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

SH7211 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer SuperH™ RISC engine Family

SH7211 R5F72115D160FPV
R5F72114D160FPV

Renesas Electronics
www.renesas.com

Rev.3.00 2009.03

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

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

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Re Technology Corp., its affiliated companies and their officers, directors, and employees against any and a damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with res to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunct 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 spec 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 inju injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design fo hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final produc system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is high. You should implement safety measures so that Renesas products may not be easily detached from 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 writte approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in t document, Renesas semiconductor products, or if you have any other inquiries.

Generally, the input pins of CMOS products are high-impedance input pins. If s
are in their open states, intermediate levels are induced by noise in the vicinity, a
through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.
The states of internal circuits are undefined until full power is supplied throughc
chip and a low level is input on the reset pin. During the period where the states
undefined, the register settings and the output state of each pin are also undefine
your system so that it does not malfunction because of processing while it is in t
undefined state. For those products which have a reset function, reset the LSI im
after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test re
may have been be allocated to these addresses. Do not access these registers; the
operation is not guaranteed if they are accessed.

- CPU and System-Control Modules
- On-Chip Peripheral Modules

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

i) Feature
ii) Input/Output Pin
iii) Register Description
iv) Operation
v) Usage Note

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

7. List of Registers
8. Electrical Characteristics
9. Appendix
   - Product Type, Package Dimensions, etc.
10. Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier ver
This does not include all of the revised contents. For details, see the actual locations in th
manual.

11. Index

characteristics of this LSI to the target users.
Refer to the SH-2A, SH2A-FPU Software Manual for a detailed descriptio
instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip

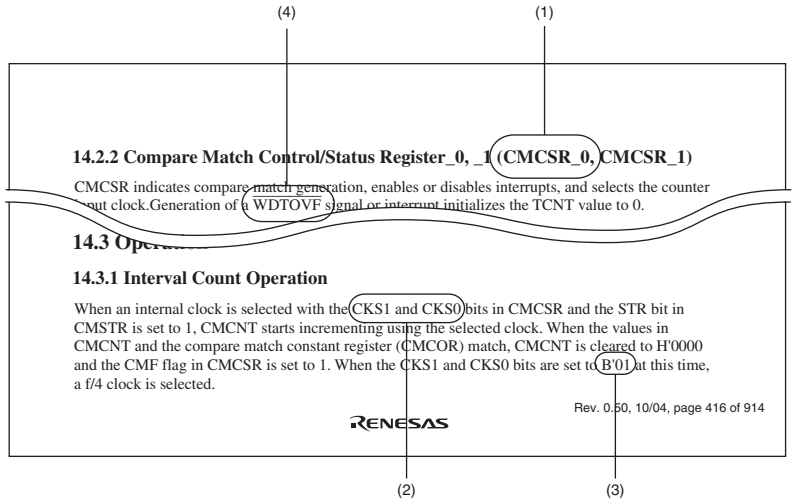
  Read the manual according to the contents. This manual can be roughly categorized
  on the CPU, system control functions, peripheral functions and electrical characteris
- In order to understand the details of the CPU's functions

  Read the SH-2A, SH2A-FPU Software Manual.
- In order to understand the details of a register when its name is known

  Read the index that is the final part of the manual to find the page number of the entr
  register. The addresses, bits, and initial values of the registers are summarized in sec
  List of Registers.

(3) Numbers
Binary numbers are given as B'xxxx, hexadecimal are given as H'xxxx, and decimal are given as xxxx.
Examples: B'11 or 11, H'EFA0, 1234

(4) Symbols
An overbar is added to the names of active-low signals.
Example: $\overline{\text{WDTOVF}}$



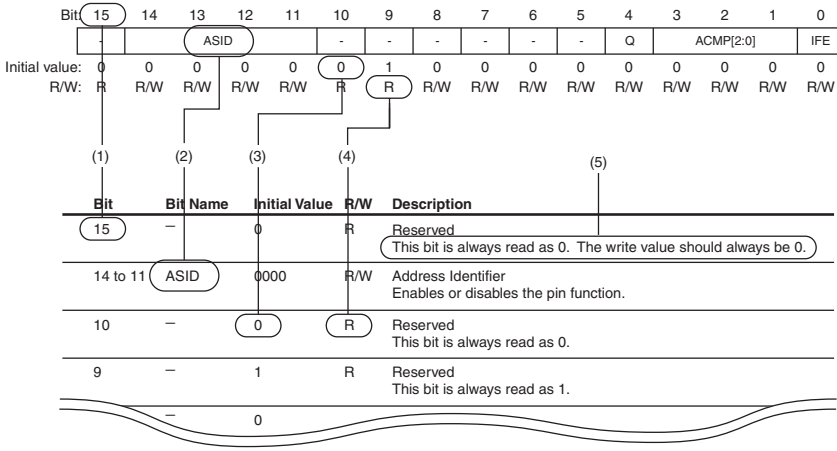
Note: The bit names and sentences in the above figure are examples, and have nothing to do with the contents of this manual.

Instead of a bit name, a blank is used for some bits, such as those of timer counters.

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

(4) R/W
Indicates whether each bit is readable or writable, or either writing to or reading from the bit is prohibited.
The notation is as follows:
R/W: Bit or field is readable and writable.
R/(W): Bit or field is readable and writable.
However, writing is only performed to clear the flag.
R: Bit or field is readable and writable.
However, "R" is indicated for all reserved bits. When writing to the bit is required, write the value stated in the bit table or the initial value.
W: Bit or field is readable and writable.
However, only the value in the bit table is guaranteed when reading from the bit.

(5) Description
Describes the function enabled by setting the bit.



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

With this CPU, it has become possible to assemble low-cost, high-performance, and hig
functioning systems, even for applications that were previously impossible with micropr
such as realtime control, which demands high speeds.

In addition, this LSI includes on-chip peripheral functions necessary for system configu
such as a large-capacity ROM, a ROM cache, a RAM, a direct memory access controlle
(DMAC), multi-function timer pulse units 2 (MTU2 and MTU2S), a serial communicati
interface with FIFO (SCIF), an A/D converter, a D/A converter, an interrupt controller (
I/O ports, and I^2C bus interface 3 (IIC3).

This LSI also provides an external memory access support function to enable direct con
various memory devices or peripheral LSIs.

These on-chip functions significantly reduce costs of designing and manufacturing appli
systems.

The features of this LSI are listed in table 1.1.

	— Register bank for high-speed response to interrupts • RISC-type instruction set (upward compatible with SH series) — Instruction length: 16-bit fixed-length basic instructions for code efficiency and 32-bit instructions for high performanc usability — Load/store architecture — Delayed branch instructions — Instruction set based on C language • Superscalar architecture to execute two instructions at one tim • Instruction execution time: Up to two instructions/cycle • Address space: 4 Gbytes • Internal multiplier • Five-stage pipeline
Operating modes	• Operating modes Extended ROM enabled mode Single-chip mode • Processing states Program execution state Exception handling state Bus mastership release state • Power-down modes Sleep mode Software standby mode Module standby mode

	• 16 priority levels available • Register bank enabling fast register saving and restoring in i processing
Bus state controller (BSC)	• Address space divided into eight areas (0 to 7), each a maxi Mbytes • External bus: 8 or 16 bits • The following features settable for each area independently — Supports both big endian and little endian for data acces — Bus size (8 or 16 bits): Available sizes depend on the are — Number of access wait cycles (different wait cycles can b specified for read and write access cycles in some areas) — Idle wait cycle insertion (between same area access cycl different area access cycles) — Specifying the memory to be connected to each area ena direct connection to SRAM, SRAM with byte selection, SD and burst ROM (clocked synchronous or asynchronous). address/data multiplexed I/O (MPX) interface is also avai — Outputs a chip select signal ($\overline{CS0}$ to $\overline{CS7}$) according to th area ($\overline{CS}$ assert or negate timing can be selected by softw • SDRAM refresh Auto refresh or self refresh mode selectable • SDRAM burst access

	• Five types of clocks generated: — CPU clock: Maximum 160 MHz — Bus clock: Maximum 40 MHz — Peripheral clock: Maximum 40 MHz — Timer clock: Maximum 80 MHz — AD clock: Maximum 40 MHz
Watchdog timer (WDT)	• On-chip one-channel watchdog timer • A counter overflow can reset the LSI
Power-down modes	• Three power-down modes provided to reduce the current cons in this LSI — Sleep mode — Software standby mode — Module standby mode

	— Non-overlapping waveforms output for 3-phase inverter c — Automatic dead time setting — 0% to 100% PWM duty value specifiable — A/D conversion delaying function — Interrupt skipping at crest or trough • Reset-synchronized PWM mode Three-phase PWM waveforms in positive and negative phas output with a required duty value • Phase counting mode Two-phase encoder pulse counting available
Multi-function timer pulse unit 2S (MTU2S)	• Subset of MTU2, included in channels 3 to 5 • Operating at 80 MHz max.
Port output enable 2 (POE2)	• High-impedance control of high-current pins at a falling edge level input on the $\overline{\text{POE}}$ pin
Compare match timer (CMT)	• Two-channel 16-bit counters • Four types of clock can be selected (Pϕ/8, Pϕ/32, Pϕ/128, an • DMA transfer request or interrupt request can be issued whe compare match occurs
Serial communication interface with FIFO (SCIF)	• Four channels • Clocked synchronous or asynchronous mode selectable • Simultaneous transmission and reception (full-duplex commu supported • Dedicated baud rate generator • Separate 16-byte FIFO registers for transmission and recept

	• Two output channels
User break controller (UBC)	• Four break channels • Addresses, type of access, and data size can all be set as bre conditions
User debugging interface (H-UDI)	• E10A emulator support • JTAG-standard pin assignment • Realtime branch trace
Advanced user debugger II (AUD- II)	• Six output pins • Branch source address/destination address trace • Window data trace • Full trace All trace data can be output by interrupting CPU operation • Realtime trace Trace data can be output within the range where CPU operati interrupted
WAVE interface (WAVEIF)	• Myway Labs realtime CPU scope “WAVE™” (WAVE1.0 Level supported
On-chip ROM	• 384/512 Kbytes (See B. Product Lineup)
On-chip RAM	• Three/Four pages • 24/32 Kbytes (See B. Product Lineup)
Power supply voltage	• Vcc: 1.4 to 1.6 V • VccQ: 3.0 to 3.6 V • AVcc: 4.5 to 5.5 V
Packages	• LQFP2020-144 (0.5 pitch)

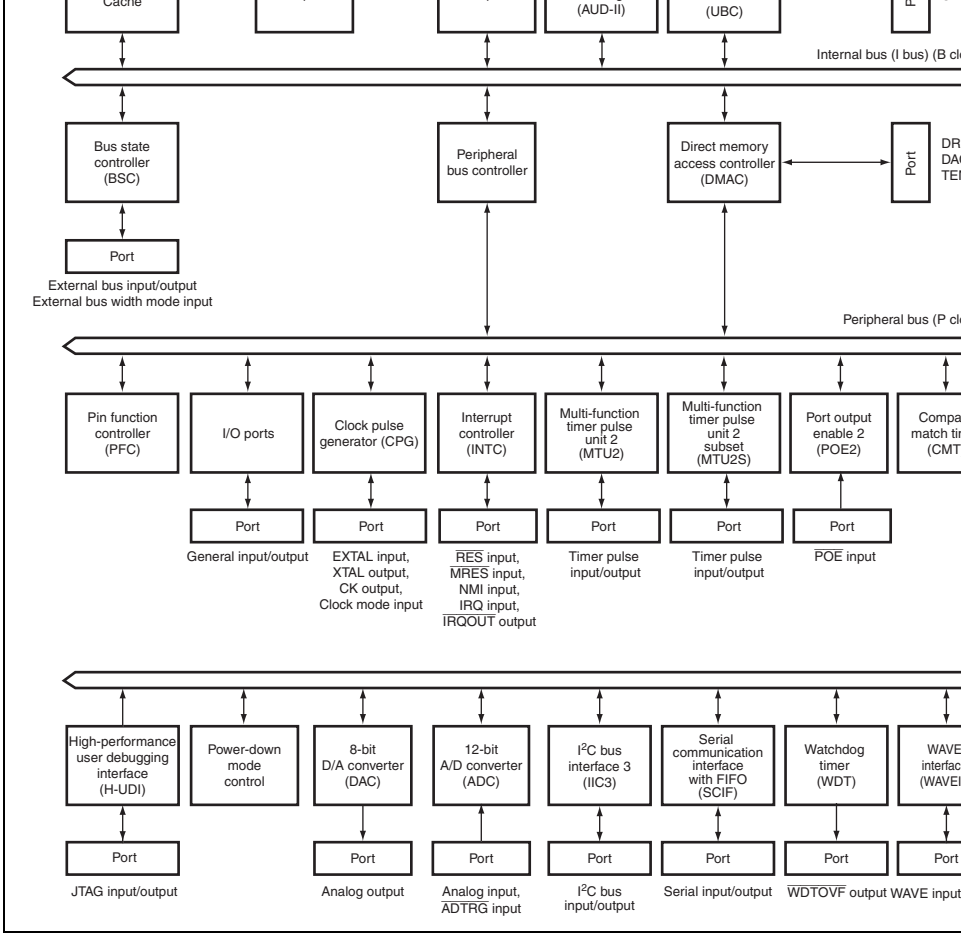
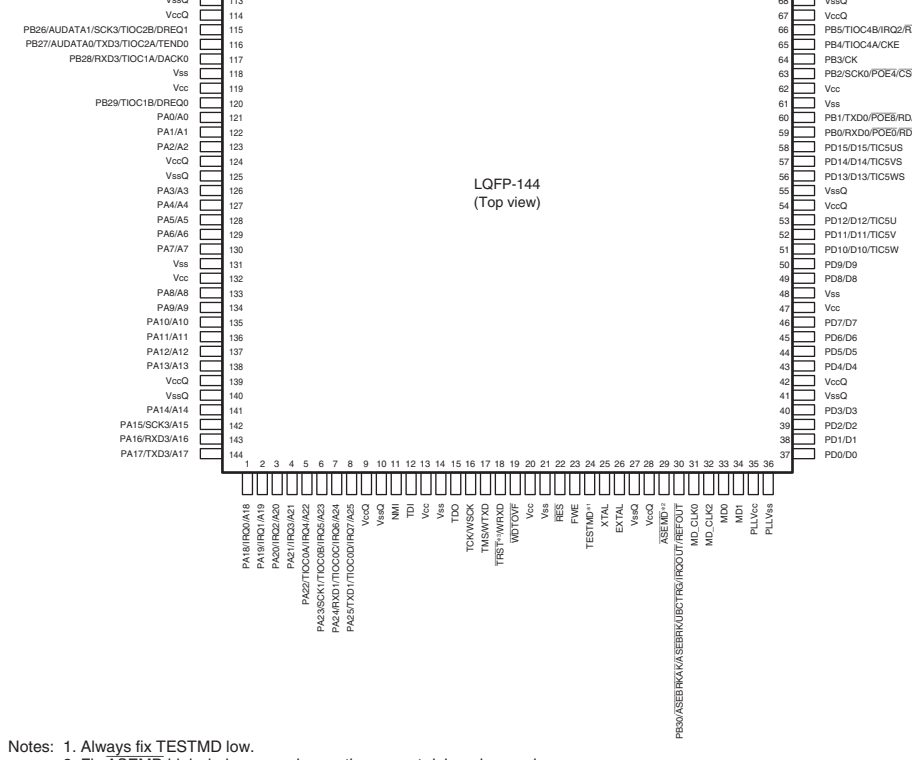


Figure 1.1 Block Diagram



Notes: 1. Always fix TESTMD low.
2. Fix ASEMD high during normal operation except debugging mode.
3. Fix TRST low when H-UDI is not used.

Figure 1.2 Pin Arrangement

	Vss	I	Ground	Ground pins. All the Vss pi be connected to the syster supply (0 V). This LSI does operate correctly if there is open.
	VccQ	I	Power supply for I/O circuits	Power supply for I/O pins. VccQ pins must be connec system power supply. This not operate correctly if ther left open.
	VssQ	I	Ground for I/O circuits	Ground pins for I/O pins. A VssQ pins must be connec system power supply (0 V) does not operate correctly a pin left open.
	PLLVcc	I	Power supply for PLL	Power supply for the on-ch oscillator.
	PLLVss	I	Ground for PLL	Ground pin for the on-chip oscillator.
Clock	EXTAL	I	External clock	Connected to a crystal res An external clock signal m input to the EXTAL pin.
	XTAL	O	Crystal	Connected to a crystal res
	CK	O	System clock	Supplies the system clock devices.

				functions.
				Input a high level to operate in normal mode (not in debu mode). To operate it in debu mode, apply a low level to th the user system board.
	TESTMD	I	Test mode	Always fix this input pin low
				Do not input a high level to t This may cause malfunction permanent failure of this LS
System control	$\overline{\text{RES}}$	I	Power-on reset	This LSI enters the power-o state when this signal goes
	$\overline{\text{MRES}}$	I	Manual reset	This LSI enters the manual state when this signal goes
	$\overline{\text{WDTOVF}}$	O	Watchdog timer overflow	Outputs an overflow signal f WDT.
	$\overline{\text{BREQ}}$	I	Bus-mastership request	A low level is input to this pi an external device requests release of the bus mastersh
	$\overline{\text{BACK}}$	O	Bus-mastership request acknowledge	Indicates that the bus maste has been released to an ext device. Reception of the $\overline{\text{BA}}$ signal informs the device wh output the $\overline{\text{BREQ}}$ signal that acquired the bus.

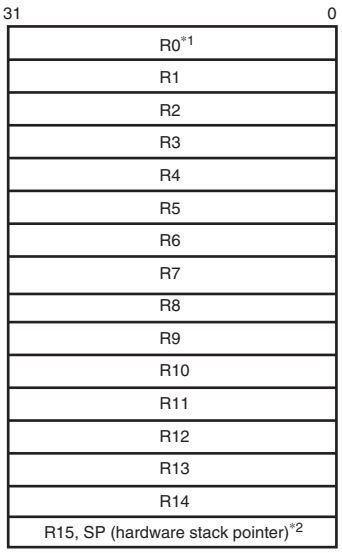
				to be informed of an interru occurrence even while the mastership is released.
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D15 to D0	I/O	Data bus	Bidirectional data bus.
Bus control	$\overline{CS7}$ to $\overline{CS0}$	O	Chip select 7 to 0	Chip-select signals for exte memory or devices.
	$\overline{RD}$	O	Read	Indicates that data is read external device.
	RD/$\overline{WR}$	O	Read/write	Read/write signal.
	$\overline{BS}$	O	Bus start	Bus-cycle start signal.
	$\overline{AH}$	O	Address hold	Address hold timing signal device that uses the addre multiplexed bus.
	$\overline{WAIT}$	I	Wait	Input signal for inserting a into the bus cycles during the external space.
	$\overline{WE0}$	O	Byte select	Indicates a write access to of data of external memory device.
	$\overline{WE1}$	O	Byte select	Indicates a write access to 8 of data of external memo device.

	$\overline{\text{REFOUT}}$	O	Refresh request	Request signal for refresh e
Direct memory access controller (DMAC)	DREQ3 to DREQ0	I	DMA-transfer request	Input pins to receive externa requests for DMA transfer.
	DACK3 to DACK0	O	DMA-transfer request accept	Output pins for signals indic acceptance of external requ from external devices.
	TEND1, TEND0	O	DMA-transfer end output	Output pins for DMA transfe
Multi-function timer pulse unit 2 (MTU2)	TCLKA, TCLKB, TCLKC, TCLKD	I	MTU2 timer clock input	External clock input pins for timer.
	TIOC0A, TIOC0B, TIOC0C, TIOC0D	I/O	MTU2 input capture/output compare (channel 0)	The TGRA_0 to TGRD_0 in capture input/output compa output/PWM output pins.
	TIOC1A, TIOC1B	I/O	MTU2 input capture/output compare (channel 1)	The TGRA_1 and TGRB_1 capture input/output compa output/PWM output pins.
	TIOC2A, TIOC2B	I/O	MTU2 input capture/output compare (channel 2)	The TGRA_2 and TGRB_2 capture input/output compa output/PWM output pins.
	TIOC3A, TIOC3B, TIOC3C, TIOC3D	I/O	MTU2 input capture/output compare (channel 3)	The TGRA_3 to TGRD_3 in capture input/output compa output/PWM output pins.

	POE7, POE4	I	Port output control	Request signal input to pla MTU2S waveform output p high impedance state.
Multi-function timer pulse unit 2S (MTU2S)	TIOC3AS, TIOC3BS, TIOC3CS, TIOC3DS	I/O	MTU2S input capture/output compare (channel 3)	The TGRA_3S to TGRD_3 capture input/output comp output/PWM output pins.
	TIOC4AS, TIOC4BS, TIOC4CS, TIOC4DS	I/O	MTU2S input capture/output compare (channel 4)	The TGRA_4S and TGRB capture input/output comp output/PWM output pins.
	TIOC5US, TIOC5VS, TIOC5WS	I	MTU2S input capture (channel 5)	The TGRU_5S, TGRV_5S TGRW_5S input capture in time compensation input p
	$\overline{\text{RTS3}}$	O	Transmit request	Modem control pin.
	$\overline{\text{CTS3}}$	I	Transmit enable	Modem control pin.
I^2C bus interface 3 (IIC3)	SCL	I/O	Serial clock pin	Serial clock input/output pi
	SDA	I/O	Serial data pin	Serial data input/output pin
A/D converter (ADC)	AN7 to AN0	I	Analog input pins	Analog input pins.
	$\overline{\text{ADTRG}}$	I	A/D conversion trigger input	External trigger input pin fo A/D conversion.
	AVcc	I	Analog power supply	Power supply pin for the A converter. Connect this pin system power supply (Vcc the A/D converter is not us
	AVREF	I	Analog reference power supply	Reference voltage pin for t converter. Connect this pin system power supply (Vcc the A/D converter is not us

I/O ports	PA25 to PA0	I/O	General port	26-bit general input/output p
	PB30 to PB0	I/O	General port	31-bit general input/output p
	PD15 to PD0	I/O	General port	16-bit general input/output p
	PF1, PF0	I	General port	2-bit general input/output po
User debugging interface (H-UDI)	TCK	I	Test clock	Test-clock input pin.
	TMS	I	Test mode select	Test-mode select signal inp
	TDI	I	Test data input	Serial input pin for instructio data.
	TDO	O	Test data output	Serial output pin for instruct data.
	$\overline{\text{TRST}}$	I	Test reset	Initialization-signal input pin low level when not using th
Advanced user debugger (AUD)	AUDATA3 to AUDATA0	I/O	AUD data	Branch destination/source a output pin
	AUDCK	I/O	AUD clock	Sync clock output pin
	AUDSYNC	I/O	AUD sync signal	Data start-position acknowl signal output pin
Emulator interface	$\overline{\text{ASEBRKAK}}$	O	Break mode acknowledge	Indicates that the E10A-US emulator has entered its br mode.
	$\overline{\text{ASEBRK}}$	I	Break request	E10A-USB emulator break
User break controller (UBC)	$\overline{\text{UBCTRG}}$	O	User break trigger output	Trigger output pin for UBC match.
WAVE interface (WAVEIF)	WSCK	O	Clock output	Interface pin to support Myw realtime CPU scope “WAVE (WAVE1.0 Level C)
	WRXD	I	Receive data	
	WTXD	O	Transmit data	

The sixteen 32-bit general registers are numbered R0 to R15. General registers are used
processing and address calculation. R0 is also used as an index register. Several instruct
R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Sa
restoring the status register (SR) and program counter (PC) in exception handling is acc
by referencing the stack using R15.



Notes: 1. R0 functions as an index register in the indexed register indirect addressing mode and indexed GBR i
addressing mode. In some instructions, R0 functions as a fixed source register or destination register.
2. R15 functions as a hardware stack pointer (SP) during exception processing.

Figure 2.1 General Registers

The jump table base register functions as the base address of the function table area.

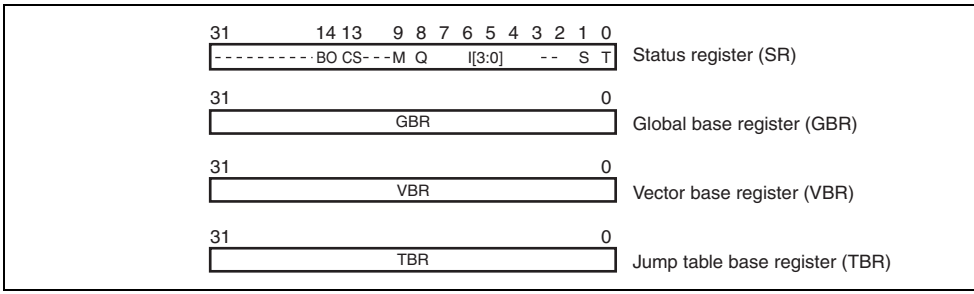


Figure 2.2 Control Registers

(1) Status Register (SR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	BO	CS	-	-	-	M	Q	I[3:0]				-	-
Initial value:	0	0	0	0	0	0	-	-	1	1	1	1	0	0
R/W:	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

				below the saturation lower-limit value.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructi
7 to 4	I[3:0]	1111	R/W	Interrupt Mask Level
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
1	S	—	R/W	S Bit Specifies a saturation operation for a MAC inst
0	T	—	R/W	T Bit True/false condition or carry/borrow bit

(2) Global Base Register (GBR)

GBR is referenced as the base address in a GBR-referencing MOV instruction.

(3) Vector Base Register (VBR)

VBR is referenced as the branch destination base address in the event of an exception or interrupt.

(4) Jump Table Base Register (TBR)

TBR is referenced as the start address of a function table located in memory in a JSR/N@@(disp8,TBR) table-referencing subroutine call instruction.

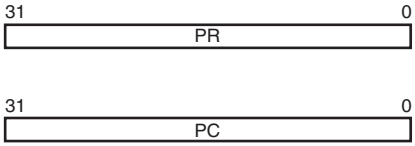


Figure 2.3 System Registers

(1) Multiply and Accumulate Register High (MACH) and Multiply and Accumula Register Low (MACL)

MACH and MACL are used as the addition value in a MAC instruction, and store the res MAC or MUL instruction.

(2) Procedure Register (PR)

PR stores the return address of a subroutine call using a BSR, BSRF, or JSR instruction, a referenced by a subroutine return instruction (RTS).

(3) Program Counter (PC)

PC indicates the address of the instruction being executed.

2.1.3 Initial Values of Registers

Table 2.1 lists the values of the registers after a reset.

Table 2.1 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0 to R14	Undefined
	R15 (SP)	Value of the stack pointer in the address table
Control registers	SR	Bits I[3:0] are 1111 (H'F), BO and 0, reserved bits are 0, and other undefined
	GBR, TBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in address table

Figure 2.4 Data Format in Registers

2.2.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be ac
8-bit bytes, 16-bit words, or 32-bit longwords. A memory operand of fewer than 32 bits i
in a register in sign-extended or zero-extended form.

A word operand should be accessed at a word boundary (an even address of multiple of t
address 2n), and a longword operand at a longword boundary (an even address of multipl
bytes: address 4n). Otherwise, an address error will occur. A byte operand can be accesse
address.

Only big-endian byte order can be selected for the data format.

Data formats in memory are shown in figure 2.5.

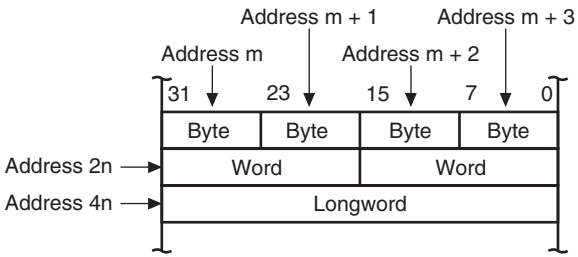


Figure 2.5 Data Formats in Memory

in the destination register in sign-extended form.

Word or longword immediate data is not located in the instruction code, but rather is sto
memory table. The memory table is accessed by an immediate data transfer instruction (
using the PC relative addressing mode with displacement.

See examples given in section 2.3.1 (10), Immediate Data.

The SH-2A additionally features 32-bit fixed-length instructions, improving performance of use.

(3) One Instruction per State

Each basic instruction can be executed in one cycle using the pipeline system.

(4) Data Length

Longword is the standard data length for all operations. Memory can be accessed in bytes or longwords. Byte or word data in memory is sign-extended and handled as longword da Immediate data is sign-extended for arithmetic operations or zero-extended for logic oper is also handled as longword data.

Table 2.2 Sign Extension of Word Data

SH2-A CPU		Description	Example of Other C
MOV.W	@(disp,PC),R1	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	ADD.W #H'1234,
ADD	R1,R0		
	.....		
.DATA.W	H'1234		

Note: @(disp, PC) accesses the immediate data.

(5) Load-Store Architecture

Basic operations are executed between registers. For operations that involve memory acc is loaded to the registers and executed (load-store architecture). Instructions such as AND manipulate bits, however, are executed directly in memory.

slot, the branch destination address remains as the register contents prior to the change.

Table 2.3 Delayed Branch Instructions

SH-2A CPU		Description	Example of Other CPU	
BRA	TRGET	Executes the ADD before branching to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

(7) Unconditional Branch Instructions with No Delay Slot

The SH-2A additionally features unconditional branch instructions in which a delay slot instruction is not executed. This eliminates unnecessary NOP instructions, and so reduce size.

(8) Multiply/Multiply-and-Accumulate Operations

16-bit × 16-bit → 32-bit multiply operations are executed in one to two cycles. 16-bit × 64-bit → 64-bit multiply-and-accumulate operations are executed in two to three cycles. 32-bit → 64-bit multiply and 32-bit × 32-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to four cycles.

(9) T Bit

The T bit in the status register (SR) changes according to the result of the comparison. W conditional branch is taken or not taken depends upon the T bit condition (true/false). Th of instructions that change the T bit is kept to a minimum to improve the processing spe

(10) Immediate Data

Byte immediate data is located in an instruction code. Word or longword immediate data
located in instruction codes but in a memory table. The memory table is accessed by an in
data transfer instruction (MOV) using the PC relative addressing mode with displacement

With the SH-2A, 17- to 28-bit immediate data can be located in an instruction code. How
21- to 28-bit immediate data, an OR instruction must be executed after the data is transfe
register.

Table 2.5 Immediate Data Accessing

Classification	SH-2A CPU		Example of Other CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOVI20	#H'1234,R0	MOV.W	#H'1234,R0
20-bit immediate	MOVI20	#H'12345,R0	MOV.L	#H'12345,R0
28-bit immediate	MOVI20S	#H'12345,R0	MOV.L	#H'1234567,
	OR	#H'67,R0		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678
		.....		
	.DATA.L	H'12345678		

Note: @(disp, PC) accesses the immediate data.

Table 2.6 Absolute Address Accessing

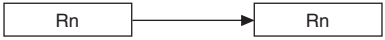
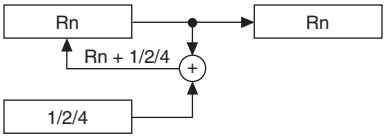
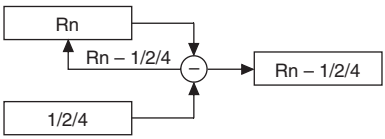
Classification	SH-2A CPU		Example of Other CPU	
Up to 20 bits	MOVI20	#H'12345,R1	MOV.B	@H'12345,R0
	MOV.B	@R1,R0		
21 to 28 bits	MOVI20S	#H'12345,R1	MOV.B	@H'1234567,R
	OR	#H'67,R1		
	MOV.B	@R1,R0		
29 bits or more	MOV.L	@(disp,PC),R1	MOV.B	@H'12345678,
	MOV.B	@R1,R0		
		.....		
	.DATA.L	H'12345678		

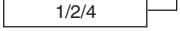
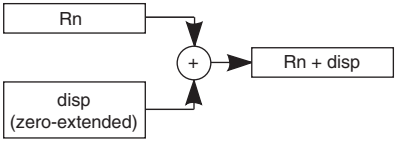
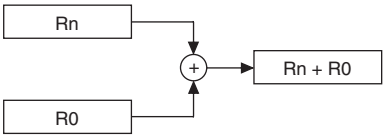
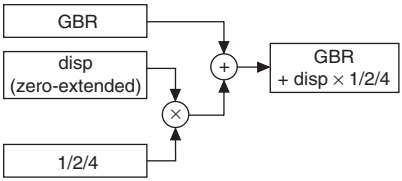
(12) 16-Bit/32-Bit Displacement

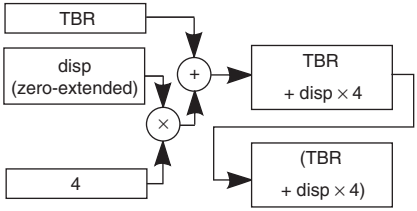
When data is accessed by 16-bit or 32-bit displacement, the displacement value should b
in the memory table in advance. That value is transferred to the register by loading the i
data during the execution of the instruction, and the data is accessed in the indexed indir
register addressing mode.

