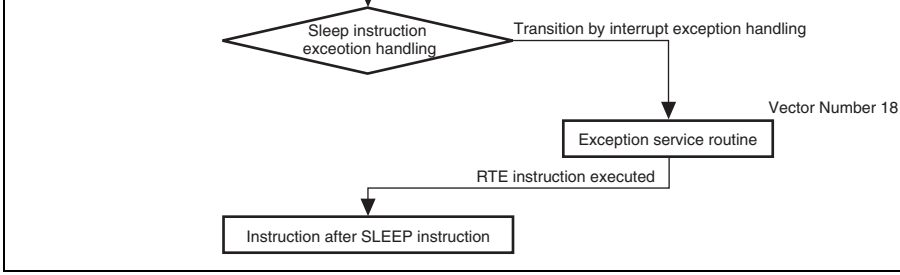


Figure 24.12 When Canceling Factor Interrupt is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Initiat

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.

24.12.3 Module Stop State of DMAC or DTC

Depending on the operating state of the DMAC and DTC, bits MSTPA13 and MSTPA1 be set to 1, respectively. The module stop state setting for the DMAC or DTC should be out only when the DMAC or DTC is not activated.

For details, refer to section 9, DMA Controller (DMAC), and section 10, Data Transfer (DTC).

24.12.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in a module stop state. Consequently stop state is entered when an interrupt has been requested, it will not be possible to clear interrupt source or the DMAC or DTC activation source. Interrupts should therefore be before entering a module stop state.

24.12.5 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 excep
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 a deep software standby mode transition and IRQ0 to IRQ15 interru
occurs, a transition to deep software standby mode can be made without executing the int
execution handling by clearing the SSIn bits in SSIER to 0 beforehand.

24.12.8 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 24.8.4, Bϕ Operation after Exit from Deep Software Standby Mode.

clock. For details, refer to section 8.5.4, 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 correspond 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.

A/D data register F	ADDRF	16	H'FEA6A	A/D	16	3P
A/D data register G	ADDRG	16	H'FEA6C	A/D	16	3P
A/D data register H	ADDRH	16	H'FEA6E	A/D	16	3P
A/D control/status register	ADCSR	8	H'FEA70	A/D	16	3P
A/D control register	ADCR	8	H'FEA71	A/D	16	3P
ΔΣ A/D data register 0	DSADDR0	16	H'FEC00	ΔΣA/D	16	3P
ΔΣ A/D data register 1	DSADDR1	16	H'FEC02	ΔΣA/D	16	3P
ΔΣ A/D data register 2	DSADDR2	16	H'FEC04	ΔΣA/D	16	3P
ΔΣ A/D data register 3	DSADDR3	16	H'FEC06	ΔΣA/D	16	3P
ΔΣ A/D data register 4	DSADDR4	16	H'FEC08	ΔΣA/D	16	3P
ΔΣ A/D data register 5	DSADDR5	16	H'FEC0A	ΔΣA/D	16	3P
ΔΣ A/D offset cancel DAC input 0	DSADOF0	16	H'FEC10	ΔΣA/D	16	3P
ΔΣ A/D offset cancel DAC input 1	DSADOF1	16	H'FEC12	ΔΣA/D	16	3P
ΔΣ A/D offset cancel DAC input 2	DSADOF2	16	H'FEC14	ΔΣA/D	16	3P
ΔΣ A/D offset cancel DAC input 3	DSADOF3	16	H'FEC16	ΔΣA/D	16	3P
ΔΣ A/D control/status register	DSADCSR	16	H'FEC18	ΔΣA/D	16	3P
ΔΣ A/D control register	DSADCR	16	H'FEC1A	ΔΣA/D	16	3P
ΔΣ A/D mode register	DSADMR	8	H'FEC24	ΔΣA/D	16	3P
Break address register AH	BARAH	16	H'FFA00	UBC	16	2Iφ
Break address register AL	BARAL	16	H'FFA02	UBC	16	2Iφ
Break address mask register AH	BAMRAH	16	H'FFA04	UBC	16	2Iφ
Break address mask register AL	BAMRAL	16	H'FFA06	UBC	16	2Iφ

Break address mask register CL	BAMRCL	16	H'FFA16	UBC	16	2
Break address register DH	BARDH	16	H'FFA18	UBC	16	2
Break address register DL	BARDL	16	H'FFA1A	UBC	16	2
Break address mask register DH	BAMRDH	16	H'FFA1C	UBC	16	2
Break address mask register DL	BAMRDL	16	H'FFA1E	UBC	16	2
Break control register A	BRCRA	16	H'FFA28	UBC	16	2
Break control register B	BRCRB	16	H'FFA2C	UBC	16	2
Break control register C	BRCRC	16	H'FFA30	UBC	16	2
Break control register D	BRCRD	16	H'FFA34	UBC	16	2
Timer control register_6	TCR_6	8	H'FFAB0	TMR_6	16	2
Timer control register_7	TCR_7	8	H'FFAB1	TMR_7	16	2
Timer control/status register_6	TCSR_6	8	H'FFAB2	TMR_6	16	2
Timer control/status register_7	TCSR_7	8	H'FFAB3	TMR_7	16	2
Time constant register A_6	TCORA_6	8	H'FFAB4	TMR_6	16	2
Time constant register A_7	TCORA_7	8	H'FFAB5	TMR_7	16	2
Time constant register B_6	TCORB_6	8	H'FFAB6	TMR_6	16	2
Time constant register B_7	TCORB_7	8	H'FFAB7	TMR_7	16	2
Timer counter_6	TCNT_6	8	H'FFAB8	TMR_6	16	2
Timer counter_7	TCNT_7	8	H'FFAB9	TMR_7	16	2
Timer counter control register_6	TCCR_6	8	H'FFABA	TMR_6	16	2
Timer counter control register_7	TCCR_7	8	H'FFABB	TMR_7	16	2
Port 1 data direction register	P1DDR	8	H'FFB80	I/O port	8	2

Port 1 input buffer control register	P1ICR	8	H'FFB90	I/O port	8	2P
Port 2 input buffer control register	P2ICR	8	H'FFB91	I/O port	8	2P
Port 3 input buffer control register	P3ICR	8	H'FFB92	I/O port	8	2P
Port 4 input buffer control register	P4ICR	8	H'FFB93	I/O port	8	2P
Port 5 input buffer control register	P5ICR	8	H'FFB94	I/O port	8	2P
Port 6 input buffer control register	P6ICR	8	H'FFB95	I/O port	8	2P
Port A input buffer control register	PAICR	8	H'FFB99	I/O port	8	2P
Port D input buffer control register	PDICR	8	H'FFB9C	I/O port	8	2P
Port E input buffer control register	PEICR	8	H'FFB9D	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 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 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 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 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 function control register 6	PFCR6	8	H'FFBC6	I/O port	8	2
Port function control register 7	PFCR7	8	H'FFBC7	I/O port	8	2
Port function control register 9	PFCR9	8	H'FFBC9	I/O port	8	2
Port function control register B	PFCRB	8	H'FFBCB	I/O port	8	2
Port function control register C	PFCRC	8	H'FFBCC	I/O port	8	2
Software standby release IRQ enable register	SSIER	16	H'FFBCE	INTC	8	2
Deep standby backup register 0	DPSBKR0	8	H'FFBF0	SYSTEM	8	2
Deep standby backup register 1	DPSBKR1	8	H'FFBF1	SYSTEM	8	2
Deep standby backup register 2	DPSBKR2	8	H'FFBF2	SYSTEM	8	2
Deep standby backup register 3	DPSBKR3	8	H'FFBF3	SYSTEM	8	2
Deep standby backup register 4	DPSBKR4	8	H'FFBF4	SYSTEM	8	2
Deep standby backup register 5	DPSBKR5	8	H'FFBF5	SYSTEM	8	2
Deep standby backup register 6	DPSBKR6	8	H'FFBF6	SYSTEM	8	2
Deep standby backup register 7	DPSBKR7	8	H'FFBF7	SYSTEM	8	2
Deep standby backup register 8	DPSBKR8	8	H'FFBF8	SYSTEM	8	2
Deep standby backup register 9	DPSBKR9	8	H'FFBF9	SYSTEM	8	2
Deep standby backup register 10	DPSBKR10	8	H'FFBFA	SYSTEM	8	2
Deep standby backup register 11	DPSBKR11	8	H'FFBFB	SYSTEM	8	2
Deep standby backup register 12	DPSBKR12	8	H'FFBFC	SYSTEM	8	2
Deep standby backup register 13	DPSBKR13	8	H'FFBFD	SYSTEM	8	2
Deep standby backup register 14	DPSBKR14	8	H'FFBFE	SYSTEM	8	2
Deep standby backup register 15	DPSBKR15	8	H'FFBFF	SYSTEM	8	2

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 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
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 D	IPRD	16	H'FFD46	INTC	16	2Iϕ
Interrupt priority register E	IPRE	16	H'FFD48	INTC	16	2Iϕ
Interrupt priority register F	IPRF	16	H'FFD4A	INTC	16	2Iϕ
Interrupt priority register G	IPRG	16	H'FFD4C	INTC	16	2Iϕ
Interrupt priority register H	IPRH	16	H'FFD4E	INTC	16	2Iϕ
Interrupt priority register I	IPRI	16	H'FFD50	INTC	16	2Iϕ
Interrupt priority register K	IPRK	16	H'FFD54	INTC	16	2Iϕ
Interrupt priority register L	IPRL	16	H'FFD56	INTC	16	2Iϕ
Interrupt priority register P	IPRP	16	H'FFD5E	INTC	16	2Iϕ
Interrupt priority register Q	IPRQ	16	H'FFD60	INTC	16	2Iϕ

Wait control register B	WTCRB	16	H'FFD8A	BSC	16	2
Read strobe timing control register	RDNCR	16	H'FFD8C	BSC	16	2
$\overline{CS}$ assertion period control register	CSACR	16	H'FFD8E	BSC	16	2
Idle control register	IDLCR	16	H'FFD90	BSC	16	2
Bus control register 1	BCR1	16	H'FFD92	BSC	16	2
Bus control register 2	BCR2	8	H'FFD94	BSC	16	2
Endian control register	ENDIANCR	8	H'FFD95	BSC	16	2
SRAM mode control register	SRAMCR	16	H'FFD98	BSC	16	2
Burst ROM interface control register	BROMCR	16	H'FFD9A	BSC	16	2
Address/data multiplexed I/O control register	MPXCR	16	H'FFD9C	BSC	16	2
RAM emulation register	RAMER	8	H'FFD9E	BSC	16	2
Mode control register	MDCR	16	H'FFDC0	SYSTEM	16	2
System control register	SYSCR	16	H'FFDC2	SYSTEM	16	2
System clock control register	SCKCR	16	H'FFDC4	SYSTEM	16	2
Standby control register	SBYCR	16	H'FFDC6	SYSTEM	16	2
Module stop control register A	MSTPCRA	16	H'FFDC8	SYSTEM	16	2
Module stop control register B	MSTPCRB	16	H'FFDCA	SYSTEM	16	2
Module stop control register C	MSTPCRC	16	H'FFDCC	SYSTEM	16	2
Flash code control/status register	FCCS	8	H'FFDE8	FLASH	16	2
Flash program code select register	FPCS	8	H'FFDE9	FLASH	16	2
Flash erase code select register	FECS	8	H'FFDEA	FLASH	16	2
Flash key code register	FKEY	8	H'FFDEC	FLASH	16	2

Deep standby interrupt edge register	DPSIEGR	8	H'FFE74	SYSTEM	8	2Iϕ
Reset status register	RSTSR	8	H'FFE75	SYSTEM	8	2Iϕ
Serial extended mode register_2	SEMR_2	8	H'FFE84	SCI_2	8	2P
Serial mode register_3	SMR_3	8	H'FFE88	SCI_3	8	2P
Bit rate register_3	BRR_3	8	H'FFE89	SCI_3	8	2P
Serial control register_3	SCR_3	8	H'FFE8A	SCI_3	8	2P
Transmit data register_3	TDR_3	8	H'FFE8B	SCI_3	8	2P
Serial status register_3	SSR_3	8	H'FFE8C	SCI_3	8	2P
Receive data register_3	RDR_3	8	H'FFE8D	SCI_3	8	2P
Smart card mode register_3	SCMR_3	8	H'FFE8E	SCI_3	8	2P
Serial mode register_4	SMR_4	8	H'FFE90	SCI_4	8	2P
Bit rate register_4	BRR_4	8	H'FFE91	SCI_4	8	2P
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

Slave address register_1	SAR_1	8	H'FFEBD	IIC2_1	8	2
I^2C bus transmit data register_1	ICDRT_1	8	H'FFEBE	IIC2_1	8	2
I^2C bus receive data register_1	ICDRR_1	8	H'FFEBF	IIC2_1	8	2
Timer control register_2	TCR_2	8	H'FFEC0	TMR_2	16	2
Timer control register_3	TCR_3	8	H'FFEC1	TMR_3	16	2
Timer control/status register_2	TCSR_2	8	H'FFEC2	TMR_2	16	2
Timer control/status register_3	TCSR_3	8	H'FFEC3	TMR_3	16	2
Time constant register A_2	TCORA_2	8	H'FFEC4	TMR_2	16	2
Time constant register A_3	TCORA_3	8	H'FFEC5	TMR_3	16	2
Time constant register B_2	TCORB_2	8	H'FFEC6	TMR_2	16	2
Time constant register B_3	TCORB_3	8	H'FFEC7	TMR_3	16	2
Timer counter_2	TCNT_2	8	H'FFEC8	TMR_2	16	2
Timer counter_3	TCNT_3	8	H'FFEC9	TMR_3	16	2
Timer counter control register_2	TCCR_2	8	H'FFECA	TMR_2	16	2
Timer counter control register_3	TCCR_3	8	H'FFECB	TMR_3	16	2
Timer control register_4	TCR_4	8	H'FFED0	TMR_4	16	2
Timer control register_5	TCR_5	8	H'FFED1	TMR_5	16	2
Timer control/status register_4	TCSR_4	8	H'FFED2	TMR_4	16	2
Timer control/status register_5	TCSR_5	8	H'FFED3	TMR_5	16	2
Time constant register A_4	TCORA_4	8	H'FFED4	TMR_4	16	2
Time constant register A_5	TCORA_5	8	H'FFED5	TMR_5	16	2
Time constant register B_4	TCORB_4	8	H'FFED6	TMR_4	16	2

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	DTC	16	2Iϕ
DTC enable register B	DTCERB	16	H'FFF22	DTC	16	2Iϕ
DTC enable register C	DTCERC	16	H'FFF24	DTC	16	2Iϕ
DTC enable register D	DTCERD	16	H'FFF26	DTC	16	2Iϕ
DTC enable register E	DTCERE	16	H'FFF28	DTC	16	2Iϕ
DTC enable register F	DTCERF	16	H'FFF2A	DTC	16	2Iϕ
DTC enable register G	DTCERG	16	H'FFF2C	DTC	16	2Iϕ
DTC control register	DTCCR	8	H'FFF30	DTC	16	2Iϕ

Port 4 register	PORT4	8	H'FFF43	I/O port	8	2
Port 5 register	PORT5	8	H'FFF44	I/O port	8	2
Port 6 register	PORT6	8	H'FFF45	I/O port	8	2
Port A register	PORTA	8	H'FFF49	I/O port	8	2
Port D register	PORTD	8	H'FFF4C	I/O port	8	2
Port E register	PORTE	8	H'FFF4D	I/O port	8	2
Port F register	PORTF	8	H'FFF4E	I/O port	8	2
Port 1 data register	P1DR	8	H'FFF50	I/O port	8	2
Port 2 data register	P2DR	8	H'FFF51	I/O port	8	2
Port 3 data register	P3DR	8	H'FFF52	I/O port	8	2
Port 6 data register	P6DR	8	H'FFF55	I/O port	8	2
Port A data register	PADR	8	H'FFF59	I/O port	8	2
Port D data register	PDDR	8	H'FFF5C	I/O port	8	2
Port E data register	PEDR	8	H'FFF5D	I/O port	8	2
Port F data register	PFDR	8	H'FFF5E	I/O port	8	2
Serial mode register_2	SMR_2	8	H'FFF60	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FFF61	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FFF62	SCI_2	8	2
Transmit data register_2	TDR_2	8	H'FFF63	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FFF64	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FFF65	SCI_2	8	2
Smart card mode register_2	SCMR_2	8	H'FFF66	SCI_2	8	2

Output data register H	PODRH	8	H'FFF7A	PPG	8	2P
Output data register L	PODRL	8	H'FFF7B	PPG	8	2P
Next data register H*	NDRH	8	H'FFF7C	PPG	8	2P
Next data register L*	NDRL	8	H'FFF7D	PPG	8	2P
Next data register H*	NDRH	8	H'FFF7E	PPG	8	2P
Next data register L*	NDRL	8	H'FFF7F	PPG	8	2P
Serial mode register_0	SMR_0	8	H'FFF80	SCI_0	8	2P
Bit rate register_0	BRR_0	8	H'FFF81	SCI_0	8	2P
Serial control register_0	SCR_0	8	H'FFF82	SCI_0	8	2P
Transmit data register_0	TDR_0	8	H'FFF83	SCI_0	8	2P
Serial status register_0	SSR_0	8	H'FFF84	SCI_0	8	2P
Receive data register_0	RDR_0	8	H'FFF85	SCI_0	8	2P
Smart card mode register_0	SCMR_0	8	H'FFF86	SCI_0	8	2P
Serial mode register_1	SMR_1	8	H'FFF88	SCI_1	8	2P
Bit rate register_1	BRR_1	8	H'FFF89	SCI_1	8	2P
Serial control register_1	SCR_1	8	H'FFF8A	SCI_1	8	2P
Transmit data register_1	TDR_1	8	H'FFF8B	SCI_1	8	2P
Serial status register_1	SSR_1	8	H'FFF8C	SCI_1	8	2P
Receive data register_1	RDR_1	8	H'FFF8D	SCI_1	8	2P
Smart card mode register_1	SCMR_1	8	H'FFF8E	SCI_1	8	2P
Timer control/status register	TCSR	8	H'FFFA4	WDT		2P
Timer counter	TCNT	8	H'FFFA5	WDT		2P