Table 2.7 Displacement Accessing

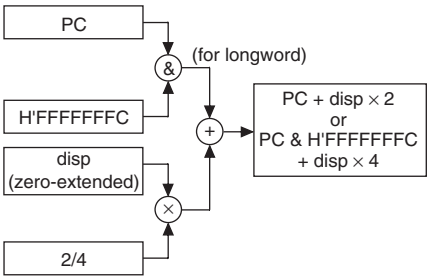
Classification	SH-2A CPU		Example of Other CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

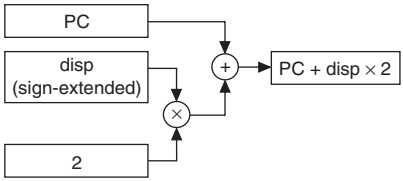
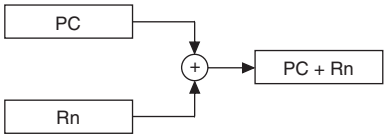
		
Register indirect `@Rn+` with post-increment	The effective address is the contents of register Rn. A constant is added to the contents of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (Afte instru exec Byte Rn + Word Rn + Long Rn +
Register indirect `@-Rn` with pre-decrement	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Byte Rn – Word Rn – Long Rn – (Inst exec Rn a calcu

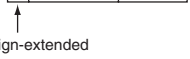
			
Register indirect with displacement	@(disp:12, Rn)	The effective address is the sum of Rn and a 12-bit displacement (disp). The value of disp is zero-extended. 	Byte Rn + Wor Rn + Long Rn +
Indexed register indirect	@(R0,Rn)	The effective address is the sum of Rn and R0. 	Rn +
GBR indirect with displacement	@(disp:8, GBR)	The effective address is the sum of GBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte GBR Wor GBR Long GBR



PC indirect with displacement	`@(disp:8, PC)`	The effective address is the sum of PC value and an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked.	Word PC + Longw PC & H'FFF disp ×

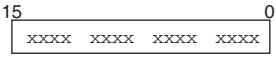
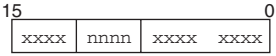


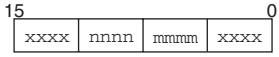
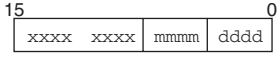
disp:12	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp).  	PC -
Rn	The effective address is the sum of PC value and Rn.  	PC -

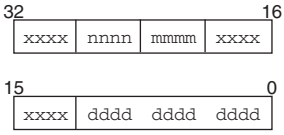
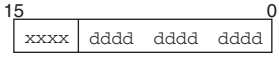
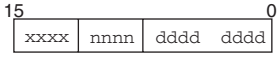


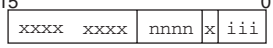
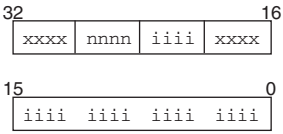
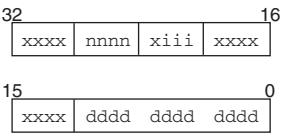
#imm:8	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.	—
#imm:8	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions is sign-extended.	—
#imm:8	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and then quadrupled.	—
#imm:3	The 3-bit immediate data (imm) for the BAND, BOR, BXOR, BST, BLD, BSET, and BCLR instructions indicates the target bit location.	—

Table 2.9 Instruction Formats

Instruction Formats	Source Operand	Destination Operand	Example
0 format 	—	—	NOP
n format 	—	nnnn: Register direct	MOVT
	Control register or system register	nnnn: Register direct	STS
	R0 (Register direct)	nnnn: Register direct	DIVU
	Control register or system register	nnnn: Register indirect with pre-decrement	STC.L
	mmmm: Register direct	R15 (Register indirect with pre-decrement)	MOVMU.L
	R15 (Register indirect with post-increment)	nnnn: Register direct	MOVMU.L
	R0 (Register direct)	nnnn: (Register indirect with post-increment)	MOV.L

	mmmm: PC relative using Rm	—	BRAF Rm
nm format  	mmmm: Register direct	nnnn: Register direct	ADD Rm,
	mmmm: Register direct	nnnn: Register indirect	MOV.L Rm,
	mmmm: Register indirect with post-increment (multiply-and-accumulate) nnnn*: Register indirect with post-increment (multiply-and-accumulate)	MACH, MACL	MAC.W @Rn
	mmmm: Register indirect with post-increment	nnnn: Register direct	MOV.L @R
	mmmm: Register direct	nnnn: Register indirect with pre-decrement	MOV.L Rm
	mmmm: Register direct	nnnn: Indexed register indirect	MOV.L Rm,@(R0,R
md format  	mmmmdddd: Register indirect with displacement	R0 (Register direct)	MOV.B @(disp,Rm

nmd12 format 	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(dis
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp12
d format 	dddddddd: GBR indirect with displacement	R0 (Register direct)	MOV.L @(disp,G
	R0 (Register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(dis
	dddddddd: PC relative with displacement	R0 (Register direct)	MOVA @(disp,P
	dddddddd: TBR duplicate indirect with displacement	—	JSR/N @@(disp8
	dddddddd: PC relative	—	BF l
d12 format 	ddddddddddd: PC relative	—	BRA l (label = PC)
nd8 format 	dddddddd: PC relative with displacement	nnnn: Register direct	MOV.L @(disp,P

	iii: Immediate		
	—	nnnn: Register direct iii: Immediate	BST #imm
ni20 format 	iiiiiiiiiiiiiiiii: Immediate	nnnn: Register direct	MOVI20 #imm20, Rn
nid format 	nnnnddddddddddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(dis
	—	nnnnddddddddddd: Register indirect with displacement iii: Immediate	BST.B #imm3,@(dis

Note: * In multiply-and-accumulate instructions, nnnn is the source register.

	Immediate data transfer
	Peripheral module data transfer
	Structure data transfer
	Reverse stack transfer
MOVA	Effective address transfer
MOVI20	20-bit immediate data transfer
MOVI20S	20-bit immediate data transfer
	8-bit left-shit
MOVML	R0−Rn register save/restore
MOVMU	Rn−R14 and PR register save/restore
MOVRT	T bit inversion and transfer to Rn
MOVT	T bit transfer
MOVU	Unsigned data transfer
NOTT	T bit inversion
PREF	Prefetch to operand cache
SWAP	Swap of upper and lower bytes
XTRCT	Extraction of the middle of registers connected

DIV0	Unsigned division (32 ÷ 32)
DIV1	One-step division
DIV0S	Initialization of signed one-step division
DIV0U	Initialization of unsigned one-step division
DMULS	Signed double-precision multiplication
DMULU	Unsigned double-precision multiplication
DT	Decrement and test
EXTS	Sign extension
EXTU	Zero extension
MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate operation
MUL	Double-precision multiply operation
MULR	Signed multiplication with result storage in Rn
MULS	Signed multiplication
MULU	Unsigned multiplication
NEG	Negation
NEGC	Negation with borrow
SUB	Binary subtraction
SUBC	Binary subtraction with borrow
SUBV	Binary subtraction with underflow

		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAD	Dynamic arithmetic shift	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLD	Dynamic logical shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	10	BF	Conditional branch, conditional delayed branch (branch when T = 0)	15
		BT	Conditional branch, conditional delayed branch (branch when T = 1)	
		BRA	Unconditional delayed branch	
		BRAF	Unconditional delayed branch	
		BSR	Delayed branch to subroutine procedure	
		BSRF	Delayed branch to subroutine procedure	
		JMP	Unconditional delayed branch	
		JSR	Branch to subroutine procedure	
			Delayed branch to subroutine procedure	
		RTS	Return from subroutine procedure	
			Delayed return from subroutine procedure	
		RTV/N	Return from subroutine procedure with Rm → R0 transfer	

		RTE	Return from exception handling	
		SETT	T bit set	
		SLEEP	Transition to power-down mode	
		STBANK	Register save to specified register bank entry	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	
Bit manipulation	10	BAND	Bit AND	14
		BCLR	Bit clear	
		BLD	Bit load	
		BOR	Bit OR	
		BSET	Bit set	
		BST	Bit store	
		BXOR	Bit exclusive OR	
		BANDNOT	Bit NOT AND	
		BORNOT	Bit NOT OR	
		BLDNOT	Bit NOT load	
Total:	91			19

SZ: Size
SRC: Source
DEST: Destination
Rm: Source register
Rn: Destination register
imm: Immediate data
disp: Displacement*2

0000: R0
0001: R1
·····
1111: R15
iiii: Immediate data
dddd: Displacement

M/Q/T: Flag bits in SR
&: Logical AND of each bit
|: Logical OR of each bit
^: Exclusive logical OR of each bit
~: Logical NOT of each bit
<<n: n-bit left shift
>>n: n-bit right shift

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are min
practice, the number of instruction execution states will be increased in cases
the following:
a. When there is a conflict between an instruction fetch and a data access
b. When the destination register of a load instruction (memory → register) is
as the register used by the next instruction.
2. Depending on the operand size, displacement is scaled by ×1, ×2, or ×4. For
refer to the SH-2A, SH2A-FPU Software Manual.

MOV.L	@(disp,PC),Rn	1101nnnndddddddd	(disp × 4 + PC) → Rn	1	—	Yes	Ye
MOV	Rm,Rn	0110nnnnmmmm0011	Rm → Rn	1	—	Yes	Ye
MOV.B	Rm,@Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—	Yes	Ye
MOV.W	Rm,@Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—	Yes	Ye
MOV.L	Rm,@Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—	Yes	Ye
MOV.B	@Rm,Rn	0110nnnnmmmm0000	(Rm) → sign extension → Rn	1	—	Yes	Ye
MOV.W	@Rm,Rn	0110nnnnmmmm0001	(Rm) → sign extension → Rn	1	—	Yes	Ye
MOV.L	@Rm,Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—	Yes	Ye
MOV.B	Rm,@-Rn	0010nnnnmmmm0100	Rn-1 → Rn, Rm → (Rn)	1	—	Yes	Ye
MOV.W	Rm,@-Rn	0010nnnnmmmm0101	Rn-2 → Rn, Rm → (Rn)	1	—	Yes	Ye
MOV.L	Rm,@-Rn	0010nnnnmmmm0110	Rn-4 → Rn, Rm → (Rn)	1	—	Yes	Ye
MOV.B	@Rm+,Rn	0110nnnnmmmm0100	(Rm) → sign extension → Rn, Rm + 1 → Rm	1	—	Yes	Ye
MOV.W	@Rm+,Rn	0110nnnnmmmm0101	(Rm) → sign extension → Rn, Rm + 2 → Rm	1	—	Yes	Ye
MOV.L	@Rm+,Rn	0110nnnnmmmm0110	(Rm) → Rn, Rm + 4 → Rm	1	—	Yes	Ye
MOV.B	R0,@(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—	Yes	Ye
MOV.W	R0,@(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—	Yes	Ye
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—	Yes	Ye
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp + Rm) → sign extension → R0	1	—	Yes	Ye
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → sign extension → R0	1	—	Yes	Ye
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—	Yes	Ye
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—	Yes	Ye

MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—	Yes	Y
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—	Yes	Y
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—	Yes	Y
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → sign extension → R0	1	—	Yes	Y
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → sign extension → R0	1	—	Yes	Y
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—	Yes	Y
MOV.B	R0,@Rn+	0100nnnn10001011	R0 → (Rn), Rn + 1 → Rn	1	—		
MOV.W	R0,@Rn+	0100nnnn10011011	R0 → (Rn), Rn + 2 → Rn	1	—		
MOV.L	R0,@Rn+	0100nnnn10101011	R0 → Rn), Rn + 4 → Rn	1	—		
MOV.B	@-Rm,R0	0100mmmm11001011	Rm-1 → Rm, (Rm) → sign extension → R0	1	—		
MOV.W	@-Rm,R0	0100mmmm11011011	Rm-2 → Rm, (Rm) → sign extension → R0	1	—		
MOV.L	@-Rm,R0	0100mmmm11101011	Rm-4 → Rm, (Rm) → R0	1	—		
MOV.B	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0000dddddddddddd	Rm → (disp + Rn)	1	—		
MOV.W	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0001dddddddddddd	Rm → (disp × 2 + Rn)	1	—		
MOV.L	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0010dddddddddddd	Rm → (disp × 4 + Rn)	1	—		
MOV.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 0100dddddddddddd	(disp + Rm) → sign extension → Rn	1	—		

		iiiiiiiiiiiiiiii		
MOVI20S #imm20,Rn	0000nnnniiii0001 iiiiiiiiiiiiiiii	imm << 8 → sign extension → Rn	1	—
MOVML.L Rm,@-R15	0100mmmm11110001	R15-4 → R15, Rm → (R15) R15-4 → R15, Rm-1 → (R15) : R15-4 → R15, R0 → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—
MOVML.L @R15+,Rn	0100nnnn11110101	(R15) → R0, R15 + 4 → R15 (R15) → R1, R15 + 4 → R15 : (R15) → Rn Note: When Rn = R15, read Rm as PR	1 to 16	—
MOVMU.L Rm,@-R15	0100mmmm11110000	R15-4 → R15, PR → (R15) R15-4 → R15, R14 → (R15) : R15-4 → R15, Rm → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—
MOVMU.L @R15+,Rn	0100nnnn11110100	(R15) → Rn, R15 + 4 → R15 (R15) → Rn + 1, R15 + 4 → R15 : (R15) → R14, R15 + 4 → R15 (R15) → PR Note: When Rn = R15, read Rm as PR	1 to 16	—

NOT		0000000000101000	T → T	1	Operation result		
PREF	@Rn	0000nnnn10000011	(Rn) → operand cache	1	—		
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → swap lower 2 bytes → Rn	1	—	Yes	
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → swap upper and lower words → Rn	1	—	Yes	
XTRCT	Rm,Rn	0010nnnnmmmm1101	Middle 32 bits of Rm:Rn → Rn	1	—	Yes	

ADDV	Rm,Rn	0011nnnnmmmm1111	Rn + Rm → Rn, overflow → T	1	Over-flow	Yes	Y
CMP/EQ	#imm,R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y
CMP/GE	Rm,Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y
CMP/HI	Rm,Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y
CMP/STR	Rm,Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Y

CLIPU.B	Rn	0100nnnn10000001	When Rn > (H'000000FF), (H'000000FF) → Rn, 1 → CS	1	—	
CLIPU.W	Rn	0100nnnn10000101	When Rn > (H'0000FFFF), (H'0000FFFF) → Rn, 1 → CS	1	—	
DIV1	Rm,Rn	0011nnnnmmmm0100	1-step division (Rn ÷ Rm)	1	Calculation result	Yes
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calculation result	Yes
DIV0U		0000000000011001	0 → M/Q/T	1	0	Yes
DIVS	R0,Rn	0100nnnn10010100	Signed operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	36	—	
DIVU	R0,Rn	0100nnnn10000100	Unsigned operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	34	—	
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes
DT	Rn	0100nnnn00010000	Rn – 1 → Rn When Rn is 0, 1 → T When Rn is not 0, 0 → T	1	Comparison result	Yes
EXTS.B	Rm,Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended → Rn	1	—	Yes
EXTS.W	Rm,Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—	Yes

			(Rm) + MAC → MAC 16 × 16 + 64 → 64 bits				
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	2	—	Yes	Y
MULR	R0,Rn	0100nnnn10000000	R0 × Rn → Rn 32 × 32 → 32 bits	2			
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Y
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Y
NEG	Rm,Rn	0110nnnnmmmm1011	0-Rm → Rn	1	—	Yes	Y
NEGC	Rm,Rn	0110nnnnmmmm1010	0-Rm-T → Rn, borrow → T	1	Borrow	Yes	Y
SUB	Rm,Rn	0011nnnnmmmm1000	Rn-Rm → Rn	1	—	Yes	Y
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn-Rm-T → Rn, borrow → T	1	Borrow	Yes	Y
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn-Rm → Rn, underflow → T	1	Over-flow	Yes	Y

			(R0 + GBR)				
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	1	—	Yes	Y
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—	Yes	Y
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	1	—	Yes	Y
OR.B	#imm,@(R0,GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—	Yes	Y
TAS.B	@Rn	0100nnnn00011011	When (Rn) is 0, 1 → T Otherwise, 0 → T, 1 → MSB of(Rn)	3	Test result	Yes	Y
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Y
TST	#imm,R0	11001000iiiiiiii	R0 & imm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Y
TST.B	#imm,@(R0,GBR)	11001100iiiiiiii	(R0 + GBR) & imm When the result is 0, 1 → T Otherwise, 0 → T	3	Test result	Yes	Y
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—	Yes	Y
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	1	—	Yes	Y
XOR.B	#imm,@(R0,GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—	Yes	Y

ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB	Yes	Ye
SHAD	Rm,Rn	0100nnnnmmmm1100	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [MSB → Rn]	1	—		Ye
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB	Yes	Ye
SHAR	Rn	0100nnnn00100001	MSB → Rn → T	1	LSB	Yes	Ye
SHLD	Rm,Rn	0100nnnnmmmm1101	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [0 → Rn]	1	—		Ye
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB	Yes	Ye
SHLR	Rn	0100nnnn00000001	0 → Rn → T	1	LSB	Yes	Ye
SHLL2	Rn	0100nnnn00001000	Rn << 2 → Rn	1	—	Yes	Ye
SHLR2	Rn	0100nnnn00001001	Rn >> 2 → Rn	1	—	Yes	Ye
SHLL8	Rn	0100nnnn00011000	Rn << 8 → Rn	1	—	Yes	Ye
SHLR8	Rn	0100nnnn00011001	Rn >> 8 → Rn	1	—	Yes	Ye
SHLL16	Rn	0100nnnn00101000	Rn << 16 → Rn	1	—	Yes	Ye
SHLR16	Rn	0100nnnn00101001	Rn >> 16 → Rn	1	—	Yes	Ye

			When T = 0, disp × 2 + PC → PC, When T = 1, nop				
BT	label	10001001dddddddd	When T = 1, disp × 2 + PC → PC, When T = 0, nop	3/1*	—	Yes	Y
BT/S	label	10001101dddddddd	Delayed branch When T = 1, disp × 2 + PC → PC, When T = 0, nop	2/1*	—	Yes	Y
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—	Yes	Y
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—	Yes	Y
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—	Yes	Y
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2	—	Yes	Y
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—	Yes	Y
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—	Yes	Y
JSR/N	@Rm	0100mmmm01001011	PC-2 → PR, Rm → PC	3	—		
JSR/N	@@(disp8,TBR)	10000011dddddddd	PC-2 → PR, (disp × 4 + TBR) → PC	5	—		
RTS		0000000000001011	Delayed branch, PR → PC	2	—	Yes	Y
RTS/N		0000000001101011	PR → PC	3	—		
RTV/N	Rm	0000mmmm01111011	Rm → R0, PR → PC	3	—		

Note: * One cycle when the program does not branch.

LDC	Rm,SR	0100mmmm00001110	Rm → SR	3	LSB	Yes	Ye
LDC	Rm,TBR	0100mmmm01001010	Rm → TBR	1	—		
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—	Yes	Ye
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—	Yes	Ye
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	5	LSB	Yes	Ye
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	1	—	Yes	Ye
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	1	—	Yes	Ye
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—	Yes	Ye
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—	Yes	Ye
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—	Yes	Ye
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—	Yes	Ye
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—	Yes	Ye
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—	Yes	Ye
NOP		0000000000001001	No operation	1	—	Yes	Ye
RESBANK		0000000001011011	Bank → R0 to R14, GBR, MACH, MACL, PR	9*	—		
RTE		0000000000101011	Delayed branch, stack area → PC/SR	6	—	Yes	Ye
SETT		0000000000011000	1 → T	1	1	Yes	Ye
SLEEP		0000000000011011	Sleep	5	—	Yes	Ye
STBANK	R0,@Rn	0100nnnn11100001	R0 → (specified register bank entry)	7	—		
STC	SR,Rn	0000nnnn00000010	SR → Rn	2	—	Yes	Ye
STC	TBR,Rn	0000nnnn01001010	TBR → Rn	1	—		

STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—	Yes	Y
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—	Yes	Y
STS.L	MACH,@-Rn	0100nnnn00000010	Rn-4 → Rn, MACH → (Rn)	1	—	Yes	Y
STS.L	MACL,@-Rn	0100nnnn00010010	Rn-4 → Rn, MACL → (Rn)	1	—	Yes	Y
STS.L	PR,@-Rn	0100nnnn00100010	Rn-4 → Rn, PR → (Rn)	1	—	Yes	Y
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	5	—	Yes	Y

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are min
practice, the number of instruction execution states in cases such as the follo

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is as the register used by the next instruction.

* In the event of bank overflow, the number of cycles is 19.

		1100ddddddddddddd			ration result
BCLR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0000ddddddddddddd	0 → (imm of (disp + Rn))	3	—
BCLR	#imm3,Rn	10000110nnnn0iii	0 → imm of Rn	1	—
BLD.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0011ddddddddddddd	(imm of (disp + Rn)) →	3	Operation result
BLD	#imm3,Rn	10000111nnnn1iii	imm of Rn → T	1	Operation result
BLDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1011ddddddddddddd	~(imm of (disp + Rn)) → T	3	Operation result
BOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0101ddddddddddddd	(imm of (disp + Rn)) | T → T	3	Operation result
BORNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1101ddddddddddddd	~(imm of (disp + Rn)) | T → T	3	Operation result
BSET.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0001ddddddddddddd	1 → (imm of (disp + Rn))	3	—
BSET	#imm3,Rn	10000110nnnn1iii	1 → imm of Rn	1	—
BST.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0010ddddddddddddd	T → (imm of (disp + Rn))	3	—
BST	#imm3,Rn	10000111nnnn0iii	T → imm of Rn	1	—

and power-down. Figure 2.6 shows the transitions between the states.

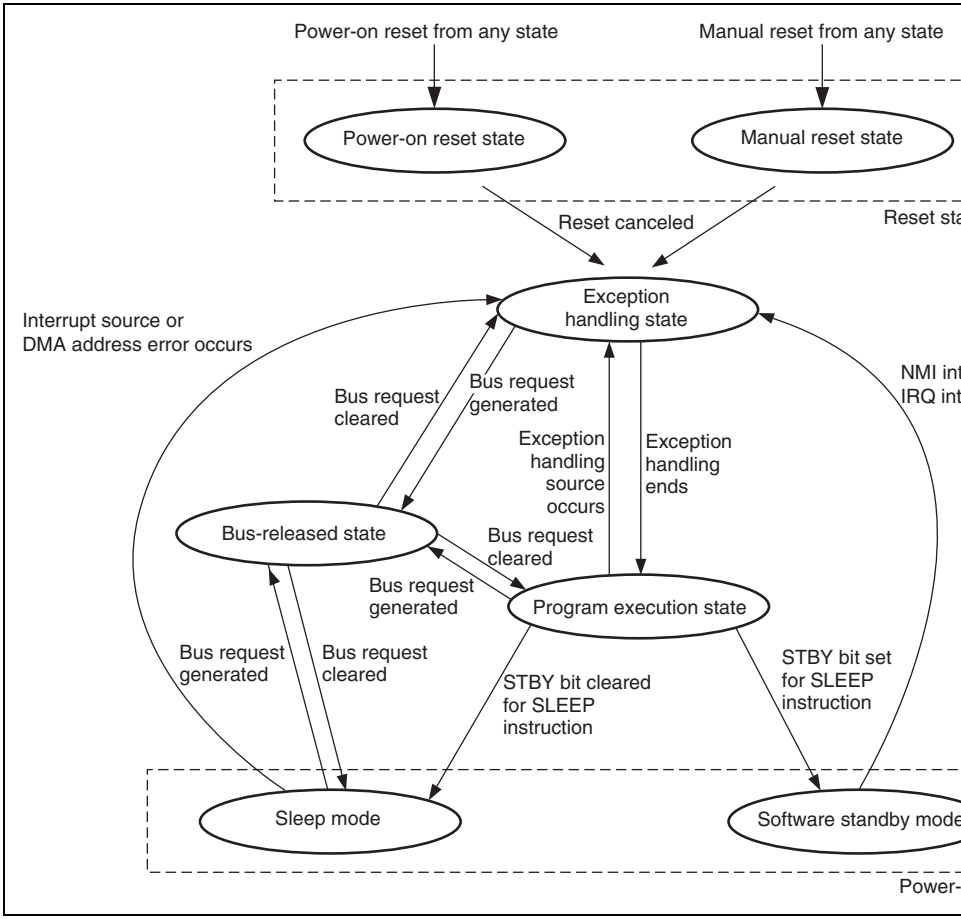


Figure 2.6 Transitions between Processing States

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and stat
register (SR) are saved to the stack area. The exception service routine start address is fet
from the exception handling vector table; the CPU then branches to that address and the p
starts executing, thereby entering the program execution state.

(3) Program Execution State

In the program execution state, the CPU sequentially executes the program.

(4) Power-Down State

In the power-down state, the CPU stops operating to reduce power consumption. The SL
instruction places the CPU in the sleep mode or the software standby mode.

(5) Bus-Released State

In the bus-released state, the CPU releases bus to a device that has requested it.

The MCU operating mode can be selected from MCU extension modes 0 to 2 and single mode. For the on-chip flash memory programming mode, boot mode, user boot mode, a program mode which are on-chip programming modes are available.

Table 3.1 Selection of Operating Modes

	Pin Setting					Bus Width of CS0
Mode No.	FWE	MD1	MD0	Mode Name	On-Chip ROM	SH7211F
Mode 0	0	0	0	MCU extension mode 0	Not active	16
Mode 1	0	0	1	MCU extension mode 1	Not active	8
Mode 2	0	1	0	MCU extension mode 2	Active	Set by CS0BCR
Mode 3	0	1	1	Single chip mode	Active	—
Mode 4*	1	0	0	Boot mode	Active	Set by CS0BCR
Mode 5*	1	0	1	User boot mode	Active	Set by CS0BCR
Mode 6*	1	1	0	User program mode	Active	Set by CS0BCR

Note: * Flash memory programming mode.

3.3.3 Mode 2 (MCU Extension Mode 2)

The on-chip ROM is active and CS0 space can be used in this mode.

3.3.4 Mode 3 (Single Chip Mode)

All ports can be used in this mode, however the external address cannot be used.

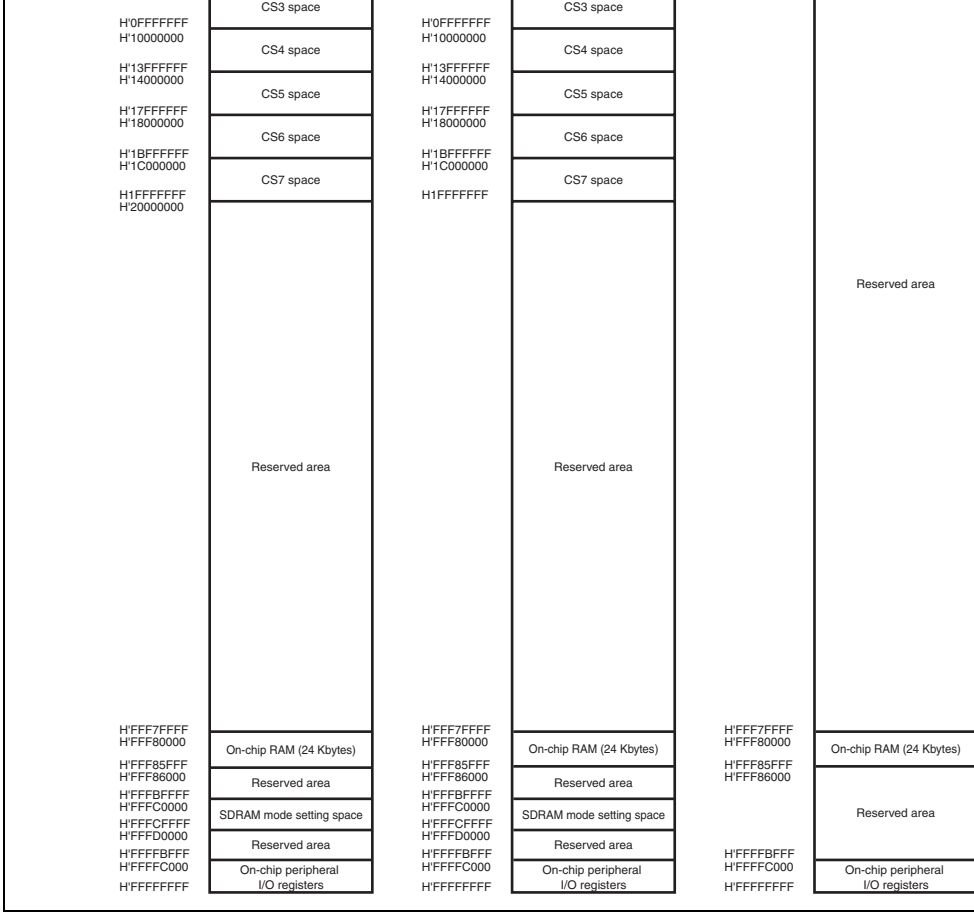


Figure 3.1 Address Map for Each Operating Mode (384-Kbyte On-Chip ROM Version)

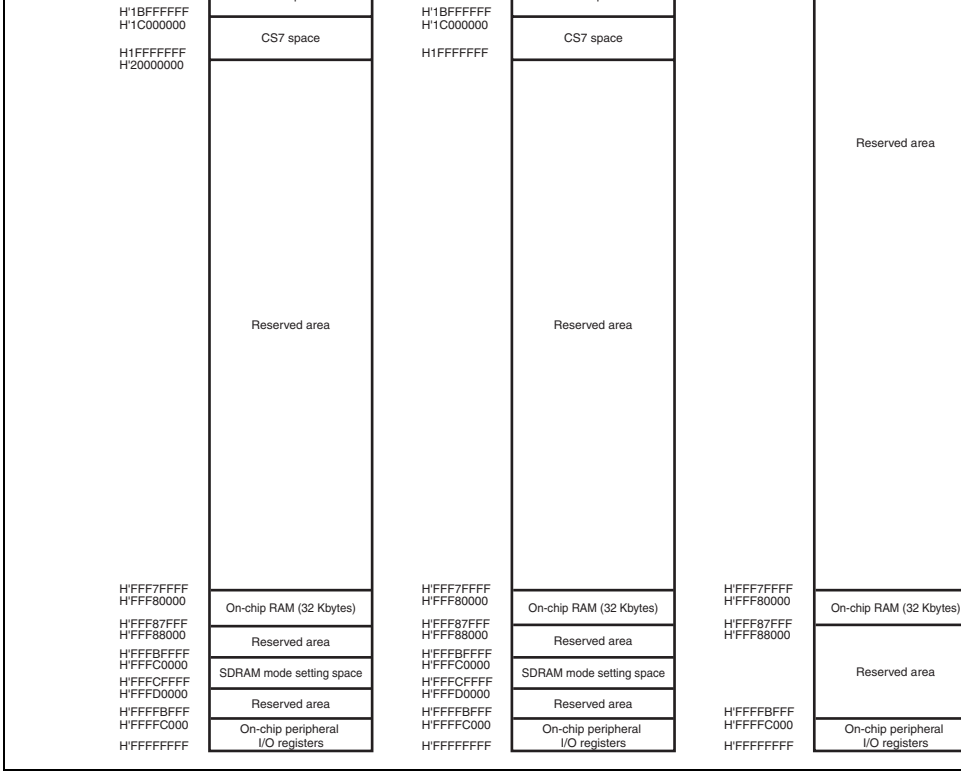
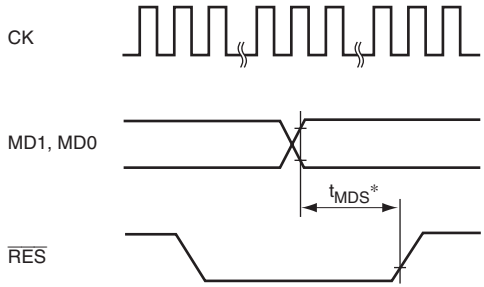


Figure 3.2 Address Map for Each Operating Mode (512-Kbyte On-Chip ROM Version)



Note: * See section 27.4.2, Control Signal Timing.

Figure 3.3 Reset Input Timing when Changing Operating Mode

selected.

- Five clocks generated independently

  An internal clock (Iϕ) for the CPU, a peripheral clock (Pϕ) for the peripheral module
  clock (Bϕ = CK) for the external bus interface, an MTU2S clock (Mϕ) for the MTU2
  and an AD clock (Aϕ) for the ADC module can be generated independently.

- Frequency change function

  Internal and peripheral clock frequencies can be changed independently using the PL
  locked loop) circuits and divider circuits within the CPG. Frequencies are changed b
  using frequency control register (FRQCR) settings.

- Power-down mode control

  The clock can be stopped for sleep mode and software standby mode, and specific m
  be stopped using the module standby function. For details on clock control in the pow
  modes, see section 23, Power-Down Modes.

Figure 4.1 shows a block diagram of the clock pulse generator.

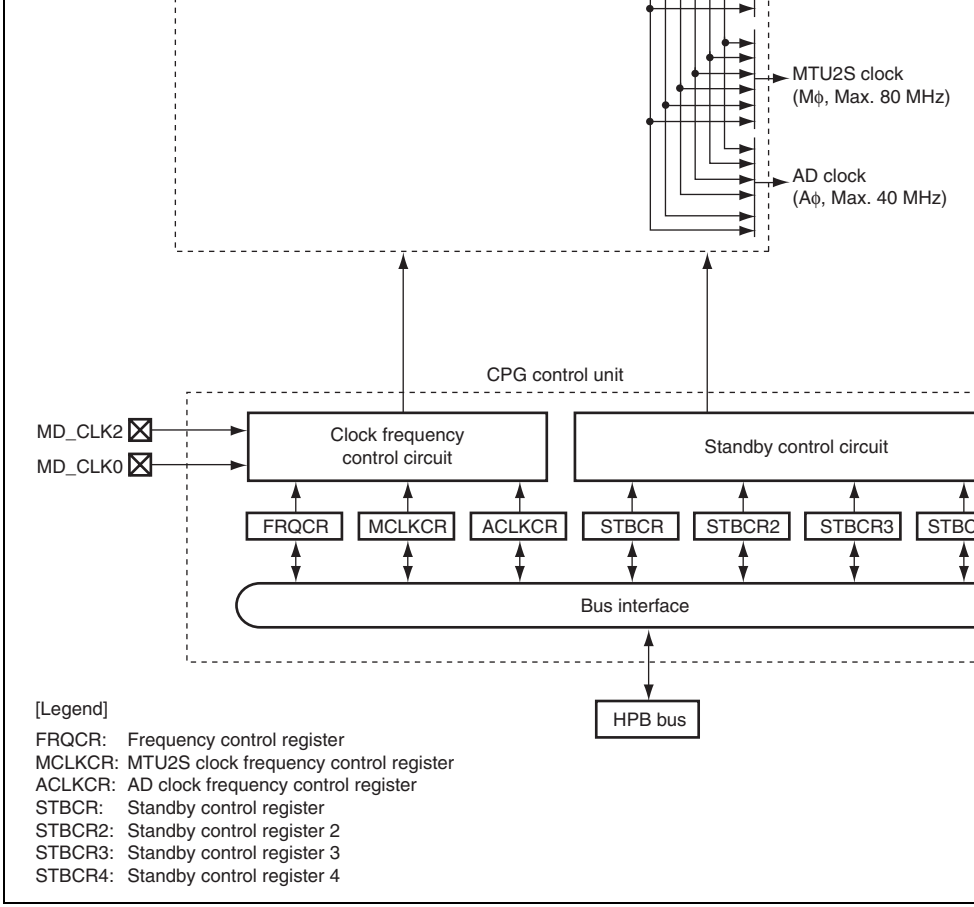


Figure 4.1 Block Diagram of Clock Pulse Generator

The multiplication rate is fixed according to the clock operating mode. The clock operat
is specified by the MD_CLK0 and MD_CLK2 pins. For details on the clock operating m
table 4.2.