Time constant register B_0	TCORB_0	8	H'FFFB6	TMR_0	16	2
Time constant register B_1	TCORB_1	8	H'FFFB7	TMR_1	16	2
Timer counter_0	TCNT_0	8	H'FFFB8	TMR_0	16	2
Timer counter_1	TCNT_1	8	H'FFFB9	TMR_1	16	2
Timer counter control register_0	TCCR_0	8	H'FFFBA	TMR_0	16	2
Timer counter control register_1	TCCR_1	8	H'FFFBB	TMR_1	16	2
Timer start register	TSTR	8	H'FFFBC	TPU	16	2
Timer synchronous register	TSYR	8	H'FFFBD	TPU	16	2
Timer control register_0	TCR_0	8	H'FFFC0	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FFFC1	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FFFC2	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FFFC3	TPU_0	16	2
Timer interrupt enable register_0	TIER_0	8	H'FFFC4	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FFFC5	TPU_0	16	2
Timer counter_0	TCNT_0	16	H'FFFC6	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FFFC8	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FFFCA	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FFFCC	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FFFCE	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FFFD0	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FFFD1	TPU_1	16	2
Timer I/O control register _1	TIOR_1	8	H'FFFD2	TPU_1	16	2

Timer I/O control register_2	TIOR_2	8	H'FFFE2	TPU_2	16	2P
Timer interrupt enable register_2	TIER_2	8	H'FFFE4	TPU_2	16	2P
Timer status register_2	TSR_2	8	H'FFFE5	TPU_2	16	2P
Timer counter_2	TCNT_2	16	H'FFFE6	TPU_2	16	2P
Timer general register A_2	TGRA_2	16	H'FFFE8	TPU_2	16	2P
Timer general register B_2	TGRB_2	16	H'FFFEA	TPU_2	16	2P
Timer control register_3	TCR_3	8	H'FFFF0	TPU_3	16	2P
Timer mode register_3	TMDR_3	8	H'FFFF1	TPU_3	16	2P
Timer I/O control register H_3	TIORH_3	8	H'FFFF2	TPU_3	16	2P
Timer I/O control register L_3	TIORL_3	8	H'FFFF3	TPU_3	16	2P
Timer interrupt enable register_3	TIER_3	8	H'FFFF4	TPU_3	16	2P
Timer status register_3	TSR_3	8	H'FFFF5	TPU_3	16	2P
Timer counter_3	TCNT_3	16	H'FFFF6	TPU_3	16	2P
Timer general register A_3	TGRA_3	16	H'FFFF8	TPU_3	16	2P
Timer general register B_3	TGRB_3	16	H'FFFFA	TPU_3	16	2P
Timer general register C_3	TGRC_3	16	H'FFFFC	TPU_3	16	2P
Timer general register D_3	TGRD_3	16	H'FFFFE	TPU_3	16	2P

Note: * When the same output trigger is specified for pulse output groups 2 and 3 by t
setting, the NDRH address is H'FFF7C. When different output triggers are spe
NDRH addresses for pulse output groups 2 and 3 are H'FFF7E and H'FFF7C,
respectively. Similarly, when the same output trigger is specified for pulse outp
0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When different ou
triggers are specified, the NDRL addresses for pulse output groups 0 and 1 ar
H'FFF7F and H'FFF7D, respectively.

ADDRC								
ADDRD								
ADDRE								
ADDRF								
ADDRG								
ADDRH								
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS
DSADDR0								
DSADDR1								
DSADDR2								

DSADOF1								
DSADOF2								
DSADOF3								
DSADCSR	ADF	ADIE	ADST	—	SCANE	—	TRGS1	TRGS0
	—	—	CH5	CH4	CH3	CH2	CH1	CH0
DSADCR	CKS	—	GAIN1	GAIN0	—	—	—	—
	DSE	—	—	—	—	—	—	—
DSADMR	BIASE	—	—	—	—	ACK2	ACK1	ACK0
BARAH	BARA31	BARA30	BARA29	BARA28	BARA27	BARA26	BARA25	BARA24
	BARA23	BARA22	BARA21	BARA20	BARA19	BARA18	BARA17	BARA16
BARAL	BARA15	BARA14	BARA13	BARA12	BARA11	BARA10	BARA9	BARA8
	BARA7	BARA6	BARA5	BARA4	BARA3	BARA2	BARA1	BARA0
BAMRAH	BAMRA31	BAMRA30	BAMRA29	BAMRA28	BAMRA27	BAMRA26	BAMRA25	BAMRA24
	BAMRA23	BAMRA22	BAMRA21	BAMRA20	BAMRA19	BAMRA18	BAMRA17	BAMRA16
BAMRAL	BAMRA15	BAMRA14	BAMRA13	BAMRA12	BAMRA11	BAMRA10	BAMRA9	BAMRA8
	BAMRA7	BAMRA6	BAMRA5	BAMRA4	BAMRA3	BAMRA2	BAMRA1	BAMRA0
BARBH	BARB31	BARB30	BARB29	BARB28	BARB27	BARB26	BARB25	BARB24
	BARB23	BARB22	BARB21	BARB20	BARB19	BARB18	BARB17	BARB16

	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	BAMRC
	BAMRC23	BAMRC22	BAMRC21	BAMRC20	BAMRC19	BAMRC18	BAMRC17	BAMRC
BAMRCL	BAMRC15	BAMRC14	BAMRC13	BAMRC12	BAMRC11	BAMRC10	BAMRC9	BAMRC
	BAMRC7	BAMRC6	BAMRC5	BAMRC4	BAMRC3	BAMRC2	BAMRC1	BAMRC
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
BAMRDH	BAMRD31	BAMRD30	BAMRD29	BAMRD28	BAMRD27	BAMRD26	BAMRD25	BAMRD
	BAMRD23	BAMRD22	BAMRD21	BAMRD20	BAMRD19	BAMRD18	BAMRD17	BAMRD
BAMRDL	BAMRD15	BAMRD14	BAMRD13	BAMRD12	BAMRD11	BAMRD10	BAMRD9	BAMRD
	BAMRD7	BAMRD6	BAMRD5	BAMRD4	BAMRD3	BAMRD2	BAMRD1	BAMRD
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	—	—
BRCRD	—	—	DMFCPD	—	CPD2	CPD1	CPD0	—
	—	—	IDD1	IDD0	RWD1	RWD0	—	—

TCORB_7								
TCNT_6								
TCNT_7								
TCCR_6	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_7	—	—	—	—	TMRIS	—	ICKS1	ICKS0
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
P6DDR	—	—	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
PFDDR	—	—	—	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
P1ICR	P17ICR	P16ICR	P15ICR	P14ICR	P13ICR	P12ICR	P11ICR	P10ICR
P2ICR	P27ICR	P26ICR	P25ICR	P24ICR	P23ICR	P22ICR	P21ICR	P20ICR
P3ICR	P37ICR	P36ICR	P35ICR	P34ICR	P33ICR	P32ICR	P31ICR	P30ICR
P4ICR	P47ICR	P46ICR	P45ICR	P44ICR	P43ICR	P42ICR	P41ICR	P40ICR
P5ICR	P57ICR	P56ICR	P55ICR	P54ICR	P53ICR	P52ICR	P51ICR	P50ICR
P6ICR	—	—	P65ICR	P64ICR	P63ICR	P62ICR	P61ICR	P60ICR
PAICR	PA7ICR	PA6ICR	PA5ICR	PA4ICR	PA3ICR	PA2ICR	PA1ICR	PA0ICR
PDICR	PD7ICR	PD6ICR	PD5ICR	PD4ICR	PD3ICR	PD2ICR	PD1ICR	PD0ICR
PEICR	PE7ICR	PE6ICR	PE5ICR	PE4ICR	PE3ICR	PE2ICR	PE1ICR	PE0ICR
PFICR	—	—	—	PF4ICR	PF3ICR	PF2ICR	PF1ICR	PF0ICR

PIICR	PI7ICR	PI6ICR	PI5ICR	PI4ICR	PI3ICR	PI2ICR	PI1ICR	PI0ICR
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
PFPCR	—	—	—	PF4PCR	PF3PCR	PF2PCR	PF1PCR	PF0PCR
PHPCR	PH7PCR	PH6PCR	PH5PCR	PH4PCR	PH3PCR	PH2PCR	PH1PCR	PH0PCR
PIPCR	PI7PCR	PI6PCR	PI5PCR	PI4PCR	PI3PCR	PI2PCR	PI1PCR	PI0PCR
P2ODR	P27ODR	P26ODR	P25ODR	P24ODR	P23ODR	P22ODR	P21ODR	P20ODR
PFODR	—	—	—	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0ODR
PFCR0	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
PFCR1	CS7SA	CS7SB	CS6SA	CS6SB	CS5SA	CS5SB	CS4SA	CS4SB
PFCR2	—	CS2S	BSS	BSE	—	RDWRE	ASOE	—
PFCR4	—	—	—	A20E	A19E	A18E	A17E	A16E
PFCR6	—	LHWROE	—	—	TCLKS	—	—	—
PFCR7	—	—	—	—	DMAS1A	DMAS1B	DMAS0A	DMAS0B
PFCR9	TPUMS5	TPUMS4	TPUMS3A	TPUMS3B	TPUMS2	TPUMS1	TPUMS0A	TPUMS0
PFCRB	—	—	ITS13	ITS12	ITS11	ITS10	ITS9	ITS8
PFCRC	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0
SSIER	SSI15	SSI14	SSI13	SSI12	SSI11	SSI10	SSI9	SSI8
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0
DPSBKR0								
DPSBKR1								
DPSBKR2								
DPSBKR3								

DPSBKR11	
DPSBKR12	
DPSBKR13	
DPSBKR14	
DPSBKR15	
DSAR_0	
DDAR_0	
DOFR_0	
DTCR_0	

	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
DACR_0	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0
DSAR_1								
DDAR_1								
DOFR_1								
DTCR_1								

	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
DACR_1	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0
DMRSR_0								
DMRSR_1								
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
IPRD	—	IPRD14	IPRD13	IPRD12	—	IPRD10	IPRD9	IPRD8
	—	IPRD6	IPRD5	IPRD4	—	IPRD2	IPRD1	IPRD0
IPRE	—	—	—	—	—	IPRE10	IPRE9	IPRE8
	—	—	—	—	—	—	—	—
IPRF	—	—	—	—	—	—	—	—
	—	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

	—	IPRP6	IPRP5	IPRP4	—	IPRP2	IPRP1	IPRP0
IPRQ	—	IPRQ14	IPRQ13	IPRQ12	—	—	—	—
	—	IPRQ6	IPRQ5	IPRQ4	—	IPRQ2	IPRQ1	IPRQ0
IPRR	—	IPRR14	IPRR13	IPRR12	—	—	—	—
	—	—	—	—	—	—	—	—
ISCRH	IRQ15SR	IRQ15SF	IRQ14SR	IRQ14SF	IRQ13SR	IRQ13SF	IRQ12SR	IRQ12SF
	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR	IRQ8SF
ISCRL	IRQ7SR	IRQ7SF	IRQ6SR	IRQ6SF	IRQ5SR	IRQ5SF	IRQ4SR	IRQ4SF
	IRQ3SR	IRQ3SF	IRQ2SR	IRQ2SF	IRQ1SR	IRQ1SF	IRQ0SR	IRQ0SF
DTCVBR								
ABWCR	ABWH7	ABWH6	ABWH5	ABWH4	ABWH3	ABWH2	ABWH1	ABWH0
	ABWL7	ABWL6	ABWL5	ABWL4	ABWL3	ABWL2	ABWL1	ABWL0
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
	—	—	—	—	—	—	—	—
WTCRA	—	W72	W71	W70	—	W62	W61	W60
	—	W52	W51	W50	—	W42	W41	W40
WTCRB	—	W32	W31	W30	—	W22	W21	W20
	—	W12	W11	W10	—	W02	W01	W00
RDNCR	RDN7	RDN6	RDN5	RDN4	RDN3	RDN2	RDN1	RDN0
	—	—	—	—	—	—	—	—

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	—	POSEL1	—	—	ICK2	ICK1	ICK0
	—	PCK2	PCK1	PCK0	—	BCK2	BCK1	BCK0
SBYCR	SSBY	OPE	—	STS4	STS3	STS2	STS1	STS0
	SLPIE	—	—	—	—	—	—	—
MSTPCRA	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9	MSTPA8
	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
MSTPCRB	MSTPB15	MSTPB14	MSTPB13	MSTPB12	MSTPB11	MSTPB10	MSTPB9	MSTPB8
	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
MSTPCRC	MSTPC15	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	MSTPC9	MSTPC8
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0

DPSWCR	—	—	WTSTS5	WTSTS4	WTSTS3	WTSTS2	WTSTS1	WTSTS0
DPSIER	—	—	—	—	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
DPSIFR	DNMIF	—	—	—	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
DPSIEGR	DNMIEG	—	—	—	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0E
RSTSR	DPSRSTF	—	—	—	—	—	—	—
SEMR_2	—	—	—	—	ABCS	ACS2	ACS1	ACS0
SMR_3*1	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP0)	MP (BCP0)	CKS1	CKS0
BRR_3								
SCR_3*1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_3								
SSR_3*1	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_3								
SCMR_3	—	—	—	—	SDIR	SINV	—	SMIF
SMR_4*1	C/$\overline{A}$ (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

ICDRR_0								
ICCRA_1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0
ICCRB_1	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—
ICMR_1	—	WAIT	—	—	BCWP	BC2	BC1	BC0
ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ
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	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_5	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0

TCNT_5								
TCCR_4	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_5	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_4								
TGRA_4								
TGRB_4								
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_5								
TGRA_5								

	DTCEC7	DTCEC6	DTCEC5	DTCEC4	—	—	—	—
DTCERD	—	—	DTCED13	DTCED12	—	—	—	—
	—	—	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0
DTCERE	DTCEE15	DTCEE14	DTCEE13	DTCEE12	—	—	—	—
	—	—	—	—	—	—	—	—
DTCERF	—	—	—	—	—	—	—	—
	—	—	—	—	DTCEF3	DTCEF2	DTCEF1	DTCEF0
DTCERG	DTCEG15	DTCEG14	DTCEG13	DTCEG12	—	—	—	—
	—	—	—	—	—	—	—	—
DTCCR	—	—	—	RRS	RCHNE	—	—	ERR
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—
CPUPCR	CPUPCE	DTCP2	DTCP1	DTCP0	IPSETE	CPUP2	CPUP1	CPUP0
IER	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
ISR	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
PORT1	P17	P16	P15	P14	P13	P12	P11	P10
PORT2	P27	P26	P25	P24	P23	P22	P21	P20
PORT3	P37	P36	P35	P34	P33	P32	P31	P30
PORT4	P47	P46	P45	P44	P43	P42	P41	P40
PORT5	P57	P56	P55	P54	P53	P52	P51	P50
PORT6	—	—	P65	P64	P63	P62	P61	P60

P6DR	—	—	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
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
DADR0								
DADR1								
DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS
PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0

TDR_0								
SSR_0*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_0								
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF
SMR_1*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_1								
SCR_1*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1								
SSR_1*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_1								
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF
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								
TCORB_0								

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
TCNT_1								

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								
TGRC_3								

H1F1F7F and H1F1F7B, respectively.

ADDRE	Initialized	—	—	—	—	Initialized*	Initialized
ADDRF	Initialized	—	—	—	—	Initialized*	Initialized
ADDRG	Initialized	—	—	—	—	Initialized*	Initialized
ADDRH	Initialized	—	—	—	—	Initialized*	Initialized
ADCSR	Initialized	—	—	—	—	Initialized*	Initialized
ADCR	Initialized	—	—	—	—	Initialized*	Initialized
DSADDR0	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADDR1	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADDR2	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADDR3	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADDR4	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADDR5	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADOF0	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADOF1	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADOF2	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADOF3	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADCSR	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADCR	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
DSADMR	Initialized	—	—	—	—	Initialized*	Initialized
BARAH	Initialized	—	—	—	—	Initialized*	Initialized
BARAL	Initialized	—	—	—	—	Initialized*	Initialized
BAMRAH	Initialized	—	—	—	—	Initialized*	Initialized
BAMRAL	Initialized	—	—	—	—	Initialized*	Initialized

BAMRCH	Initialized	—	—	—	—	Initialized*	Initialized
BAMRCL	Initialized	—	—	—	—	Initialized*	Initialized
BARDH	Initialized	—	—	—	—	Initialized*	Initialized
BARDL	Initialized	—	—	—	—	Initialized*	Initialized
BAMRDH	Initialized	—	—	—	—	Initialized*	Initialized
BAMRDL	Initialized	—	—	—	—	Initialized*	Initialized
BRCRA	Initialized	—	—	—	—	Initialized*	Initialized
BRCRB	Initialized	—	—	—	—	Initialized*	Initialized
BRCRC	Initialized	—	—	—	—	Initialized*	Initialized
BRCRD	Initialized	—	—	—	—	Initialized*	Initialized
TCR_6	Initialized	—	—	—	—	Initialized*	Initialized
TCR_7	Initialized	—	—	—	—	Initialized*	Initialized
TCSR_6	Initialized	—	—	—	—	Initialized*	Initialized
TCSR_7	Initialized	—	—	—	—	Initialized*	Initialized
TCORA_6	Initialized	—	—	—	—	Initialized*	Initialized
TCORA_7	Initialized	—	—	—	—	Initialized*	Initialized
TCORB_6	Initialized	—	—	—	—	Initialized*	Initialized
TCORB_7	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_6	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_7	Initialized	—	—	—	—	Initialized*	Initialized
TCCR_6	Initialized	—	—	—	—	Initialized*	Initialized
TCCR_7	Initialized	—	—	—	—	Initialized*	Initialized

PEDDR	Initialized	—	—	—	—	Initialized*	Initialized
PFDDR	Initialized	—	—	—	—	Initialized*	Initialized
P1ICR	Initialized	—	—	—	—	Initialized*	Initialized
P2ICR	Initialized	—	—	—	—	Initialized*	Initialized
P3ICR	Initialized	—	—	—	—	Initialized*	Initialized
P4ICR	Initialized	—	—	—	—	Initialized*	Initialized
P5ICR	Initialized	—	—	—	—	Initialized*	Initialized
P6ICR	Initialized	—	—	—	—	Initialized*	Initialized
PAICR	Initialized	—	—	—	—	Initialized*	Initialized
PDICR	Initialized	—	—	—	—	Initialized*	Initialized
PEICR	Initialized	—	—	—	—	Initialized*	Initialized
PFICR	Initialized	—	—	—	—	Initialized*	Initialized
PORTH	Initialized	—	—	—	—	Initialized*	Initialized
PORTI	Initialized	—	—	—	—	Initialized*	Initialized
PHDR	Initialized	—	—	—	—	Initialized*	Initialized
PIDR	Initialized	—	—	—	—	Initialized*	Initialized
PHDDR	Initialized	—	—	—	—	Initialized*	Initialized
PIDDR	Initialized	—	—	—	—	Initialized*	Initialized
PHICR	Initialized	—	—	—	—	Initialized*	Initialized
PIICR	Initialized	—	—	—	—	Initialized*	Initialized
PDPCR	Initialized	—	—	—	—	Initialized*	Initialized
PEPCR	Initialized	—	—	—	—	Initialized*	Initialized
PFPCR	Initialized	—	—	—	—	Initialized*	Initialized

PFCR2	Initialized	—	—	—	—	Initialized*	Initialized
PFCR4	Initialized	—	—	—	—	Initialized*	Initialized
PFCR6	Initialized	—	—	—	—	Initialized*	Initialized
PFCR7	Initialized	—	—	—	—	Initialized*	Initialized
PFCR9	Initialized	—	—	—	—	Initialized*	Initialized
PFCRB	Initialized	—	—	—	—	Initialized*	Initialized
PFCRC	Initialized	—	—	—	—	Initialized*	Initialized
SSIER	Initialized	—	—	—	—	Initialized*	Initialized
DPSBKR0	Initialized	—	—	—	—	Retained	Initialized
DPSBKR1	Initialized	—	—	—	—	Retained	Initialized
DPSBKR2	Initialized	—	—	—	—	Retained	Initialized
DPSBKR3	Initialized	—	—	—	—	Retained	Initialized
DPSBKR4	Initialized	—	—	—	—	Retained	Initialized
DPSBKR5	Initialized	—	—	—	—	Retained	Initialized
DPSBKR6	Initialized	—	—	—	—	Retained	Initialized
DPSBKR7	Initialized	—	—	—	—	Retained	Initialized
DPSBKR8	Initialized	—	—	—	—	Retained	Initialized
DPSBKR9	Initialized	—	—	—	—	Retained	Initialized
DPSBKR10	Initialized	—	—	—	—	Retained	Initialized
DPSBKR11	Initialized	—	—	—	—	Retained	Initialized
DPSBKR12	Initialized	—	—	—	—	Retained	Initialized
DPSBKR13	Initialized	—	—	—	—	Retained	Initialized
DPSBKR14	Initialized	—	—	—	—	Retained	Initialized

DMDR_0	Initialized	—	—	—	—	Initialized*	Initialized
DACR_0	Initialized	—	—	—	—	Initialized*	Initialized
DSAR_1	Initialized	—	—	—	—	Initialized*	Initialized
DDAR_1	Initialized	—	—	—	—	Initialized*	Initialized
DOFR_1	Initialized	—	—	—	—	Initialized*	Initialized
DTCR_1	Initialized	—	—	—	—	Initialized*	Initialized
DBSR_1	Initialized	—	—	—	—	Initialized*	Initialized
DMDR_1	Initialized	—	—	—	—	Initialized*	Initialized
DACR_1	Initialized	—	—	—	—	Initialized*	Initialized
DMRSR_0	Initialized	—	—	—	—	Initialized*	Initialized
DMRSR_1	Initialized	—	—	—	—	Initialized*	Initialized
IPRA	Initialized	—	—	—	—	Initialized*	Initialized
IPRB	Initialized	—	—	—	—	Initialized*	Initialized
IPRC	Initialized	—	—	—	—	Initialized*	Initialized
IPRD	Initialized	—	—	—	—	Initialized*	Initialized
IPRE	Initialized	—	—	—	—	Initialized*	Initialized
IPRF	Initialized	—	—	—	—	Initialized*	Initialized
IPRG	Initialized	—	—	—	—	Initialized*	Initialized
IPRH	Initialized	—	—	—	—	Initialized*	Initialized
IPRI	Initialized	—	—	—	—	Initialized*	Initialized
IPRK	Initialized	—	—	—	—	Initialized*	Initialized
IPRL	Initialized	—	—	—	—	Initialized*	Initialized
IPRP	Initialized	—	—	—	—	Initialized*	Initialized

ASTCR	Initialized	—	—	—	—	Initialized*	Initialized
WTCRA	Initialized	—	—	—	—	Initialized*	Initialized
WTCRB	Initialized	—	—	—	—	Initialized*	Initialized
RDNCR	Initialized	—	—	—	—	Initialized*	Initialized
CSACR	Initialized	—	—	—	—	Initialized*	Initialized
IDLCR	Initialized	—	—	—	—	Initialized*	Initialized
BCR1	Initialized	—	—	—	—	Initialized*	Initialized
BCR2	Initialized	—	—	—	—	Initialized*	Initialized
ENDIANCR	Initialized	—	—	—	—	Initialized*	Initialized
SRAMCR	Initialized	—	—	—	—	Initialized*	Initialized
BROMCR	Initialized	—	—	—	—	Initialized*	Initialized
MPXCR	Initialized	—	—	—	—	Initialized*	Initialized
RAMER	Initialized	—	—	—	—	Initialized*	Initialized
MDCR	Initialized	—	—	—	—	Initialized*	Initialized
SYSCR	Initialized	—	—	—	—	Initialized*	Initialized
SCKCR	Initialized	—	—	—	—	Initialized*	Initialized
SBYCR	Initialized	—	—	—	—	Initialized*	Initialized
MSTPCRA	Initialized	—	—	—	—	Initialized*	Initialized
MSTPCRB	Initialized	—	—	—	—	Initialized*	Initialized
MSTPCRC	Initialized	—	—	—	—	Initialized*	Initialized
FCCS	Initialized	—	—	—	—	Initialized*	Initialized
FPCS	Initialized	—	—	—	—	Initialized*	Initialized
FECS	Initialized	—	—	—	—	Initialized*	Initialized

DPSIFR	Initialized	—	—	—	—	—	Initialized
DPSIEGR	Initialized	—	—	—	—	—	Initialized
RSTSR	Initialized	—	—	—	—	—	Initialized
SEMR_2	Initialized	—	—	—	—	Initialized*	Initialized
SMR_3	Initialized	—	—	—	—	Initialized*	Initialized
BRR_3	Initialized	—	—	—	—	Initialized*	Initialized
SCR_3	Initialized	—	—	—	—	Initialized*	Initialized
TDR_3	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SSR_3	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
RDR_3	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SCMR_3	Initialized	—	—	—	—	Initialized*	Initialized
SMR_4	Initialized	—	—	—	—	Initialized*	Initialized
BRR_4	Initialized	—	—	—	—	Initialized*	Initialized
SCR_4	Initialized	—	—	—	—	Initialized*	Initialized
TDR_4	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SSR_4	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
RDR_4	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SCMR_4	Initialized	—	—	—	—	Initialized*	Initialized
ICCRA_0	Initialized	—	—	—	—	Initialized*	Initialized
ICCRB_0	Initialized	—	—	—	—	Initialized*	Initialized
ICMR_0	Initialized	—	—	—	—	Initialized*	Initialized
ICIER_0	Initialized	—	—	—	—	Initialized*	Initialized
ICSR_0	Initialized	—	—	—	—	Initialized*	Initialized

ICIER_1	Initialized	—	—	—	—	Initialized*	Initialized
ICSR_1	Initialized	—	—	—	—	Initialized*	Initialized
SAR_1	Initialized	—	—	—	—	Initialized*	Initialized
ICDRT_1	Initialized	—	—	—	—	Initialized*	Initialized
ICDRR_1	Initialized	—	—	—	—	Initialized*	Initialized
TCR_2	Initialized	—	—	—	—	Initialized*	Initialized
TCR_3	Initialized	—	—	—	—	Initialized*	Initialized
TCSR_2	Initialized	—	—	—	—	Initialized*	Initialized
TCSR_3	Initialized	—	—	—	—	Initialized*	Initialized
TCORA_2	Initialized	—	—	—	—	Initialized*	Initialized
TCORA_3	Initialized	—	—	—	—	Initialized*	Initialized
TCORB_2	Initialized	—	—	—	—	Initialized*	Initialized
TCORB_3	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_2	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_3	Initialized	—	—	—	—	Initialized*	Initialized
TCCR_2	Initialized	—	—	—	—	Initialized*	Initialized
TCCR_3	Initialized	—	—	—	—	Initialized*	Initialized
TCR_4	Initialized	—	—	—	—	Initialized*	Initialized
TCR_5	Initialized	—	—	—	—	Initialized*	Initialized
TCSR_4	Initialized	—	—	—	—	Initialized*	Initialized
TCSR_5	Initialized	—	—	—	—	Initialized*	Initialized
TCORA_4	Initialized	—	—	—	—	Initialized*	Initialized
TCORA_5	Initialized	—	—	—	—	Initialized*	Initialized

TCR_4	Initialized	—	—	—	—	Initialized*	Initialized
TMDR_4	Initialized	—	—	—	—	Initialized*	Initialized
TIOR_4	Initialized	—	—	—	—	Initialized*	Initialized
TIER_4	Initialized	—	—	—	—	Initialized*	Initialized
TSR_4	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_4	Initialized	—	—	—	—	Initialized*	Initialized
TGRA_4	Initialized	—	—	—	—	Initialized*	Initialized
TGRB_4	Initialized	—	—	—	—	Initialized*	Initialized
TCR_5	Initialized	—	—	—	—	Initialized*	Initialized
TMDR_5	Initialized	—	—	—	—	Initialized*	Initialized
TIOR_5	Initialized	—	—	—	—	Initialized*	Initialized
TIER_5	Initialized	—	—	—	—	Initialized*	Initialized
TSR_5	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_5	Initialized	—	—	—	—	Initialized*	Initialized
TGRA_5	Initialized	—	—	—	—	Initialized*	Initialized
TGRB_5	Initialized	—	—	—	—	Initialized*	Initialized
DTCERA	Initialized	—	—	—	—	Initialized*	Initialized
DTCERB	Initialized	—	—	—	—	Initialized*	Initialized
DTCERC	Initialized	—	—	—	—	Initialized*	Initialized
DTCERD	Initialized	—	—	—	—	Initialized*	Initialized
DTCERE	Initialized	—	—	—	—	Initialized*	Initialized
DTCERF	Initialized	—	—	—	—	Initialized*	Initialized
DTCERG	Initialized	—	—	—	—	Initialized*	Initialized

PORT2	—	—	—	—	—	—	—
PORT3	—	—	—	—	—	—	—
PORT4	—	—	—	—	—	—	—
PORT5	—	—	—	—	—	—	—
PORT6	—	—	—	—	—	—	—
PORTA	—	—	—	—	—	—	—
PORTD	—	—	—	—	—	—	—
PORTE	—	—	—	—	—	—	—
PORTF	—	—	—	—	—	—	—
P1DR	Initialized	—	—	—	—	Initialized*	Initialized
P2DR	Initialized	—	—	—	—	Initialized*	Initialized
P3DR	Initialized	—	—	—	—	Initialized*	Initialized
P6DR	Initialized	—	—	—	—	Initialized*	Initialized
PADR	Initialized	—	—	—	—	Initialized*	Initialized
PDDR	Initialized	—	—	—	—	Initialized*	Initialized
PEDR	Initialized	—	—	—	—	Initialized*	Initialized
PFDR	Initialized	—	—	—	—	Initialized*	Initialized
SMR_2	Initialized	—	—	—	—	Initialized*	Initialized
BRR_2	Initialized	—	—	—	—	Initialized*	Initialized
SCR_2	Initialized	—	—	—	—	Initialized*	Initialized
TDR_2	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SSR_2	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
RDR_2	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized

NDERH	Initialized	—	—	—	—	Initialized*	Initialized
NDERL	Initialized	—	—	—	—	Initialized*	Initialized
PODRH	Initialized	—	—	—	—	Initialized*	Initialized
PODRL	Initialized	—	—	—	—	Initialized*	Initialized
NDRH	Initialized	—	—	—	—	Initialized*	Initialized
NDRL	Initialized	—	—	—	—	Initialized*	Initialized
SMR_0	Initialized	—	—	—	—	Initialized*	Initialized
BRR_0	Initialized	—	—	—	—	Initialized*	Initialized
SCR_0	Initialized	—	—	—	—	Initialized*	Initialized
TDR_0	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SSR_0	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
RDR_0	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SCMR_0	Initialized	—	—	—	—	Initialized*	Initialized
SMR_1	Initialized	—	—	—	—	Initialized*	Initialized
BRR_1	Initialized	—	—	—	—	Initialized*	Initialized
SCR_1	Initialized	—	—	—	—	Initialized*	Initialized
TDR_1	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SSR_1	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
RDR_1	Initialized	—	Initialized	Initialized	Initialized	Initialized*	Initialized
SCMR_1	Initialized	—	—	—	—	Initialized*	Initialized
TCSR	Initialized	—	—	—	—	Initialized*	Initialized
TCNT	Initialized	—	—	—	—	Initialized*	Initialized
RSTCSR	Initialized	—	—	—	—	Initialized*	Initialized

TCORB_0	Initialized	—	—	—	—	Initialized*	Initialized
TCORB_1	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_0	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_1	Initialized	—	—	—	—	Initialized*	Initialized
TCCR_0	Initialized	—	—	—	—	Initialized*	Initialized
TCCR_1	Initialized	—	—	—	—	Initialized*	Initialized
TSTR	Initialized	—	—	—	—	Initialized*	Initialized
TSYR	Initialized	—	—	—	—	Initialized*	Initialized
TCR_0	Initialized	—	—	—	—	Initialized*	Initialized
TMDR_0	Initialized	—	—	—	—	Initialized*	Initialized
TIORH_0	Initialized	—	—	—	—	Initialized*	Initialized
TIORL_0	Initialized	—	—	—	—	Initialized*	Initialized
TIER_0	Initialized	—	—	—	—	Initialized*	Initialized
TSR_0	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_0	Initialized	—	—	—	—	Initialized*	Initialized
TGRA_0	Initialized	—	—	—	—	Initialized*	Initialized
TGRB_0	Initialized	—	—	—	—	Initialized*	Initialized
TGRC_0	Initialized	—	—	—	—	Initialized*	Initialized
TGRD_0	Initialized	—	—	—	—	Initialized*	Initialized
TCR_1	Initialized	—	—	—	—	Initialized*	Initialized
TMDR_1	Initialized	—	—	—	—	Initialized*	Initialized
TIOR_1	Initialized	—	—	—	—	Initialized*	Initialized
TIER_1	Initialized	—	—	—	—	Initialized*	Initialized

TIOR_2	Initialized	—	—	—	—	Initialized*	Initialized
TIER_2	Initialized	—	—	—	—	Initialized*	Initialized
TSR_2	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_2	Initialized	—	—	—	—	Initialized*	Initialized
TGRA_2	Initialized	—	—	—	—	Initialized*	Initialized
TGRB_2	Initialized	—	—	—	—	Initialized*	Initialized
TCR_3	Initialized	—	—	—	—	Initialized*	Initialized
TMDR_3	Initialized	—	—	—	—	Initialized*	Initialized
TIORH_3	Initialized	—	—	—	—	Initialized*	Initialized
TIORL_3	Initialized	—	—	—	—	Initialized*	Initialized
TIER_3	Initialized	—	—	—	—	Initialized*	Initialized
TSR_3	Initialized	—	—	—	—	Initialized*	Initialized
TCNT_3	Initialized	—	—	—	—	Initialized*	Initialized
TGRA_3	Initialized	—	—	—	—	Initialized*	Initialized
TGRB_3	Initialized	—	—	—	—	Initialized*	Initialized
TGRC_3	Initialized	—	—	—	—	Initialized*	Initialized
TGRD_3	Initialized	—	—	—	—	Initialized*	Initialized