(3) Crystal Oscillator

The crystal oscillator is an oscillation circuit in which a crystal resonator is connected to
XTAL pin or EXTAL pin. This can be used according to the clock operating mode.

(4) Divider 1

Divider 1 generates a clock signal at the operating frequency used by the internal clock (
bus clock (Bϕ), the peripheral clock (Pϕ), the MTU2S clock (Mϕ), or the AD clock (Aϕ)
operating frequency can be 1, 1/2, 1/4, or 1/8 times the output frequency of PLL circuit
However, set the internal clock (Iϕ) so that its frequency is not less than the clock freque
CK pin, and set the peripheral clock (Pϕ) so that its frequency is not more than the clock
of the CK pin. The division ratio is set in the frequency control register (FRQCR).

(5) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the MD_CLK0 a
MD_CLK2 pins and the frequency control register (FRQCR).

(8) MTU2S Clock Frequency Control Register (MCLKCR)

The MTU2S clock frequency control register (MCLKCR) has control bits assigned for th
following functions: MTU2S clock (Mϕ) output/non-output and the frequency division ra

(9) AD Clock Frequency Control Register (ACLKCR)

The AD clock frequency control register (ACLKCR) has control bits assigned for the foll
functions: AD clock (Aϕ) output/non-output and the frequency division ratio.

(10) Standby Control Register

The standby control register has bits for controlling the power-down modes. See section 2
Power-Down Modes, for more information.

pins)	EXTAL	Input	Connected to the crystal resonator or used to input an e clock.
Clock output pin	CK	Output	Clock output pin. This pin can be high impedance.

- Mode 6

  The frequency of the signal received from the EXTAL pin or crystal oscillator is quad
  by the PLL circuit 2 before it is supplied to the LSI as the clock signal. This allows a
  with a lower frequency to be used. Either a crystal resonator with a frequency in the r
  from 8 to 10 MHz or an external signal in the same frequency range input on the EXT
  can be used. When the CK output is in use, the frequency range is from 32 to 40 MHz
  an input signal on the EXTAL pin is in use, the XTAL pin should be left open.

H'1103	On (× 2)	On (×4)	8:4:2	8 to 10	32 to 40	64 to 80	32 to 40
H'1105	On (× 2)	On (×4)	8:4:1	8 to 10	32 to 40	64 to 80	32 to 40
H'1111	On (× 2)	On (×4)	4:4:4	8 to 10	32 to 40	32 to 40	32 to 40
H'1113	On (× 2)	On (×4)	4:4:2	8 to 10	32 to 40	32 to 40	32 to 40
H'1115	On (× 2)	On (×4)	4:4:1	8 to 10	32 to 40	32 to 40	32 to 40
H'1303	On (× 4)	On (×4)	16:4:4	8 to 10	32 to 40	128 to 160	32 to 40
H'1305	On (× 4)	On (×4)	16:4:2	8 to 10	32 to 40	128 to 160	32 to 40
H'1313	On (× 4)	On (×4)	8:4:4	8 to 10	32 to 40	64 to 80	32 to 40
H'1315	On (× 4)	On (×4)	8:4:2	8 to 10	32 to 40	64 to 80	32 to 40
H'1333	On (× 4)	On (×4)	4:4:4	8 to 10	32 to 40	32 to 40	32 to 40
H'1335	On (× 4)	On (×4)	4:4:2	8 to 10	32 to 40	32 to 40	32 to 40

Notes: 1. The ratio of clock frequencies, where the input clock frequency is assumed to be 1.

2. The frequency of the clock input from the EXTAL pin or the frequency of the crystal

Caution: 1. The frequency of the internal clock (Iϕ) is the frequency of the signal input to the CK multiplication by the frequency-multiplier of PLL circuit 1 and division by the divider's the frequency of the internal clock to 160 MHz or less but not less than the frequenc signal on the CK pin.

2. The frequency of the peripheral clock (Pϕ) is the frequency of the signal input to the multiplication by the frequency-multiplier of PLL circuit 1 and division by the divider's the frequency of the peripheral clock to 40 MHz or less. In addition, do not set a high frequency for the internal clock than the frequency on the CK pin.

3. The frequency multiplier of PLL circuit 1 can be selected as ×1, ×2, or ×4. The divisc divider can be selected as ×1, ×1/2, ×1/4, or ×1/8. The settings are made in the frequ control register (FRQCR).

4. The signal output by PLL circuit 1 is the signal on the CK pin multiplied by the freque multiplier of PLL circuit 1. Ensure that the frequency of the signal from PLL circuit 1 than 160 MHz.

4.4.1 Frequency Control Register (FRQCR)

FRQCR is a 16-bit readable/writable register used to specify whether a clock is output fro
CK pin in software standby mode, the frequency multiplication ratio of PLL circuit 1, and
frequency division ratio of the internal clock (Iϕ) and peripheral clock (Pϕ). Only word ac
be used on FRQCR.

FRQCR is initialized to H'1003 only by a power-on reset. FRQCR retains its previous va
manual reset or in software standby mode. The previous value is also retained when an in
reset is triggered by an overflow of the WDT.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	-	-	-	CKOEN	-	-	STC[1:0]		-	IFC[2:0]			RNGS	PF	
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				prevented. 0: The CK pin is fixed to the low level during standby mode or when exiting software s mode. 1: Clock is output from the CK pin (placed in impedance state during software standby
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
9, 8	STC[1:0]	00	R/W	Frequency multiplication ratio of PLL circuit 00: × 1 time 01: × 2 times 10: Setting prohibited 11: × 4 times
7	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

				Other than above: Setting prohibited
3	RNGS	0	R/W	Set this bit according to the output frequency circuit 1. 0: High-frequency mode 1: Low-frequency mode Always specify high-frequency mode for this Do not set this bit to 1.
2 to 0	PFC[2:0]	011	R/W	Peripheral Clock (Pϕ) Frequency Division Ra These bits specify the frequency division rati peripheral clock with respect to the output fre of PLL circuit 1. If a prohibited value is specified, correct oper cannot be guaranteed. 000: × 1 time 001: × 1/2 time 011: × 1/4 time 101: × 1/8 time Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	MSSCS[1:0]	01	R/W	Source Clock Select These bits select the source clock. 00: Clock stop 01: PLL1 output clock 10: Reserved (setting prohibited) 11: Reserved (setting prohibited)
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
1, 0	MSDIVS[1:0]	11	R/W	Division Ratio Select These bits specify the frequency division rat source clock. Set these bits so that the outp 80 MHz or less, and also an integer multiple peripheral clock frequency (Pϕ). 00: × 1 time 01: × 1/2 time 10: Setting prohibited 11: × 1/4 time

Bit	Bit Name	Initial Value	R/W	Description
7, 6	ASSCS[1:0]	01	R/W	Source Clock Select These bits select the source clock. 00: Clock stoppage 01: PLL1 output clock 10: Reserved (setting prohibited) 11: Reserved (setting prohibited)
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
1, 0	ASDIVS[1:0]	11	R/W	Division Ratio Select These bits specify the frequency division ratio source clock. Set these bits so that the outpu 40 MHz or less, and also an integer multiple peripheral clock frequency (Pϕ). 00: × 1 time 01: × 1/2 time 10: Setting prohibited 11: × 1/4 time

multiplication rate is changed, the LSI temporarily stops automatically and the internal
timer (WDT) starts counting the settling time. When the count of the WDT overflows, t
restarts operating with the set clock frequency. The following shows this setting procedu

1. In the initial state, the multiplication rate of PLL circuit 1 is 1 time.
2. Set a value that will become the specified oscillation settling time in the WDT and s
   WDT. The following must be set:

   WTCSR.TME = 0: WDT stops

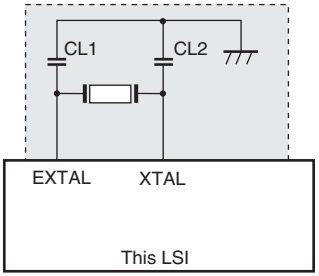
   WTCSR.CKS[2:0]: Division ratio of WDT count clock

   WTCNT counter: Initial counter value

   For setting of the counter, determine the overflow period with the frequency after th
   peripheral clock (Pϕ) setting change.
3. Set the desired value in the STC[1:0] bits. The division ratio can also be set in the IF
   PFC[2:0] bits.
4. This LSI pauses temporarily and the WDT starts incrementing. The internal and peri
   clocks both stop and the WDT is supplied with the clock. The clock will continue to
   at the CK pin. This state is the same as software standby mode. Whether or not regis
   initialized depends on the module. For details, see table 23.4 in section 23, Power-D
   Modes.
5. Supply of the clock that has been set begins at WDT count overflow, and this LSI be
   operating again. The WDT stops counting after it overflows.

division ratio.

Note: When executing the SLEEP instruction after the frequency has been changed, be
read the frequency control register (FRQCR) three times before executing the SL
instruction.



Reference value
CL1 = 20 pF
CL2 = 20 pF

Note: The values for CL1 and CL2 should be determined after consultation with the crystal resonator manufacturer.

Figure 4.2 Note on Using a Crystal Resonator

4.6.2 Note on Bypass Capacitor

A multilayer ceramic capacitor should be inserted for each pair of Vss and Vcc as a byp capacitor as many as possible. The bypass capacitor must be inserted as close to the pow pins of the LSI as possible. Note that the capacitance and frequency characteristics of th capacitor must be appropriate for the operating frequency of the LSI.

4.6.3 Note on Using a PLL Oscillation Circuit

In the PLLVcc and PLLVss connection pattern for the PLL, signal lines from the board supply pins must be as short as possible and pattern width must be as wide as possible to inductive interference.

Since the analog power supply pins of the PLL are sensitive to the noise, the system may malfunction due to inductive interference at the other power supply pins. To prevent such malfunction, the analog power supply pin Vcc and digital power supply pin VccQ should supply the same resources on the board if at all possible.

Table 5.1 Types of Exception Handling and Priority Order

Type	Exception Handling	
Reset	Power-on reset	
	Manual reset	
Address error	CPU address error	
	DMAC address error	
Instruction	Integer division exception (division by zero)	
	Integer division exception (overflow)	
Register bank error	Bank underflow	
	Bank overflow	
Interrupt	NMI	
	User break	
	H-UDI	
	IRQ	
	On-chip peripheral modules	A/D converter (ADC)
		Direct memory access controller (DMAC)
		Compare match timer (CMT)
		Bus state controller (BSC)
		Watchdog timer (WDT)
		Multi-function timer pulse unit 2 (MTU2)
		Port output enable 2 (POE2): OEI1 and OEI2 interrupts

branch instruction*1, instructions that rewrite the PC*2, 32-bit instructions*3, RESBANK instruction, DIVS instruction, and DIVU instruction)

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, B BRAF.

2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TR BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N.

3. 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR. BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.

	Manual reset	Starts when the MRES pin changes from low to high the WDT overflows.
Address error		Detected when instruction is decoded and starts whe previous executing instruction finishes executing.
Interrupts		Detected when instruction is decoded and starts whe previous executing instruction finishes executing.
Register bank error	Bank underflow	Starts upon attempted execution of a RESBANK instr when saving has not been performed to register bank
	Bank overflow	In the state where saving has been performed to all r bank areas, starts when acceptance of register bank exception has been set by the interrupt controller (the in IBNR of the INTC is 1) and an interrupt that uses a bank has occurred and been accepted by the CPU.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime e immediately after a delayed branch instruction (delay
	Slot illegal instructions	Starts from the decoding of undefined code placed im after a delayed branch instruction (delay slot), of instr that rewrite the PC, of 32-bit instructions, of the RESE instruction, of the DIVS instruction, or of the DIVU ins
	Integer division instructions	Starts when detecting division-by-zero exception or o exception caused by division of the negative maximu (H'80000000) by −1.

from the PC address fetched from the exception handling vector table.

(2) Exception Handling Triggered by Address Errors, Register Bank Errors, Inter and Instructions

SR and PC are saved to the stack indicated by R15. In the case of interrupt exception han other than NMI or UBC with usage of the register banks enabled, general registers R0 to control register GBR, system registers MACH, MACL, and PR, and the vector number o interrupt exception handling to be executed are saved to the register banks. In the case of exception handling due to an address error, register bank error, NMI interrupt, UBC inter instruction, saving to a register bank is not performed. When saving is performed to all re banks, automatic saving to the stack is performed instead of register bank saving. In this c interrupt controller setting must have been made so that register bank overflow exception accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept register bank ov exceptions has been made (the BOVE bit in IBNR of the INTC is 1), register bank overfl exception will be generated. In the case of interrupt exception handling, the interrupt prio is written to the I3 to I0 bits in SR. In the case of exception handling due to an address er instruction, the I3 to I0 bits are not affected. The start address is then fetched from the ex handling vector table and the program begins running from that address.

Table 5.3 shows the vector numbers and vector table address offsets. Table 5.4 shows ho
table addresses are calculated.

Table 5.3 Exception Handling Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset
Power-on reset	PC	0	H'00000000 to H'00000003
	SP	1	H'00000004 to H'00000007
Manual reset	PC	2	H'00000008 to H'0000000B
	SP	3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved by system)		5	H'00000014 to H'00000017
Slot illegal instruction		6	H'00000018 to H'0000001B
(Reserved by system)		7	H'0000001C to H'0000001F
		8	H'00000020 to H'00000023
CPU address error		9	H'00000024 to H'00000027
DMAC address error		10	H'00000028 to H'0000002B
Interrupts	NMI	11	H'0000002C to H'0000002F
	User break	12	H'00000030 to H'00000033
(Reserved by system)		13	H'00000034 to H'00000037
H-UDI		14	H'00000038 to H'0000003B
Bank overflow		15	H'0000003C to H'0000003F
Bank underflow		16	H'00000040 to H'00000043

	63	H'000000FC to H'000000FF
External interrupts (IRQ), on-chip peripheral module interrupts*	64	H'00000100 to H'00000103
	:	:
	511	H'000007FC to H'000007FF

Note: * The vector numbers and vector table address offsets for each external interrup chip peripheral module interrupt are given in table 6.4 in section 6, Interrupt Co (INTC).

Table 5.4 Calculating Exception Handling Vector Table Addresses

Exception Source	Vector Table Address Calculation
Resets	Vector table address = (vector table address offset) = (vector number) × 4
Address errors, register bank errors, interrupts, instructions	Vector table address = VBR + (vector table address offs = VBR + (vector number) × 4

Notes: 1. Vector table address offset: See table 5.3.
2. Vector number: See table 5.3.

	Conditions for Transition to Reset State			Internal States		
Type	$\overline{\text{RES}}$ or $\overline{\text{MRES}}$	H-UDI Command	WDT Overflow	CPU	On-Chip Peripheral Modules, I/O Port	WRCS FRQCI
Power-on reset	Low	—	—	Initialized	Initialized	Initializ
	High	H-UDI reset assert command is set	—	Initialized	Initialized	Initializ
	High	Command other than H-UDI reset assert is set	Power-on reset	Initialized	Initialized	Not init
Manual reset	Low	—	—	Initialized	Not initialized*	Not init
	High	—	Manual reset	Initialized	Not initialized*	Not init

Note: * The BN bit in IBNR of the INTC is initialized.

In the power-on reset state, power-on reset exception handling starts when the $\overline{\text{RES}}$ pin is driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vecto
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits a initialized. The BN bit in IBNR of the INTC is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, program begins executing.

Be certain to always perform power-on reset processing when turning the system power o

(2) Power-On Reset by Means of H-UDI Reset Assert Command

When the H-UDI reset assert command is set, this LSI enters the power-on reset state. Po reset by means of an H-UDI reset assert command is equivalent to power-on reset by mea $\overline{\text{RES}}$ pin. Setting the H-UDI reset negate command cancels the power-on reset state. The required between an H-UDI reset assert command and H-UDI reset negate command is th as the time to keep the $\overline{\text{RES}}$ pin low to initiate a power-on reset. In the power-on reset sta generated by an H-UDI reset assert command, setting the H-UDI reset negate command s power-on reset exception handling. The CPU operates in the same way as when a power- was caused by the $\overline{\text{RES}}$ pin.

exception processing is started by the WDT, the CPU operates in the same way as when on reset was caused by the $\overline{\text{RES}}$ pin.

exception handling vector table.

2. The initial value of the stack pointer (SP) is fetched from the exception handling vecto
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits a initialized. The BN bit in IBNR of the INTC is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, program begins executing.

(2) Manual Reset Initiated by WDT

When a setting is made for a manual reset to be generated in the WDT's watchdog timer and WTCNT of the WDT overflows, this LSI enters the manual reset state.

When manual reset exception processing is started by the WDT, the CPU operates in the way as when a manual reset was caused by the $\overline{\text{MRES}}$ pin.

When a manual reset is generated, the bus cycle is retained, but if a manual reset occurs v bus is released or during DMAC burst transfer, manual reset exception handling will be d until the CPU acquires the bus. However, if the interval from generation of the manual re the end of the bus cycle is equal to or longer than the fixed internal manual reset interval the internal manual reset source is ignored instead of being deferred, and manual reset ex handling is not executed.

Instruction fetch	CPU	Instruction fetched from even address	None (norma
		Instruction fetched from odd address	Address erro
		Instruction fetched from other than on-chip peripheral module space* or H'F0000000 to H'F5FFFFFF in on-chip RAM space*	None (norma
		Instruction fetched from on-chip peripheral module space* or H'F0000000 to H'F5FFFFFF in on-chip RAM space*	Address erro
		Instruction fetched from external memory space in single-chip mode	Address erro
Data read/write	CPU or DMAC	Word data accessed from even address	None (norma
		Word data accessed from odd address	Address erro
		Longword data accessed from a longword boundary	None (norma
		Longword data accessed from other than a long-word boundary	Address erro
		Byte or word data accessed in on-chip peripheral module space*	None (norma
		Longword data accessed in 16-bit on-chip peripheral module space*	None (norma
		Longword data accessed in 8-bit on-chip peripheral module space*	None (norma
		Instruction fetched from external memory space in single-chip mode	Address erro

Note: * See section 8, Bus State Controller (BSC), for details of the on-chip peripher space and on-chip RAM space.

4. After jumping to the address fetched from the exception handling vector table, progra
execution starts. The jump that occurs is not a delayed branch.

Note: * This is the case in which an address error was caused by data read or write. W
address error is caused by an instruction fetch, and if the bus cycle in which th
error occurred does not end by step 3 above, the CPU restarts the address erro
exception handling until the bus cycle ends.

(2) Bank Underflow

Bank underflow occurs when an attempt is made to execute a RESBANK instruction wh
has not been performed to register banks.

5.4.2 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. The CP
as follows:

1. The exception service routine start address which corresponds to the register bank er
   occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start addr
   instruction to be executed after the last executed instruction for a bank overflow, and
   address of the executed RESBANK instruction for a bank underflow.

   To prevent multiple interrupts from occurring at a bank overflow, the interrupt priori
   that caused the bank overflow is written to the interrupt mask level bits (I3 to I0) of t
   register (SR).
4. After jumping to the address fetched from the exception handling vector table, progr
   execution starts. The jump that occurs is not a delayed branch.

NMI	NMI pin (external input)	1
User break	User break controller (UBC)	1
H-UDI	User debugging interface (H-UDI)	1
IRQ	IRQ0 to IRQ7 pins (external input)	8
On-chip peripheral module	A/D converter (ADC)	1
	Direct memory access controller (DMAC)	16
	Compare match timer (CMT)	2
	Bus state controller (BSC)	1
	Watchdog timer (WDT)	1
	Multi-function timer pulse unit 2 (MTU2)	26
	Multi-function timer pulse unit 2S (MTU2S)	13
	Port output enable 2 (POE2)	3
	I^2C bus interface 3 (IIC3)	5
	Serial communication interface with FIFO (SCIF)	16

Each interrupt source is allocated a different vector number and vector table offset. See ta
in section 6, Interrupt Controller (INTC), for more information on vector numbers and ve
address offsets.

in table 5.8. The priority levels that can be set are 0 to 15. Level 16 cannot be set. See se
6.3.1, Interrupt Priority Registers 01, 02, 05 to 15 (IPR01, IPR02, IPR05 to IPR15), for
IPR01, IPR02, and IPR05 to IPR15.

Table 5.8 Interrupt Priority Order

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level.
H-UDI	15	Fixed priority level.
IRQ	0 to 15	Set with interrupt priority registers 01, 0
On-chip peripheral module		to 15 (IPR01, IPR02, and IPR05 to IPR

MACH, MACL, and PR, and the vector number of the interrupt exception handling to be
are saved in the register banks. In the case of exception handling due to an address error,
interrupt, UBC interrupt, or instruction, saving is not performed to the register banks. If s
been performed to all register banks (0 to 14), automatic saving to the stack is performed
of register bank saving. In this case, an interrupt controller setting must have been made s
register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the INTC i
setting to accept register bank overflow exceptions has been made (the BOVE bit in IBN
INTC is 1), register bank overflow exception occurs. Next, the priority level value of the
interrupt is written to the I3 to I0 bits in SR. For NMI, however, the priority level is 16, b
value set in the I3 to I0 bits is H'F (level 15). Then, after jumping to the start address fetc
the exception handling vector table, program execution starts. The jump that occurs is no
delayed branch. See section 6.6, Operation, for further details of interrupt exception hand

Trap instruction	TRAPA	
Slot illegal instructions	Undefined code placed immediately after a delayed branch instruction (delay slot), instructions that rewrite the PC, 32-bit instructions, RESBANK instruction, DIVS instruction, and DIVU instruction	Delayed branch instructions: JMP BRA, BSR, RTS, RTE, BF/S, BT/S BRAF Instructions that rewrite the PC: JI BRA, BSR, RTS, RTE, BT, BF, TF BF/S, BT/S, BSRF, BRAF, JSR/N 32-bit instructions: BAND.B, BANI BCLR.B, BLD.B, BLDNOT.B, BOR BORNOT.B, BSET.B, BST.B, BXO MOV.B@disp12, MOV.W@disp12 MOV.L@disp12, MOVI20, MOVI2 MOVU.B, MOVU.W.
General illegal instructions	Undefined code anywhere besides in a delay slot	
Integer division exceptions	Division by zero	DIVU, DIVS
	Negative maximum value ÷ (–1)	DIVS

4. After jumping to the address fetched from the exception handling vector table, progra
execution starts. The jump that occurs is not a delayed branch.

5.6.3 Slot Illegal Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed
slot. When the instruction placed in the delay slot is undefined code, an instruction that re
the PC, a 32-bit instruction, an RESBANK instruction, a DIVS instruction, or a DIVU ins
slot illegal exception handling starts when such kind of instruction is decoded. The CPU
as follows:

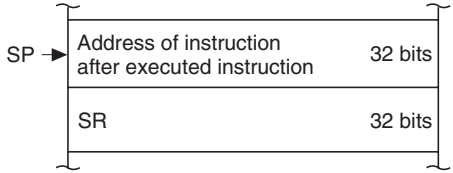
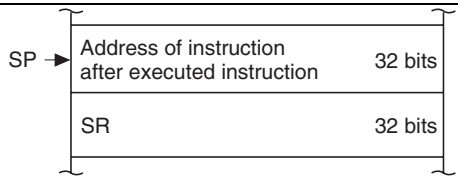
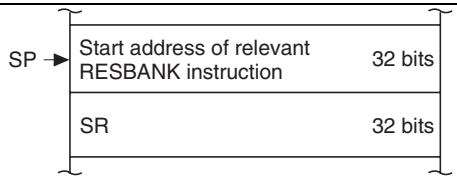
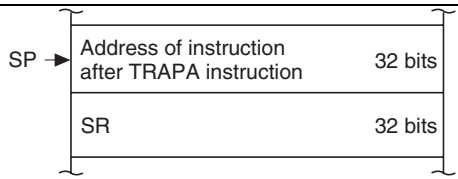
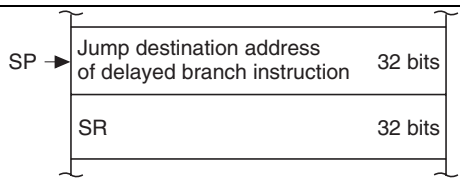
1. The exception service routine start address is fetched from the exception handling vec
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the jump addre
delayed branch instruction immediately before the undefined code, the instruction tha
the PC, the 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the
instruction.
4. After jumping to the address fetched from the exception handling vector table, progra
execution starts. The jump that occurs is not a delayed branch.

When an integer division instruction performs division by zero or the result of integer di
overflows, integer division instruction exception handling starts. The instructions that m
the source of division-by-zero exception are DIVU and DIVS. The only source instructi
overflow exception is DIVS, and overflow exception occurs only when the negative ma
value is divided by −1. The CPU operates as follows:

1. The exception service routine start address which corresponds to the integer division instruction exception that occurred is fetched from the exception handling vector tab
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start addr integer division instruction at which the exception occurred.
4. After jumping to the address fetched from the exception handling vector table, progr execution starts. The jump that occurs is not a delayed branch.

Point of Occurrence	Address Error	(Overflow)	Interrupt
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accep

Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, B BRAF

Interrupt	
Register bank error (overflow)	
Register bank error (underflow)	
Trap instruction	
Slot illegal instruction	

will occur when the stack is accessed during exception handling.

5.9.3 Address Errors Caused by Stacking of Address Error Exception Handlin

When the stack pointer is not a multiple of four, an address error will occur during stack exception handling (interrupts, etc.) and address error exception handling will start up as the first exception handling is ended. Address errors will then also occur in the stacking address error exception handling. To ensure that address error exception handling does n an endless loop, no address errors are accepted at that point. This allows program contro shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle executed. During stacking of the status register (SR) and program counter (PC), the SP i decremented by 4 for both, so the value of SP will not be a multiple of four after the stac either. The address value output during stacking is the SP value, so the address where th occurred is itself output. This means the write data stacked will be undefined.

peripheral module interrupts can be selected from 16 levels for request sources.

- NMI noise canceller function

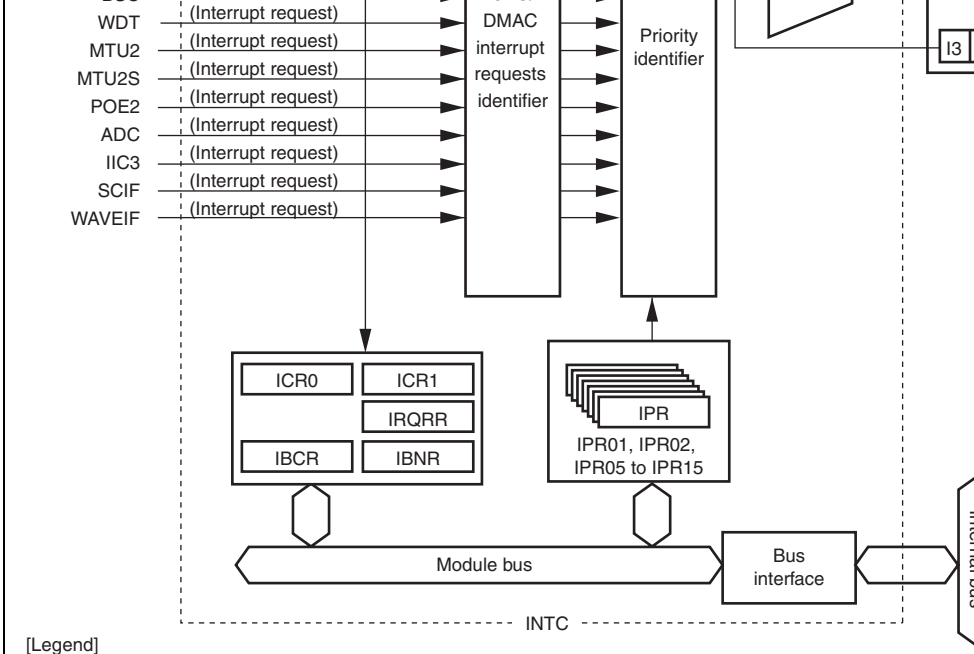
  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interru exception service routine, the pin state can be checked, enabling it to be used as the n canceller function.

- Occurrence of interrupt can be reported externally ($\overline{\text{IRQOUT}}$ pin)

  For example, when this LSI has released the bus mastership, this LSI can inform the bus master of occurrence of an on-chip peripheral module interrupt and request for th mastership.

- Register banks

  This LSI has register banks that enable register saving and restoration required in the processing to be performed at high speed.



[Legend]

UBC:	User break controller
H-UDI:	User debugging interface
DMAC:	Direct memory access controller
CMT:	Compare match timer
BSC:	Bus state controller
WDT:	Watchdog timer
MTU2:	Multi-function timer pulse unit 2
MTU2S:	Multi-function timer pulse unit 2S
POE2:	Port output enable 2
ADC:	A/D converter
IIC3:	I^2C bus interface 3
SCIF:	Serial communication interface with FIFO
WAVEIF:	WAVE interface

ICR0:	Interrupt control register 0
ICR1:	Interrupt control register 1
IRQRR:	IRQ interrupt request registe
IBCR:	Bank control register
IBNR:	Bank number register
IPR01, 02, 05 to 15:	Interrupt priority registers 01, 05 to 15

Figure 6.1 Block Diagram of INTC

Interrupt request output pin	IRQOUT	Output	Output of signal to report occ of interrupt source

IRQ interrupt request register	IRQRR	R/(W)*[2]	H'0000	H'FFFE0806	1
Bank control register	IBCR	R/W	H'0000	H'FFFE080C	1
Bank number register	IBNR	R/W	H'0000	H'FFFE080E	1
Interrupt priority register 01	IPR01	R/W	H'0000	H'FFFE0818	1
Interrupt priority register 02	IPR02	R/W	H'0000	H'FFFE081A	1
Interrupt priority register 05	IPR05	R/W	H'0000	H'FFFE0820	1
Interrupt priority register 06	IPR06	R/W	H'0000	H'FFFE0C00	1
Interrupt priority register 07	IPR07	R/W	H'0000	H'FFFE0C02	1
Interrupt priority register 08	IPR08	R/W	H'0000	H'FFFE0C04	1
Interrupt priority register 09	IPR09	R/W	H'0000	H'FFFE0C06	1
Interrupt priority register 10	IPR10	R/W	H'0000	H'FFFE0C08	1
Interrupt priority register 11	IPR11	R/W	H'0000	H'FFFE0C0A	1
Interrupt priority register 12	IPR12	R/W	H'0000	H'FFFE0C0C	1
Interrupt priority register 13	IPR13	R/W	H'0000	H'FFFE0C0E	1
Interrupt priority register 14	IPR14	R/W	H'0000	H'FFFE0C10	1
Interrupt priority register 15	IPR15	R/W	H'0000	H'FFFE0C12	1

Notes: Two access cycles are needed for word access, and four access cycles for longwo access.

1. When the NMI pin is high, becomes H'8000; when low, becomes H'0000.
2. Only 0 can be written after reading 1, to clear the flag.

Table 6.3 Interrupt Request Sources and IPR01, IPR02, and IPR05 to IPR15

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 t
Interrupt priority register 01	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register 02	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register 05	Reserved	Reserved	ADI	Reserv
Interrupt priority register 06	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register 07	DMAC4	DMAC5	DMAC6	DMAC7
Interrupt priority register 08	CMT0	CMT1	BSC	WDT
Interrupt priority register 09	MTU0 (TGI0A to TGI0D)	MTU0 (TCI0V, TGI0E, TGI0F)	MTU1 (TGI1A, TGI1B)	MTU1 (TCI1V
Interrupt priority register 10	MTU2 (TGI2A, TGI2B)	MTU2 (TCI2V, TCI2U)	MTU3 (TGI3A to TGI3D)	MTU3 (TCI3V
Interrupt priority register 11	MTU4 (TGI4A to TGI4D)	MTU4 (TCI4V)	MTU5 (TGI5U, TGI5V, TGI5W)	POE2 (OEI1,
Interrupt priority register 12	MTU3S (TGI3A to TGI3D)	MTU3S (TCI3V)	MTU4S (TGI4A to TGI4D)	MTU4S (TCI4V

to 0) with values from H'0 (0000) to H'F (1111), the priority of each corresponding interr
Setting of H'0 means priority level 0 (the lowest level) and H'F means priority level 15 (tl
highest level).

IPR01, IPR02, and IPR05 to IPR15 are initialized to H'0000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	*	R	NMI Input Level Sets the level of the signal input at the NMI pi NMI pin level can be obtained by reading this bit cannot be modified. 0: Low level is input to NMI pin 1: High level is input to NMI pin
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edge of th interrupt request signal on the NMI pin is dete 0: Interrupt request is detected on falling edge input 1: Interrupt request is detected on rising edge input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ Sense Select
14	IRQ70S	0	R/W	These bits select whether interrupt signals corresponding to pins IRQ7 to IRQ0 are detect low level, falling edge, rising edge, or both edg
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	00: Interrupt request is detected on low level o input
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	01: Interrupt request is detected on falling edge input
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	10: Interrupt request is detected on rising edge input
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	11: Interrupt request is detected on both edges input
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

[Legend]

n = 7 to 0

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

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

0	IRQ0F	0	R/(W)*	[Setting condition] • IRQn input is low Edge detection: 0: IRQn interrupt request is not detected [Clearing conditions] • Cleared by reading IRQnF while IRQnF = writing 0 to IRQnF • Cleared by executing IRQn interrupt exce handling 1: IRQn interrupt request is detected [Setting condition] • Edge corresponding to IRQn1S or IRQn0 ICR1 has occurred at IRQn pin

[Legend]
n = 7 to 0
Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Value	R/W	Description
15	E15	0	R/W	Enable
14	E14	0	R/W	These bits enable or disable use of register b interrupt priority levels 15 to 1. However, use banks is always disabled for the user break in
13	E13	0	R/W	
12	E12	0	R/W	0: Use of register banks is disabled
11	E11	0	R/W	1: Use of register banks is enabled
10	E10	0	R/W	
9	E9	0	R/W	
8	E8	0	R/W	
7	E7	0	R/W	
6	E6	0	R/W	
5	E5	0	R/W	
4	E4	0	R/W	
3	E3	0	R/W	
2	E2	0	R/W	
1	E1	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

R/W:	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BE[1:0]	00	R/W	Register Bank Enable These bits enable or disable use of register ba 00: Use of register banks is disabled for all inte The setting of IBCR is ignored. 01: Use of register banks is enabled for all inte except NMI and user break. The setting of ignored. 10: Reserved (setting prohibited) 11: Use of register banks is controlled by the s IBCR.
13	BOVE	0	R/W	Register Bank Overflow Enable Enables of disables register bank overflow exc 0: Generation of register bank overflow excepti disabled 1: Generation of register bank overflow except enabled
12 to 4	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

selects whether the rising edge or falling edge is detected.