Note: * This register is not initialized in deep software standby mode, though, initialize
clearing the deep software standby mode. It is because a reset exception hand
carried out by an internal reset when the deep software standby mode is clear

Input voltage (except ports 4 and 5)	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (port 4)	V_{in}	–0.3 to $AV_{CC}P$ +0.3
Input voltage (port 5)	V_{in}	–0.3 to AV_{CC} +0.3
Reference power supply voltage (V_{ref})	V_{ref}	–0.3 to AV_{CC} +0.3
Reference power supply voltage ($AV_{ref}T$)	$AV_{ref}T$	–0.3 to $AV_{cc}A$ +0.3
Analog power supply voltage (AV_{cc})	AV_{CC}	–0.3 to +4.6
Analog power supply voltage ($AV_{cc}P$, $AV_{cc}A$, $AV_{cc}D$)	$AV_{cc}P$ = $AV_{cc}A$ = $AV_{cc}D$	–0.3 to +4.6
Analog input voltage (AN0 to 7)	V_{AN}	–0.3 to AV_{CC} +0.3
Analog input voltage (ANDS0 to 3, ANDS4P/4N, ANDS5P/5N)	V_{AN}	–0.3 to $AV_{cc}P$ +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 when programming or erasing the flash memory is
Regular specifications: 0 to +75°C
Wide-range specifications: 0 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Cond
Schmitt trigger input voltage	$\overline{IRQ}$ input pin, TPU input pin, TMR input pin, port 2, port 3	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.06$	—	—	V	
	Port 5*2	VT^-	$AV_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$AV_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$AV_{CC} \times 0.06$	—	—	V	
Input high voltage (except Schmitt trigger input pin)	MD, $\overline{RES}$, $\overline{STBY}$, EMLE, NMI	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Port 4		$AV_{cc}P \times 0.7$	—	$AV_{cc}P + 0.3$	V	
	Port 5		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V	
	Other input pins		$V_{cc} \times 0.7$	—	$V_{cc} + 0.3$	V	
Input low voltage (except Schmitt trigger input pin)	MD, $\overline{RES}$, $\overline{STBY}$, EMLE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, NMI		–0.3	—	$V_{CC} \times 0.2$	V	
	Port 4		–0.3	—	$AV_{cc}P \times 0.2$	V	
	Port 5		–0.3	—	$AV_{cc} \times 0.2$	V	
	Other input pins		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	$V_{CC} - 0.5$	—	—	V	I_{OH} = -
			$V_{CC} - 1.0$	—	—		I_{OH} = -
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1
	Port 3		—	—	1.0		I_{OL} = 1

(off state)							
Input pull-up MOS current	Ports D to F, H, I	$-I_p$	10	—	300	µA	V_{CC} = 3.0 V_{in} = 0 V
Input capacitance	ANDS[3:0], ANDS4P,4N ANDS5P,5N	C_{in}	—	—	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	Other input pins		—	—	15	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
Supply current*3	Normal operation	I_{CC}*5	—	48	76	mA	Iϕ = Bϕ = Pϕ = 25 M
			—	46	66		Iϕ = Bϕ = 35 MHz*4
	Sleep mode		—	39	45		Iϕ = Bϕ = Pϕ = 25 M
			—	38	43		Iϕ = Bϕ = 35 MHz*4
	Software standby mode		—	0.5	1	mA	Ta ≤ 50°C
			—	—	3.2		Ta > 50°C
	Deep software standby mode*4 (RAM retained)		—	19	55	µA	Ta ≤ 50°C
			—	—	190		Ta > 50°C
	Deep software standby mode*4 (RAM power supply stopped)		—	4	7		Ta ≤ 50°C
			—	—	16		Ta > 50°C
	Hardware standby mode*4		—	3	5		Ta ≤ 50°C
			—	—	15		Ta > 50°C
	All-module-clock-stop mode*6		—	22	29	mA	

Notes: 1. When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins s be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

When the $\Delta\Sigma$ A/D converter is not used, the $AV_{cc}P$, $AV_{cc}A$, $AV_{cc}D$, $AV_{ref}T$, $AV_{ss}P$ $AV_{ss}D$ and $AV_{ref}B$ pins should not be open.

Connect the $AV_{cc}P$, $AV_{cc}A$, $AV_{cc}D$ and $AV_{ref}T$ pins to V_{cc}, and the $AV_{ss}P$, $AV_{ss}A$, $AV_{ref}B$ pins to V_{ss}.

2. The case where port 5 is used as $\overline{IRQ0}$ to $\overline{IRQ7}$.
3. Supply current values are for $V_{IH}min = V_{CC} - 0.5$ V and $V_{IL}max = 0.5$ V with all pins unloaded and all input pull-up MOSs in the off state.
4. The values are for $V_{IH}min = V_{CC} \times 0.9$ and $V_{IL}max = 0.3$ V.
5. I_{CC} depends on f as follows.
   Normal operation:
   $I_{CC}max = 16\ (mA) + 1.20\ (mA/MHz) \times f\ \ (I\phi = B\phi,\ P\phi = 1/2I\phi)$
   $I_{CC}max = 16\ (mA) + 1.44\ (mA/MHz) \times f\ \ (I\phi = B\phi = P\phi)$
   Sleep mode:
   $I_{CC}max = 16\ (mA) + 0.57\ (mA/MHz) \times f\ \ (I\phi = B\phi,\ P\phi = 1/2I\phi)$
   $I_{CC}max = 16\ (mA) + 0.78\ (mA/MHz) \times f\ \ (I\phi = B\phi = P\phi)$
6. The values are for reference.
7. This can be applied when the $\overline{RES}$ pin is held low at power-on.

Permissible output low current (per pin)	Port 3	I_{OL}	—	—	10
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	80
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40

Caution: To protect the LSI's reliability, do not exceed the output current values in table

Note: * When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins sh be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

When the $\Delta\Sigma$ A/D converter is not used, the $AV_{cc}P$, $AV_{cc}A$, $AV_{cc}D$, $AV_{ref}T$, $AV_{ss}P$, $AV_{ss}D$ and $AV_{ref}B$ pins should not be open.

Connect the $AV_{cc}P$, $AV_{cc}A$, $AV_{cc}D$ and $AV_{ref}T$ pins to V_{cc}, and the $AV_{ss}P$, $AV_{ss}A$, A $AV_{ref}B$ pins to V_{ss}.

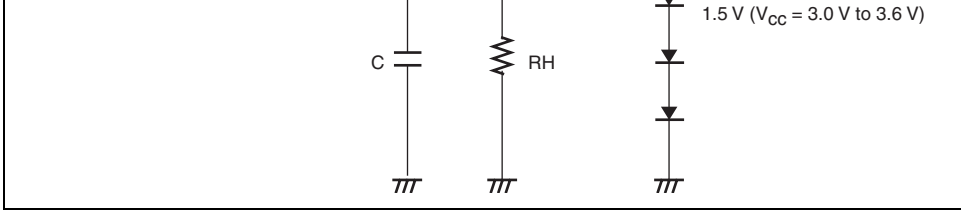


Figure 26.1 Output Load Circuit

(1) Clock Timing

Table 26.4 Clock Timing

Conditions: Vcc = PLLVcc = 3.0 to 3.6 V, AVcc = 3.0 to 3.6 V, AVccP = AVccA = AV
to 3.6 V, Vref = 3.0 V to AVcc, AVrefT = AVccA, Vss = PLLVss = AVss =
AVssA = AVssD = AVrefB = 0 V
Iϕ = 8 to 50 MHz, Bϕ = 8 to 50 MHz, Pϕ = 8 to 35 MHz,
Ta = –20 to + 75 °C (regular specifications),
Ta = –40 to + 85 °C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit.	Test Con
Clock cycle time	t_{cyc}	20.0	125	ns	Figure 26
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 26

External clock falling time	t_{EXf}	—	5	ns

(2) Control Signal Timing

Table 26.5 Control Signal Timing

Conditions: Vcc = PLLVcc = 3.0 to 3.6 V, AVcc = 3.0 to 3.6 V, AVccP = AVccA = AVc
to 3.6 V, Vref = 3.0 V to AVcc, AVrefT = AVccA, Vss = PLLVss = AVss =
AVssA = AVssD = AVrefB = 0 V
Iϕ = 8 to 50 MHz, Ta = –20 to + 75 °C (regular specifications),
Ta = –40 to + 85 °C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Con
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figure 26
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 26
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (after leaving software standby mode or deep software standby mode)	t_{NMIW}	200	—	ns	
$\overline{\text{IRQ}}$ setup time	t_{IRQS}	150	—	ns	
$\overline{\text{IRQ}}$ hold time	t_{IRQH}	10	—	ns	
$\overline{\text{IRQ}}$ pulse width (after leaving software standby mode or deep software standby mode)	t_{IRQW}	200	—	ns	

Address delay time	t_{AD}	—	15	ns	Figure 26.21
Address setup time 1	t_{AS1}	$0.5 \times t_{cyc} - 8$	—	ns	
Address setup time 2	t_{AS2}	$1.0 \times t_{cyc} - 8$	—	ns	
Address setup time 3	t_{AS3}	$1.5 \times t_{cyc} - 8$	—	ns	
Address setup time 4	t_{AS4}	$2.0 \times t_{cyc} - 8$	—	ns	
Address hold time 1	t_{AH1}	$0.5 \times t_{cyc} - 8$	—	ns	
Address hold time 2	t_{AH2}	$1.0 \times t_{cyc} - 8$	—	ns	
Address hold time 3	t_{AH3}	$1.5 \times t_{cyc} - 8$	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	15	ns	
$\overline{AS}$ delay time	t_{ASD}	—	15	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	15	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	15	ns	
Read data setup time 1	t_{RDS1}	15	—	ns	
Read data setup time 2	t_{RDS2}	15	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	0	—	ns	
Read data access time 2	t_{AC2}	—	$1.5 \times t_{cyc} - 20$	ns	
Read data access time 4	t_{AC4}	—	$2.5 \times t_{cyc} - 20$	ns	
Read data access time 5	t_{AC5}	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 6	t_{AC6}	—	$2.0 \times t_{cyc} - 20$	ns	

(from address) 5	t_{AAS}			t_{cyc}	
$\overline{WR}$ delay time 1	t_{WRD1}	—	15	ns	Figures 26.21
$\overline{WR}$ delay time 2	t_{WRD2}	—	15	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 13$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	t_{WDD}	—	20	ns	
Write data setup time 1	t_{WDS1}	$0.5 \times t_{cyc} - 13$	—	ns	
Write data setup time 2	t_{WDS2}	$1.0 \times t_{cyc} - 13$	—	ns	
Write data setup time 3	t_{WDS3}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 1	t_{WDH1}	$0.5 \times t_{cyc} - 8$	—	ns	
Write data hold time 3	t_{WDH3}	$1.5 \times t_{cyc} - 8$	—	ns	
Byte control delay time	t_{UBD}	—	15	ns	Figures 26.15
Byte control pulse width 1	t_{UBW1}	—	$1.0 \times t_{cyc} - 15$	ns	Figure
Byte control pulse width 2	t_{UBW2}	—	$2.0 \times t_{cyc} - 15$	ns	Figure
Multiplexed address delay time 1	t_{MAD1}	—	15	ns	Figures 26.19
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	

RD/WR delay time	t_{RWD}	15	—	ns	26.15

(4) DMAC Timing

Table 26.7 DMAC Timing

Conditions: Vcc = PLLVcc = 3.0 to 3.6 V, AVcc = 3.0 to 3.6 V, AVccP = AVccA = AV
to 3.6 V, Vref = 3.0 V to AVcc, AVrefT = AVccA, Vss = PLLVss = AVss =
AVssA = AVssD = AVrefB = 0 V
Bϕ = 8 to 50 MHz, Ta = –20 to + 75 °C (regular specifications),
Ta = –40 to + 85 °C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Cond
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	20	—	ns	Figure
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	5	—	ns	
$\overline{\text{TEND}}$ delay time	t_{TED}	—	20	ns	Figure
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	20	ns	Figure
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	20	ns	26.25

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}	
WDT	Overflow output delay time		t_{WOVD}	—	40	ns	Figure 2

	Receive data hold time (clocked synchronous)	t_{RXH}	40	—	ns	
A/D converter	Trigger input setup time	t_{TRGS}	30	—	ns	Figure
ΔΣ A/D converter	Trigger input setup time	t_{DSTRS}	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	
	Retransmit 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	

Conversion time	5.33	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.5	LSB
Offset error	—	—	±3.5	LSB
Full-scale error	—	—	±3.5	LSB
Quantization error	—	±0.5	—	LSB
Absolute accuracy	—	—	±4.0	LSB

26.1.5 D/A Conversion Characteristics

Table 26.10 D/A Conversion Characteristics

Conditions: Vcc = PLLVcc = 3.0 to 3.6 V, AVcc = 3.0 to 3.6 V, Vref = 3.0 V to AVcc, V
PLLVss = AVss = 0 V, Pϕ = 8 to 35 MHz,
Ta = –20 to + 75 °C (regular specifications),
Ta = –40 to + 85 °C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit	Test Conditi
Resolution	8	8	8	Bit	
Conversion time	—	—	10	µs	20 pF capaci
Absolute accuracy	—	±2.0	±3.0	LSB	2 MΩ resistiv
	—	—	±2.0	LSB	4 MΩ resistiv

Number of states for conversion (in fos)			—	286	—
Oversampling frequency (fos)			2.5	—	3.3
Conversion time			86.67	—	114.40
Input frequency			—	—	1.0
Input voltage range (with respect to input offset voltage)	Single-ended	×8 gain mode	—	—	±1/12 × AVrefT
		×4 gain mode	—	—	±1/6 × AVrefT
		×2 gain mode	—	—	±1/3 × AVrefT
		×1 gain mode	—	—	±1/2 × AVrefT
	Differential	×8 gain mode	—	—	±1/24 × AVrefT
		×4 gain mode	—	—	±1/12 × AVrefT
		×2 gain mode	—	—	±1/6 × AVrefT
		×1 gain mode	—	—	±1/3 × AVrefT
Input offset voltage	×8 gain mode		1/4 × AVrefT	—	3/4 × AVrefT
	×4 gain mode		1/4 × AVrefT	—	3/4 × AVrefT
	×2 gain mode		—	1/2 × AVrefT	—
	×1 gain mode		—	1/2 × AVrefT	—

Note: * It is recommended to use a 1%-error resistor as the external biasing resistor on the REXT pin.

	ended		AVrefT	AVrefT	up to 1 kHz			
		×4	±1/6 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	fos= 3.125 MHz	—	±2	—
		×2	±1/3 × AVrefT	1/2 × AVrefT		—	±2	—
		×1	±1/2 × AVrefT	1/2 × AVrefT		—	±2	—
	Differential	×8	±1/24 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	Sine wave input= up to 1 kHz	—	±2	—
		×4	±1/12 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	fos= 3.125 MHz	—	±2	—
		×2	±1/6 × AVrefT	1/2 × AVrefT		—	±2	—
		×1	±1/3 × AVrefT	1/2 × AVrefT		—	±2	—
SNDR	Single-ended	×8	±1/12 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	Sine wave input= up to 1 kHz	—	83	—
		×4	±1/6 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	fos= 3.125 MHz	—	84	—
		×2	±1/3 × AVrefT	1/2 × AVrefT		—	87	—
		×1	±1/2 × AVrefT	1/2 × AVrefT		—	84	—
	Differential	×8	±1/24 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	Sine wave input= up to 1 kHz	—	85	—
		×4	±1/12 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	fos= 3.125 MHz	—	86	—
		×2	±1/6 × AVrefT	1/2 × AVrefT		—	88	—
		×1	±1/3 × AVrefT	1/2 × AVrefT		—	88	—

			AVrefT	AVrefT	fos = 3.125 MHz		
		×4	±1/12 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT		—	±1.0
		×2	±1/6 × AVrefT	1/2 × AVrefT		—	±0.8
		×1	±1/3 × AVrefT	1/2 × AVrefT		—	±0.8
INL	Single-ended	×8	±1/12 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	Ramp wave input fos = 3.125 MHz	—	±5.0
		×4	±1/6 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT		—	±4.5
		×2	±1/3 × AVrefT	1/2 × AVrefT		—	±3.0
		×1	±1/2 × AVrefT	1/2 × AVrefT		—	±3.0
	Differential	×8	±1/24 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	Ramp wave input fos = 3.125 MHz	—	±4.0
		×4	±1/12 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT		—	±3.5
		×2	±1/6 × AVrefT	1/2 × AVrefT		—	±2.5
		×1	±1/3 × AVrefT	1/2 × AVrefT		—	±2.5

	AVrefT	AVrefT			
×4	±1/12 × AVrefT	1/4 × AVrefT to 3/4 × AVrefT	40	—	—
×2	±1/6 × AVrefT	1/2 × AVrefT	80	—	—
×1	±1/3 × AVrefT	1/2 × AVrefT	160	—	—

Note: * It is recommended to use a 1%-error resistor as the external biasing resistor c
on the REXT pin.

Table 26.11 ΔΣ A/D Conversion Characteristics (Reference Value) (3)

Conditions: Vcc = PLLVcc = 3.0 to 3.6 V, AVccP = AVccA = AVccD = 3.0 to 3.6 V,
AVrefT = AVccA, Vss = PLLVss = AVssP = AVssA = AVssD = AV
0 V, Pϕ = 8 to 35 MHz, Ta = –20 to +75 °C (regular specifications),
Ta = –40 to +85 °C (wide-range specifications), REXT = 51K Ω*

Item	Conditions	min	typ	ma
Offset cancellation resolution	×8 gain mode	—	10	—
Offset cancellation absolute accuracy	×8 gain mode	—	±2.0	—

Note: * It is recommended to use a 1%-error resistor as the external biasing resistor c
on the REXT pin.

(standby state)					
Stabilization time (AVCM)	Stabilization time from the point the BIASE bit is set	AVCM = 0.1 μF	20	—	—

Note: * It is recommended to use a 1%-error resistor as the external biasing resistor on the REXT pin.

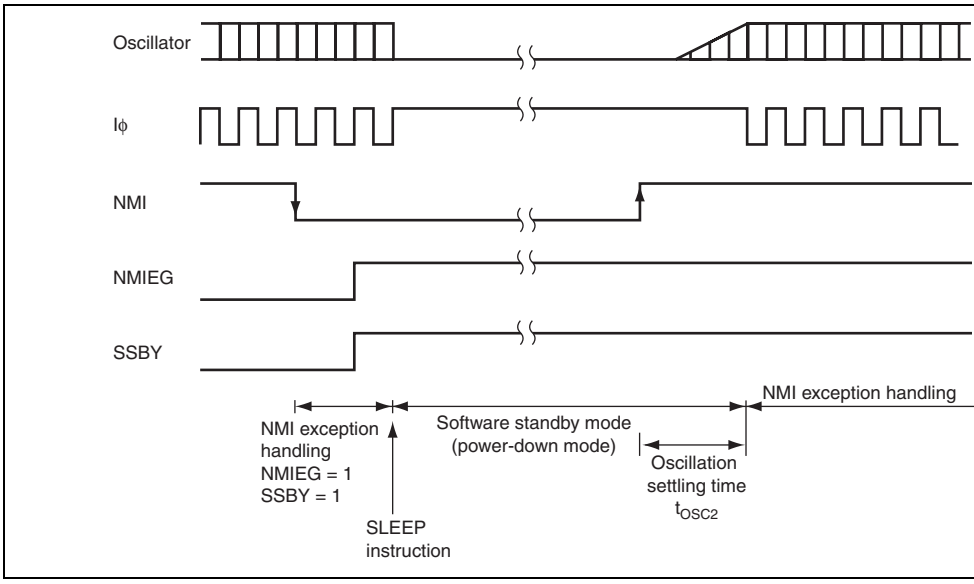


Figure 26.3 Oscillation Settling Timing after Software Standby Mode

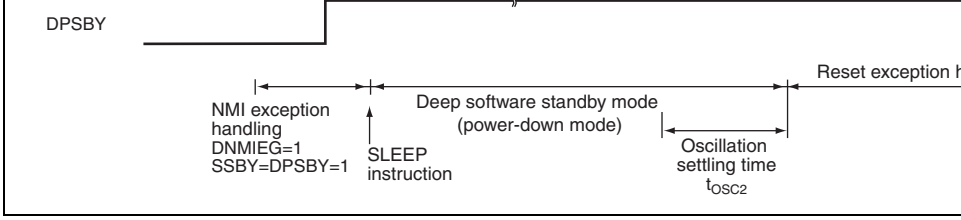


Figure 26.4 Oscillation Settling Timing after Deep Software Standby Mod

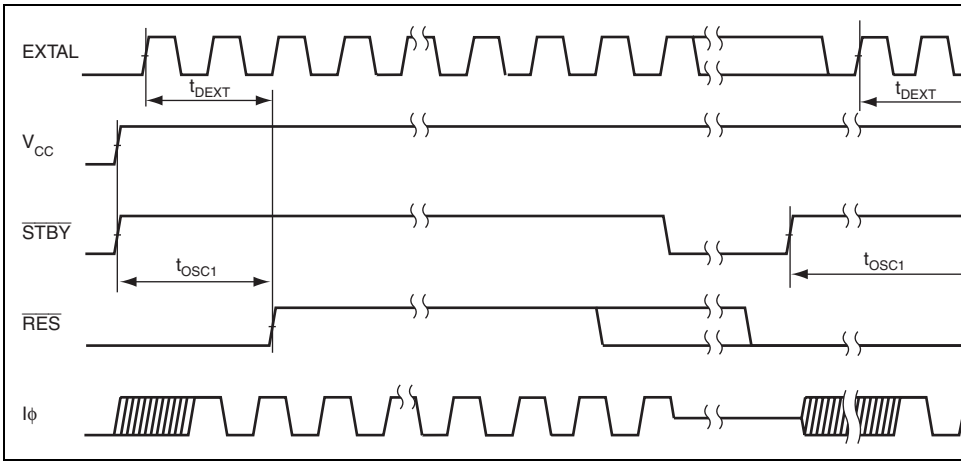


Figure 26.5 Oscillation Settling Timing

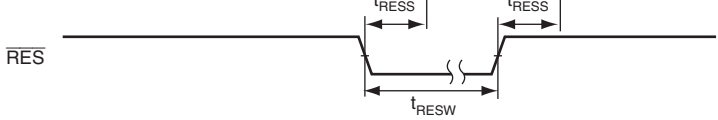
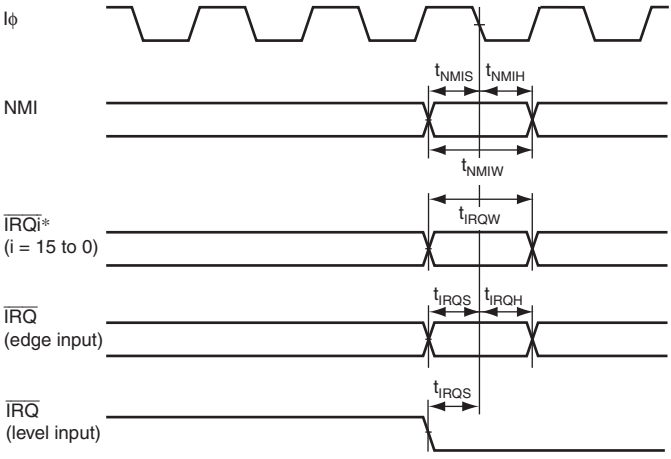


Figure 26.7 Reset Input Timing



Note: * SSIER must be set to cancel software standby mode.
DIRQE[3:0] must be set to cancel deep software standby mode.

Figure 26.8 Interrupt Input Timing

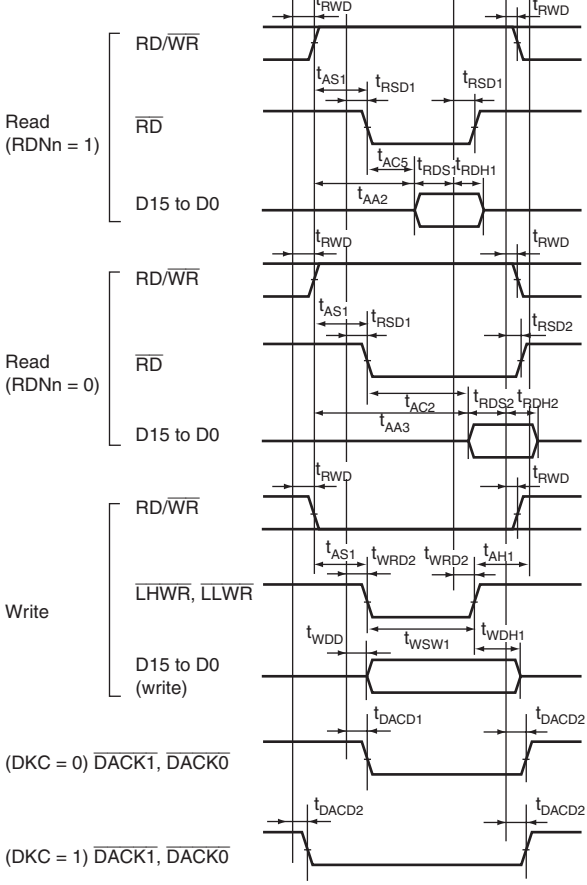


Figure 26.9 Basic Bus Timing: 2-State Access

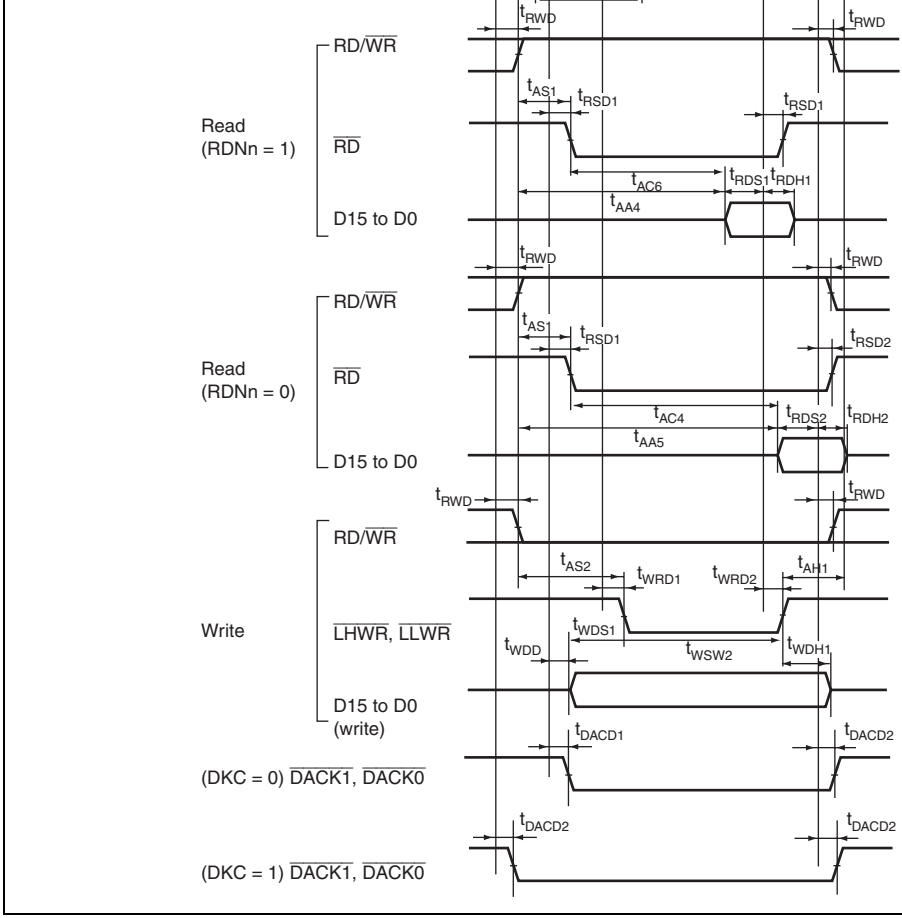


Figure 26.10 Basic Bus Timing: 3-State Access

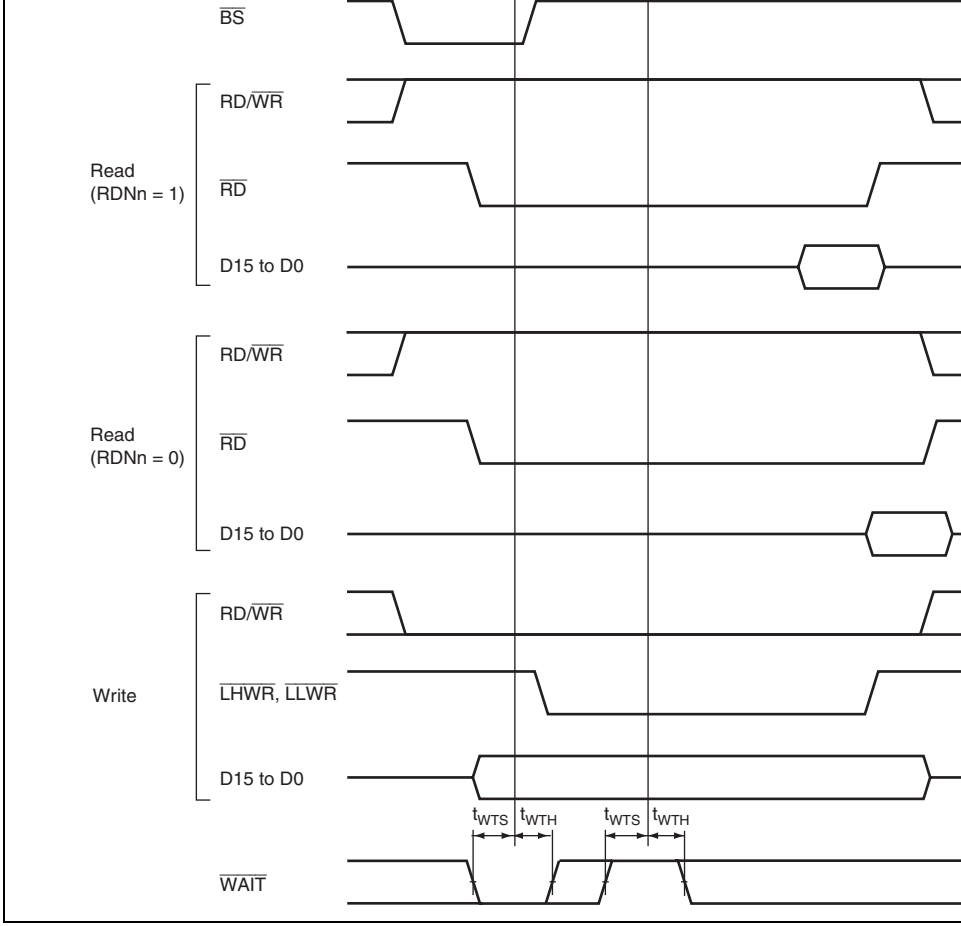


Figure 26.11 Basic Bus Timing: Three-State Access, One Wait

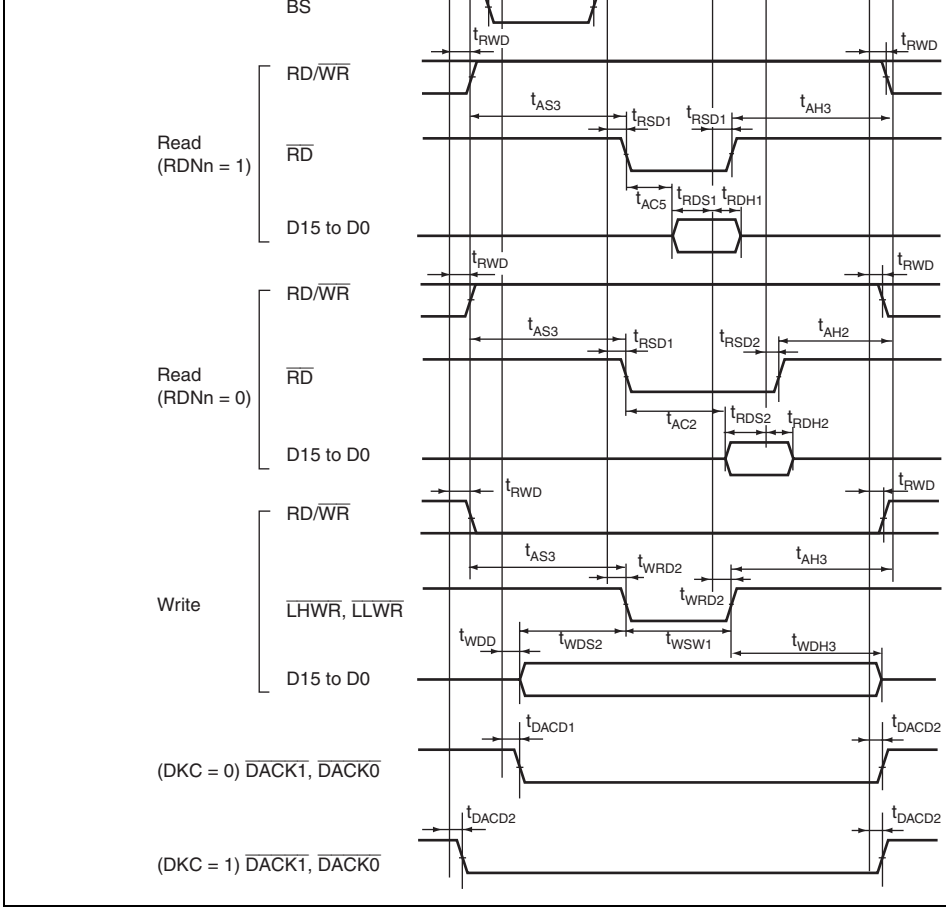


Figure 26.12 Basic Bus Timing: 2-State Access ($\overline{CS}$ Assertion Period Extende