Though the priority level of the NMI interrupt is 16, the NMI interrupt exception handlin interrupt mask level bits (I3 to I0) in the status register (SR) to level 15.

6.4.2 User Break Interrupt

A user break interrupt which occurs when a break condition set in the user break controll matches has a priority level of 15. The user break interrupt exception handling sets the I3 in SR to level 15. For user break interrupts, see section 7, User Break Controller (UBC).

6.4.3 H-UDI Interrupt

The user debugging interface (H-UDI) interrupt has a priority level of 15, and occurs at s input of an H-UDI interrupt instruction. H-UDI interrupt requests are edge-detected and r until they are accepted. The H-UDI interrupt exception handling sets the I3 to I0 bits in S 15. For H-UDI interrupts, see section 24, User Debugging Interface (H-UDI).

checked by reading the IRQ interrupt request bits (IRQ7F to IRQ0F) in the IRQ interrup
register (IRQRR).

When using edge-sensing for IRQ interrupts, an interrupt request is detected due to chan
IRQ7 to IRQ0 pin states, and an interrupt request signal is sent to the INTC. The result o
interrupt request detection is retained until that interrupt request is accepted. Whether IR
interrupt requests have been detected or not can be checked by reading the IRQ7F to IR
IRQRR. Writing 0 to these bits after reading them as 1 clears the result of IRQ interrupt
detection.

The IRQ interrupt exception handling sets the I3 to I0 bits in SR to the priority level of t
accepted IRQ interrupt.

- Multi-function timer pulse unit 2S (MTU2S)
- Port output enable 2 (POE2)
- I^2C bus interface 3 (IIC3)
- Serial communication interface with FIFO (SCIF)
- WAVE interface (WAVEIF)

As every source is assigned a different interrupt vector, the source does not need to be ide
the exception service routine. A priority level in a range from 0 to 15 can be set for each
by interrupt priority registers 05 to 15 (IPR05 to IPR15). The on-chip peripheral module i
exception handling sets the I3 to I0 bits in SR to the priority level of the accepted on-chip
peripheral module interrupt.

The priorities of IRQ interrupts and on-chip peripheral module interrupts can be set free
0 and 15 for each pin or module by setting interrupt priority registers 01, 02, and 05 to 1
IPR02, and IPR05 to IPR15). However, if two or more interrupts specified by the same
among IPR05 to IPR15 occur, the priorities are defined as shown in the IPR setting unit
priority of table 6.4, and the priorities cannot be changed. A power-on reset assigns prio
to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is a
two or more interrupt sources and interrupts from those sources occur simultaneously, th
processed by the default priorities indicated in table 6.4.

			H'0000003B			
IRQ	IRQ0	64	H'00000100 to H'00000103	0 to 15 (0)	IPR01 (15 to 12)	—
	IRQ1	65	H'00000104 to H'00000107	0 to 15 (0)	IPR01 (11 to 8)	—
	IRQ2	66	H'00000108 to H'0000010B	0 to 15 (0)	IPR01 (7 to 4)	—
	IRQ3	67	H'0000010C to H'0000010F	0 to 15 (0)	IPR01 (3 to 0)	—
	IRQ4	68	H'00000110 to H'00000113	0 to 15 (0)	IPR02 (15 to 12)	—
	IRQ5	69	H'00000114 to H'00000117	0 to 15 (0)	IPR02 (11 to 8)	—
	IRQ6	70	H'00000118 to H'0000011B	0 to 15 (0)	IPR02 (7 to 4)	—
	IRQ7	71	H'0000011C to H'0000011F	0 to 15 (0)	IPR02 (3 to 0)	—
ADC	ADI	92	H'00000170 to H'00000173	0 to 15 (0)	IPR05 (7 to 4)	—

	HEI1	113	H'000001C4 to H'000001C7			2
DMAC2	DEI2	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPR06 (7 to 4)	1
	HEI2	117	H'000001D4 to H'000001D7			2
DMAC3	DEI3	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPR06 (3 to 0)	1
	HEI3	121	H'000001E4 to H'000001E7			2
DMAC4	DEI4	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPR07 (15 to 12)	1
	HEI4	125	H'000001F4 to H'000001F7			2
DMAC5	DEI5	128	H'00000200 to H'00000203	0 to 15 (0)	IPR07 (11 to 8)	1
	HEI5	129	H'00000204 to H'00000207			2
DMAC6	DEI6	132	H'00000210 to H'00000213	0 to 15 (0)	IPR07 (7 to 4)	1
	HEI6	133	H'00000214 to H'00000217			2
DMAC7	DEI7	136	H'00000220 to H'00000223	0 to 15 (0)	IPR07 (3 to 0)	1
	HEI7	137	H'00000224 to H'00000227			2

WDT	ITI		152	H'00000260 to H'00000263	0 to 15 (0)	IPR08 (3 to 0)	—
MTU2	MTU0	TGI0A	156	H'00000270 to H'00000273	0 to 15 (0)	IPR09 (15 to 12)	1
		TGI0B	157	H'00000274 to H'00000277			2
		TGI0C	158	H'00000278 to H'0000027B			3
		TGI0D	159	H'0000027C to H'0000027F			4
		TCI0V	160	H'00000280 to H'00000283	0 to 15 (0)	IPR09 (11 to 8)	1
		TGI0E	161	H'00000284 to H'00000287			2
		TGI0F	162	H'00000288 to H'0000028B			3
	MTU1	TGI1A	164	H'00000290 to H'00000293	0 to 15 (0)	IPR09 (7 to 4)	1
		TGI1B	165	H'00000294 to H'00000297			2
		TCI1V	168	H'000002A0 to H'000002A3	0 to 15 (0)	IPR09 (3 to 0)	1
		TCI1U	169	H'000002A4 to H'000002A7			2

	TCI2U	177	H'000002C4 to H'000002C7			2
MTU3	TGI3A	180	H'000002D0 to H'000002D3	0 to 15 (0)	IPR10 (7 to 4)	1
	TGI3B	181	H'000002D4 to H'000002D7			2
	TGI3C	182	H'000002D8 to H'000002DB			3
	TGI3D	183	H'000002DC to H'000002DF			4
	TCI3V	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPR10 (3 to 0)	—
MTU4	TGI4A	188	H'000002F0 to H'000002F3	0 to 15 (0)	IPR11 (15 to 12)	1
	TGI4B	189	H'000002F4 to H'000002F7			2
	TGI4C	190	H'000002F8 to H'000002FB			3
	TGI4D	191	H'000002FC to H'000002FF			4
	TCI4V	192	H'00000300 to H'00000303	0 to 15 (0)	IPR11 (11 to 8)	—
MTU5	TGI5U	196	H'00000310 to H'00000313	0 to 15 (0)	IPR11 (7 to 4)	1
	TGI5V	197	H'00000314 to H'00000317			2
	TGI5W	198	H'00000318 to H'0000031B			3

		TGI3B	205	H'00000334 to H'00000337			2
		TGI3C	206	H'00000338 to H'0000033B			3
		TGI3D	207	H'0000033C to H'0000033F			4
		TCI3V	208	H'00000340 to H'00000343	0 to 15 (0)	IPR12 (11 to 8)	—
	MTU4S	TGI4A	212	H'00000350 to H'00000353	0 to 15 (0)	IPR12 (7 to 4)	1
		TGI4B	213	H'00000354 to H'00000357			2
		TGI4C	214	H'00000358 to H'0000035B			3
		TGI4D	215	H'0000035C to H'0000035F			4
		TCI4V	216	H'00000360 to H'00000363	0 to 15 (0)	IPR12 (3 to 0)	—
	MTU5S	TGI5U	220	H'00000370 to H'00000373	0 to 15 (0)	IPR13 (15 to 12)	1
		TGI5V	221	H'00000374 to H'00000377			2
		TGI5W	222	H'00000378 to H'0000037B			3
POE2	OEI3		224	H'00000380 to H'00000383	0 to 15 (0)	IPR13 (11 to 8)	—

		TXI	231	H'0000039C to H'0000039F			4
		TEI	232	H'000003A0 to H'000003A3			5
SCIF	SCIF0	BRI0	240	H'000003C0 to H'000003C3	0 to 15 (0)	IPR14 (15 to 12)	1
		ERI0	241	H'000003C4 to H'000003C7			2
		RXI0	242	H'000003C8 to H'000003CB			3
		TXI0	243	H'000003CC to H'000003CF			4
	SCIF1	BRI1	244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR14 (11 to 8)	1
		ERI1	245	H'000003D4 to H'000003D7			2
		RXI1	246	H'000003D8 to H'000003DB			3
		TXI1	247	H'000003DC to H'000003DF			4
	SCIF2	BRI2	248	H'000003E0 to H'000003E3	0 to 15 (0)	IPR14 (7 to 4)	1
		ERI2	249	H'000003E4 to H'000003E7			2
		RXI2	250	H'000003E8 to H'000003EB			3
		TXI2	251	H'000003EC to H'000003EF			4

		TXI3	255	H'000003FC to H'000003FF			4
WAVEIF	ERR		256	H'00000400 to H'00000403	0 to 15 (0)	IPR15 (15 to 12)	1
	WRXI		257	H'00000404 to H'00000407			2
	WTXI		258	H'00000408 to H'0000040B			3

have the same priority level or if multiple interrupts occur within a single IPR, the in
with the highest priority is selected, according to the default priority and IPR setting
internal priority shown in table 6.4.

3. The priority level of the interrupt selected by the interrupt controller is compared wit
interrupt level mask bits (I3 to I0) in the status register (SR) of the CPU. If the interr
request priority level is equal to or less than the level set in bits I3 to I0, the interrupt
ignored. If the interrupt request priority level is higher than the level in bits I3 to I0,
interrupt controller accepts the interrupt and sends an interrupt request signal to the C
4. When the interrupt controller accepts an interrupt, a low level is output from the $\overline{\text{IRQ}}$
5. The CPU detects the interrupt request sent from the interrupt controller when the CP
the instruction to be executed. Instead of executing the decoded instruction, the CPU
interrupt exception handling (figure 6.4).
6. The interrupt exception service routine start address is fetched from the exception ha
vector table corresponding to the accepted interrupt.
7. The status register (SR) is saved onto the stack, and the priority level of the accepted
is copied to bits I3 to I0 in SR.
8. The program counter (PC) is saved onto the stack.
9. The CPU jumps to the fetched interrupt exception service routine start address and s
executing the program. The jump that occurs is not a delayed branch.
10. A high level is output from the $\overline{\text{IRQOUT}}$ pin. However, if the interrupt controller acc
interrupt with a higher priority than the interrupt just being accepted, the $\overline{\text{IRQOUT}}$ p
low level.

Interrupts held pending due to edge-sensing are cleared by a power-on reset.

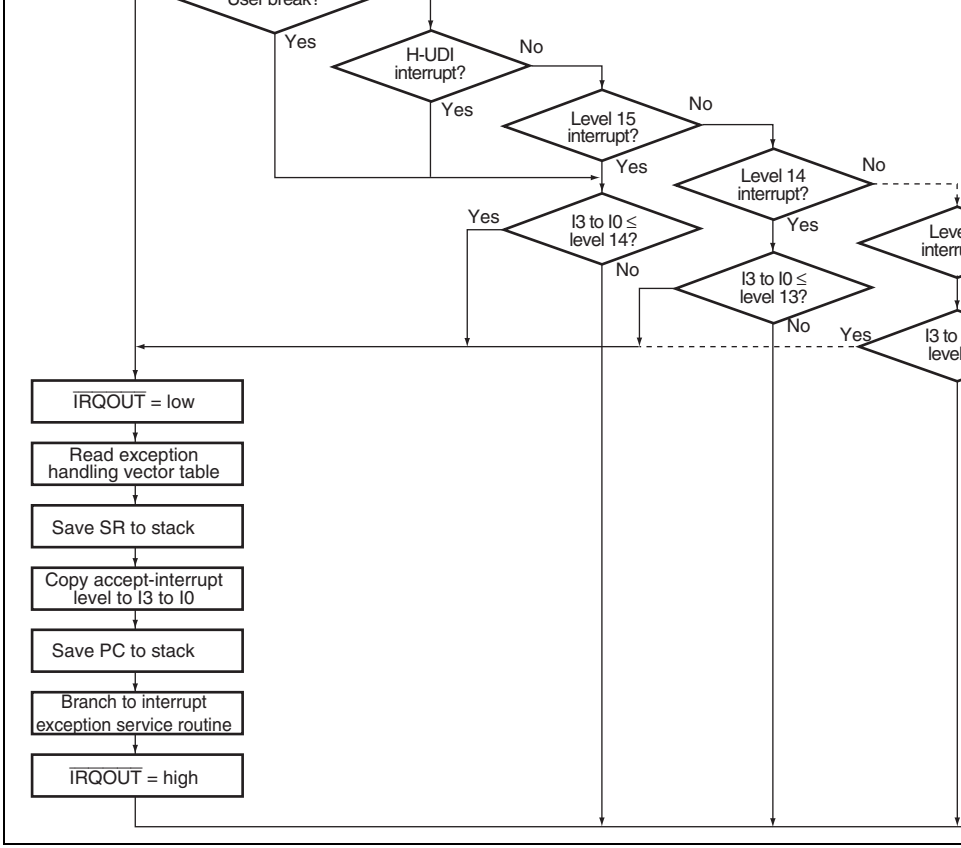


Figure 6.2 Interrupt Operation Flow

Notes: 1. PC: Start address of the next instruction (return destination instruction) after the executed instruction
2. Always make sure that SP is a multiple of 4.

Figure 6.3 Stack after Interrupt Exception Handling

Table 6.5 Interrupt Response Time

Item			Number of States					Remarks
			NMI	User Break	H-UDI	IRQ	Peripheral Module	
Time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU			2 Icyc + 2 Bcyc + 1 Pcyc	3 Icyc	2 Icyc + 1 Pcyc	2 Icyc + 3 Bcyc + 1 Pcyc	2 Icyc + 1 Bcyc + 1 Pcyc	
Time from input of interrupt request signal to CPU until sequence currently being executed is completed, interrupt exception handling starts, and first instruction in interrupt exception service routine is fetched	No register banking	Min.	3 Icyc + m1 + m2					Min. is when wait time is z
		Max.	4 Icyc + 2(m1 + m2) + m3					Max. is when priority interru occurred duri exception ha
	Register banking without register bank overflow	Min.	—		3 Icyc + m1 + m2			Min. is when wait time is z
		Max.	—		12 Icyc + m1 + m2			Max. is when request has o execution of t instruction.
	Register banking with register bank overflow	Min.	—		3 Icyc + m1 + m2			Min. is when wait time is z
		Max.	—		3 Icyc + m1 + m2 + 19(m4)			Max. is when request has o execution of t instruction.

banking without register bank overflow				m1 + m2	1 Pcyc + m1 + m2	1 Pcyc + m1 + m2	
	Max.	—	—	14 Icyc + 1 Pcyc + m1 + m2	14 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	14 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	160-MHz opera 0.106 to 0.163
Register banking with register bank overflow	Min.	—	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	160-MHz opera 0.050 to 0.106
	Max.	—	—	5 Icyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	160-MHz opera 0.169 to 0.225

Notes: m1 to m4 are the number of states needed for the following memory accesses.

m1: Vector address read (longword read)

m2: SR save (longword write)

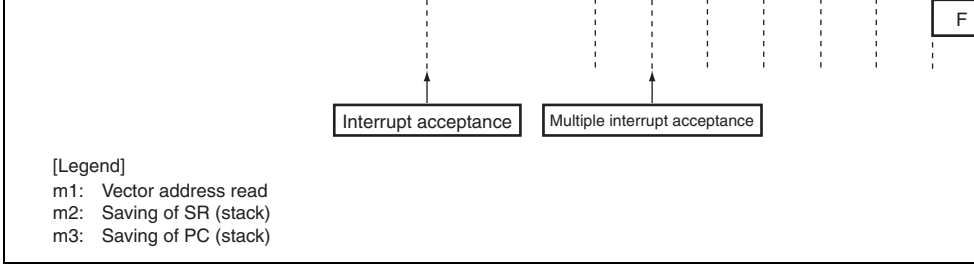
m3: PC save (longword write)

m4: Banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from stack.

1. In the case that m1 = m2 = m3 = m4 = 1 Icyc.
2. In the case that (Iϕ, Bϕ, Pϕ) = (160 MHz, 40 MHz, 40 MHz).

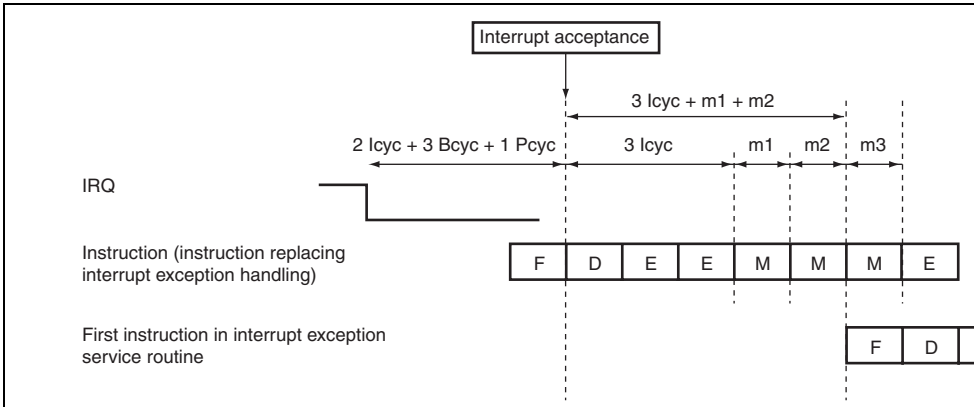
[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)
F: Instruction fetch. Instruction is fetched from memory in which program is stored.
D: Instruction decoding. Fetched instruction is decoded.
E: Instruction execution. Data operation or address calculation is performed in accordance with the result of
M: Memory access. Memory data access is performed.

Figure 6.4 Example of Pipeline Operation when IRQ Interrupt is Accepte (No Register Banking)



[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)

Figure 6.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

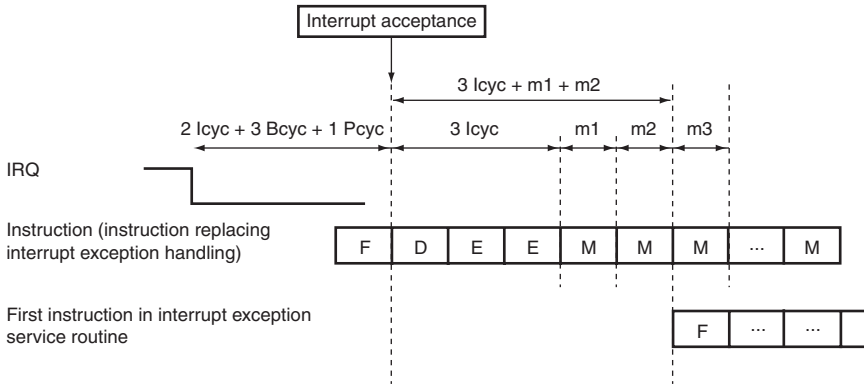


[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)

Figure 6.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)

Figure 6.7 Example of Pipeline Operation when Interrupt is Accepted during RI Instruction Execution (Register Banking without Register Bank Overflow



[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)

Figure 6.8 Example of Pipeline Operation when IRQ Interrupt is Accepte (Register Banking with Register Bank Overflow)

[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)
m4: Restoration of banked registers

Figure 6.9 Example of Pipeline Operation when Interrupt is Accepted during RE Instruction Execution (Register Banking with Register Bank Overflow)

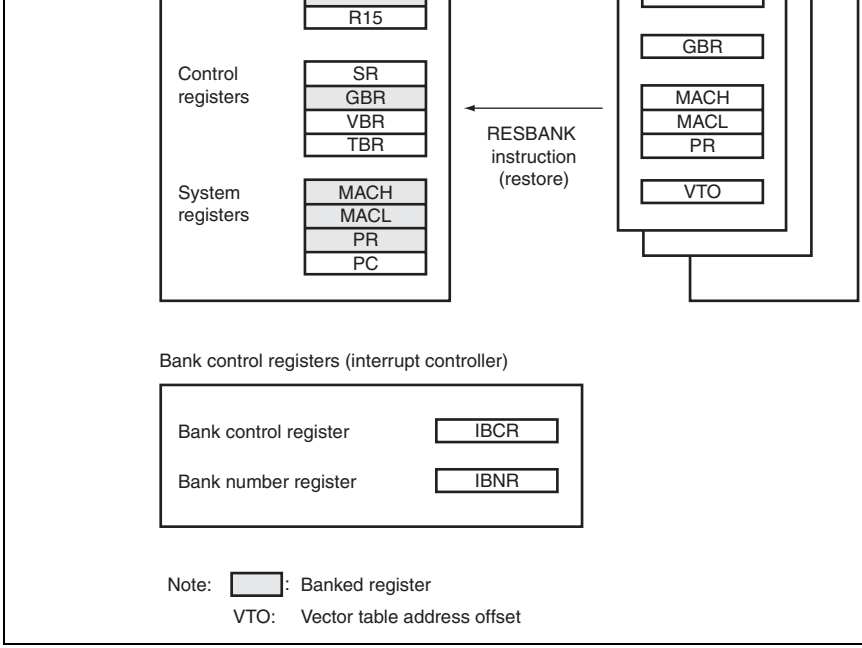


Figure 6.10 Overview of Register Bank Configuration

out (FILO) sequence. Saving takes place in order, beginning from bank 0, and restoration place in the reverse order, beginning from the last bank saved to.

6.8.2 Bank Save and Restore Operations

(1) Saving to Bank

Figure 6.11 shows register bank save operations. The following operations are performed interrupt for which usage of register banks is allowed is accepted by the CPU:

a. Assume that the bank number bit value in the bank number register (IBNR), BN, is i b interrupt is generated.
b. The contents of registers R0 to R14, GBR, MACH, MACL, and PR, and the interrupt table address offset (VTO) of the accepted interrupt are saved in the bank indicated by bank i.
c. The BN value is incremented by 1.

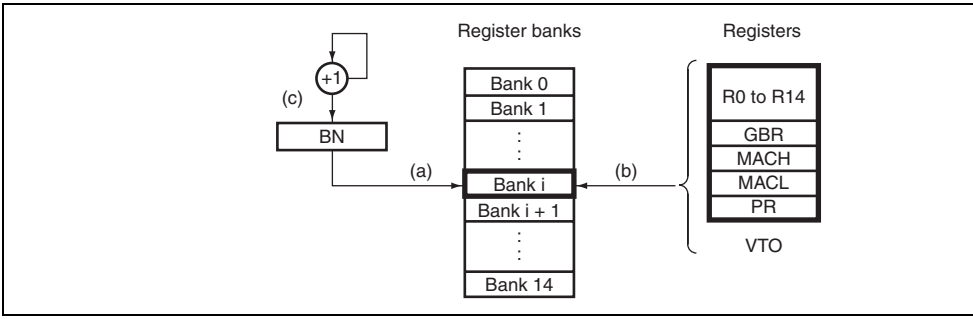


Figure 6.11 Bank Save Operations

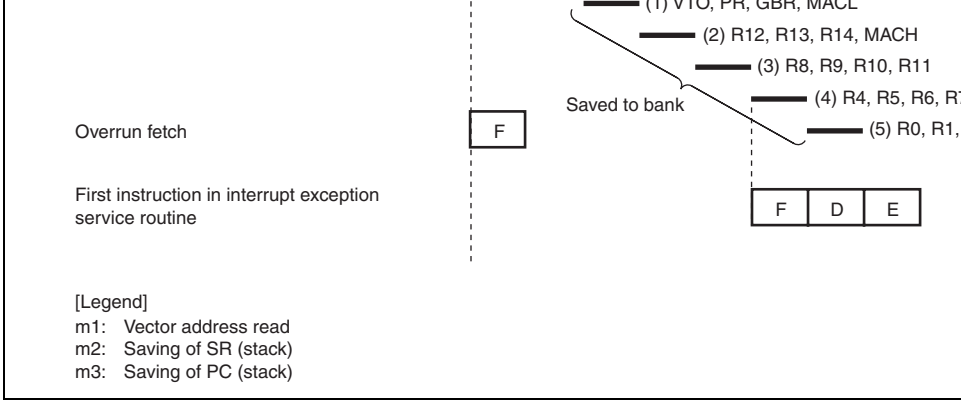


Figure 6.12 Bank Save Timing

(2) Restoration from Bank

The RESBANK (restore from register bank) instruction is used to restore data saved in a bank. After restoring data from the register banks with the RESBANK instruction at the interrupt service routine, execute the RTE instruction to return from the exception handl

1. The status register (SR) and program counter (PC) are saved to the stack during interr
   exception handling.
2. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are s
   the stack. The registers are saved to the stack in the order of MACL, MACH, GBR, P
   R13, …, R1, and R0.
3. The register bank overflow bit (BO) in SR is set to 1.
4. The bank number bit (BN) value in the bank number register (IBNR) remains set to th
   maximum value of 15.

(2) Restoration from Stack

When the RESBANK (restore from register bank) instruction is executed with the registe
overflow bit (BO) in SR set to 1, the CPU operates as follows:

1. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are r
   from the stack. The registers are restored from the stack in the order of R0, R1, …, R1
   PR, GBR, MACH, and MACL.
2. The bank number bit (BN) value in the bank number register (IBNR) remains set to th
   maximum value of 15.

bank.

(2) Register Bank Underflow

This exception occurs if the RESBANK (restore from register bank) instruction is execu
no data has been saved to the register banks. In this case, the values of R0 to R14, GBR,
MACL, and PR do not change. In addition, the bank number bit (BN) value in the bank
register (IBNR) remains set to 0.

6.8.5 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. When t
happens, the CPU operates as follows:

1. The exception service routine start address which corresponds to the register bank er
occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start addr
instruction to be executed after the last executed instruction for a register bank overf
the start address of the executed RESBANK instruction for a register bank underflo
prevent multiple interrupts from occurring at a register bank overflow, the interrupt
level that caused the register bank overflow is written to the interrupt mask level bits
of the status register (SR).
4. Program execution starts from the exception service routine start address.

Figure 6.13 shows a block diagram of interrupt control.

Here, DME is bit 0 in DMAOR of the DMAC, and DEn (n = 0 to 7) is bit 0 in CHCR0 to
of the DMAC. For details, see section 9, Direct Memory Access Controller (DMAC).

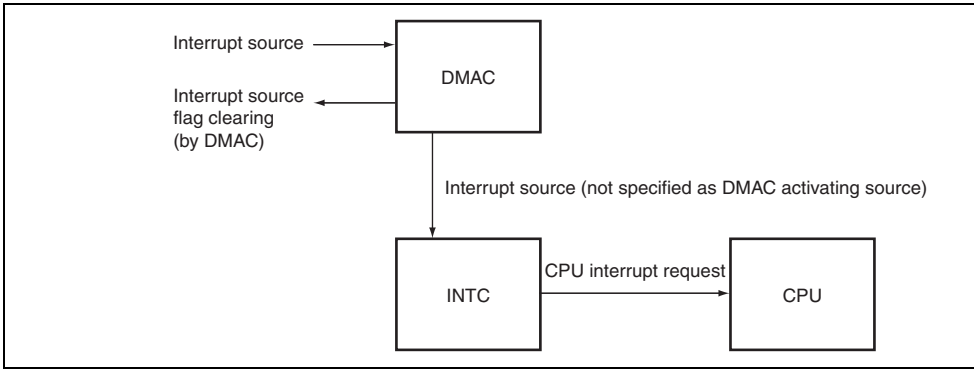


Figure 6.13 Interrupt Control Block Diagram

Interrupt

1. Select DMAC activating sources and set both the DE and DME bits to 1. This masks interrupt sources regardless of the interrupt priority register settings.
2. Activating sources are applied to the DMAC when interrupts occur.
3. The DMAC clears the interrupt sources when starting transfer.

6.10 Usage Note

6.10.1 Timing to Clear an Interrupt Source

The interrupt source flags should be cleared in the interrupt exception service routine. A clearing the interrupt source flag, "time from occurrence of interrupt request until interru controller identifies priority, compares it with mask bits in SR, and sends interrupt reque to CPU" shown in table 6.5 is required before the interrupt source sent to the CPU is act cancelled. To ensure that an interrupt request that should have been cleared is not inadve accepted again, read the interrupt source flag after it has been cleared, and then execute instruction.

and internal bus (I bus).

7.1 Features

1. The following break comparison conditions can be set.
   Number of break channels: four channels (channels 0 to 3)
   User break can be requested as the independent condition on channels 0, 1, 2, and 3.

- Address
  Comparison of the 32-bit address is maskable in 1-bit units.
  One of the three address buses (F address bus (FAB), M address bus (MAB), and I a
  (IAB)) can be selected.
- Bus master when I bus is selected
  Selection of CPU cycles or DMAC cycles
- Bus cycle
  Instruction fetch (only when C bus is selected) or data access
- Read/write
- Operand size
  Byte, word, and longword

2. Exception handling routine for user-specified break conditions can be executed.
3. In an instruction fetch cycle, it can be selected whether PC breaks are set before or a
   instruction is executed.
4. When a break condition is satisfied, a trigger signal is output from the $\overline{\text{UBCTRG}}$ pin

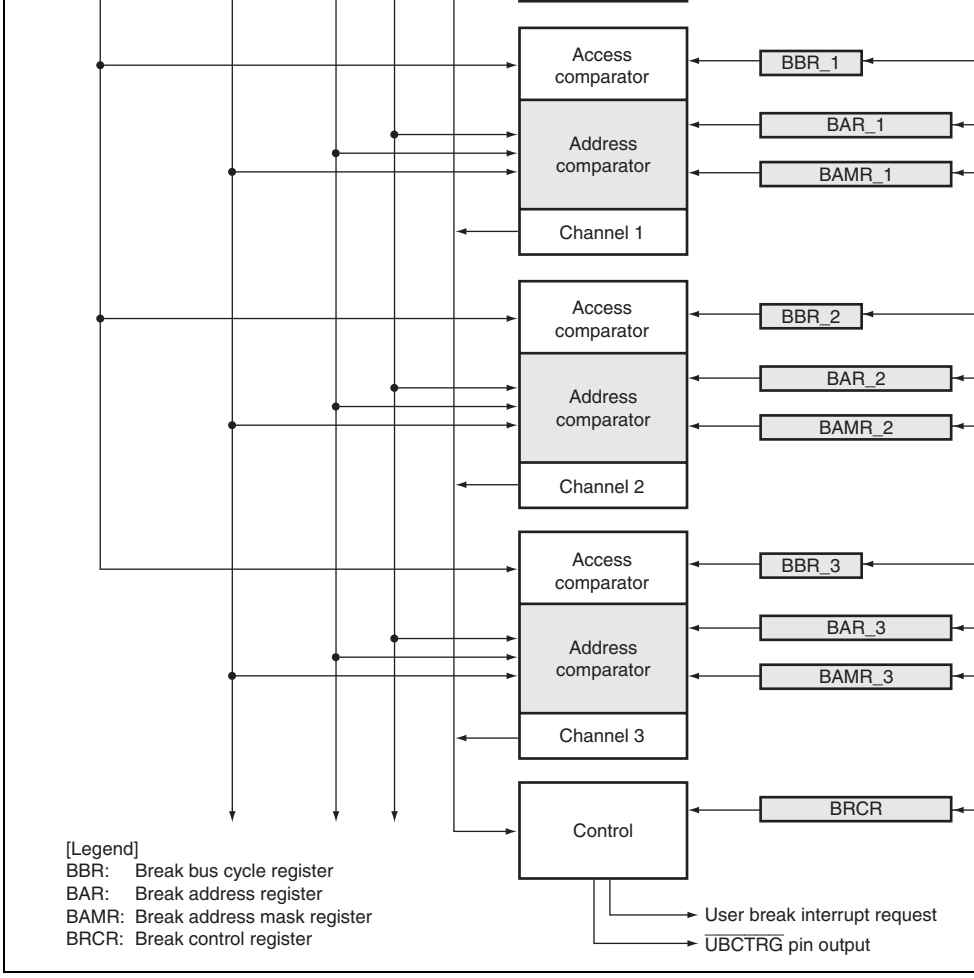


Figure 7.1 Block Diagram of UBC

7.3 Register Descriptions

The UBC has the following registers.

Table 7.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address
0	Break address register_0	BAR_0	R/W	H'00000000	H'FFFC0400
	Break address mask register_0	BAMR_0	R/W	H'00000000	H'FFFC0404
	Break bus cycle register_0	BBR_0	R/W	H'0000	H'FFFC04A0
1	Break address register_1	BAR_1	R/W	H'00000000	H'FFFC0410
	Break address mask register_1	BAMR_1	R/W	H'00000000	H'FFFC0414
	Break bus cycle register_1	BBR_1	R/W	H'0000	H'FFFC04B0
2	Break address register_2	BAR_2	R/W	H'00000000	H'FFFC0420
	Break address mask register_2	BAMR_2	R/W	H'00000000	H'FFFC0424
	Break bus cycle register_2	BBR_2	R/W	H'0000	H'FFFC04A4
3	Break address register_3	BAR_3	R/W	H'00000000	H'FFFC0430
	Break address mask register_3	BAMR_3	R/W	H'00000000	H'FFFC0434
	Break bus cycle register_3	BBR_3	R/W	H'0000	H'FFFC04B4
Common	Break control register	BRCR	R/W	H'00000000	H'FFFC04C0

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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	BA0_15	BA0_14	BA0_13	BA0_12	BA0_11	BA0_10	BA0_9	BA0_8	BA0_7	BA0_6	BA0_5	BA0_4	BA0_3	BA0_2	BA
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA0_31 to BA0_0	All 0	R/W	Break Address 0 Store an address on the CPU address bus (FA MAB) or IAB specifying break conditions of ch When the C bus and instruction fetch cycle ar selected by BBR_0, specify an FAB address i BA0_31 to BA0_0. When the C bus and data access cycle are se BBR_0, specify an MAB address in bits BA0_ BA0_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR

	BAM0_15	BAM0_14	BAM0_13	BAM0_12	BAM0_11	BAM0_10	BAM0_9	BAM0_8	BAM0_7	BAM0_6	BAM0_5	BAM0_4	BAM0_3	BAM0_2
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM0_31 to BAM0_0	All 0	R/W	Break Address Mask 0 Specify bits masked in the channel-0 break a bits specified by BAR_0 (BA0_31 to BA0_0). 0: Break address bit BA0_n is included in the condition 1: Break address bit BA0_n is masked and n included in the break condition

Note: n = 31 to 0

R/W:	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
13	UBID0	0	R/W	User Break Interrupt Disable 0 Disables or enables user break interrupt requ when a channel-0 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
9, 8	CP0[1:0]	00	R/W	I-Bus Bus Master Select 0 Select the bus master when the bus cycle of t channel-0 break condition is the I bus cycle. H when the C bus cycle is selected, this bit is inv (only the CPU cycle). x1: CPU cycle is included in break conditions 1x: DMAC cycle is included in break condition

				Select the instruction fetch cycle or data acc as the bus cycle of the channel-0 break cond the instruction fetch cycle is selected, select cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cy 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cy data access cycle
3, 2	RW0[1:0]	00	R/W	Read/Write Select 0 Select the read cycle or write cycle as the bu the channel-0 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write
1, 0	SZ0[1:0]	00	R/W	Operand Size Select 0 Select the operand size of the bus cycle for t channel-0 break condition. 00: Break condition does not include operan 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	BA1_15	BA1_14	BA1_13	BA1_12	BA1_11	BA1_10	BA1_9	BA1_8	BA1_7	BA1_6	BA1_5	BA1_4	BA1_3	BA1_2	BA
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA1_31 to BA1_0	All 0	R/W	Break Address 1 Store an address on the CPU address bus (FA MAB) or IAB specifying break conditions of ch When the C bus and instruction fetch cycle ar selected by BBR_1, specify an FAB address i BA1_31 to BA1_0. When the C bus and data access cycle are se BBR_1, specify an MAB address in bits BA1_ BA1_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR

	BAM1_15	BAM1_14	BAM1_13	BAM1_12	BAM1_11	BAM1_10	BAM1_9	BAM1_8	BAM1_7	BAM1_6	BAM1_5	BAM1_4	BAM1_3	BAM1_2
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM1_31 to BAM1_0	All 0	R/W	Break Address Mask 1 Specify bits masked in the channel-1 break a bits specified by BAR_1 (BA1_31 to BA1_0). 0: Break address bit BA1_n is included in the condition 1: Break address bit BA1_n is masked and n included in the break condition

Note: n = 31 to 0

R/W:	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
13	UBID1	0	R/W	User Break Interrupt Disable 1 Disables or enables user break interrupt requ when a channel-1 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
9, 8	CP1[1:0]	00	R/W	I-Bus Bus Master Select 1 Select the bus master when the bus cycle of t channel-1 break condition is the I bus cycle. H when the C bus cycle is selected, this bit is inv (only the CPU cycle). x1: CPU cycle is included in break conditions 1x: DMAC cycle is included in break condition

				Select the instruction fetch cycle or data acc as the bus cycle of the channel-1 break cond the instruction fetch cycle is selected, select cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cy 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cy data access cycle
3, 2	RW1[1:0]	00	R/W	Read/Write Select 1 Select the read cycle or write cycle as the bu the channel-1 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write
1, 0	SZ1[1:0]	00	R/W	Operand Size Select 1 Select the operand size of the bus cycle for t channel-1 break condition. 00: Break condition does not include operan 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	BA2_15	BA2_14	BA2_13	BA2_12	BA2_11	BA2_10	BA2_9	BA2_8	BA2_7	BA2_6	BA2_5	BA2_4	BA2_3	BA2_2	BA
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA2_31 to BA2_0	All 0	R/W	Break Address 2 Store an address on the CPU address bus (FA MAB) or IAB specifying break conditions of ch When the C bus and instruction fetch cycle ar selected by BBR_2, specify an FAB address i BA2_31 to BA2_0. When the C bus and data access cycle are se BBR_2, specify an MAB address in bits BA2_ BA0_2.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR

	BAM2_15	BAM2_14	BAM2_13	BAM2_12	BAM2_11	BAM2_10	BAM2_9	BAM2_8	BAM2_7	BAM2_6	BAM2_5	BAM2_4	BAM2_3	BAM2_2
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM2_31 to BAM2_0	All 0	R/W	Break Address Mask 2 Specify bits masked in the channel-2 break a bits specified by BAR_2 (BA2_31 to BA2_0). 0: Break address bit BA2_n is included in the condition 1: Break address bit BA2_n is masked and n included in the break condition

Note: n = 31 to 0

R/W:	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
13	UBID2	0	R/W	User Break Interrupt Disable 2 Disables or enables user break interrupt requ when a channel-2 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
9, 8	CP2[1:0]	00	R/W	I-Bus Bus Master Select 2 Select the bus master when the bus cycle of t channel-2 break condition is the I bus cycle. H when the C bus cycle is selected, this bit is inv (only the CPU cycle). x1: CPU cycle is included in break conditions 1x: DMAC cycle is included in break condition

				Select the instruction fetch cycle or data acc as the bus cycle of the channel-2 break cond the instruction fetch cycle is selected, select cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cy 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cy data access cycle
3, 2	RW2[1:0]	00	R/W	Read/Write Select 2 Select the read cycle or write cycle as the bu the channel-2 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write
1, 0	SZ2[1:0]	00	R/W	Operand Size Select 2 Select the operand size of the bus cycle for t channel-2 break condition. 00: Break condition does not include operan 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	BA3_15	BA3_14	BA3_13	BA3_12	BA3_11	BA3_10	BA3_9	BA3_8	BA3_7	BA3_6	BA3_5	BA3_4	BA3_3	BA3_2	BA
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA3_31 to BA3_0	All 0	R/W	Break Address 3 Store an address on the CPU address bus (FA MAB) or IAB specifying break conditions of ch When the C bus and instruction fetch cycle ar selected by BBR_3, specify an FAB address i BA3_31 to BA3_0. When the C bus and data access cycle are se BBR_3, specify an MAB address in bits BA3_ BA3_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR

	BAM3_15	BAM3_14	BAM3_13	BAM3_12	BAM3_11	BAM3_10	BAM3_9	BAM3_8	BAM3_7	BAM3_6	BAM3_5	BAM3_4	BAM3_3	BAM3_2
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM3_31 to BAM3_0	All 0	R/W	Break Address Mask 3 Specify bits masked in the channel-3 break a bits specified by BAR_3 (BA3_31 to BA3_0). 0: Break address bit BA3_n is included in the condition 1: Break address bit BA3_n is masked and n included in the break condition

Note: n = 31 to 0

R/W:	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
13	UBID3	0	R/W	User Break Interrupt Disable 3 Disables or enables user break interrupt requ when a channel-3 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
9, 8	CP3[1:0]	00	R/W	I-Bus Bus Master Select 3 Select the bus master when the bus cycle of t channel-3 break condition is the I bus cycle. H when the C bus cycle is selected, this bit is inv (only the CPU cycle). x1: CPU cycle is included in break conditions 1x: DMAC cycle is included in break condition

				Select the instruction fetch cycle or data acc as the bus cycle of the channel-3 break cond the instruction fetch cycle is selected, select cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cy 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cy data access cycle
3, 2	RW3[1:0]	00	R/W	Read/Write Select 3 Select the read cycle or write cycle as the bu the channel-3 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write
1, 0	SZ3[1:0]	00	R/W	Operand Size Select 3 Select the operand size of the bus cycle for t channel-3 break condition. 00: Break condition does not include operan 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

reset, but retains its previous value by a manual reset or in software standby mode or slee

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
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	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	SCMFC 0	SCMFC 1	SCMFC 2	SCMFC 3	SCMFD 0	SCMFD 1	SCMFD 2	SCMFD 3	PCB3	PCB2	PCB1	PCB0	-	-	
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	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
17, 16	CKS[1:0]	00	R/W	Clock Select These bits specify the pulse width output to th $\overline{\text{UBCTRG}}$ pin when a break condition is satisfi 00: Pulse width of $\overline{\text{UBCTRG}}$ is one bus clock 01: Pulse width of $\overline{\text{UBCTRG}}$ is two bus clock 10: Pulse width of $\overline{\text{UBCTRG}}$ is four bus clock 11: Pulse width of $\overline{\text{UBCTRG}}$ is eight bus clock

				When the C bus cycle condition in the break set for channel 1 is satisfied, this flag is set t order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 1 do match 1: The C bus cycle condition for channel 1 m
13	SCMFC2	0	R/W	C Bus Cycle Condition Match Flag 2 When the C bus cycle condition in the break set for channel 2 is satisfied, this flag is set t order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 2 do match 1: The C bus cycle condition for channel 2 m
12	SCMFC3	0	R/W	C Bus Cycle Condition Match Flag 3 When the C bus cycle condition in the break set for channel 3 is satisfied, this flag is set t order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 3 do match 1: The C bus cycle condition for channel 3 m
11	SCMFD0	0	R/W	I Bus Cycle Condition Match Flag 0 When the I bus cycle condition in the break c set for channel 0 is satisfied, this flag is set t order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 0 do match 1: The I bus cycle condition for channel 0 ma

				When the I bus cycle condition in the break co set for channel 2 is satisfied, this flag is set to order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 2 does match 1: The I bus cycle condition for channel 2 mat
8	SCMFD3	0	R/W	I Bus Cycle Condition Match Flag 3 When the I bus cycle condition in the break co set for channel 3 is satisfied, this flag is set to order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 3 does match 1: The I bus cycle condition for channel 3 mat
7	PCB3	0	R/W	PC Break Select 3 Selects the break timing of the instruction fetc for channel 3 as before or after instruction exe 0: PC break of channel 3 is generated before instruction execution 1: PC break of channel 3 is generated after in execution
6	PCB2	0	R/W	PC Break Select 2 Selects the break timing of the instruction fetc for channel 2 as before or after instruction exe 0: PC break of channel 2 is generated before instruction execution 1: PC break of channel 2 is generated after in execution

				Selects the break timing of the instruction fet for channel 0 as before or after instruction ex 0: PC break of channel 0 is generated before instruction execution 1: PC break of channel 0 is generated after i execution
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

fetch/data access select, and read/write select) are each set. No user break will be gen
even one of these groups is set to 00. The relevant break control conditions are set in t
the break control register (BRCR). Make sure to set all registers related to breaks befo
BBR, and branch after reading from the last written register. The newly written regist
become valid from the instruction at the branch destination.

2. In the case where the break conditions are satisfied, the UBC sends a user break inter
request to the CPU, sets the C bus condition match flag (SCMFC) or I bus condition m
flag (SCMFD) for the appropriate channel, and outputs a pulse to the $\overline{\text{UBCTRG}}$ pin w
width set by the CKS1 and CKS0 bits. Setting the UBID bit in BBR to 1 enables exte
monitoring of the trigger output without requesting user break interrupts.
3. On receiving a user break interrupt request signal, the INTC determines its priority. S
user break interrupt has a priority level of 15, it is accepted when the priority level set
interrupt mask level bits (I3 to I0) of the status register (SR) is 14 or lower. If the I3 to
are set to a priority level of 15, the user break interrupt is not accepted, but the conditi
checked, and condition match flags are set if the conditions match. For details on asce
the priority, see section 6, Interrupt Controller (INTC).
4. Condition match flags (SCMFC and SCMFD) can be used to check which condition h
satisfied. They are set when the conditions match, but are not reset. To use these flags
write 0 to the corresponding bit of the flags.
5. It is possible that the breaks set in channels 0 to 3 occur around the same time. In this
there will be only one user break request to the CPU, but these four break channel ma
may be set at the same time.
6. When selecting the I bus as the break condition, note as follows:
   — Several bus masters, including the CPU and DMAC, are connected to the I bus. Th
   monitors bus cycles generated by the bus master specified by BBR, and determine
   condition match.

1. When C bus/instruction fetch/read/word or longword is set in the break bus cycle reg
(BBR), the break condition is the FAB bus instruction fetch cycle. Whether PC breal
before or after the execution of the instruction can then be selected with the PCB0 or
of the break control register (BRCR) for the appropriate channel. If an instruction fet
set as a break condition, clear LSB in the break address register (BAR) to 0. A break
generated as long as this bit is set to 1.
2. A break for instruction fetch which is set as a break before instruction execution occ
is confirmed that the instruction has been fetched and will be executed. This means a
does not occur for instructions fetched by overrun (instructions fetched at a branch o
an interrupt transition, but not to be executed). When this kind of break is set for the
of a delayed branch instruction, the break is not generated until the execution of the
instruction at the branch destination.

   Note: If a branch does not occur at a delayed branch instruction, the subsequent ins
   not recognized as a delay slot.

3. When setting a break condition for break after instruction execution, the instruction s
the break condition is executed and then the break is generated prior to execution of
instruction. As with pre-execution breaks, a break does not occur with overrun fetch
instructions. When this kind of break is set for a delayed branch instruction and its d
the break is not generated until the first instruction at the branch destination.
4. When an instruction fetch cycle is set, the break data register (BDR) is ignored. Ther
break data cannot be set for the break of the instruction fetch cycle.
5. If the I bus is set for a break of an instruction fetch cycle, the setting is invalidated.

operand size is listed in table 7.3.

Table 7.3 Data Access Cycle Addresses and Operand Size Comparison Condition

Access Size	Address Compared
Longword	Compares break address register bits 31 to 2 to address bus bits 3
Word	Compares break address register bits 31 to 1 to address bus bits 3
Byte	Compares break address register bits 31 to 0 to address bus bits 3

This means that when address H'00001003 is set in the break address register (BAR), example, the bus cycle in which the break condition is satisfied is as follows (where c conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. If the data access cycle is selected, the instruction at which the break will occur canno determined.

The address of the instruction that matched the break condition is saved to the stack.
instruction that matched the condition is not executed, and the break occurs before it
when a delay slot instruction matches the condition, the instruction is executed, and
destination address is saved to the stack.

2. When C bus (FAB)/instruction fetch (after instruction execution) is specified as a br
condition:

   The address of the instruction following the instruction that matched the break condi
saved to the stack. The instruction that matches the condition is executed, and the br
before the next instruction is executed. However when a delayed branch instruction
slot matches the condition, the instruction is executed, and the branch destination ad
saved to the stack.

3. When C bus/data access cycle or I bus/data access cycle is specified as a break cond

   The address after executing several instructions of the instruction that matched the b
condition is saved to the stack.

Address: H'00000404, Address mask: H'00000000

Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is
included in the condition)

<Channel 1>

Address: H'00008010, Address mask: H'00000006

Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size i
included in the condition)

A user break occurs after an instruction of address H'00000404 is executed or before
instructions of addresses H'00008010 to H'00008016 are executed.

(Example 1-2)

- Register specifications

  BAR_0 = H'00027128, BAMR_0 = H'00000000, BBR_0 = H'005A, BAR_1= H'0003
  BAMR_1 = H'00000000, BBR_1 = H'0054, BRCR = H'00000000

  <Channel 0>

  Address: H'00027128, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/write/word

  <Channel 1>

  Address: H'00031415, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size i
  included in the condition)

  On channel 0, a user break does not occur since instruction fetch is not a write cycle.
  channel 1, a user break does not occur since instruction fetch is performed for an even

Address: H'00008010, Address mask: H'00000006

Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size included in the condition)

A user break occurs after an instruction with addresses H'00008000 to H'00008FFE or before an instruction with addresses H'00008010 to H'00008016 are executed.

(2) Break Condition Specified for C Bus Data Access Cycle

(Example 2-1)

- Register specifications

  BBR_0 = H'0064, BAR_0 = H'00123456, BAMR_0 = H'00000000,
  BBR_1 = H'006A, BAR_1 = H'000ABCDE, BAMR_1 = H'000000FF, BRCR = H'0

  <Channel 0>

  Address: H'00123456, Address mask: H'00000000

  Bus cycle: C bus/data access/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'000ABCDE, Address mask: H'000000FF

  Bus cycle: C bus/data access/write/word

  On channel 0, a user break occurs with longword read from address H'00123456, wo from address H'00123456, or byte read from address H'00123456. On channel 1, a u occurs when word is written in addresses H'000ABC00 to H'000ABCFE.

<Channel 1>

Address: H'00055555, Address mask: H'00000000

Bus cycle: I bus/data access/write/byte

On channel 0, the setting of I bus/instruction fetch is ignored.

On channel 1, a user break occurs when the DMAC writes byte data in address H'000 the I bus (write by the CPU does not generate a user break).

table 5.1 in section 5, Exception Handling. If an exception source with higher priorit
the user break interrupt request is not received.

4. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at a delay slot instruction, the user break interrupt requ
   received immediately before execution of the branch destination.

5. User breaks are disabled during UBC module standby mode. Do not read from or wr
   UBC registers during UBC module standby mode; the values are not guaranteed.

6. Do not set an address within an interrupt exception handling routine whose interrupt
   level is at least 15 (including user break interrupts) as a break address.

7. Do not set break after instruction execution for the SLEEP instruction or for the dela
   branch instruction where the SLEEP instruction is placed at its delay slot.

8. When setting a break for a 32-bit instruction, set the address where the upper 16 bits
   placed. If the address of the lower 16 bits is set and a break before instruction execut
   as a break condition, the break is handled as a break after instruction execution.

9. Do not set a user break before instruction execution for the instruction following the
   DIVS instruction. If a user break before instruction execution is set for the instructio
   following the DIVU or DIVS instruction and an exception or interrupt occurs during
   of the DIVU or DIVS instruction, a user break occurs before instruction execution ev
   execution of the DIVU or DIVS instruction is halted.

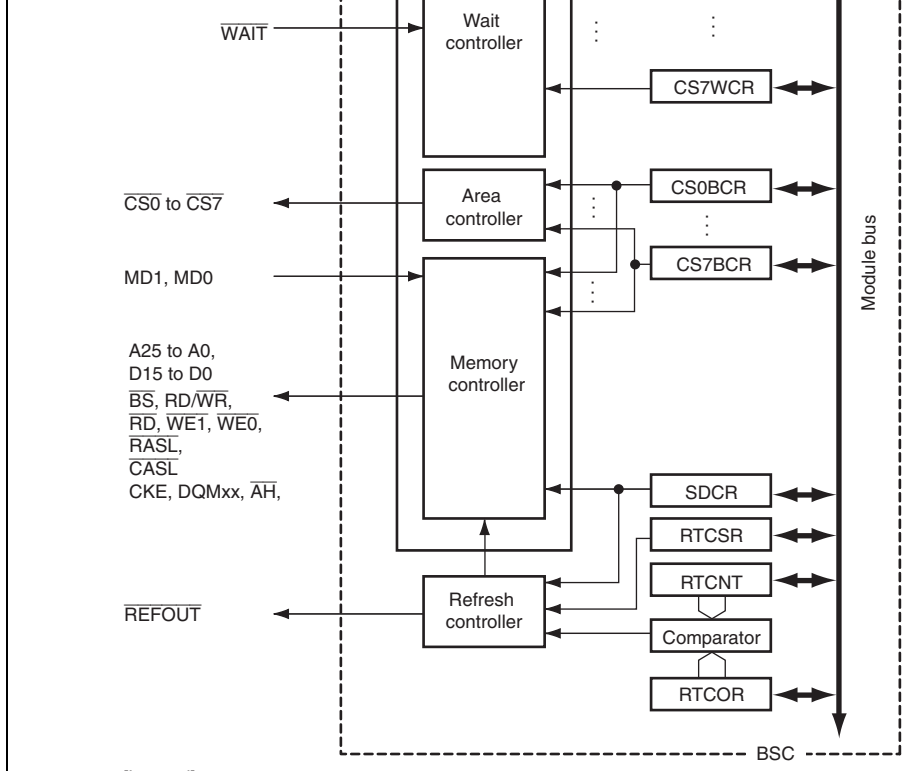
10. Do not set a user break both before instruction execution and after instruction execut
    instruction of the same address. If, for example, a user break before instruction execu
    channel 0 and a user break after instruction on channel 1 are set at the instruction of
    address, the condition match flag for the channel 1 is set even though a user break or
    occurs before instruction execution.

1. External address space
   - A maximum of 64 Mbytes for each of areas CS0 to CS7.
   - Can specify the normal space interface, SRAM interface with byte selection, burs (clock synchronous or asynchronous), MPX-I/O, and SDRAM for each address s
   - Can select the data bus width (8 or 16 bits) for each address space.
   - Controls insertion of wait cycles for each address space.
   - Controls insertion of wait cycles for each read access and write access.
   - Can set independent idle cycles during the continuous access for five cases: read- same space/different spaces), read-read (in same space/different spaces), the first write access.
2. Normal space interface
   - Supports the interface that can directly connect to the SRAM.
3. Burst ROM interface (clock asynchronous)
   - High-speed access to the ROM that has the page mode function.
4. MPX-I/O interface
   - Can directly connect to a peripheral LSI that needs an address/data multiplexing.
5. SDRAM interface
   - Can set the SDRAM in up to two areas.
   - Multiplex output for row address/column address.
   - Efficient access by single read/single write.
   - High-speed access in bank-active mode.
   - Supports an auto-refresh and self-refresh.
   - Supports low-frequency and power-down modes.
   - Issues MRS and EMRS commands.

— Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8)

10. Usage as interval timer for refresh counter

— Generates an interrupt request at compare match.



[Legend]
CMNCR: Common control register
CSnWCR: CSn space wait control register (n = 0 to 7)
CSnBCR: CSn space bus control register (n = 0 to 7)
SDCR: SDRAM control register
RTCSR: Refresh timer control/status register
RTCNT: Refresh timer counter
RTCOR: Refresh time constant register

Figure 8.1 Block Diagram of BSC

CS0 to CS7	Output	Chip select
RD/$\overline{WR}$	Output	Read/write Connects to $\overline{WE}$ pins when SDRAM or SRAM with byte selecti connected.
$\overline{RD}$	Output	Read pulse signal (read data output enable signal) Functions as a strobe signal for indicating memory read cycles PCMCIA is used.
$\overline{AH}$	Output	A signal used to hold an address when MPX-I/O is in use
$\overline{WE1}$/$\overline{DQMLU}$	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a SRAM with byte se connected. Functions as the select signals for D15 to D8 when SDRAM is connected.
$\overline{WE0}$/$\overline{DQMLL}$	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a SRAM with byte se connected. Functions as the select signals for D7 to D0 when SDRAM is c
$\overline{RASL}$	Output	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{CASL}$	Output	Connects to $\overline{CAS}$ pin when SDRAM is connected.
CKE	Output	Connects to CKE pin when SDRAM is connected.
$\overline{WAIT}$	Input	External wait input
$\overline{BREQ}$	Input	Bus request input
$\overline{BACK}$	Output	Bus enable output
$\overline{REFOUT}$	Output	Refresh request output in bus-released state
MD1, MD0	Input	Select bus width (8 or 16 bits) of area 0 and modes including enabling/disabling of the on-chip ROM.

Table 8.2 Address Map in On-Chip ROM-Enabled Mode

Address	Space	Memory to be Connected
H'0000 0000 to H'0007 FFFF	On-chip ROM	On-chip ROM
H'0008 0000 to H'01FF FFFF	Other	Reserved area
H'0200 0000 to H'03FF FFFF	CS0	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)
H'0400 0000 to H'07FF FFFF	CS1	Normal space, SRAM with byte selection
H'0800 0000 to H'0BFF FFFF	CS2	Normal space, SRAM with byte selection, SDRAM
H'0C00 0000 to H'0FFF FFFF	CS3	Normal space, SRAM with byte selection, SDRAM
H'1000 0000 to H'13FF FFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM, reserved area*
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip peripheral modules, reserved area*

Note: * For the on-chip RAM space, access the addresses shown in section 22, On-C
For the on-chip peripheral module space, access the addresses shown in sec
List of Registers. Do not access addresses which are not described in these s
Otherwise, the correct operation cannot be guaranteed.

		burst ROM (asynchronous)	
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O	6
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection	6
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection	6
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area	
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM, reserved area*	
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip peripheral modules, reserved area*	

Note: * For the on-chip RAM space, access the addresses shown in section 22, On-Cl
For the on-chip I/O register space, access the addresses shown in section 26,
Registers. Do not access addresses which are not described in these sections
Otherwise, the correct operation cannot be guaranteed.

8.3.2 Setting Operating Modes

This LSI can set the following modes of operation at the time of power-on reset using the
pins.

- Single-Chip Mode

  In single-chip mode, no access is made to the external bus, and the LSI is activated by
  chip ROM program upon a power-on reset. The BSC module enters the module stand
  to reduce power consumption.

- On-Chip ROM-Enabled Mode/On-Chip ROM-Disabled Mode

  In on-chip ROM-enabled mode, since the first half of area 0 is allocated to the on-chip
  the LSI can be activated by the on-chip ROM program upon a power-on reset. The sec
  of area 0 is the external memory space.

In on-chip ROM-disabled mode, the data bus width of area 0 cannot be changed fron setting after a power-on reset, but the data bus widths of areas 1 to 7 can be changed register settings in the program. In on-chip ROM-enabled mode, all the data bus wid areas 0 to 7 can be changed by register settings in the program. Note that data bus wi be restricted depending on memory types.

- Initial Settings of Endianness

  The initial settings of byte-data alignment of areas 0 to 7 can be selected as big endia endian. In on-chip ROM-disabled mode, the endianness of area 0 cannot be changed initial setting after a power-on reset, but the endianness of areas 1 to 7 can be change register settings in the program. In on-chip ROM-enabled mode, all the endianness o to 7 can be changed by register settings in the program. Little endian cannot be selec 0. Since both 32-bit and 16-bit accesses are included in instruction fetches, no instru be assigned in little endian area. Accordingly, instructions should be executed in big area.

For details of mode settings, see section 3, MCU Operating Modes.

CSn space bus control register	CSnBCR	R/W	H'36DB0400 (in activation with 16-bit bus width)	H'FFFC 0004 to H'FFFC 0020	32
CSn space wait control register	CSnWCR	R/W	H'00000500	H'FFFC0028 to H'FFFC 0044	32
SDRAM control register	SDCR	R/W	H'00000000	H'FFFC004C	32
Refresh timer control/status register	RTCSR	R/W	H'00000000	H'FFFC0050	32
Refresh timer counter	RTCNT	R/W	H'00000000	H'FFFC0054	32
Refresh time constant register	RTCOR	R/W	H'00000000	H'FFFC0058	32

	-	-	-	-	BLOCK	DPRTY[1:0]		DMAIW[2:0]			DMA IWA	-	-	HIZ CKIO
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
12	—	1	R	Reserved This bit is always read as 1. The write value always be 1.
11	BLOCK	0	R/W	Bus Lock Specifies whether or not the $\overline{\text{BREQ}}$ signal is 0: Receives $\overline{\text{BREQ}}$. 1: Does not receive $\overline{\text{BREQ}}$.
10, 9	DPRTY[1:0]	00	R/W	DMA Burst Transfer Priority Specify the priority for a refresh request/bus mastership request during DMA burst transfe 00: Accepts a refresh request and bus maste request during DMA burst transfer. 01: Accepts a refresh request but does not a bus mastership request during DMA burs 10: Accepts neither a refresh request nor a b mastership request during DMA burst tra 11: Reserved (setting prohibited)

				010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
5	DMAIWA	0	R/W	Method of inserting wait states between acces when DMA single address transfer is performe Specifies the method of inserting the idle cycl specified by the DMAIW[2:0] bit. Clearing this make this LSI insert the idle cycles when anot device, which includes this LSI, drives the dat after an external device with DACK drove it. H when the external device with DACK drives th bus continuously, idle cycles are not inserted. this bit will make this LSI insert the idle cycles access to an external device with DACK, ever the continuous access cycles to an external d with DACK are performed. 0: Idle cycles inserted when another device dr data bus after an external device with DAC it. 1: Idle cycles always inserted after an access external device with DACK
4	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.

				1: CK is driven in standby mode and bus rele state.
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in standby mode for A $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WEn}$/$\overline{DQMxx}$, $\overline{AH}$, and $\overline{RD}$ released state, these pins are in high-impeda regardless of the setting value of the HIZMEI 0: High impedance in standby mode. 1: Driven in standby mode
0	HIZCNT	0	R/W	High-Z Control Specifies the state in standby mode and bus state for CKE, $\overline{RASL}$, and $\overline{CASL}$. 0: CKE, $\overline{RASL}$, and $\overline{CASL}$ are in high-impeda in standby mode and bus-released state. 1: CKE, $\overline{RASL}$, and $\overline{CASL}$ are driven in stand and bus-released state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	-	IWW[2:0]			IWRWD[2:0]			IWRWS[2:0]			IWRRD[2:0]			IWR	
Initial value:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	
R/W:	R	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	-	TYPE[2:0]			ENDIAN	BSZ[1:0]		-	-	-	-	-	-	-	
Initial value:	0	0	0	0	0	1*	1*	0	0	0	0	0	0	0	
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	

Note: * CSnBCR samples the external pins (MD1 and MD0) that specify the bus width at power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

				011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
27 to 25	IWRWD[2:0]	011	R/W	Idle Cycles for Another Space Read-Write Specify the number of idle cycles to be inser the access to a memory that is connected to space. The target access cycle is a read-writ which continuous access cycles switch betw different spaces. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

				100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
21 to 19	IWRRD[2:0]	011	R/W	Idle Cycles for Read-Read in Another Space Specify the number of idle cycles to be inserte the access to a memory that is connected to t space. The target cycle is a read-read cycle o continuous access cycles switch between diffe space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

				100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
14 to 12	TYPE[2:0]	000	R/W	Specify the type of memory connected to a s 000: Normal space 001: Burst ROM (clock asynchronous) 010: MPX-I/O 011: SRAM with byte selection 100: SDRAM 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Burst ROM (clock synchronous) For details of memory type in each area, see and 8.3.
11	ENDIAN	0	R/W	Endian Select Specifies data alignment in a space. 0: Big endian 1: Little endian

width can be specified as 8 bits o
by the address according to the S
in CS5WCR by specifying the BS
bits to 11. The fixed bus width ca
specified as 8 bits or 16 bits.

2. The initial data bus width for area
is specified by external pins. In o
ROM-disabled mode, writing to th
and BSZ0 bits in CS0BCR is igno
the bus width settings in CS1BCR
CS7BCR can be modified. In on-
ROM-enabled mode, the bus wid
settings in CS0BCR to CS7BCR
modified.
3. If area 2 or area 3 is specified as
space, the bus width can be spec
16 bits only.
4. If area 0 or 4 is specified as clock
synchronous burst ROM space, t
width can be specified as 16 bits

8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

Note: * CSnBCR samples the external pins (MD1 and MD0) that specify the bus width
power-on reset.

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	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	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	— *	All 0	R/W	Reserved These bits are always read as 0. The write v should always be 0.
20	BAS*	0	R/W	Byte Access Selection when SRAM with Byt Selection is Used Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read/write ti asserts the RD/$\overline{WR}$ signal during the write cycle. 1: Asserts the $\overline{WEn}$ signal during the read/wr cycle and asserts the RD/$\overline{WR}$ signal at the timing.

				01: 1.5 cycles
				10: 2.5 cycles
				11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles
				Specify the number of cycles that are necessa read/write access.
				0000: No cycle
				0001: 1 cycle
				0010: 2 cycles
				0011: 3 cycles
				0100: 4 cycles
				0101: 5 cycles
				0110: 6 cycles
				0111: 8 cycles
				1000: 10 cycles
				1001: 12 cycles
				1010: 14 cycles
				1011: 18 cycles
				1100: 24 cycles
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)

				should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from RD, $\overline{WEn}$ Negation to Add Negation Specify the number of delay cycles from RD negation to address and $\overline{CS0}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Note * To connect the burst ROM to the CS0 space and switch to the burst ROM int after activation in ROM-disabled mode, set the TYPE[2:0] bits in CS0BCR aft the burst number by the bits 20 and 21 and the burst wait cycle number by th and 17. Do not write 1 to the reserved bits other than above bits.

Bit	Bit Name	Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Selec Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing v SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read/write tim asserts the RD/$\overline{WR}$ signal during the write a cycle. 1: Asserts the $\overline{WEn}$ signal during the read/writ cycle and asserts the RD/$\overline{WR}$ signal at the timing.
19	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessa write access. 000: The same cycles as WR[3:0] setting (nur read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles

				10: 2.5 cycles
				11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles
				Specify the number of cycles that are neces read access.
				0000: No cycle
				0001: 1 cycle
				0010: 2 cycles
				0011: 3 cycles
				0100: 4 cycles
				0101: 5 cycles
				0110: 6 cycles
				0111: 8 cycles
				1000: 10 cycles
				1001: 12 cycles
				1010: 14 cycles
				1011: 18 cycles
				1100: 24 cycles
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification
				Specifies whether or not the external wait inp The specification by this bit is valid even whe number of access wait cycle is 0.
				0: External wait input is valid
				1: External wait input is ignored

10: 2.5 cycles
11: 3.5 cycles

Bit	Bit Name	Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Sele Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing asserts the RD/$\overline{WR}$ signal during the write cycle. 1: Asserts the $\overline{WEn}$ signal during the read ac and asserts the RD/$\overline{WR}$ signal at the write
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

				0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait inpu The specification by this bit is valid even wher number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

Bit	Bit Name	Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Sele Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing asserts the RD/$\overline{WR}$ signal during the write cycle. 1: Asserts the $\overline{WEn}$ signal during the read ac and asserts the RD/$\overline{WR}$ signal at the write
19	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necess write access. 000: The same cycles as WR[3:0] setting (nu read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles

				01: 1.5 cycles
				10: 2.5 cycles
				11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles
				Specify the number of cycles that are necessa read access.
				0000: No cycle
				0001: 1 cycle
				0010: 2 cycles
				0011: 3 cycles
				0100: 4 cycles
				0101: 5 cycles
				0110: 6 cycles
				0111: 8 cycles
				1000: 10 cycles
				1001: 12 cycles
				1010: 14 cycles
				1011: 18 cycles
				1100: 24 cycles
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)

1, 0	HW[1:0]	00	R/W	Delay Cycles from RD, $\overline{WE}$n Negation to Add Negation Specify the number of delay cycles from RD negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write val should always be 0.
21	SZSEL	0	R/W	MPX-I/O Interface Bus Width Specification Specifies an address to select the bus width w BSZ[1:0] of CS5BCR are specified as 11. This valid only when area 5 is specified as MPX-I/O 0: Selects the bus width by address A14 1: Selects the bus width by address A21 The relationship between the SZSEL bit and b selected by A14 or A21 are summarized below

SZSEL	A14	A21	Bus
0	0	Not affected	8 bit
0	1	Not affected	16 b
1	Not affected	0	8 bit
1	Not affected	1	16 b

				Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing asserts the RD/$\overline{WR}$ signal during the write cycle. 1: Asserts the $\overline{WEn}$ signal during the read ac and asserts the RD/$\overline{WR}$ signal at the write
19	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necess write access. 000: The same cycles as WR[3:0] setting (nu read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				Specify the number of cycles that are necessa read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait inpu valid. The specification by this bit is valid ever the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

Bit	Bit Name	Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Selec Specifies the $\overline{\text{WEn}}$ and RD/$\overline{\text{WR}}$ signal timing v SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WEn}}$ signal at the read timing a asserts the RD/$\overline{\text{WR}}$ signal during the write a cycle. 1: Asserts the $\overline{\text{WEn}}$ signal during the read/writ cycle and asserts the RD/$\overline{\text{WR}}$ signal at the timing.
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{\text{CS6}}$ A to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from addre assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WN	0	R/W	External Wait Mask Specification Specifies whether or not the external wait inp The specification of this bit is valid even whe number of access wait cycles is 0. 0: The external wait input is valid 1: The external wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Neg Address, $\overline{CS6}$ Negation Specify the number of delay cycles from $\overline{RD}$, negation to address, and $\overline{CS6}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. Th must not be set to B'11.