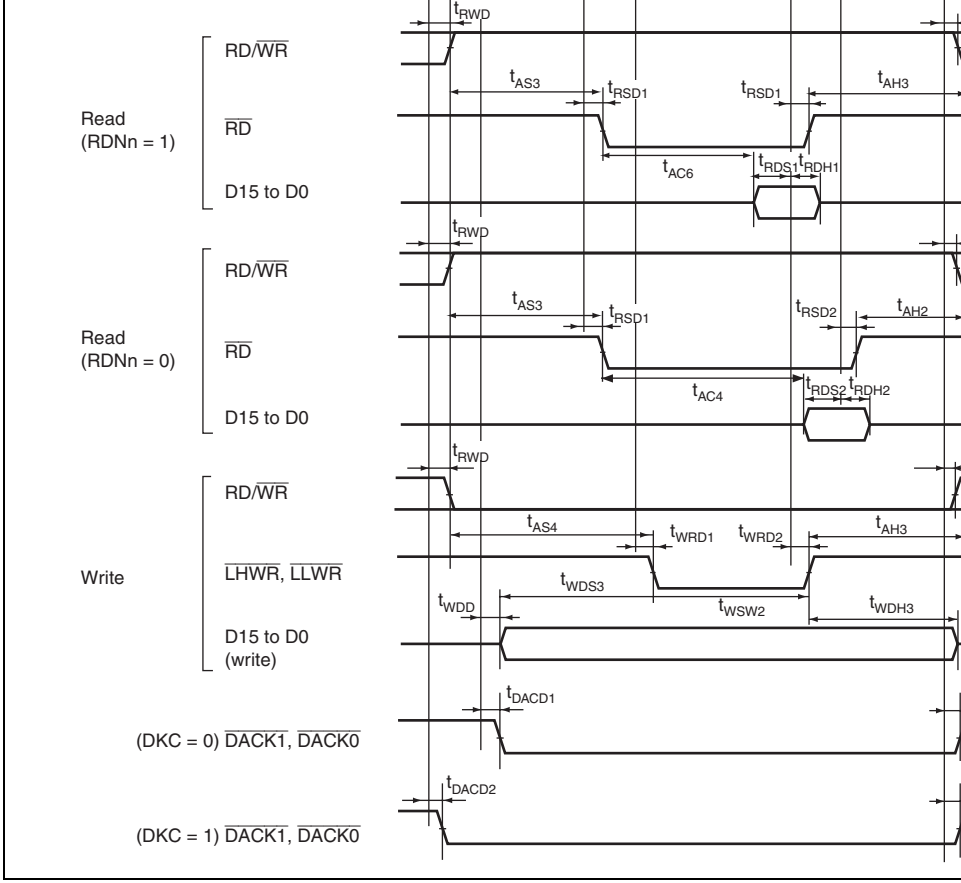


Figure 26.13 Basic Bus Timing: 3-State Access (CS Assertion Period Extend

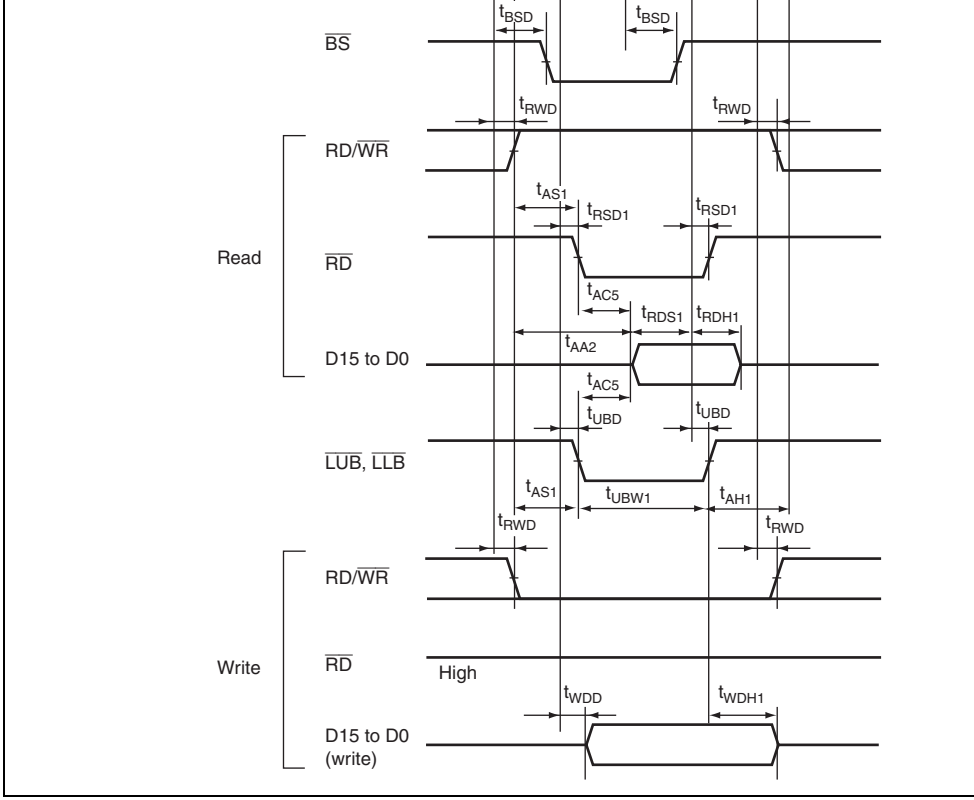


Figure 26.14 Byte Control SRAM: 2-State Read/Write Access

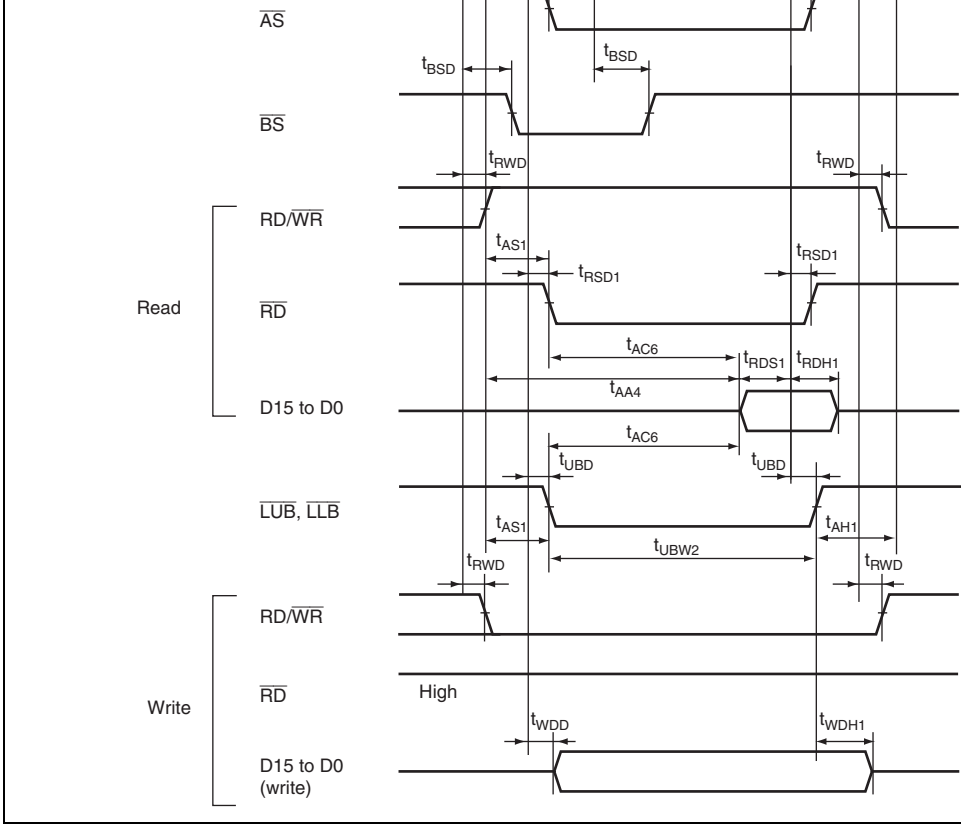


Figure 26.15 Byte Control SRAM: 3-State Read/Write Access

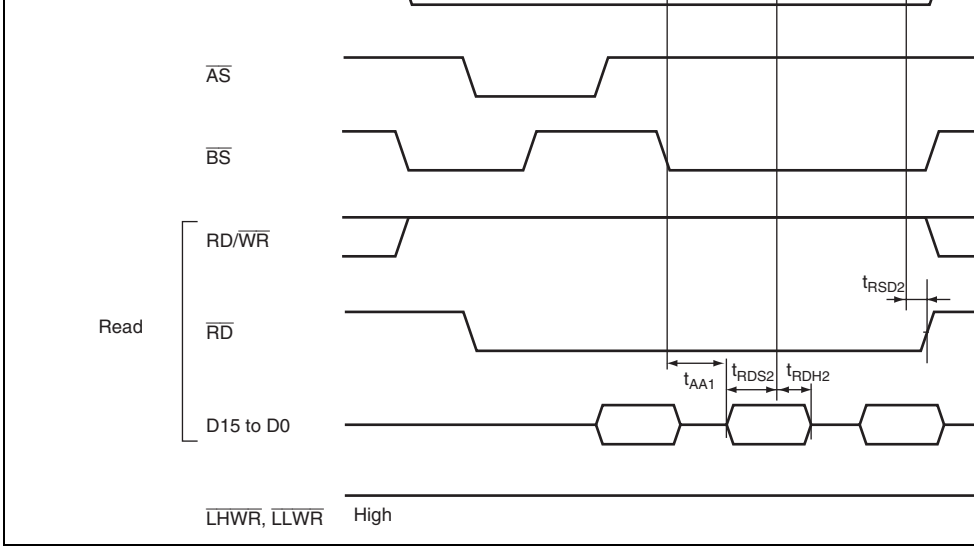


Figure 26.16 Burst ROM Access Timing: 1-State Burst Access

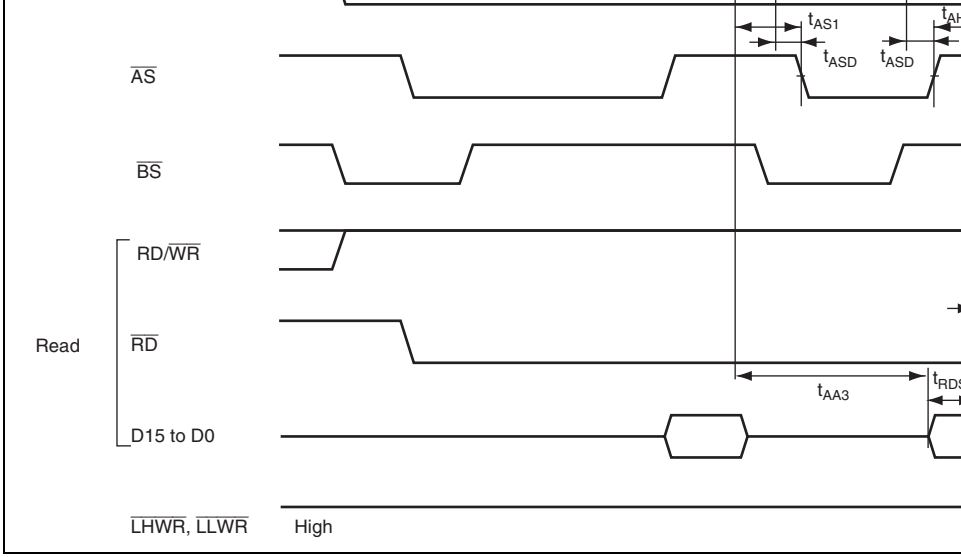


Figure 26.17 Burst ROM Access Timing: 2-State Burst Access

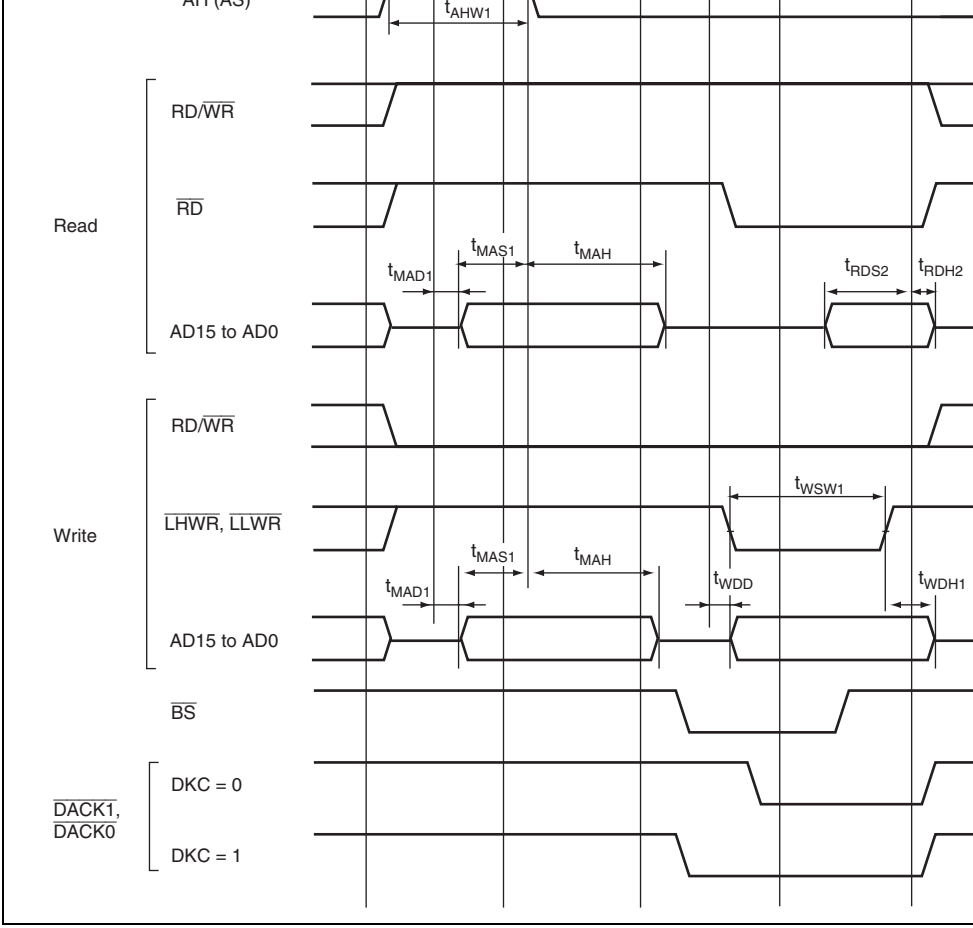


Figure 26.18 Address/Data Multiplexed Access Timing (No Wait) (Basic, 4-State

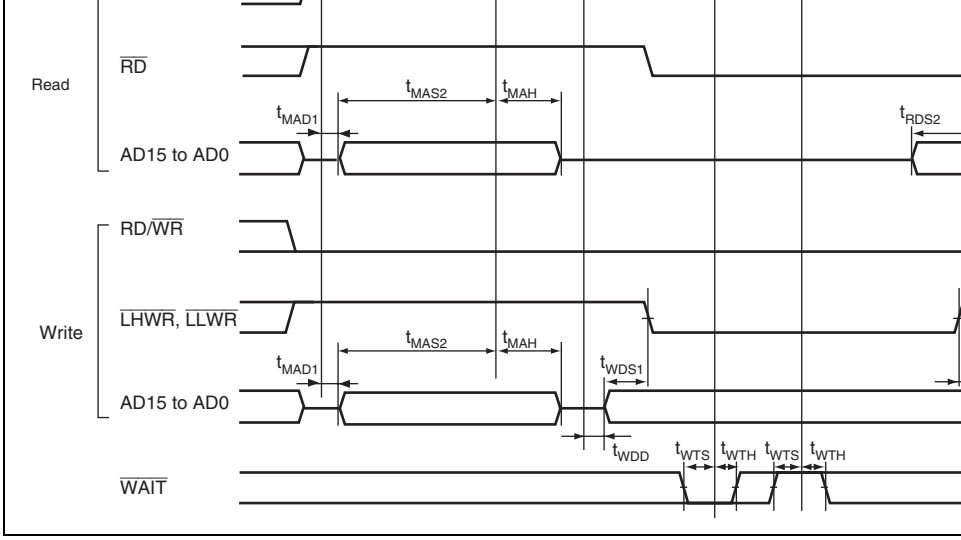


Figure 26.19 Address/Data Multiplexed Access Timing (Wait Control) (Address Cycle Program Wait × 1 + Data Cycle Program Wait × 1 + Data Cycle Pin Wait × 1)