Bus Width	BST[1:0]	Burst count
8 bits	00	16 burst × one t
	01	4 burst × four tir
16 bits	00	8 burst × one tir
	01	2 burst × four tir
	10	4-4 or 2-4-2 bur

19, 18	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				These bits are always read as 0. The write v should always be 0.
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inse first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

should always be 0.

Bit	Bit Name	Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. T must not be set to B'11.

Bus Width	BST[1:0]	Burst count
8 bits	00	16 burst × one
	01	4 burst × four t
16 bits	00	8 burst × one t
	01	2 burst × four t
	10	4-4 or 2-4-2 bu

19, 18	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inse between the second or subsequent access o burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

				01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be insert first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Ad $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
10	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.
9	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
8, 7	A2CL[1:0]	10	R/W	CAS Latency for Area 2 Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

Note: * If only one area is connected to the SDRAM, specify area 3. In this case, spec as normal space or SRAM with byte selection.

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
14, 13	WTRP[1:0]*	00	R/W	Number of Auto-Precharge Completion Wait Specify the number of minimum precharge c wait cycles as shown below. • From the start of auto-precharge and issu ACTV command for the same bank • From issuing of the PRE/PALL command of the ACTV command for the same banl • Till entering power-down mode or deep p down mode • From the issuing of PALL command to is command in auto-refresh mode • From the issuing of PALL command to is SELF command in self-refresh mode The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

				00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
9	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
8, 7	A3CL[1:0]	10	R/W	CAS Latency for Area 3 Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				are required between the WRITE comma in the SDRAM and the auto-precharge ac referring to each SDRAM data sheet. An cycle number so as not to exceed the cyc specified by this bit. • Cycle number from the issuance of the W command until the issuance of the PRE c This is the case when accessing another address in the same bank in bank active The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
2	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

The setting for areas 2 and 3 is common.

00: 2 cycles

01: 3 cycles

10: 5 cycles

11: 8 cycles

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRW and WTRC[1:0] bit settings are used in both areas in common.

If only one area is connected to the SDRAM, specify area 3. In this case, spec as normal space or SRAM with byte selection.

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inse between the second or subsequent access c burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

				0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait inpu valid. The specification by this bit is valid ever the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	-	-	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV	-	-	-	A3ROW[1:0]		-	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R/W	R/W	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
20, 19	A2ROW[1:0]	00	R/W	Number of Bits of Row Address for Area 2 Specify the number of bits of row address fo 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
17, 16	A2COL[1:0]	00	R/W	Number of Bits of Column Address for Area Specify the number of bits of column addres area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

				1: Deep power-down mode
12	SLOW	0	R/W	Low-Frequency Mode Specifies the output timing of command, addr write data for SDRAM and the latch timing of from SDRAM. Setting this bit makes the hold command, address, write and read data exter half cycle (output or read at the falling edge o This mode is suitable for SDRAM with low-fre clock. 0: Command, address, and write data for SDR output at the rising edge of CK. Read data SDRAM is latched at the rising edge of CK. 1: Command, address, and write data for SDR output at the falling edge of CK. Read data SDRAM is latched at the falling edge of CK
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation SDRAM is performed. 0: No refresh 1: Refresh

9	PDOWN	0	R	Power-Down Mode Specifies whether the SDRAM will enter pow mode after the access to the SDRAM. With t being set to 1, after the SDRAM is accessed signal is driven low and the SDRAM enters p down mode. 0: The SDRAM does not enter power-down m being accessed. 1: The SDRAM enters power-down mode aft accessed.
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharg (using READA and WRITA commands) or in active mode (using READ and WRIT comma 0: Auto-precharge mode (using READA and commands) 1: Bank active mode (using READ and WRIT commands) Note: Bank active mode can be set only in and only the 16-bit bus width can be both the CS2 and CS3 spaces are se SDRAM, specify auto-precharge mod
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

				This bit is always read as 0. The write value s always be 0.
1, 0	A3COL[1:0]	00	R/W	Number of Bits of Column Address for Area 3 Specify the number of bits of the column addr area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

other than B'000.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	-	-	-	-	-	-	-	CMF	CMIE	CKS[2:0]			R
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/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs betwe refresh timer counter (RTCNT) and refresh ti constant register (RTCOR). This bit is set or the following conditions. 0: Clearing condition: When 0 is written in CM reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTC RTCOR is satisfied.

				000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096
2 to 0	RRC[2:0]	000	R/W	Refresh Count Specify the number of continuous refresh cycl the refresh request occurs after the coincidenc values of the refresh timer counter (RTCNT) a refresh time constant register (RTCOR). Thes can make the period of occurrence of refresh 000: 1 time 001: 2 times 010: 4 times 011: 6 times 100: 8 times 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	-	-	-	-	-	-	-						
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/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Counter

section 8.5.6 (9), Relationship between Refresh Requests and Bus Cycles, and section 8.5
Arbitration.

When the CMIE bit in RTCSR is set to 1, an interrupt request is issued by this matching s
The request continues to be output until the CMF bit in RTCSR is cleared. Clearing the C
only affects the interrupt request and does not clear the refresh request. Therefore, a comb
of refresh request and interval timer interrupt can be specified so that the number of refre
requests are counted by using timer interrupts while refresh is performed periodically.

When RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel w
protection. This register is initialized to H'00000000 by a power-on reset and retains the v
manual reset and in software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	-	-	-	-	-	-	-						
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/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Register

Two data bus widths (8 bits and 16 bits) are available for normal memory and SRAM w
selection. Only 16-bit data bus width is available for SDRAM. For MPX-I/O, the data b
fixed at 8 bits or 16 bits, or 8 bits or 16 bits can be selected by the access address. Data a
is performed in accordance with the data bus width of the device. This also means that w
longword data is read from a byte-width device, the read operation must be done four ti
this LSI, data alignment and conversion of data length is performed automatically betwe
respective interfaces.

Tables 8.5 to 8.8 show the relationship between device data width and access unit. Note
addresses corresponding to the strobe signals for the 16-bit bus width differ between big
and little endian. WE1 indicates the 0 address in big-endian mode, but WE0 indicates th
address in little-endian mode.

Table 8.5 16-Bit External Device Access and Data Alignment in Big-Endian Mod

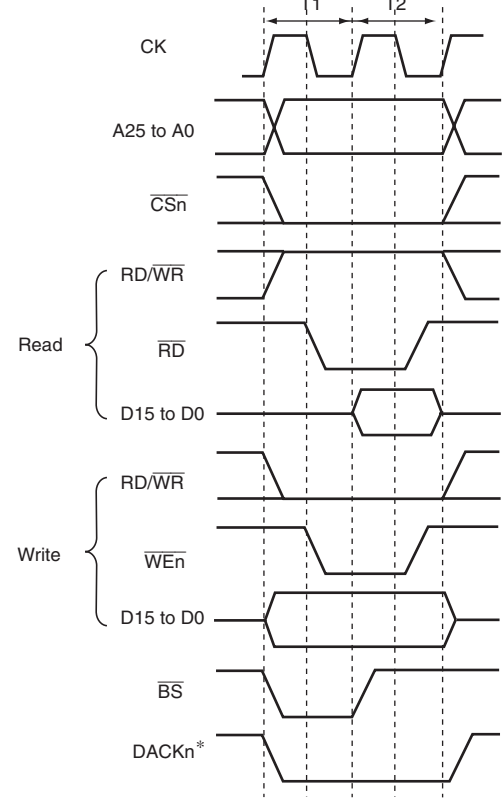
		Data Bus		Strobe Signa	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{WE1}$, $\overline{DQMLU}$**	**$\overline{WE0}$**
Byte access at 0		Data 7 to 0	—	Assert	—
Byte access at 1		—	Data 7 to 0	—	Asse
Byte access at 2		Data 7 to 0	—	Assert	—
Byte access at 3		—	Data 7 to 0	—	Asse
Word access at 0		Data 15 to 8	Data 7 to 0	Assert	Asse
Word access at 2		Data 15 to 8	Data 7 to 0	Assert	Asse
Longword access at 0	1st time at 0	Data 23 to 16	Data 31 to 24	Assert	Asse
	2nd time at 2	Data 7 to 0	Data 15 to 8	Assert	Asse

Word access at 2	1st time at 2	—	Data 15 to 8	—	Assert
	2nd time at 3	—	Data 7 to 0	—	Assert
Longword access at 0	1st time at 0	—	Data 31 to 24	—	Assert
	2nd time at 2	—	Data 23 to 16	—	Assert
	3rd time at 2	—	Data 15 to 8	—	Assert
	4th time at 3	—	Data 7 to 0	—	Assert

Longword access at 0	1st time at 0	Data 15 to 8	Data 7 to 0	Assert	Asse
	2nd time at 2	Data 31 to 24	Data 23 to 16	Assert	Asse

Table 8.8 8-Bit External Device Access and Data Alignment in Little-Endian Mo

		Data Bus		Strobe Signa	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{WE1}$, $\overline{DQMLU}$**	**$\overline{WE0}$**
Byte access at 0		—	Data 7 to 0	—	Asse
Byte access at 1		—	Data 7 to 0	—	Asse
Byte access at 2		—	Data 7 to 0	—	Asse
Byte access at 3		—	Data 7 to 0	—	Asse
Word access at 0	1st time at 0	—	Data 7 to 0	—	Asse
	2nd time at 1	—	Data 15 to 8	—	Asse
Word access at 2	1st time at 2	—	Data 7 to 0	—	Asse
	2nd time at 3	—	Data 15 to 8	—	Asse
Longword access at 0	1st time at 0	—	Data 7 to 0	—	Asse
	2nd time at 2	—	Data 15 to 8	—	Asse
	3rd time at 2	—	Data 23 to 16	—	Asse
	4th time at 3	—	Data 31 to 24	—	Asse



Note: * The waveform for DACKn is when active low is specified.

Figure 8.2 Normal Space Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is out
least significant bit of the address, but since there is no access size specification, 16 bits a

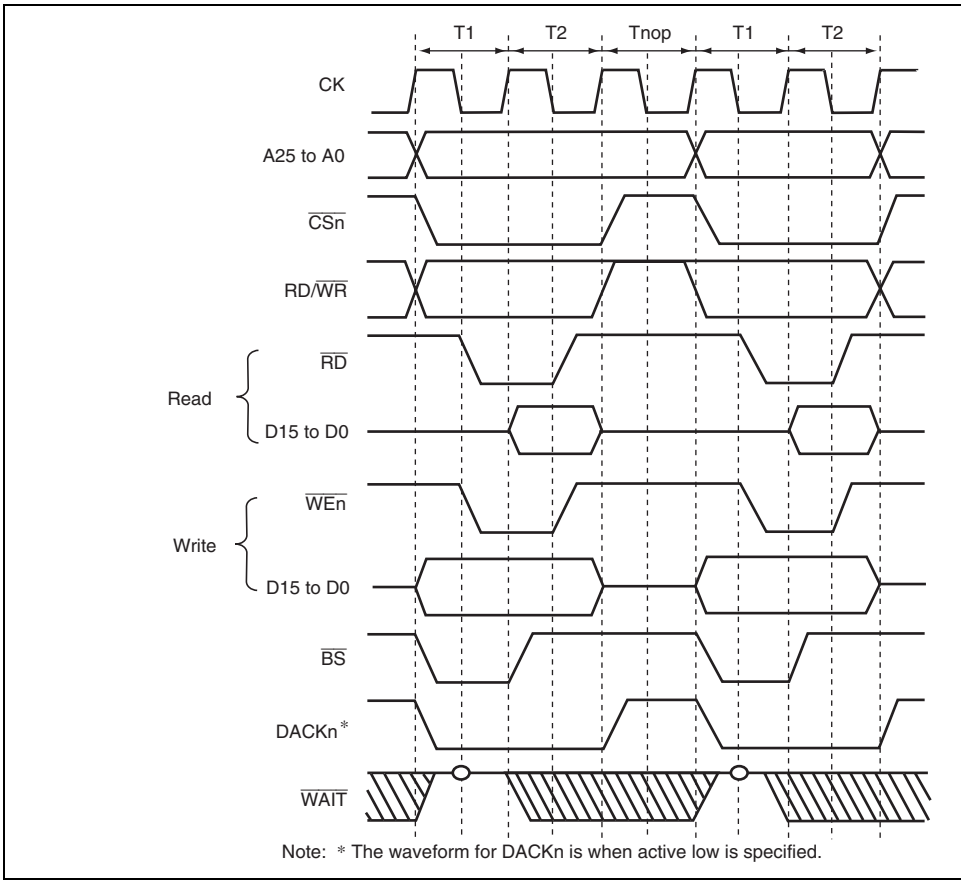


Figure 8.3 Continuous Access for Normal Space 1
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0
(Access Wait = 0, Cycle Wait = 0)

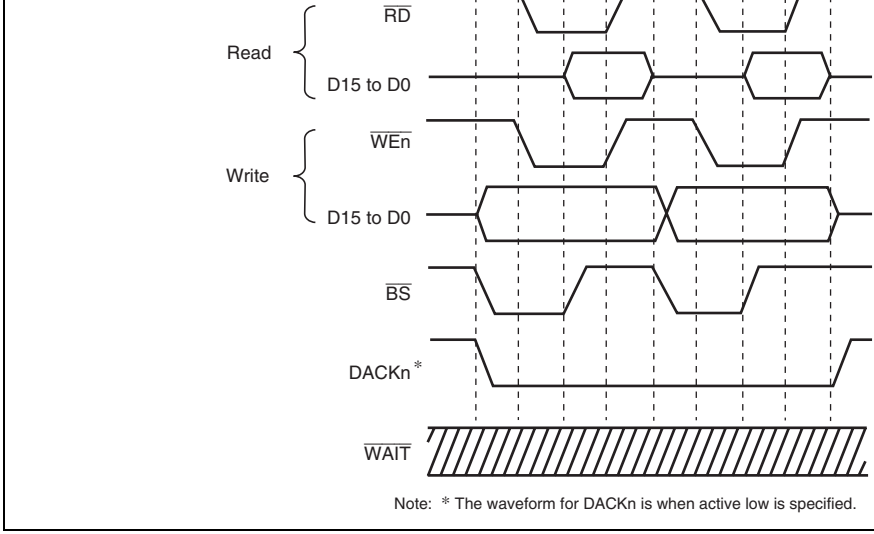


Figure 8.4 Continuous Access for Normal Space 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

Figure 8.5 Example of 16-Bit Data-Width SRAM Connection

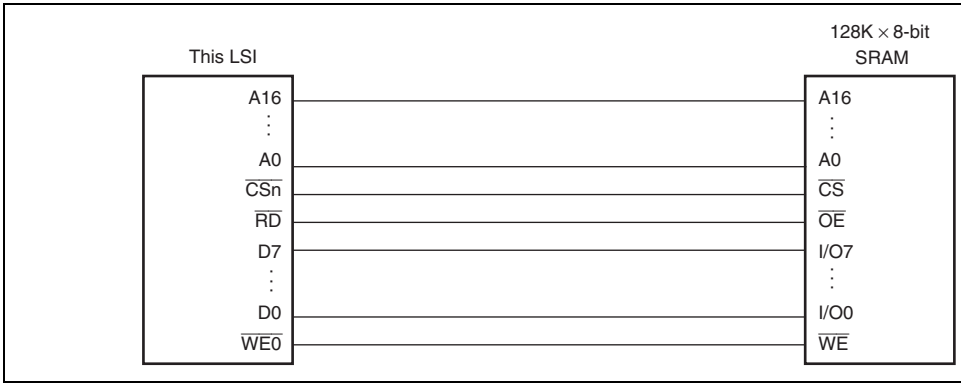
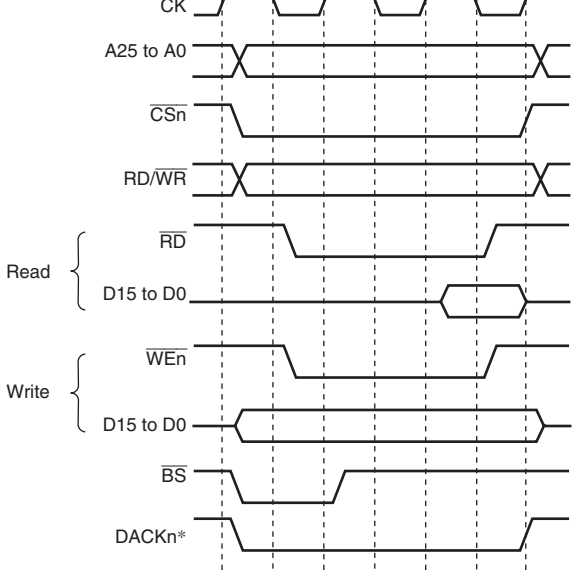
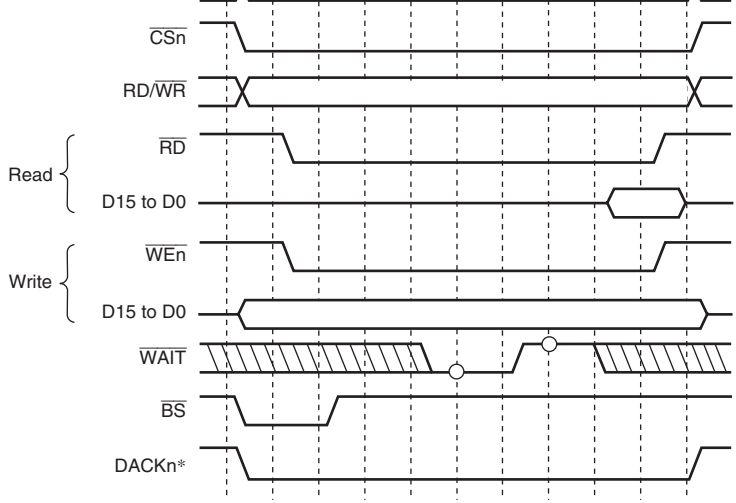


Figure 8.6 Example of 8-Bit Data-Width SRAM Connection



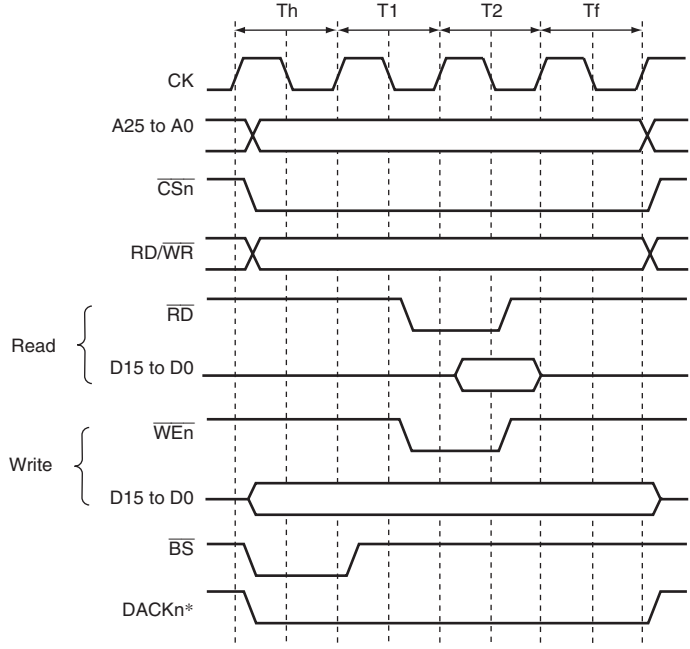
Note: * The waveform for DACKn is when active low is specified.

Figure 8.7 Wait Timing for Normal Space Access (Software Wait Only)



Note: * The waveform for DACKn is when active low is specified.

Figure 8.8 Wait Cycle Timing for Normal Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)



Note: * The waveform for DACKn is when active low is specified.

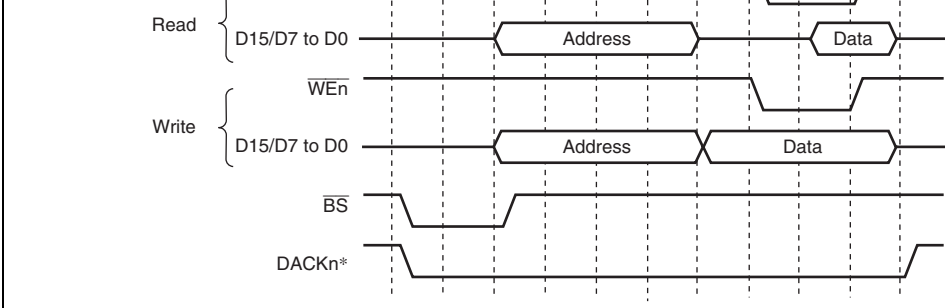
Figure 8.9 $\overline{\text{CSn}}$ Assert Period Expansion

cycles by setting the MPXW bit in CS5WCR to 1.

The RD/$\overline{WR}$ signal is output at the same time as the $\overline{CS5}$ signal; it is high in the read cy
low in the write cycle.

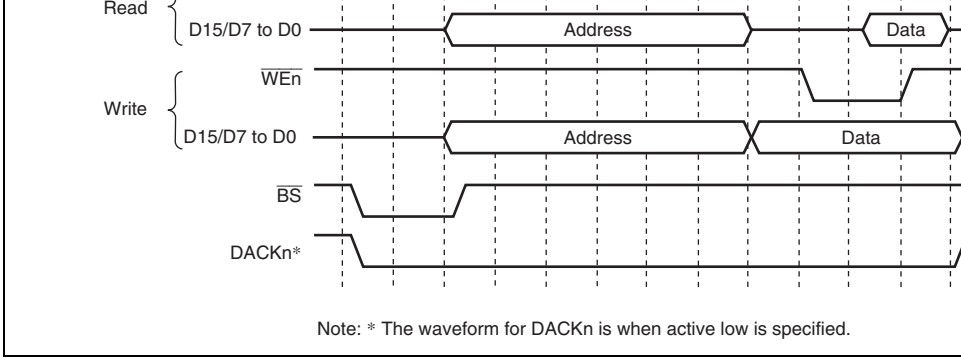
The data cycle is the same as that in a normal space access.

Timing charts are shown in figures 8.10 to 8.12.



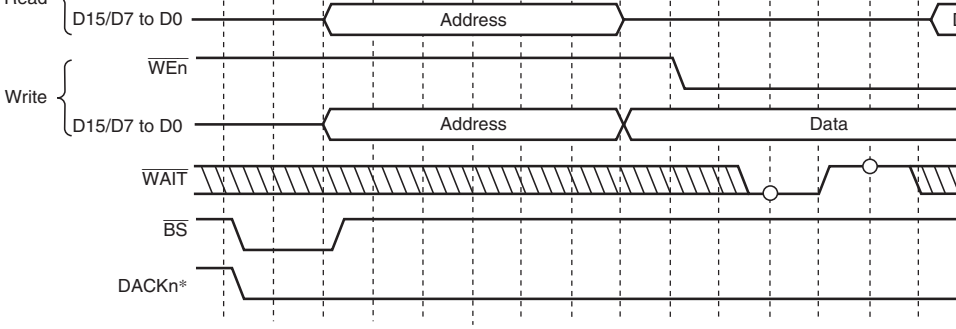
Note: * The waveform for DACKn is when active low is specified.

Figure 8.10 Access Timing for MPX Space (Address Cycle No Wait, Data Cycle N



Note: * The waveform for DACKn is when active low is specified.

Figure 8.11 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle N



Note: * The waveform for DACKn is when active low is specified.

Figure 8.12 Access Timing for MPX Space (Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

up to 2 spaces. The data bus width of the area that is connected to SDRAM can be set to
only.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are su
SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{\text{RASL}}$, $\overline{\text{CASL}}$, RD/$\overline{\text{WR}}$, and specific addre
These commands supports:

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks pre-charge (PALL)
- Specified bank pre-charge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with pre-charge (READA)
- Write (WRIT)
- Write with pre-charge (WRITA)
- Write mode register (MRS, EMRS)

The byte to be accessed is specified by $\overline{\text{DQMUL}}$ and $\overline{\text{DQMLL}}$. Reading or writing is pe
for a byte whose corresponding $\overline{\text{DQMxx}}$ is low. For details on the relationship between
and the byte to be accessed, see section 8.5.1, Endian/Access Size and Data Alignment.

Figure 8.13 shows an example of the connection of the SDRAM with the LSI.

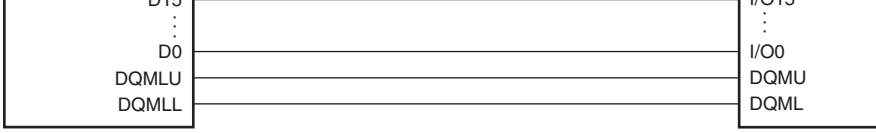


Figure 8.13 Example of 16-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

(2) Address Multiplexing

An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ[1:0] in CSnBCR, bits A2ROW and A2COL[1:0], A3ROW[1:0], and A3COL[1:0] in SDCR. Tables 8.9 to 8.11 show the relationship between the settings of bits BSZ[1:0], A2ROW[1:0], A2COL[1:0], A3ROW A3COL[1:0] and the bits output at the address pins. Do not specify those bits in the mann than this table, otherwise the operation of this LSI is not guaranteed. A29 to A18 are not multiplexed and the original values of address are always output at these pins.

The A0 pin of SDRAM specifies a word address. Therefore, connect the A0 pin of SDRA A1 pin of the LSI; then connect the A1 pin of SDRAM to the A2 pin of the LSI, and so o

A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21*2	A21*2	A12 (BA1)	Specifies
A12	A20*2	A20*2	A11 (BA0)	
A11	A19	L/H*1	A10/AP	Specifies address/
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord access mode.

2. Bank address specification

A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A13 (BA1)	Specifies
A13	A21*2	A21*2	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*1	A10/AP	Specifies address/p
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mword × 16 bits × 4 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the mode.
2. Bank address specification

A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13 (BA1)	Specifies
A13	A22*2	A22*2	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*1	A10/AP	Specifies address/p
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord access mode.

2. Bank address specification

A16	A26	A16		
A15	A25	A15		
A14	A24*[2]	A24*[2]	A13 (BA1)	Specifies
A13	A23*[2]	A23*[2]	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*[1]	A10/AP	Specifies address/p
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accordi access mode.

2. Bank address specification

A16	A25	A16		
A15	A24*[2]	A24*[2]	A14 (BA1)	Specifies
A14	A23*[2]	A23*[2]	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*[1]	A10/AP	Specifies address/
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord access mode.

2. Bank address specification

A16	A26	A16		
A15	A25*2	A25*2	A14 (BA1)	Specifies b
A14	A24*2	A24*2	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	Specifies address/pr
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accordi access mode.

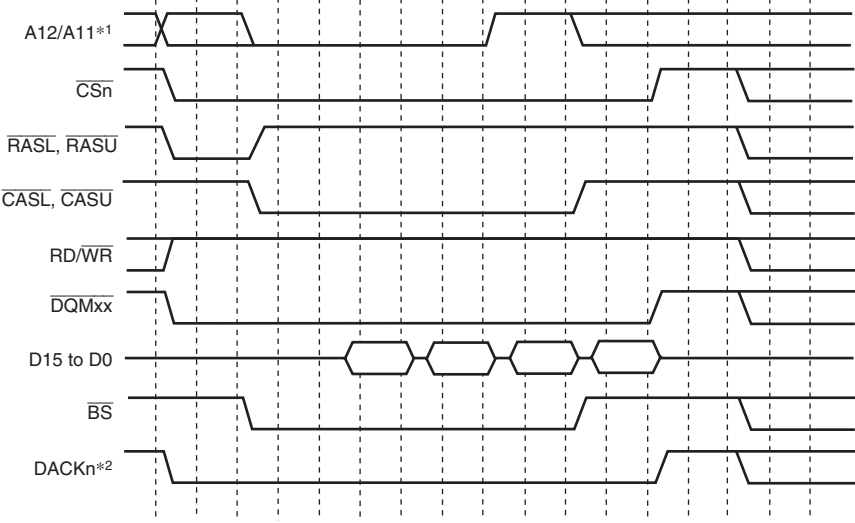
2. Bank address specification

Table 8.12 Relationship between Access Size and Number of Bursts

Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bits	8

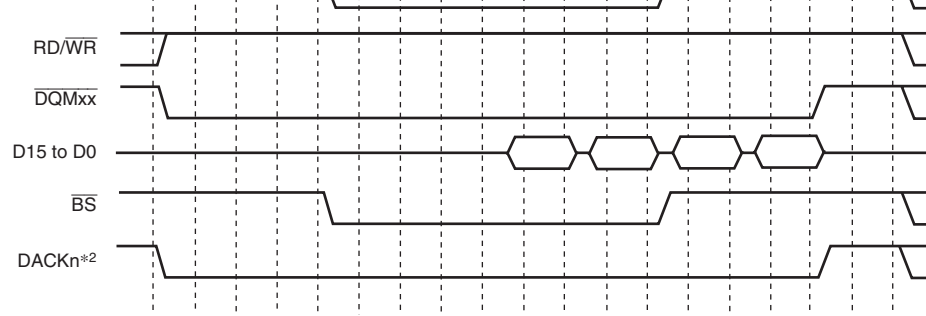
Figures 8.14 and 8.15 show a timing chart in burst read. In burst read, an ACTV comma output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, th command is issued in the Tc4 cycle, and the read data is received at the rising edge of th clock (CK) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion o precharge induced by the READA command in the SDRAM. In the Tap cycle, a new co will not be issued to the same bank. However, access to another CS space or another ban same SDRAM space is enabled. The number of Tap cycles is specified by the WTRP1 a WTRP0 bits in CS3WCR.

In this LSI, wait cycles can be inserted by specifying each bit in CS3WCR to connect th in variable frequencies. Figure 8.15 shows an example in which wait cycles are inserted number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cyc the READ command is output can be specified using the WTRCD1 and WTRCD0 bits i CS3WCR. If the WTRCD1 and WTRCD0 bits specify one cycles or more, a Trw cycle NOT command is issued is inserted between the Tr cycle and Tc1 cycle. The number of from the Tc1 cycle where the READ command is output to the Td1 cycle where the read latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 an bits in CS2WCR or the A3CL1 and A3CL0 bits in CS3WCR and WTRCD0 bit in CS3W number of cycles from Tc1 to Td1 corresponds to the SDRAM CAS latency. The CAS l the SDRAM is normally defined as up to three cycles. However, the CAS latency in this



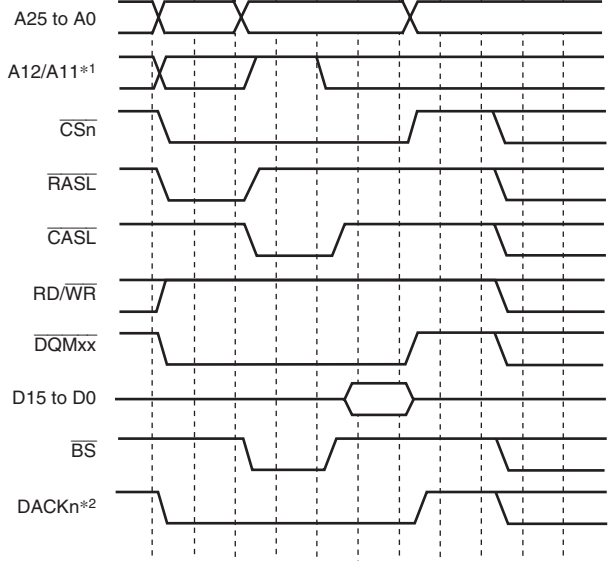
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.14 Burst Read Basic Timing (CAS Latency 1, Auto-Precharge)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

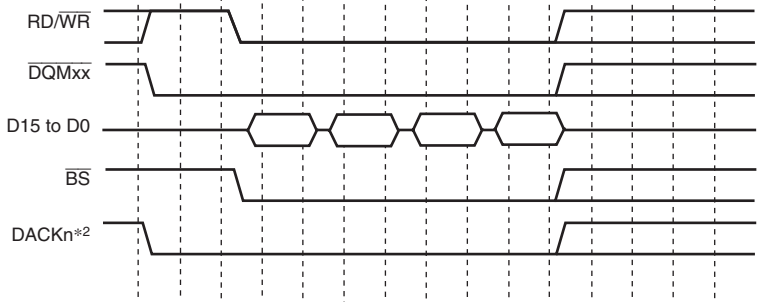
Figure 8.15 Burst Read Wait Specification Timing (CAS Latency 2, WTRCD[1:0] = 1 Cycle, Auto-Precharge)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

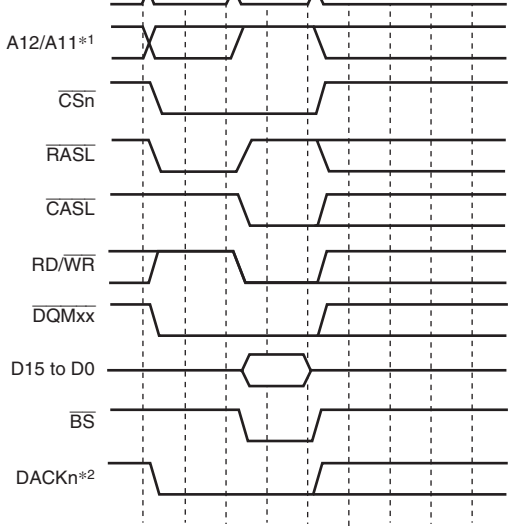
Figure 8.16 Basic Timing for Single Read (CAS Latency 1, Auto-Precharge

Figure 8.17 shows a timing chart for burst writes. In burst write, an ACTV command is
the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WR
command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the
is output simultaneously with the write command. After the write command with the aut
precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followe
Tap cycle that waits for completion of the auto-precharge induced by the WRITA comm
SDRAM. Between the Trwl and the Tap cycle, a new command will not be issued to the
bank. However, access to another CS space or another bank in the same SDRAM space
The number of Trw1 cycles is specified by the TRWL1 and TRWL0 bits in CS3WCR. T
number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.17 Basic Timing for Burst Write (Auto-Precharge)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.18 Single Write Basic Timing (Auto-Precharge)

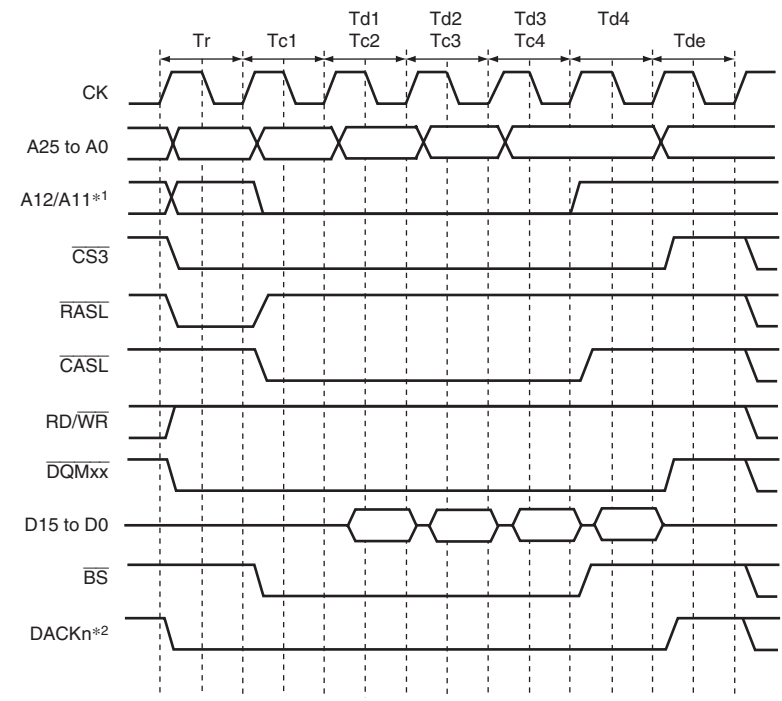
accessing the same row address in the same bank, it is possible to issue the READ or WR
command immediately, without issuing an ACTV command. As SDRAM is internally di
into several banks, it is possible to activate one row address in each bank. If the next acce
different row address, a PRE command is first issued to precharge the relevant bank, then
precharging is completed, the access is performed by issuing an ACTV command followe
READ or WRIT command. If this is followed by an access to a different row address, the
time will be longer because of the precharging performed after the access request is issue
number of cycles between issuance of the PRE command and the ACTV command is det
by the WTRP1 and WTPR0 bits in CS3WCR.