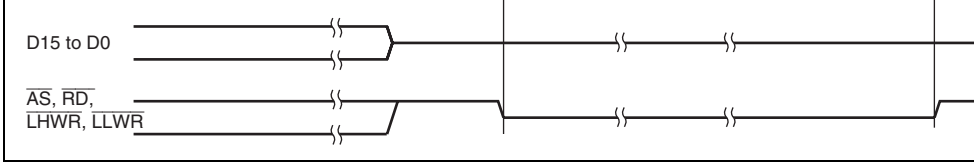


Figure 26.20 External Bus Release Timing

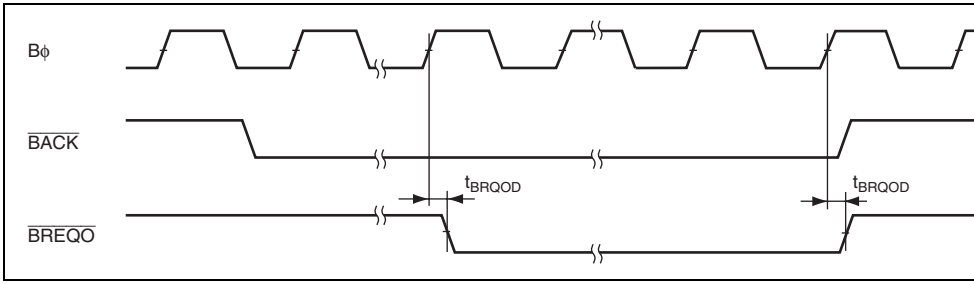


Figure 26.21 External Bus Request Output Timing

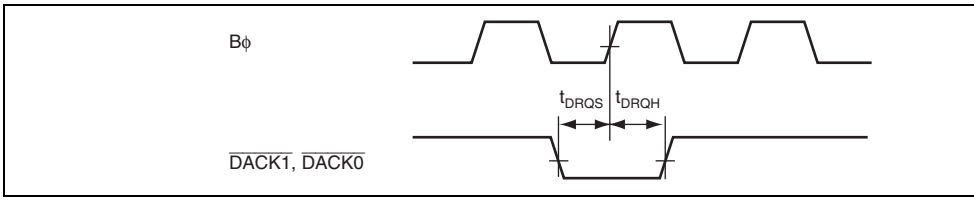


Figure 26.22 DMAC, DREQ Input Timing

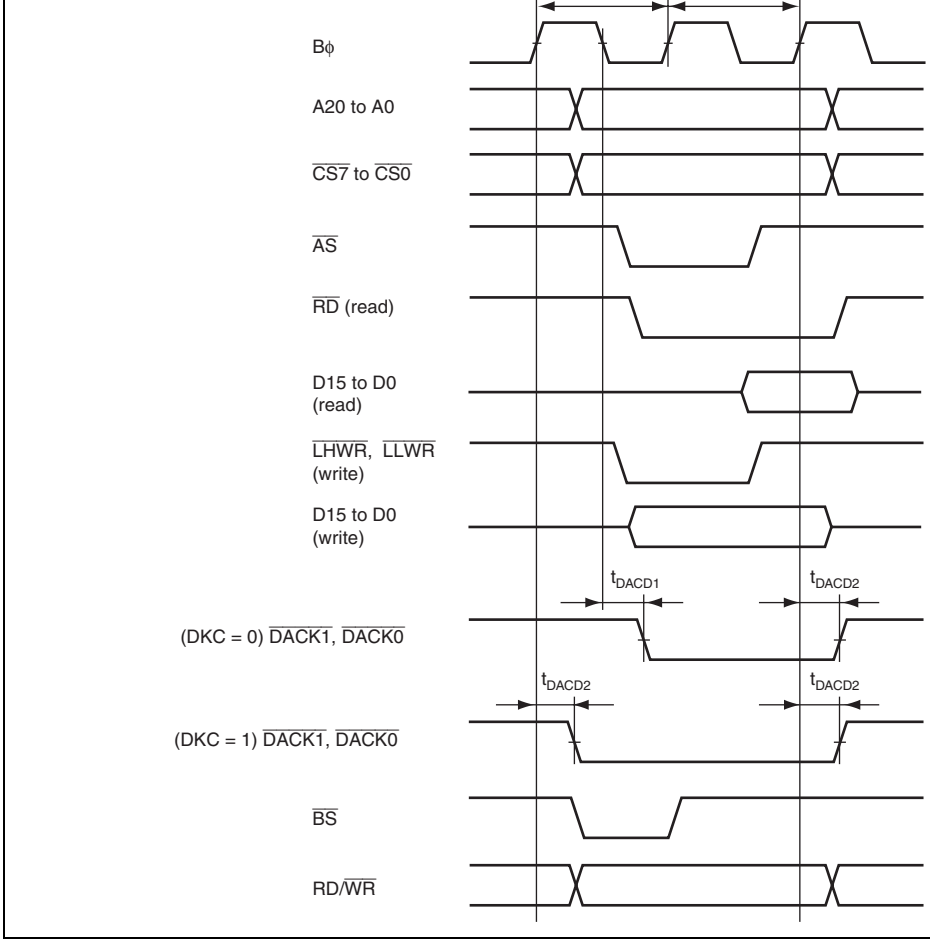


Figure 26.24 DMAC Single Address Transfer Timing: 2-State Access

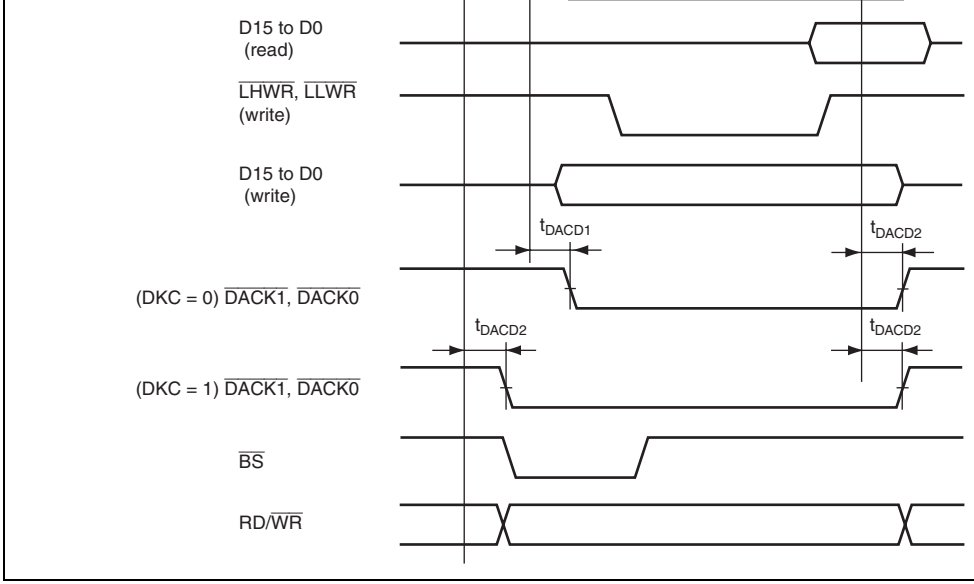


Figure 26.25 DMAC Single Address Transfer Timing: 3-State Access

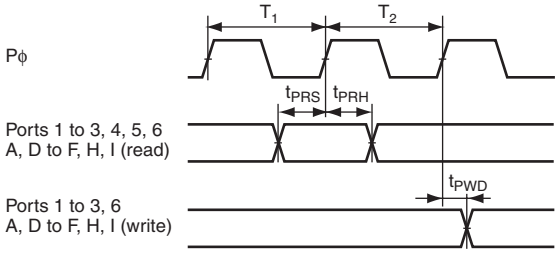


Figure 26.26 I/O Port Input/Output Timing

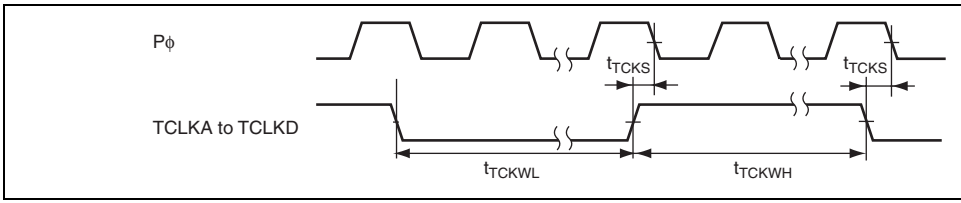


Figure 26.28 TPU Clock Input Timing

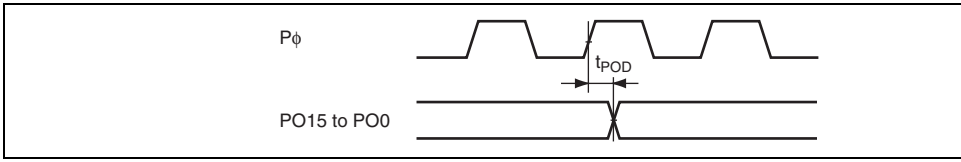


Figure 26.29 PPG Output Timing

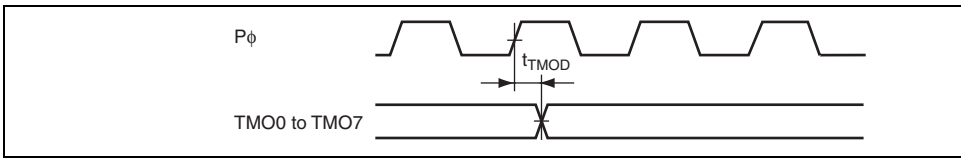


Figure 26.30 8-Bit Timer Output Timing

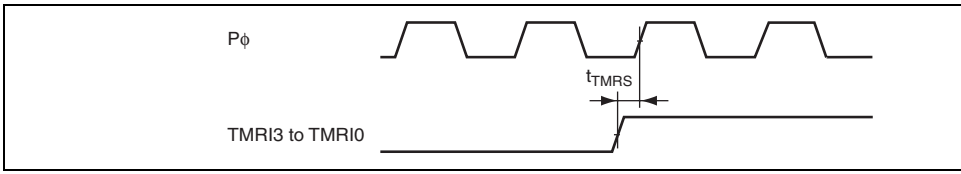


Figure 26.31 8-Bit Timer Reset Input Timing

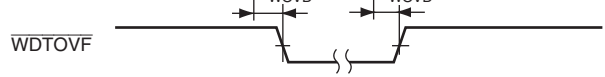


Figure 26.33 WDT Output Timing

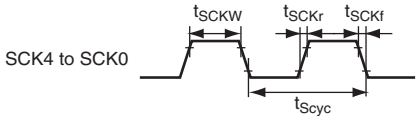


Figure 26.34 SCK Clock Input Timing

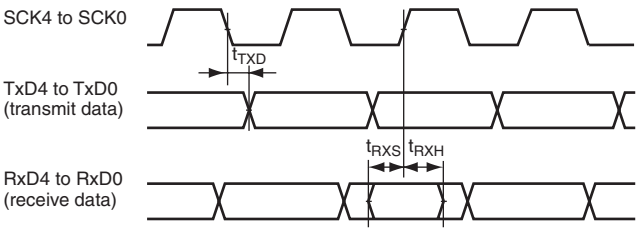


Figure 26.35 SCI Input/Output Timing: Clocked Synchronous Mode

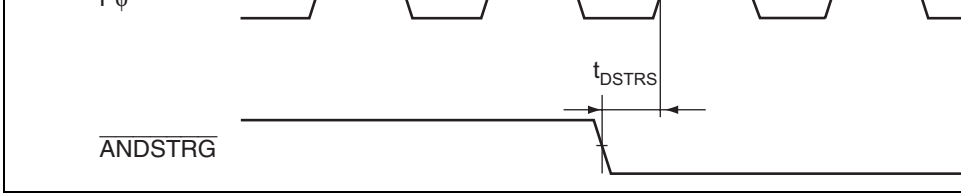
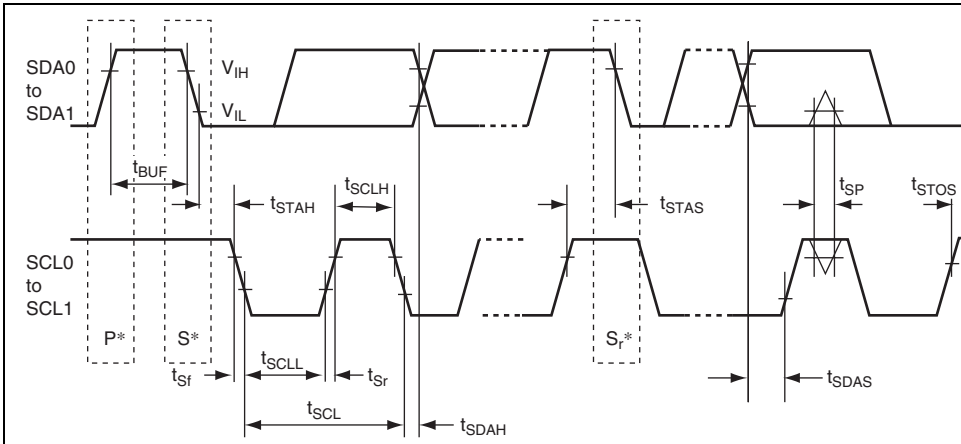


Figure 26.37 ΔΣ A/D Converter External Trigger Input Timing



Note: S, P, and Sr represent the following conditions:
S: Start condition
P: Stop condition
Sr: Retransmit start condition

Figure 26.38 I²C Bus Interface 2 Input/Output Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Test Cond
Programming time*1, *2, *4	t_P	—	1	10	ms/128 bytes	
Erasure time*1, *2, *4	t_E	—	40	130	ms/4k byte block	
		—	300	800	ms/32k byte block	
		—	600	1500	ms/64k byte block	
Programming time (total)*1, *2, *44	Σ_{tP}	—	2.3	6	s/256k bytes	T_a = 2 for al
Erasure time (total)*1, *2, *4	Σ_{tE}	—	2.3	6	s/256k bytes	T_a = 2
Programming, Erasure time (total)*1, *2, *4	Σ_{tPE}	—	4.6	12	s/256k bytes	T_a = 2
Overwrite count	N_{WEC}	100*3	—	—	times	
Data save time*4	T_{DRP}	10	—	—	Year	

Notes: 1. Programming time and erase time depend on data in the flash memory.
2. Programming time and erase time do not include time for data transfer.
3. All the characteristics after programming are guaranteed within this value (gua value is from 1 to Min. value).
4. Characteristics when programming is performed within the Min. value

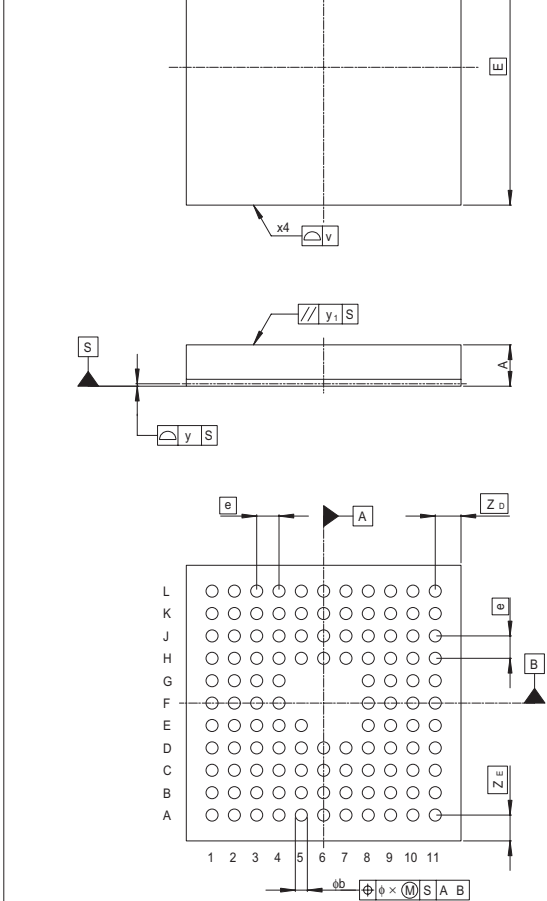
Port 1	All	HiZ	HiZ	Keep		Keep	
Port 2	All	HiZ	HiZ	Keep		Keep	
P30/PO8/TIOCA0/ DREQ0-B/CS0/ CS4-A/CS5-B	Single-chip mode (EXPE = 0)	HiZ	HiZ	[CS output] H	[CS output] HiZ	[CS output] H	[CS
	External extended mode (EXPE = 1)	H	HiZ	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other t K
P31/PO9/TIOCA0/ TIOCB0/TEND0-B/ CS1/CS2-B/CS5-A/ CS6-B/CS7-B	Single-chip mode (EXPE = 0)	HiZ	HiZ	[CS output] H	[CS output] HiZ	[CS output] H	[CS
	External extended mode (EXPE = 1)	HiZ	HiZ	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other t K
P32/PO10/TIOCC0/ TCLKA-A/DACK0-B/ CS2-A/CS6-A	Single-chip mode (EXPE = 0)	HiZ	HiZ	[CS output] H	[CS output] HiZ	[CS output] H	[CS
	External extended mode (EXPE = 1)	HiZ	HiZ	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other t K
P33/PO11/TIOCC0/ TIOCD0/TCLKB-A/ DREQ1-B/CS3/ CS7-A	Single-chip mode (EXPE = 0)	HiZ	HiZ	[CS output] H	[CS output] HiZ	[CS output] H	[CS
	External extended mode (EXPE = 1)	HiZ	HiZ	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other t K
P34 to P37	All	HiZ	HiZ	Keep		Keep	
P40 to P47	All	HiZ	HiZ	HiZ		HiZ	
P50 to P55	All	HiZ	HiZ	HiZ		HiZ	