In a write, when an auto-precharge is performed, a command cannot be issued to the sam
for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank act
is used, READ or WRIT commands can be issued successively if the row address is the s
number of cycles can thus be reduced by Trwl + Tap cycles for each write.

There is a limit on tRAS, the time for placing each bank in the active state. If there is no g
that there will not be a cache hit and another row address will be accessed within the peri
which this value is maintained by program execution, it is necessary to set auto-refresh ar
refresh cycle to no more than the maximum value of tRAS.

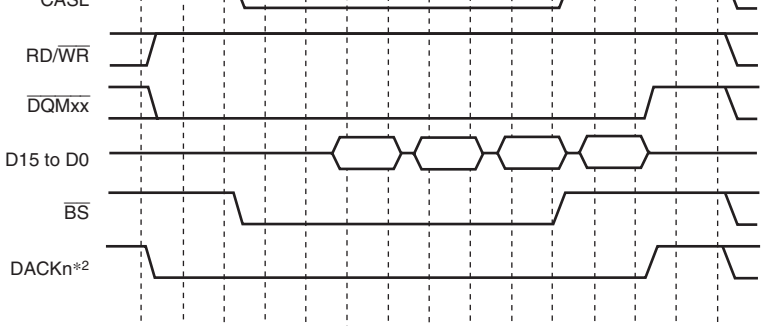
A burst read cycle without auto-precharge is shown in figure 8.19, a burst read cycle for t
row address in figure 8.20, and a burst read cycle for different row addresses in figure 8.2
Similarly, a burst write cycle without auto-precharge is shown in figure 8.22, a burst writ
for the same row address in figure 8.23, and a burst write cycle for different row addresse
figure 8.24.

In figure 8.20, a Tnop cycle in which no operation is performed is inserted before the Tc
issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS late
the $\overline{\text{DQMxx}}$ signal that specifies the read byte in the data read from the SDRAM. If the C



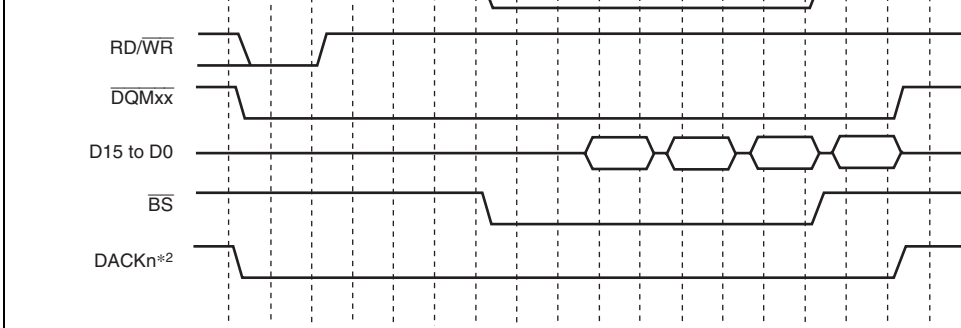
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.19 Burst Read Timing (Bank Active, Different Bank, CAS Latenc



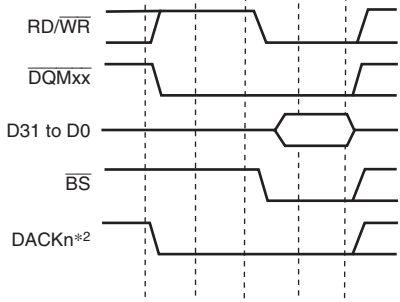
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.20 Burst Read Timing (Bank Active, Same Row Addresses in the Same Ba Latency 1)



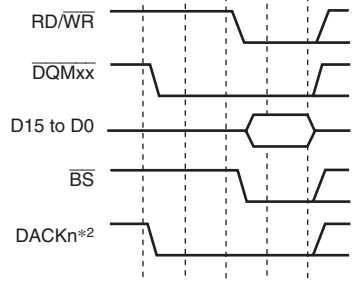
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.21 Burst Read Timing (Bank Active, Different Row Addresses in the Sa CAS Latency 1)



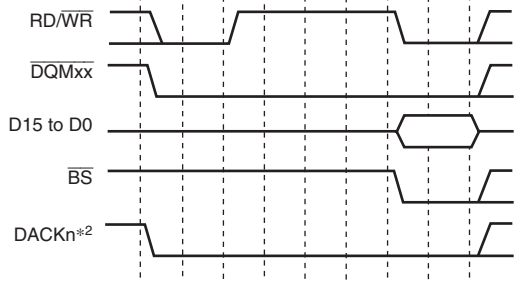
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.22 Single Write Timing (Bank Active, Different Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.23 Single Write Timing (Bank Active, Same Row Addresses in the Sam

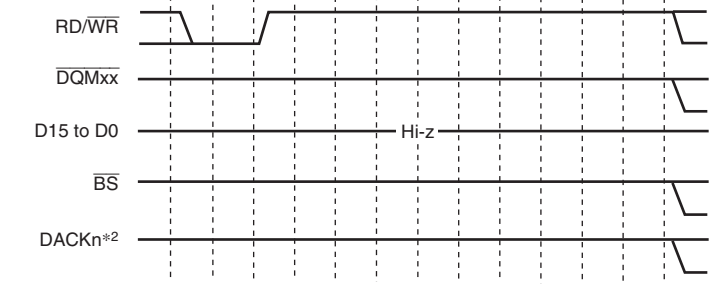


Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.24 Single Write Timing (Bank Active, Different Row Addresses in the Sar

be set so as to satisfy the refresh interval stipulation for the SDRAM used. First make th
for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, then make the CKS2 t
and RRC2 to RRC0 settings. When the clock is selected by bits CKS2 to CKS0, RTCN
counting up from the value at that time. The RTCNT value is constantly compared with
RTCOR value, and if the two values are the same, a refresh request is generated and an
refresh is performed for the number of times specified by the RRC2 to RRC0. At the sa
RTCNT is cleared to zero and the count-up is restarted.

Figure 8.25 shows the auto-refresh cycle timing. After starting, the auto refreshing, PAL
command is issued in the Tp cycle to make all the banks to pre-charged state from activ
when some bank is being pre-charged. Then REF command is issued in the Trr cycle af
inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS
new command is not issued for the duration of the number of cycles specified by the W
WTRC0 bits in CS3WCR after the Trr cycle. The WTRC1 and WTRC0 bits must be se
satisfy the SDRAM refreshing cycle time stipulation (tRC). An idle cycle is inserted be
Tp cycle and Trr cycle when the setting value of the WTRP1 and WTRP0 bits in CS3W
longer than or equal to 1 cycle.



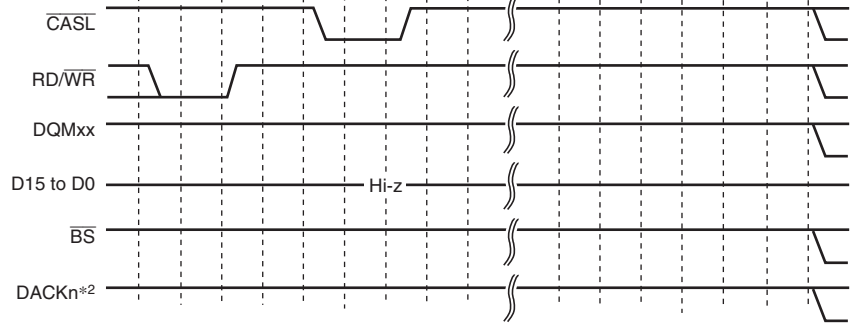
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.25 Auto-Refresh Timing

Self-refresh timing is shown in figure 8.20. Settings must be made so that self-refresh ch
data retention are performed correctly, and auto-refreshing is performed at the correct in
When self-refreshing is activated from the state in which auto-refreshing is set, or when
standby mode other than through a power-on reset, auto-refreshing is restarted if the RF
set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the tran
from clearing of self-refresh mode to the start of auto-refreshing takes time, this time sh
taken into consideration when setting the initial value of RTCNT. Making the RTCNT v
than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standb
entered using the LSI standby function, and is maintained even after recovery from stan
due to an interrupt. Note that the necessary signals such as CKE must be driven even in
state by setting the HIZCNT bit in CMNCR to 1.

The self-refresh state is not cleared by a manual reset. In case of a power-on reset, the bu
controller's registers are initialized, and therefore the self-refresh state is cleared.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.26 Self-Refresh Timing

refresh interval, refresh cannot be executed and the SDRAM contents may be lost.

If a new refresh request occurs while waiting for the previous refresh request, the previo
request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the
mastership occupation must be prevented from occurring.

If a bus mastership is requested during self-refresh, the bus will not be released until the
completed.

commands, addresses, write data, and read data are not guaranteed. Take the operating fr
and timing design into consideration when making the SLOW bit setting.

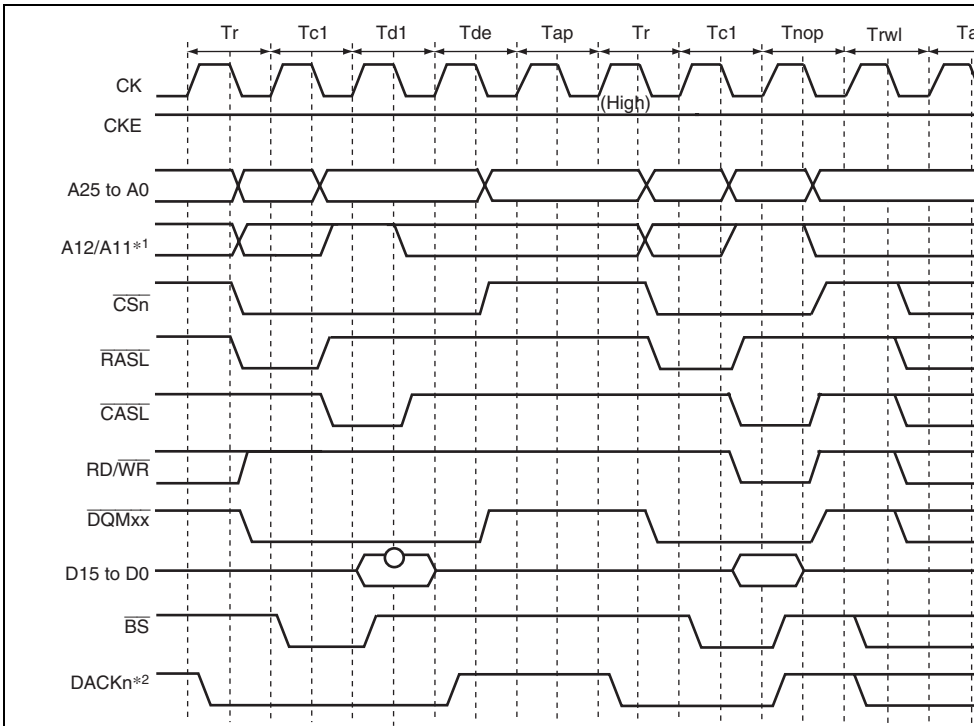
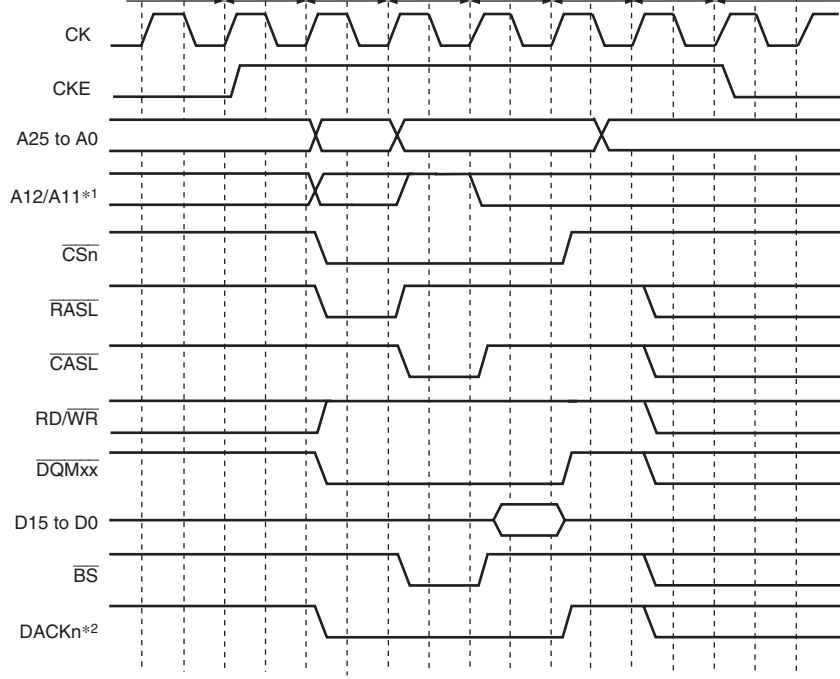


Figure 8.27 Low-Frequency Mode Access Timing



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.28 Power-Down Mode Access Timing

and to address H'FFFC3000 + X for area 3 SDRAM. In this operation the data is ignored.
mode write is performed as a byte-size access. To set burst read/single write, CAS latency
wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written i
size access to the addresses shown in table 8.13. In this time 0 is output at the external ad
pins of A12 or later.

Table 8.13 Access Address in SDRAM Mode Register Write

- Setting for Area 2

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Add
16 bits	2	H'FFFC4440	H'0000440
	3	H'FFFC4460	H'0000460

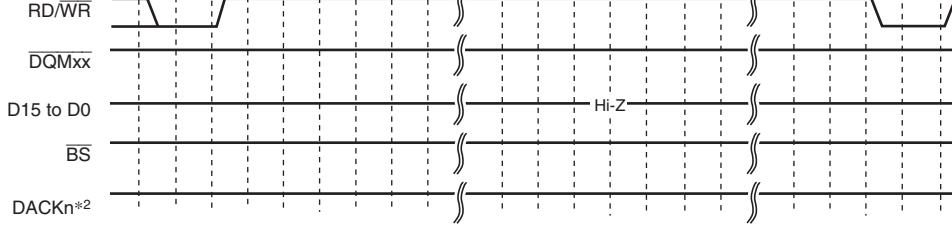
Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Add
16 bits	2	H'FFFC4040	H'0000040
	3	H'FFFC4060	H'0000060

16 bits	2	H'FFFC5040	H'0000040
	3	H'FFFC5060	H'0000060

Mode register setting timing is shown in figure 8.29. A PALL command (all bank pre-cl
command) is firstly issued. A REF command (auto refresh command) is then issued 8 ti
MRS command (mode register write command) is finally issued. Idle cycles, of which n
specified by the WTRP1 and WTRP0 bits in CS3WCR, are inserted between the PALL
first REF. Idle cycles, of which number is specified by the WTRC1 and WTRC0 bits in
are inserted between REF and REF, and between the 8th REF and MRS. Idle cycles, of
number is one or more, are inserted between the MRS and a command to be issued next

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL com
power-on. Refer to the manual of the SDRAM for the idle time to be needed. When the
width of the reset signal is longer than the idle time, mode register setting can be started
immediately after the reset, but care should be taken when the pulse width of the reset si
shorter than the idle time.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

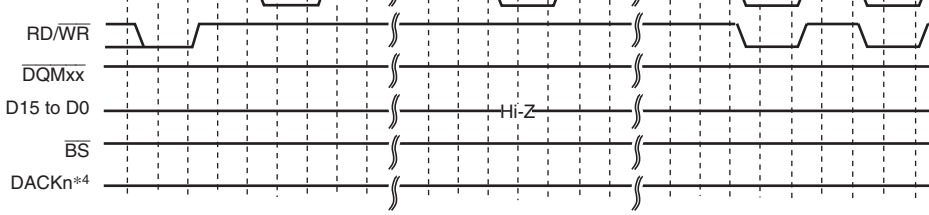
Figure 8.29 SDRAM Mode Write Timing (Based on JEDEC)

registers as the normal SDRAM. This LSI supports issuing of the EMRS command.

The EMRS command is issued according to the conditions specified in table below. For
if data H'0YYYYYYY is written to address H'FFFC5XX0 in longword, the commands a
to the CS3 space in the following sequence: PALL -> REF × 8 -> MRS -> EMRS. In thi
MRS and EMRS issue addresses are H'0000XX0 and H'YYYYYYY, respectively. If da
H'1YYYYYYY is written to address H'FFFC5XX0 in longword, the commands are issu
CS3 space in the following sequence: PALL -> MRS -> EMRS.

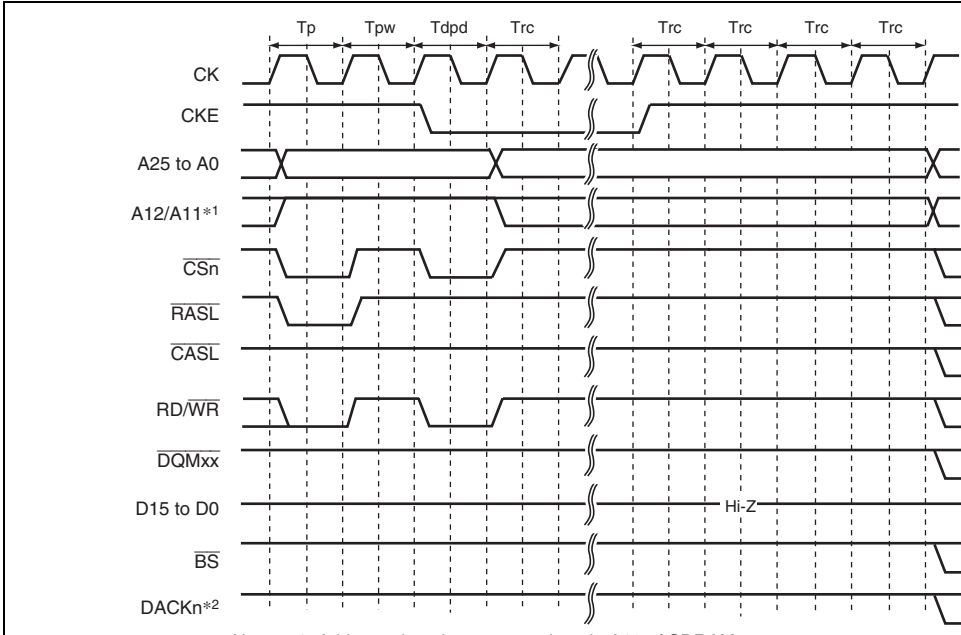
Table 8.14 Output Addresses when EMRS Command Is Issued

Command to be Issued	Access Address	Access Data	Write Access Size	MRS Command Issue Address	EMRS Comm Issue
CS2 MRS	H'FFFC4XX0	H'********	16 bits	H'0000XX0	—
CS3 MRS	H'FFFC5XX0	H'********	16 bits	H'0000XX0	—
CS2 MRS + EMRS (with refresh)	H'FFFC4XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYY
CS3 MRS + EMRS (with refresh)	H'FFFC5XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYY
CS2 MRS + EMRS (without refresh)	H'FFFC4XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYY
CS3 MRS + EMRS (without refresh)	H'FFFC5XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYY



Notes: 1. Address pin to be connected to pin BA1 of SDRAM.
2. Address pin to be connected to pin BA0 of SDRAM.
3. Address pin to be connected to pin A10 of SDRAM.
4. The waveform for DACKn is when active low is specified.

Figure 8.30 EMRS Command Issue Timing



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.31 Deep Power-Down Mode Transition Timing

to W0 bits in CSnWCR is inserted.

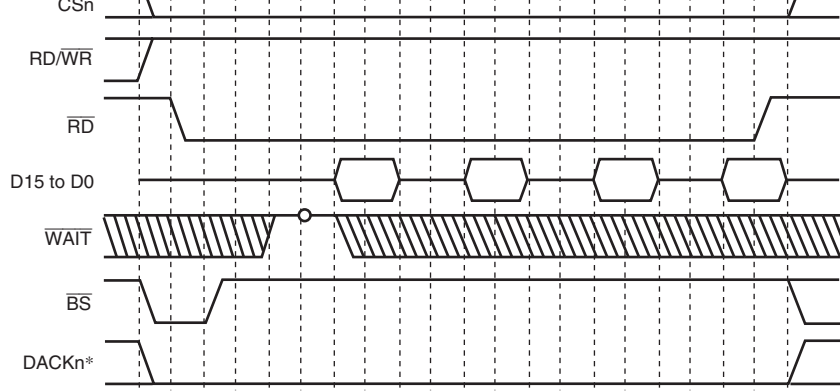
In the access to the burst ROM (clock asynchronous), the $\overline{BS}$ signal is asserted only to the access cycle. An external wait input is valid only to the first access cycle.

In the single access or write access that does not perform the burst operation in the burst R (clock asynchronous) interface, access timing is same as a normal space. In addition, there some restrictions on 16-byte write access. For details, see section 8.6, Usage Notes.

Table 8.15 lists a relationship between bus width, access size, and the number of bursts. F 8.32 shows a timing chart.

Table 8.15 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	Access Size	CSnWCR. BST[1:0] Bits	Number of Bursts	Access
8 bits	8 bits	Not affected	1	1
	16 bits	Not affected	2	1
	32 bits	Not affected	4	1
	16 bytes	00	16	1
		01	4	4
16 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	2	1
	16 bytes	00	8	1
		01	2	4
		10*	4	2
			2, 4, 2	3

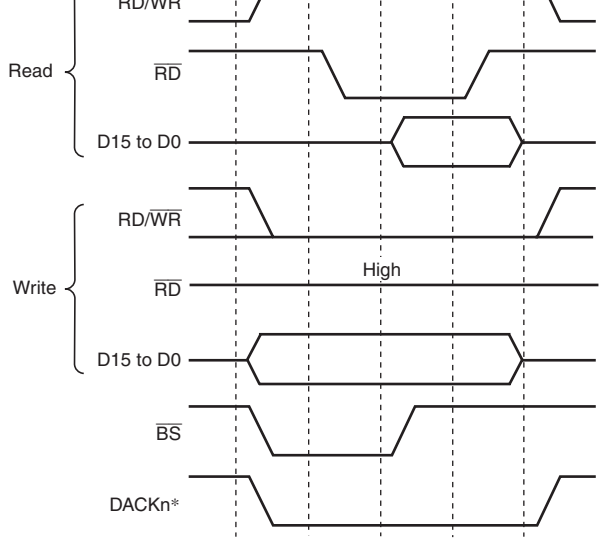


Note: * The waveform for DACKn is when active low is specified.

Figure 8.32 Burst ROM Access Timing (Clock Asynchronous)
(Bus Width = 32 Bits, 16-Byte Transfer (Number of Burst 4), Wait Cycles Inserte
Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles =

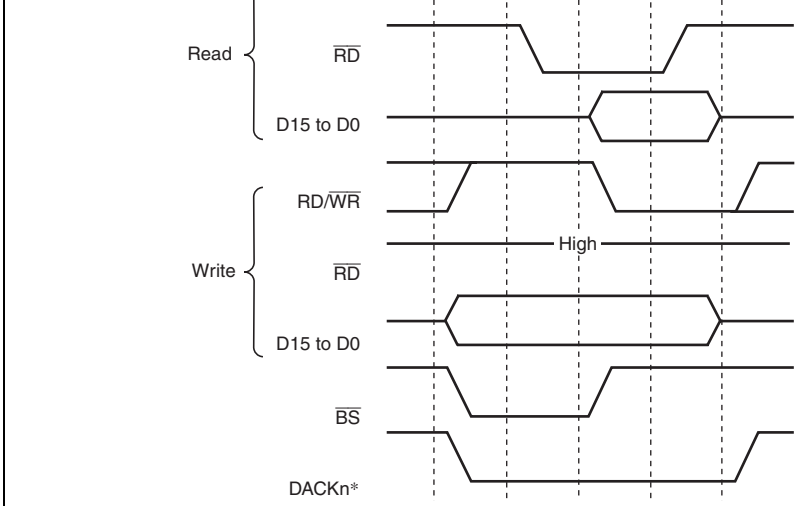
selection pin ($\overline{\text{WEn}}$). For details, please refer to the Data Sheet for the corresponding mem

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ pin and RD/$\overline{\text{WR}}$ pin timings change. Figu shows the basic access timing. In write access, data is written to the memory according to timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/$\overline{\text{WR}}$ negation to must be acquired by setting the HW1 and HW0 bits in CSnWCR. Figure 8.35 shows the a timing when a software wait is specified.



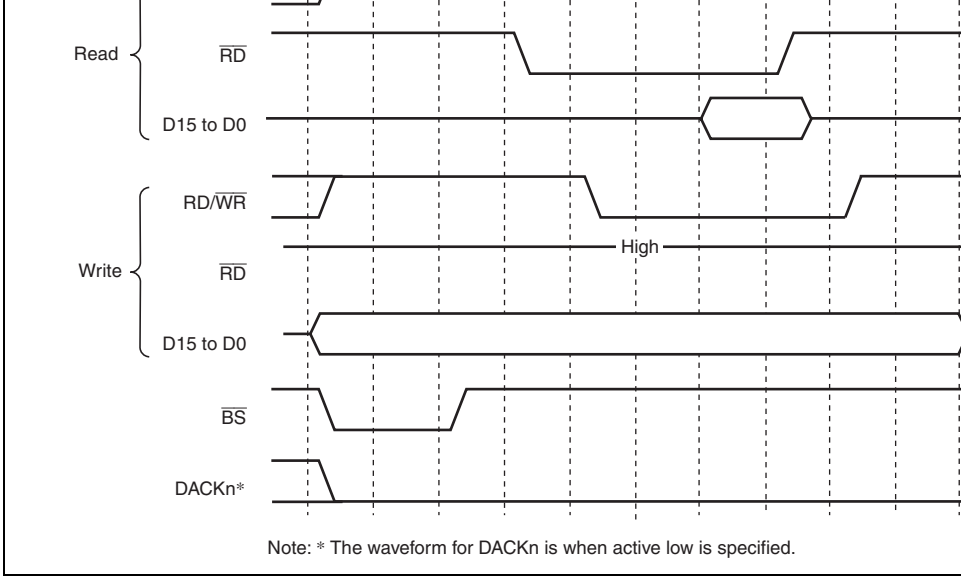
Note: * The waveform for DACKn is when active low is specified.

Figure 8.33 Basic Access Timing for SRAM with Byte Selection (BAS = 0



Note: * The waveform for DACKn is when active low is specified.

Figure 8.34 Basic Access Timing for SRAM with Byte Selection (BAS = 1)



Note: * The waveform for DACKn is when active low is specified.

Figure 8.35 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

Figure 8.36 Example of Connection with 16-Bit Data-Width SRAM with Byte Se

access cycle and an external wait input is also valid for the first access cycle.

If the bus width is 16 bits, the burst length must be specified as 8. The burst ROM interf
not support the 8-bit bus width for the burst ROM.

The burst ROM interface performs burst operations for all read access. For example, in a
longword access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit
read six times. These invalid data read cycles increase the memory access time and degr
program execution speed and DMA transfer speed. To prevent this problem, using 16-by
the DMA is recommended. The burst ROM interface performs write access in the same
normal space access.

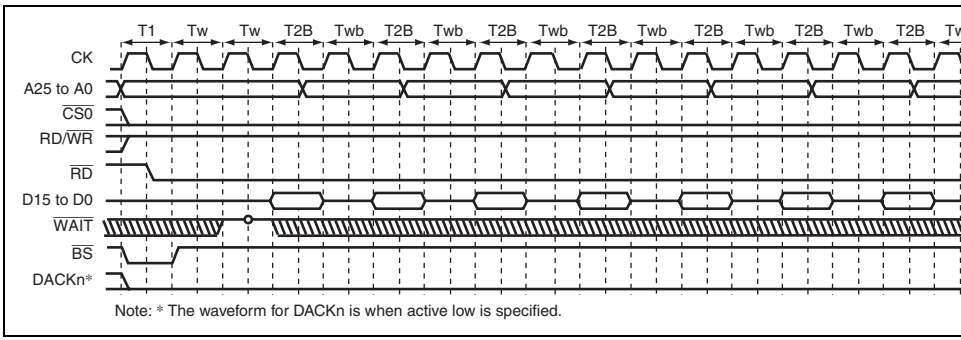


**Figure 8.37 Burst ROM Access Timing (Clock Synchronous)
(Burst Length = 8, Wait Cycles Inserted in First Access = 2,
Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)**

The conditions for setting the idle cycles between access cycles are shown below.

1. Continuous access cycles are write-read or write-write
2. Continuous access cycles are read-write for different spaces
3. Continuous access cycles are read-write for the same space
4. Continuous access cycles are read-read for different spaces
5. Continuous access cycles are read-read for the same space
6. Data output from an external device caused by DMA single address transfer is follow data output from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single address transfer is follow type of access (DMAIWA = 1)

For the specification of the number of idle cycles between access cycles described above, the description of each register.

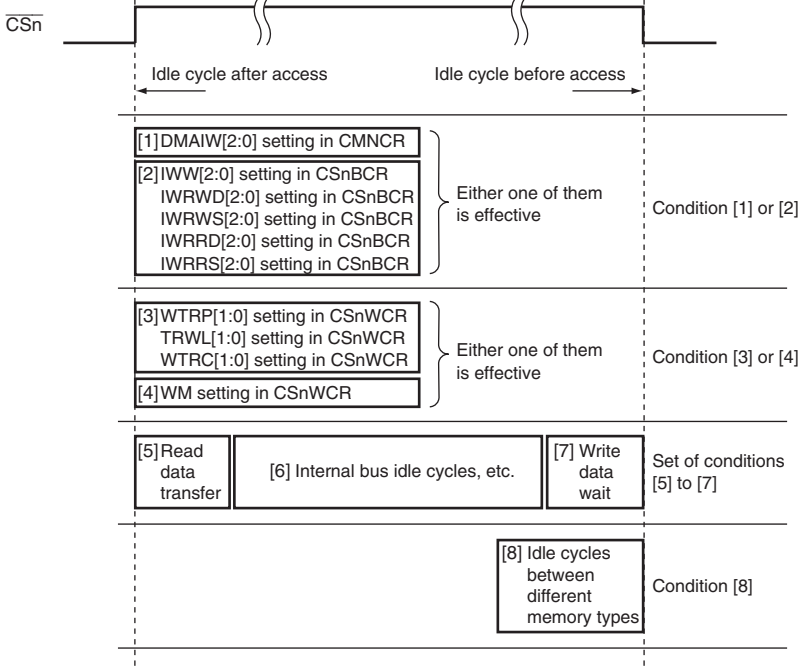
Besides the idle cycles between access cycles specified by the registers, idle cycles must inserted to interface with the internal bus or to obtain the minimum pulse width for a mul pin ($\overline{\text{WEn}}$). The following gives detailed information about the idle cycles and describes estimate the number of idle cycles.

The number of idle cycles on the external bus from $\overline{\text{CSn}}$ negation to $\overline{\text{CSn}}$ or $\overline{\text{CSm}}$ assertio described below.

There are eight conditions that determine the number of idle cycles on the external bus as in table 8.16. The effects of these conditions are shown in figure 8.38.