							HiZ
P60 to P65	All	HiZ	HiZ	Keep		Keep	
PA0/BREQO/ BS-A	All	HiZ	HiZ	[BREQO output] HiZ		[BREQO output] HiZ	
				[BS output] Keep	[BS output] HiZ	[BS output] Keep	[BS output] HiZ
				[Other than above] Keep		[Other than above] Keep	
PA1/BACK/(RD/WR)	All	HiZ	HiZ	[BACK output] HiZ		[BACK output] HiZ	
				[RD/WR output] Keep	[RD/WR output] HiZ	[RD/WR output] Keep	[RD/WR output] HiZ
				[Other than above] Keep		[Other than above] Keep	
PA2/BREQ/WAIT	All	HiZ	HiZ	[BREQ input] HiZ [WAIT input] HiZ [Other than above] Keep		[BREQ input] HiZ [WAIT input] HiZ [Other than above] Keep	
PA3/LLWR/LLB	Single-chip mode (EXPE = 0)	HiZ	HiZ	Keep		Keep	
	External extended mode (EXPE = 1)	H	HiZ	H	HiZ	H	H

				above] Keep	than above] Keep	above] Keep	
PA5/RD	Single-chip mode (EXPE = 0)	HiZ	HiZ	Keep		Keep	
	External extended mode (EXPE = 1)	H	HiZ	H	HiZ	H	
PA6/AS/AH/ BS-B	Single-chip mode (EXPE = 0)	HiZ	HiZ	[AS, BS output] H [AH output] L [Other than above] Keep	[AS, AH, BS output] HiZ [Other than above] Keep	[AS, BS output] H [AH output] L [Other than above] Keep	[AS, o [Oth a K
	External extended mode (EXPE = 1)	H	HiZ				
PA7/Bϕ	Single-chip mode (EXPE = 0)	HiZ	HiZ	[Clock output] H [Other than above] Keep		[Clock output] H [Other than above] Keep	
	External extended mode (EXPE = 1)	Clock output	HiZ				

					above] Keep		abo Ke
	Single-chip mode (EXPE = 0)	HiZ	HiZ	Keep		Keep	
Port E	External extended mode (EXPE = 1)	L	HiZ	Keep	HiZ	Keep	H
	ROM enabled extended mode	HiZ	HiZ	Keep	[Address output] HiZ [Other than above] Keep	Keep	[Add out H [Othe abo Ke
	Single-chip mode (EXPE = 0)	HiZ	HiZ	Keep		Keep	
PF0 to PF4	External extended mode (EXPE = 1)	L	HiZ	Keep	HiZ	Keep	H
	ROM enabled extended mode	HiZ	HiZ	Keep	[Address output] HiZ [Other than above] Keep	Keep	[Add out H [Othe abo Ke
	Single-chip mode (EXPE = 0)	HiZ	HiZ	Keep		Keep	

	External extended mode (EXPE = 1)	8-bit bus mode	HiZ	HiZ	Keep	Keep
		16-bit bus mode	HiZ	HiZ	HiZ	HiZ
		32-bit bus mode	HiZ	HiZ	HiZ	HiZ



Reference Symbol	Dimension in Millimet		
	Min	Nom	M
D	——	9.0	—
E	——	9.0	—
v	——	——	0.1
w	——	——	0.2
A	——	——	1
A_1	——	——	—
e	——	0.65	—
b	0.30	0.35	0.4
x	——	——	0.0
y	——	——	0.1
y_1	——	——	0.2
S_D	——	——	—
S_E	——	——	—
Z_D	——	0.6	—
Z_E	——	0.6	—

Figure C.1 Package Dimensions (TLP-145V)

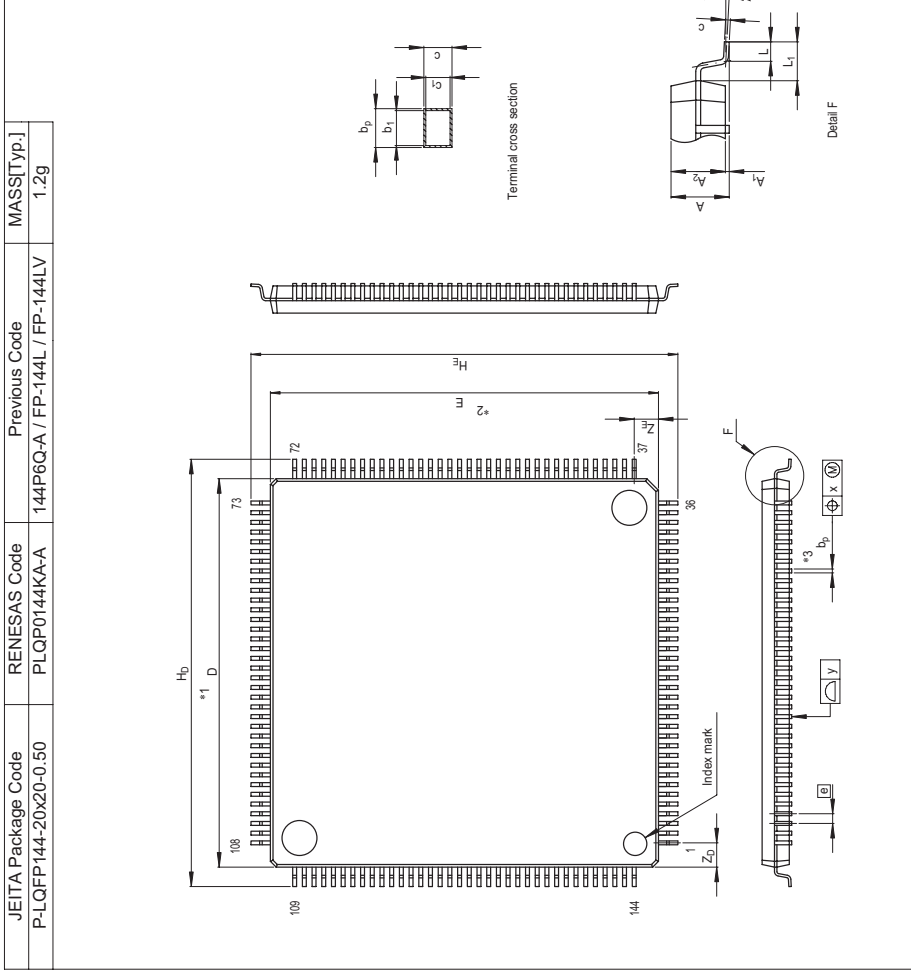


Figure C.2 Package Dimensions (FP-120BV)

MD2, MD1, 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	
Port 1, port 2, port 3, ports 37 to 31, port 6, PA2 to PA0, PF7 to PF5	Connect these pins to VCC via a pull-up resistor or to VSS via a pull-do resistor, respectively	
Port 4	Connect these pins to AVccP via a pull-up resistor or to AVssP via a pul resistor, respectively	
Port 5	Connect these pins to AVcc via a pull-up resistor or to AVss via a pull-d resistor, respectively	
PA7	Since this is the Bϕ output in its initial state, leave this pin unconnected.	Since this general-p input port initial stat each pin pull-up re connect e V_{SS} via a resistor.

		input port i initial state each pin to pull-up res connect ea V_{SS} via a p resistor.
PA4	Since this is the LHWR output in its initial state, leave this pin unconnected.	Since this general-pu input port i initial state each pin to pull-up res connect ea V_{SS} via a p resistor.
PA3	Since this is the LLWR output in its initial state, leave this pin unconnected.	Since this general-pu input port i initial state each pin to pull-up res connect ea V_{SS} via a p resistor.
P30	Since this is the CS0 output in its initial state, leave this pin unconnected	Since this is a general-purpose port in its initial state, connect e to V_{CC} via a pull-up resistor or c each pin to V_{SS} via a pull-down

	input port in its initial state, connect each pin to V_{CC} via a pull-up resistor or connect each pin to V_{SS} via a pulldown resistor.	to V_{CC} via a pull-up resistor or each pin to V_{SS} via a pull-down
Vref	Connect this pin to AVcc	
ANDS0, ANDS1, ANDS2, ANDS3, ANDS4P, ANDS4N, ANDS5P, ANDS5N	After the BIASE bit in DSADMR of the ΔΣ A/D converter is cleared to 0 a module stop bit of the ΔΣ A/D converter is set to 1, connect each pin to A a pull-up resistor or connect each pin to AVssP via a pull-down resistor.	
REXT	After the BIASE bit in DSADMR of the ΔΣ A/D converter is cleared to 0 a module stop bit of the ΔΣ A/D converter is set to 1, this pin is left open.	
AVCM	After the BIASE bit in DSADMR of the ΔΣ A/D converter is cleared to 0 a module stop bit of the ΔΣ A/D converter is set to 1, this pin is left open.	
AVrefT	Connect to $AV_{CC}A$.	
AVrefB	Connect to AVssA.	
NC	Open	

Notes: 1. Do not change the function of an unused pin from its initial state.
2. Do not change the initial value (input-buffer disabled) of PnICR, where n corre an unused pin.

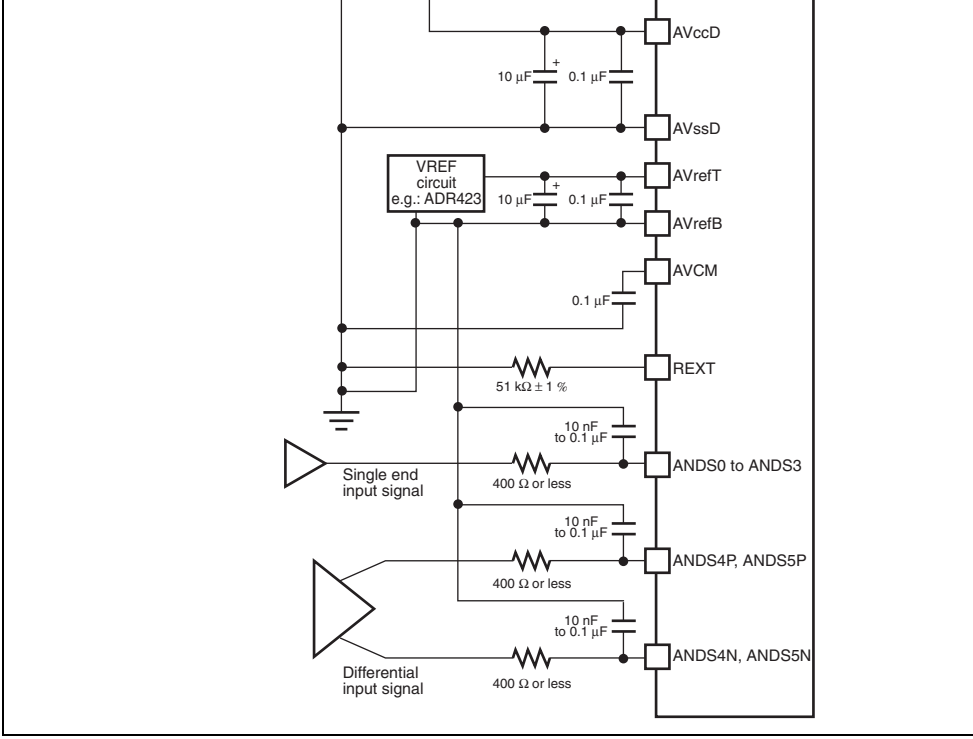
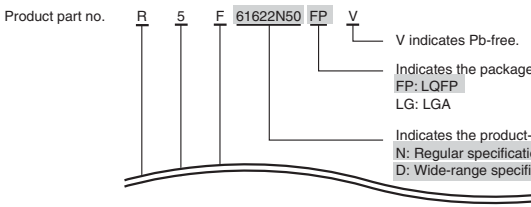


Figure E.1 Example of an External Circuit of ΔΣ A/D Converter

- Sixteen 16-bit general registers
- Eleven addressing modes
- 4-Gbyte address space
  Program: 4 Gbytes available
  Data: 4 Gbytes available
- 87 basic instructions, classifiable as bit a
  logic instructions, multiply and divide inst
  manipulation instructions, multiply-and-a
  instructions, and others
- Minimum instruction execution time: 20.0
  ADD instruction when running with syste
  50 MHz and VCC = 3.0 to 3.6 V)
- On-chip multiplier (16 ´ 16 ® 32 bits)
- Supports multiply-and-accumulate instru
  (16 × 16 + 42 → 42 bits)

1.2 List of Products	8	Replaced Table 1.2 List of Products
Figure 1.1 How to Read the Product Part No.	8	Amended 

LQFP	LGA	Modes 1, 2, 6, 7	Modes 4 and 5
56	K8	P24/PO4/TIOCA4/TIOCB4/ TMRI1/SCK1/IRQ12-A	P24/PO4/TIOCA4/TIO TMRI1/SCK1/IRQ12-A
57	N7	P25/PO5/TIOCA4/TMCI1/RxD1/ IRQ13-A	P25/PO5/TIOCA4/TMC IRQ13-A
58	M8	P26/PO6/TIOCA5/TMO1/TxD1/ IRQ14	P26/PO6/TIOCA5/TMC IRQ14
59	L7	P27/PO7/TIOCA5/TIOCB5/IRQ15	P27/PO7/TIOCA5/TIO
103	D13	V_{CL}	V_{CL}
104	D10	WDTOVF/TDO	WDTOVF/TDO

Section 6 Interrupt Controller 6.6.5 DTC and DMAC Activation by Interrupt (1) Selection of Interrupt Sources (3) Operation Order	134, 135	Amended "DTCERA to DTCERH of the DTC" is amended to "DTCERA to DTCERG of the DTC".
Section 9 DMA Controller (DMAC) 9.3.5 DMA Block Size Register (DBSR)	268	Amended
Section 10 Data Transfer Controller (DTC) 10.9.9 Points for Caution when Overwriting DTCER	375	Added Text and figure 10.17 (Example of Procedures for Overwriting DTCER) are added.

		BREQO_OE	BREQO	SYSCR.EXPE = 1, BCR BCR1.BREQOE = 1
PE	7	A15_OE	A15	SYSCR.EXPE = 1, PED = 1
	6	A14_OE	A14	SYSCR.EXPE = 1, PED = 1
	5	A13_OE	A13	SYSCR.EXPE = 1, PED = 1
	4	A12_OE	A12	SYSCR.EXPE = 1, PED = 1
	3	A11_OE	A11	SYSCR.EXPE = 1, PED = 1
	2	A10_OE	A10	SYSCR.EXPE = 1, PED = 1
	1	A9_OE	A9	SYSCR.EXPE = 1, PED = 1
	0	A8_OE	A8	SYSCR.EXPE = 1, PED = 1

Section 16 Serial Communication Interface (SCI) 16.3.7 Serial Status Register (SSR)	616	Amended

Bit	7	6	5	4	3	2	1
Bit Name	TDRE	RDRF	ORER	FER	PER	TEND	MP
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.

		TPU is enabled. 100: A/D conversion start by conversion tri TMR is enabled. 110: A/D conversion start by the $\overline{\text{ADTRG0}}$ enabled.* 001: External triggers are disabled 011: Setting prohibited 101: Setting prohibited 111: Setting prohibited
18.7.3 Notes on A/D Conversion Start by an External Trigger	732, 733	Added Text and figure 18.9 (Procedure for Changing the M When Setting for Activation by an External Trigger Use) are added.
18.7.4 Notes on Stopping the A/D Converter	734	Text and figures 18.10 (Stopping Continuous Scan Activated by Software) and 18.11 (Stopping Contin Scan Mode Activated by External Trigger) are adde
	737, 739	Figure No. amended Figure 18.12 Example of Analog Input Circuit Figure 18.13 Example of Analog Input Protection C Figure 18.14 Analog Input Pin Equivalent Circuit

1018 Product model and marking switched

Product Classification	Product Model	Marking	Packag (Packa
H8SX/1622	R5F61622	R5F61622LGV	PTLG0 (TLP-1
H8SX/1622	R5F61622	R5F61622FPV	PLQP0 (FP-14

E

H

I

R

T

U

V

W

Renesas 32-Bit CISC Microcomputer
Hardware Manual
H8SX/1622 Group

Publication Date:	Rev.1.00, Nov. 01, 2007
	Rev.2.00, Sep. 16, 2009
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

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/1622 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0414-0200

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 32-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MB91F575BHSPMC-GSE1 MB91F594BSPMC-GSE1 PIC32MX120F032B-50I/ML MB91F464AAPMC-GSE2 MB91F577BHSPMC-GSE1 SPC5604EEF2MLH MB91F528USCPMC-GSE2 MB91F248PFV-GE1 MB91F594BPMC-GSE1 MB91243PFV-GS-136E1 MB91F577BHSPMC1-GSE1 PIC32MM0032GPL020-E/ML PIC32MM0032GPL020-E/SS MEC1632X-AUE PIC32MM0016GPL020-E/ML PIC32MM0016GPL020-E/SS PIC32MM0016GPL028-E/SS PIC32MM0016GPL028-E/SO PIC32MM0016GPL028-E/ML PIC32MM0032GPL028-E/SS PIC32MM0032GPL028-E/SO PIC32MM0032GPL028-E/ML PIC32MM0032GPL036-E/MV PIC32MM0064GPL028-E/M6 PIC32MM0016GPL028-E/M6 PIC32MM0032GPL028-E/M6 MB91F526KSEPMC-GSE1 PIC32MM0064GPL028-E/SP PIC32MM0032GPL036-E/M2 TLE9872QTW40XUMA1 FT902L-T R5F564MLCDFB#31 R5F564MLCDFC#31 R5F523E5ADFL#30 R5F524TAADFF#31 MCF51AC256ACPUE PIC32MM0064GPL028-I/ML PIC32MM0064GPL028-I/SP PIC32MM0064GPL028-I/SO PIC32MX120F032D-I/TL PIC32MX130F064D-I/ML PIC32MZ2064DAB169-I/HF PIC32MZ2064DAB288-I/4J ATUC256L4U-AUT R5F56318CDBG#U0 PIC32MX150F128C-I/TL PIC32MX170F256B-50IML PIC32MX130F064C-ITL PIC32MX230F064D-IML PIC32MM0032GPL028-I/ML