(2)	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access other than single address transfer. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS1 space followed by reading other CS space, the bits IWRRD[2:0] in CS1BCR should be set to B'100 to specify six or more idle cycles. This condition is effective only for access cycles other than single address transfer and generates idle cycles after the access is completed.	0 to 12	Do not set 0 for the idle cycles between types which are not to be accessed suc
(3)	SDRAM-related bits in CSnWCR	These bits specify precharge completion and startup wait cycles and idle cycles between commands for SDRAM access. This condition is effective only for SDRAM access and generates idle cycles after the access is completed	0 to 3	Specify these bits in accordance with the specification of the SDRAM.
(4)	WM in CSnWCR	This bit enables or disables external $\overline{\text{WAIT}}$ pin input for the memory types other than SDRAM. When this bit is cleared to 0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to 1 (disabled), no idle cycle is generated.	0 or 1	

		state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the BSC when the access size is larger than the external data bus width.		show the relationship between the clock ra the minimum numbe internal bus idle cycl
(7)	Write data wait cycles	During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect).	0 or 1	For write → write or read access cycles, successive access c without idle cycles ar frequently available the write buffer effec described in the left successive access c without idle cycles ar allowed, specify the number of idle cycles between access cycl through CSnBCR.
(8)	Idle cycles between different memory types	To ensure the minimum pulse width on the signal-multiplexed pins, idle cycles may be inserted before access after memory types are switched. For some memory types, idle cycles are inserted even when memory types are not switched.	0 to 2.5	The number of idle c depends on the targe memory types. See t 8.19.



Note: A total of four conditions (condition [1] or [2], condition [3] or [4], a set of conditions [5] to [7], and condition [8]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 8.38 Idle Cycle Conditions

	Transfer Mode	
DMAC Operation	**Dual Address**	**Single Address**
Write → write	0	2
Write → read	0 or 2	0
Read → write	0	0
Read → read	0	2

Notes: 1. The write → write and read → read columns in dual address transfer indicate t in the divided access cycles.

2. For the write → read cycles in dual address transfer, 0 means different channe activated successively and 2 means when the same channel is activated succ

3. The write → read and read → write columns in single address transfer indicate when different channels are activated successively. The "write" means transfe device with DACK to external memory and the "read" means transfer from exte memory to a device with DACK.

Byte SRAM (BAS = 0)	0	0	1	0	1	1	1.5	0
Byte SRAM (BAS = 1)	1	1	2	1	0	0	1.5	1
SDRAM	1	1	2	1	0	0	—	1
SDRAM (low-frequency mode)	1.5	1.5	2.5	1.5	0.5	—	1	1.5
Burst ROM (synchronous)	0	0	1	0	1	1	1.5	0

Figure 8.39 shows sample estimation of idle cycles between access cycles. In the actual
the idle cycles may become shorter than the estimated value due to the write buffer effec
become longer due to internal bus idle cycles caused by stalling in the pipeline due to C
instruction execution or CPU register conflicts. Please consider these errors when estima
idle cycles.

Condition	R → R	R → W	W → W	W → R	Note
[1] or [2]	0	0	0	0	CSnBCR is set to 0.
[3] or [4]	0	0	0	0	The WM bit is set to 1.
[5]	1	1	0	0	Generated after a read cycle.
[6]	0	2	2	0	See the Iϕ:Bϕ = 4:1 columns in table 8.17.
[7]	0	1	0	0	No idle cycle is generated for the second time due write buffer effect.
[5] + [6] + [7]	0	4	2	0	
[8]	0	0	0	0	Value for SRAM → SRAM access
Estimated idle cycles	1	4	2	0	Maximum value among conditions [1] or [2], [3] or [5] + [6] + [7], and [8]
Actual idle cycles	1	4	2	1	The estimated value does not match the actual va the W → R cycles because the internal idle cycles condition [6] is estimated as 0 but actually an inter cycle is generated due to execution of a loop conc check instruction.

Figure 8.39 Comparison between Estimated Idle Cycles and Actual Value

signal or other bus control signals. The states that do not allow bus mastership release ar
below.

1. Between the read and write cycles of a TAS instruction, or 64-bit transfer cycle of ar
   instruction
2. Multiple bus cycles generated when the data bus width is smaller than the access siz
   example, between bus cycles when longword access is made to a memory with a dat
   width of 8 bits)
3. 16-byte transfer by the DMAC
4. Setting the BLOCK bit in CMNCR to 1

Moreover, by using DPRTY bit in CMNCR, whether the bus mastership request is recei
can be selected during DMAC burst transfer.

The LSI has the bus mastership until a bus request is received from another device. Upo
acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LS
the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the
acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the external d
released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

With the SDRAM interface, all bank pre-charge commands (PALLs) are issued when ac
exist and the bus is released after completion of a PALL command.

The bus sequence is as follows. The address bus and data bus are placed in a high-impec
synchronized with the rising edge of CK. The bus mastership enable signal is asserted 0
after the above timing, synchronized with the falling edge of CK. The bus control signal
$\overline{\text{CSn}}$, $\overline{\text{RASL}}$, $\overline{\text{CASL}}$, CKE, $\overline{\text{DQMxx}}$, $\overline{\text{WEn}}$, $\overline{\text{RD}}$, and RD/$\overline{\text{WR}}$) are placed in the high-imp
state at subsequent rising edges of CK. Bus request signals are sampled at the falling edg
CKIO. Note that CKE, $\overline{\text{RASL}}$, and $\overline{\text{CASL}}$ can continue to be driven at the previous valu
the bus-released state by setting the HIZCNT bit in CMNCR.

mastership is returned from the external device. If the bus mastership is not returned for a refreshing period or longer, the contents of SDRAM cannot be guaranteed because a refre cannot be executed.

While releasing the bus mastership, the SLEEP instruction (to enter sleep mode or standb as well as a manual reset, cannot be executed until the LSI obtains the bus mastership.

The $\overline{\text{BREQ}}$ input signal is ignored in standby mode and the $\overline{\text{BACK}}$ output signal is placed high impedance state. If the bus mastership request is required in this state, the bus maste must be released by pulling down the $\overline{\text{BACK}}$ pin to enter standby mode.

The bus mastership release ($\overline{\text{BREQ}}$ signal for high level negation) after the bus mastershi ($\overline{\text{BREQ}}$ signal for low level assertion) must be performed after the bus usage permission (signal for low level assertion). If the $\overline{\text{BREQ}}$ signal is negated before the $\overline{\text{BACK}}$ signal is a only one cycle of the $\overline{\text{BACK}}$ signal is asserted depending on the timing of the $\overline{\text{BREQ}}$ sign negated and this may cause a bus contention between the external device and the LSI.

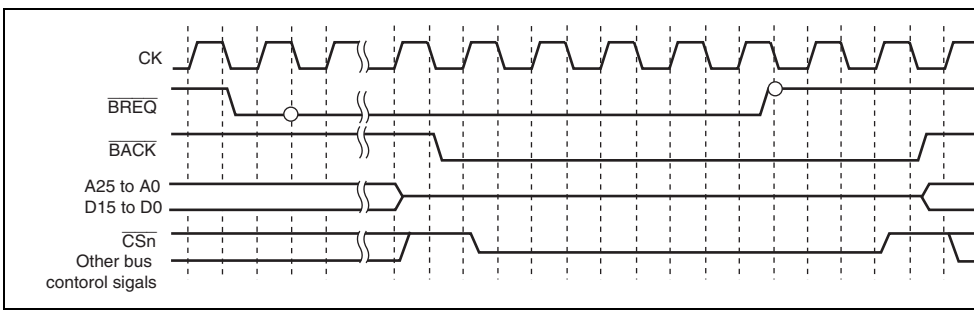


Figure 8.40 Bus Arbitration Timing (Clock Mode 7 or CMNCR.HIZCNT =

(2) Access from the Side of the LSI Internal Bus Master

Since the bus state controller (BSC) incorporates a one-stage write buffer, the BSC can
access via the internal bus before the previous external bus cycle is completed in a write
the on-chip module is read or written after the external low-speed memory is written, th
module can be accessed before the completion of the external low-speed memory write

In read cycles, the CPU is placed in the wait state until read operation has been complet
continue the process after the data write to the device has been completed, perform a du
to the same address to check for completion of the write before the next process to be ex

The write buffer of the BSC functions in the same way for an access by a bus master oth
the CPU such as the DMAC. Accordingly, to perform dual address DMA transfers, the
cycle is initiated before the previous write cycle is completed. Note, however, that if bot
DMA source and destination addresses exist in external memory space, the next write cy
not be initiated until the previous write cycle is completed.

Changing the registers in the BSC while the write buffer is operating may disrupt correc
access. Therefore, do not change the registers in the BSC immediately after a write acce
change becomes necessary, do it after executing a dummy read of the write data.

(3) On-Chip Peripheral Module Access

Access to the on-chip peripheral module registers from the internal bus requires 2 or mo
of the peripheral module clock (Pϕ). When the CPU writes to an on-chip peripheral regi
however, the CPU can execute the following instructions without waiting for the registe
complete.

This section describes the case where the system switches to software standby mode to
power consumption as an example. In this case, the code sets the STBCR register STBY

The table below lists the number of access cycles required for CPU accesses to the on-chi
peripheral module registers.

Table 8.20 On-Chip Peripheral Module Register Access Cycle Counts

	Access Cycles
Write	$(2+n) \times I\phi + (1+m) \times B\phi + 2 \times P\phi$
Read	$(2+n) \times I\phi + (1+m) \times B\phi + 2 \times P\phi + (2+l) \times I\phi$

Note: These are the numbers of cycles when the instruction is executed from internal RO
internal RAM.
When Iϕ:Bϕ is 1:1; n = 0, l = 0
When Iϕ:Bϕ is 2:1; n = 0 or 1, l = 1
When Iϕ:Bϕ is 4:1; n = 0 to 3, l = 2
When Iϕ:Bϕ is 8:1; n = 0 to 7, l = 2
When Bϕ:Pϕ is 1:1; m = 0
When Bϕ:Pϕ is 2:1; m = 0 or 1
When Bϕ:Pϕ is 4:1; m = 0 to 3
Note that n and m depend on the internal execution state.

This product adopts synchronized logic and has a hierarchical bus structure. Data input ar
for each of the busses is synchronized with the rising edge of the Iϕ clock for the C bus, t
clock for the I bus, and the Pϕ clock for the peripheral bus.

Figure 8.41 shows an example of the write timing to the peripheral bus when the relations
between the clocks is Iϕ:Bϕ:Pϕ = 4:4:1. Data is output in synchronization with Iϕ to the C
which the CPU is connected. When Iϕ:Bϕ is 1:1, 2 × Iϕ + Bϕ periods are required for data
transfers from the C bus to the I bus. For transfers from the I bus to the peripheral bus wh
is 4:1, since there are four clock cycles during a single Pϕ clock period, the timing with w
data is placed on the peripheral bus is as follows: there are four timings for Pϕ × 1, and up
periods are required for the Pϕ rising edge, which is the timing for transfers from the I bu

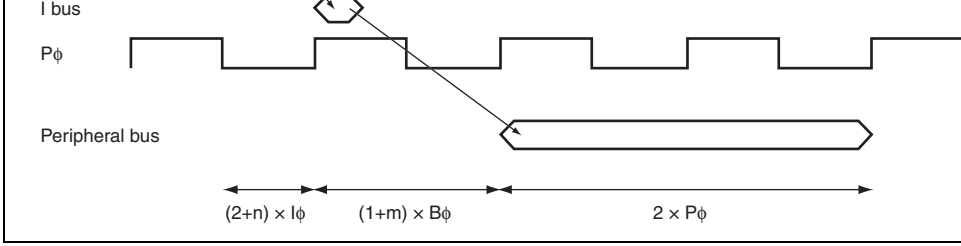


Figure 8.41 Internal Peripheral I/O Register Timing when Iϕ:Bϕ:Pϕ = 4:4

Figure 8.42 shows an example of the write timing to the peripheral bus when the relation between the clocks is Iϕ:Bϕ:Pϕ = 4:2:1. Although transfers from the C bus to the periphe are performed the same way for write, for read, the value read from the peripheral bus m transferred to the CPU. Although the transfers from the peripheral bus to the I bus and fr bus to the C bus are all performed on the corresponding bus clock rising edge, since Iϕ ≥ (2 + 1) × Iϕ periods are actually required. In the example in figure 8.42, since n = 1, m = 1, the access period will be 3 × Iϕ + 2 × Bϕ + 2 × Pϕ + 3 × Iϕ.

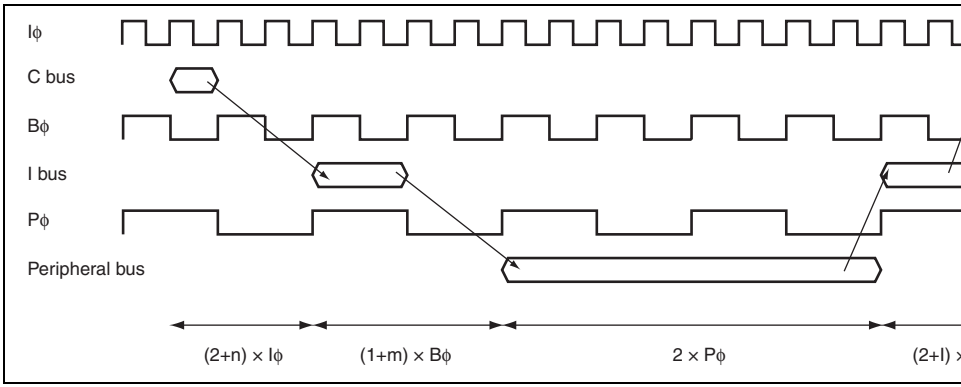
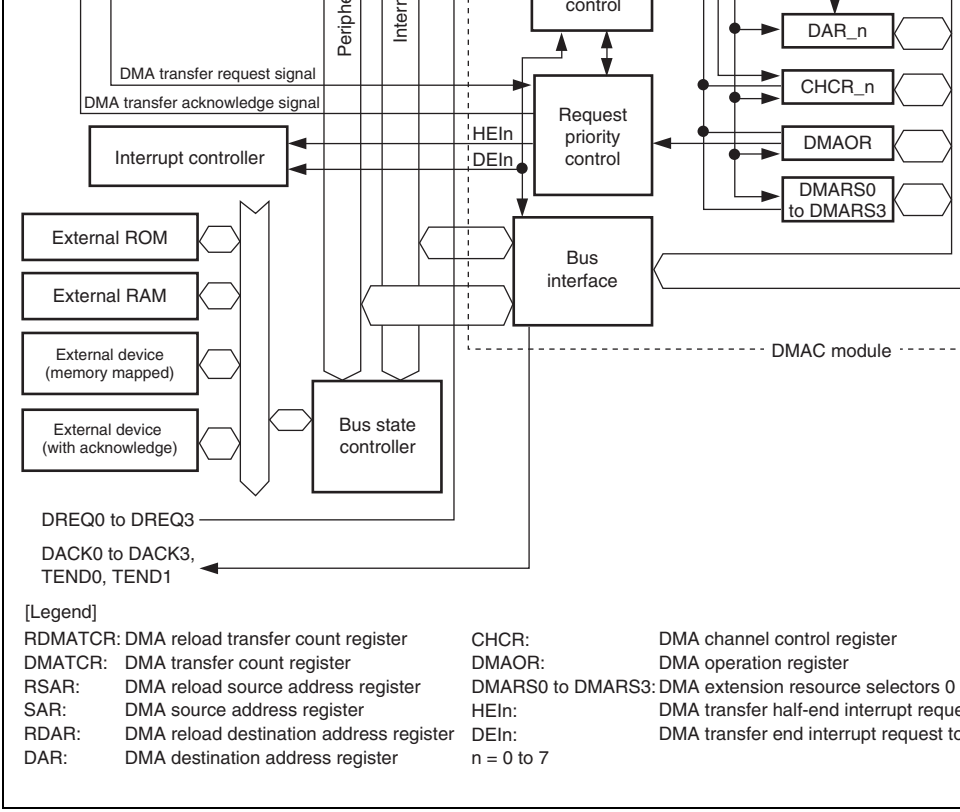


Figure 8.42 Internal Peripheral I/O Register Timing when Iϕ:Bϕ:Pϕ = 4:2

3. Write-back is performed with operand cache or 16-byte write access is performed wit
   DMAC for the burst ROM interface set as above.

- 4-Gbyte physical address space
- Transfer data length is selectable: Byte, word (two bytes), longword (four bytes), an (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode and single address mode are supported.
- Transfer requests
  - External request
  - On-chip peripheral module request
  - Auto request

  The following modules can issue on-chip peripheral module requests.
  - Eight SCIF sources, two IIC3 sources, one A/D converter source, five MTU2 sou two CMT sources
- Selectable bus modes
  - Cycle steal mode (normal mode and intermittent mode)
  - Burst mode
- Selectable channel priority levels: The channel priority levels are selectable between mode and round-robin mode.
- Interrupt request: An interrupt request can be sent to the CPU on completion of half- data transfer. Through the HE and HIE bits in CHCR, an interrupt is specified to be the CPU when half of the initially specified DMA transfer is completed.
- External request detection: There are following four types of DREQ input detection.
  - Low level detection
  - High level detection
  - Rising edge detection
  - Falling edge detection



[Legend]

RDMATCR:	DMA reload transfer count register	CHCR:	DMA channel control register
DMATCR:	DMA transfer count register	DMAOR:	DMA operation register
RSAR:	DMA reload source address register	DMARS0 to DMARS3:	DMA extension resource selectors 0
SAR:	DMA source address register	HEIn:	DMA transfer half-end interrupt reque
RDAR:	DMA reload destination address register	DEIn:	DMA transfer end interrupt request to
DAR:	DMA destination address register	n = 0 to 7	

Figure 9.1 Block Diagram of DMAC

	DMA transfer request acknowledge	DACK0	O	DMA transfer request acknowle output from channel 0 to an exte device
1	DMA transfer request	DREQ1	I	DMA transfer request input from external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer request acknowle output from channel 1 to an exte device
2	DMA transfer request	DREQ2	I	DMA transfer request input from external device to channel 2
	DMA transfer request acknowledge	DACK2	O	DMA transfer request acknowle output from channel 2 to an exte device
3	DMA transfer request	DREQ3	I	DMA transfer request input from external device to channel 3
	DMA transfer request acknowledge	DACK3	O	DMA transfer request acknowle output from channel 3 to an exte device
0	DMA transfer end	TEND0	O	DMA transfer end output for cha
1	DMA transfer end	TEND1	O	DMA transfer end output for cha

0	DMA source address register_0	SAR_0	R/W	H'00000000	H'FFFE1000
	DMA destination address register_0	DAR_0	R/W	H'00000000	H'FFFE1004
	DMA transfer count register_0	DMATCR_0	R/W	H'00000000	H'FFFE1008
	DMA channel control register_0	CHCR_0	R/W*[1]	H'00000000	H'FFFE100C
	DMA reload source address register_0	RSAR_0	R/W	H'00000000	H'FFFE1100
	DMA reload destination address register_0	RDAR_0	R/W	H'00000000	H'FFFE1104
	DMA reload transfer count register_0	RDMATCR_0	R/W	H'00000000	H'FFFE1108
1	DMA source address register_1	SAR_1	R/W	H'00000000	H'FFFE1010
	DMA destination address register_1	DAR_1	R/W	H'00000000	H'FFFE1014
	DMA transfer count register_1	DMATCR_1	R/W	H'00000000	H'FFFE1018
	DMA channel control register_1	CHCR_1	R/W*[1]	H'00000000	H'FFFE101C
	DMA reload source address register_1	RSAR_1	R/W	H'00000000	H'FFFE1110
	DMA reload destination address register_1	RDAR_1	R/W	H'00000000	H'FFFE1114
	DMA reload transfer count register_1	RDMATCR_1	R/W	H'00000000	H'FFFE1118

	address register_2				
	DMA reload destination address register_2	RDAR_2	R/W	H'00000000	H'FFFE1124
	DMA reload transfer count register_2	RDMATCR_2	R/W	H'00000000	H'FFFE1128
3	DMA source address register_3	SAR_3	R/W	H'00000000	H'FFFE1030
	DMA destination address register_3	DAR_3	R/W	H'00000000	H'FFFE1034
	DMA transfer count register_3	DMATCR_3	R/W	H'00000000	H'FFFE1038
	DMA channel control register_3	CHCR_3	R/W*[1]	H'00000000	H'FFFE103C
	DMA reload source address register_3	RSAR_3	R/W	H'00000000	H'FFFE1130
	DMA reload destination address register_3	RDAR_3	R/W	H'00000000	H'FFFE1134
	DMA reload transfer count register_3	RDMATCR_3	R/W	H'00000000	H'FFFE1138

	address register_4				
	DMA reload destination address register_4	RDAR_4	R/W	H'00000000	H'FFFE1144
	DMA reload transfer count register_4	RDMATCR_4	R/W	H'00000000	H'FFFE1148
5	DMA source address register_5	SAR_5	R/W	H'00000000	H'FFFE1050
	DMA destination address register_5	DAR_5	R/W	H'00000000	H'FFFE1054
	DMA transfer count register_5	DMATCR_5	R/W	H'00000000	H'FFFE1058
	DMA channel control register_5	CHCR_5	R/W*[1]	H'00000000	H'FFFE105C
	DMA reload source address register_5	RSAR_5	R/W	H'00000000	H'FFFE1150
	DMA reload destination address register_5	RDAR_5	R/W	H'00000000	H'FFFE1154
	DMA reload transfer count register_5	RDMATCR_5	R/W	H'00000000	H'FFFE1158

	address register_6				
	DMA reload destination address register_6	RDAR_6	R/W	H'00000000	H'FFFE1164
	DMA reload transfer count register_6	RDMATCR_6	R/W	H'00000000	H'FFFE1168
7	DMA source address register_7	SAR_7	R/W	H'00000000	H'FFFE1070
	DMA destination address register_7	DAR_7	R/W	H'00000000	H'FFFE1074
	DMA transfer count register_7	DMATCR_7	R/W	H'00000000	H'FFFE1078
	DMA channel control register_7	CHCR_7	R/W*[1]	H'00000000	H'FFFE107C
	DMA reload source address register_7	RSAR_7	R/W	H'00000000	H'FFFE1170
	DMA reload destination address register_7	RDAR_7	R/W	H'00000000	H'FFFE1174
	DMA reload transfer count register_7	RDMATCR_7	R/W	H'00000000	H'FFFE1178

Notes: 1. For the HE and TE bits in CHCRn, only 0 can be written to clear the flags afte
read.
2. For the AE and NMIF bits in DMAOR, only 0 can be written to clear the flags
read.

9.3.1 DMA Source Address Registers (SAR)

The DMA source address registers (SAR) are 32-bit readable/writable registers that spec
source address of a DMA transfer. During a DMA transfer, these registers indicate the n
address. When the data of an external device with DACK is transferred in single address
SAR is ignored.

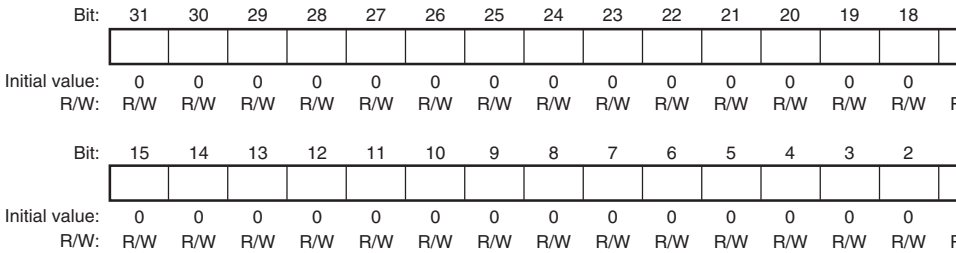
To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit addre
boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte bou

SAR is initialized to H'00000000 by a reset and retains the value in software standby mo
module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
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

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

module standby mode.



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
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

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

and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	-	-	-	-	-	-	-	-						
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/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
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

module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	TC	-	-	RLD	-	-	-	-	DO	TL	-	-	HE	HIE	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R/(W)*	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	DM[1:0]		SM[1:0]		RS[3:0]				DL	DS	TB	TS[1:0]		IE	
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/

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

Bit	Bit Name	Initial Value	R/W	Descriptions
31	TC	0	R/W	Transfer Count Mode Specifies whether to transmit data once or for count specified in DMATCR by one transfer re Note that when this bit is set to 0, the TB bit m be set to 1 (burst mode). When the SCIF or II selected for the transfer request source, this b must not be set to 1. 0: Transmits data once by one transfer reques 1: Transmits data for the count specified in DM by one transfer request
30, 29	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				Selects whether DREQ is detected by overru overrun 1. This bit is valid only in CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 an CHCR_7; it is always read as 0 and the write should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	TL	0	R/W	Transfer End Level Specifies the TEND signal output is high acti active. This bit is valid only in CHCR_0 and C This bit is reserved in CHCR_2 to CHCR_7; always read as 0 and the write value should 0. 0: Low-active output from TEND 1: High-active output from TEND
21, 20	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				DME bit in DMAOR after the HE bit is set to 1 remains set to 1. To clear the HE bit, write 0 to it after HE = 1 is 0: DMATCR > (DMATCR set before transfer s during DMA transfer or after DMA transfer i terminated [Clearing condition] • Writing 0 after reading HE = 1. 1: DMATCR ≤ (DMATCR set before transfer s
18	HIE	0	R/W	Half-End Interrupt Enable Specifies whether to issue an interrupt reques CPU when the transfer count reaches half of t DMATCR value that was specified before tran starts. When the HIE bit is set to 1, the DMAC reque interrupt to the CPU when the HE bit becomes 0: Disables an interrupt to be issued when DM = (DMATCR set before transfer starts)/2 1: Enables an interrupt to be issued when DM = (DMATCR set before transfer starts)/2

				1: DACK output in write cycle (dual address
16	AL	0	R/W	Acknowledge Level Specifies the DACK (acknowledge) signal ou high active or low active. This bit is valid only in CHCR_0 to CHCR_3. reserved in CHCR_4 to CHCR_7; it is always 0 and the write value should always be 0. 0: Low-active output from DACK 1: High-active output from DACK

				transfer, +2 in 16-bit transfer, +4 in 32-bit t +16 in 16-byte transfer) 10: Destination address is decremented (–1 in transfer, –2 in 16-bit transfer, –4 in 32-bit t setting prohibited in 16-byte transfer) 11: Setting prohibited
13, 12	SM[1:0]	00	R/W	Source Address Mode These bits select whether the DMA source ad incremented, decremented, or left fixed. (In si address mode, SM1 and SM0 bits are ignored data is transferred from an external device wit DACK.) 00: Fixed source address (Setting prohibited i byte-unit transfer) 01: Source address is incremented (+1 in byte transfer, +2 in word-unit transfer, +4 in lon unit transfer, +16 in 16-byte-unit transfer) 10: Source address is decremented (–1 in byt transfer, –2 in word-unit transfer, –4 in lon unit transfer, setting prohibited in 16-byte- transfer) 11: Setting prohibited

0011: External request/single address mode
External device with DACK → Externa
space
0100: Auto request
0101: Setting prohibited
0110: Setting prohibited
0111: Setting prohibited
1000: DMA extension resource selector
1001: Setting prohibited
1010: Setting prohibited
1011: Setting prohibited
1100: Setting prohibited
1101: Setting prohibited
1110: Setting prohibited
1111: Setting prohibited

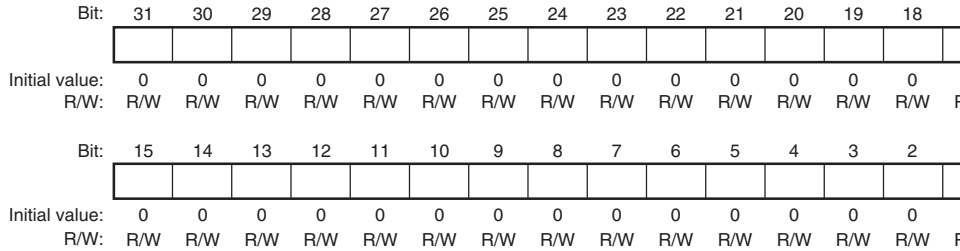
Note: External request specification is valid o
CHCR_0 to CHCR_3. If a request sou
selected in channels CHCR_4 to CHC
operation will be performed.

				specification by these bits is ignored. 00: DREQ detected in low level 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies bus mode when DMA transfers data that burst mode must not be selected when T 0: Cycle steal mode 1: Burst mode
4, 3	TS[1:0]	00	R/W	Transfer Size These bits specify the size of data to be transf Select the size of data to be transferred when source or destination is an on-chip peripheral register of which transfer size is specified. 00: Byte unit 01: Word unit (two bytes) 10: Longword unit (four bytes) 11: 16-byte unit (four longwords)
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request i generated to the CPU at the end of the DMA t Setting this bit to 1 generates an interrupt requ (DEI) to the CPU when TE bit is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request

				To clear the TE bit, write 0 after reading TE Even if the DE bit is set to 1 while this bit is transfer is not enabled. 0: During the DMA transfer or DMA transfer terminated [Clearing condition] • Writing 0 after reading TE = 1 1: DMA transfer ends by the specified count (DMATCR = 0)
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In au mode, DMA transfer starts by setting the DE DME bit in DMAOR to 1. In this case, all of t TE, NMIF in DMAOR, and AE must be 0. In external request or peripheral module reque transfer starts if DMA transfer request is ger the devices or peripheral modules after sett bits DE and DME to 1. In this case, howeve bits TE, NMIF, and AE must be 0 as in the c auto-request mode. Clearing the DE bit to 0 terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

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

RSAR is initialized to H'00000000 by a reset and retains the value in software standby m
module standby mode.



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
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

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

RDAR is initialized to H'00000000 by a reset and retains the value in software standby m
module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
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

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

As in DMATCR, the transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. To transfer 16 bytes, one 16-byte transfer (128 bits) counts one.

RDMATCR is initialized to H'00000000 by a reset and retains the value in software standby and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	-	-	-	-	-	-	-	-						
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/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
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/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R/(W)*	R/(

Note: * To clear flags, read the register and then write 0 only to the bits that were read as 1. Write 1 to the bits that w

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Steal Mode Select These bits select either normal mode or inter mode in cycle steal mode. It is necessary that the bus modes of all char set to cycle steal mode to make intermittent valid. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer for every 16 Bϕ clock. 11: Intermittent mode 64 Executes one DMA transfer for every 64 Bϕ clock.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

				11: Round-robin mode (only supported in CH0
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
2	AE	0	R/(W)*	Address Error Flag Indicates whether an address error has occur the DMAC. When this bit is set, even if the DE CHCR and the DME bit in DMAOR are set to transfer is not enabled. This bit can only be cl writing 0 after reading 1. 0: No DMAC address error 1: DMAC address error occurred [Clearing condition] • Only write 0 to the AE bit after it has been 1. If the bit's value is 0 when read, write 1

				0: No NMI interrupt 1: NMI interrupt occurred [Clearing condition] • Only write 0 to the NMIF bit after it has be as 1. If the bit's value is 0 when read, wri
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfer on all chan the DME bit and DE bit in CHCR are set to 1 transfer is enabled. However, transfer is enabled only when the T CHCR of the transfer corresponding channel bit in DMAOR, and the AE bit are all cleared Clearing the DME bit to 0 can terminate the D transfer on all channels. 0: DMA transfer is disabled on all channels 1: DMA transfer is enabled on all channels

Note: * To clear flags, read the register and then write 0 only to the bits that were rea Write 1 to the bits that were read as 0.

If the priority mode bits are modified after a DMA transfer, the channel priority is initial fixed mode 2 is specified, the channel priority is specified as CH0 > CH4 > CH1 > CH5 CH6 > CH3 > CH7. If fixed mode 1 is specified, the channel priority is specified as CH0 CH2 > CH3 > CH4 > CH5 > CH6 > CH7. If round-robin mode is specified, the transfer channel is reset.

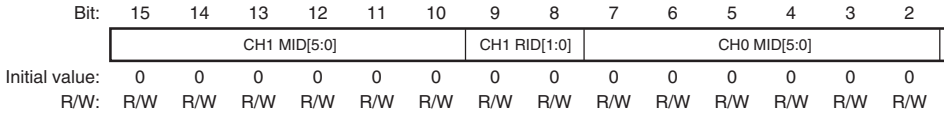
Table 9.3 show the priority change in each mode (modes 0 to 2) specified by the priority bits. In each priority mode, the channel priority to accept the next transfer request may c up to three ways according to the transfer end channel.

Table 9.3 Combinations of Priority Mode Bits

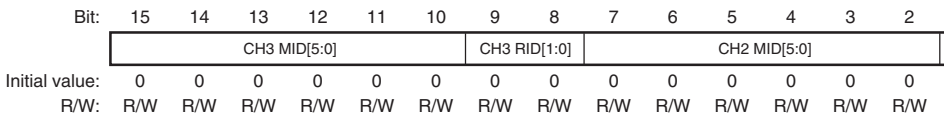
	Transfer End	Priority Mode Bits		Priority Level at the End of Transfer						
				High	◄					
Mode	**CH No.**	**PR[1]**	**PR[0]**	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Mode 0 (fixed mode 1)	Any channel	0	0	CH0	CH1	CH2	CH3	CH4	CH5	CH6
Mode 1 (fixed mode 2)	Any channel	0	1	CH0	CH4	CH1	CH5	CH2	CH6	CH3
Mode 2 (round-robin mode)	CH0	1	1	CH1	CH2	CH3	CH0	CH4	CH5	CH6
	CH1	1	1	CH2	CH3	CH0	CH1	CH4	CH5	CH6
	CH2	1	1	CH3	CH0	CH1	CH2	CH4	CH5	CH6
	CH3	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6
	CH4	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6
	CH5	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6
	CH6	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6
	CH7	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6

module standby mode.

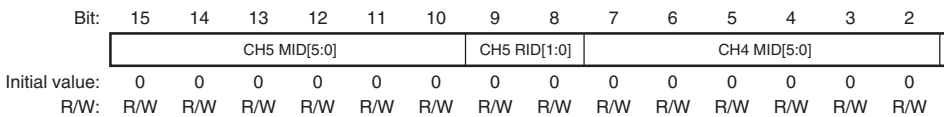
- DMARS0



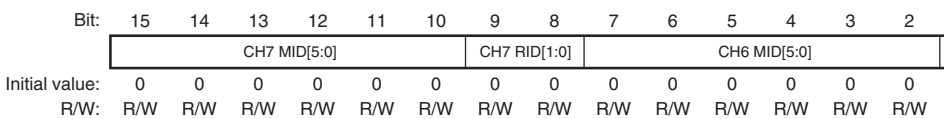
- DMARS1



- DMARS2



- DMARS3



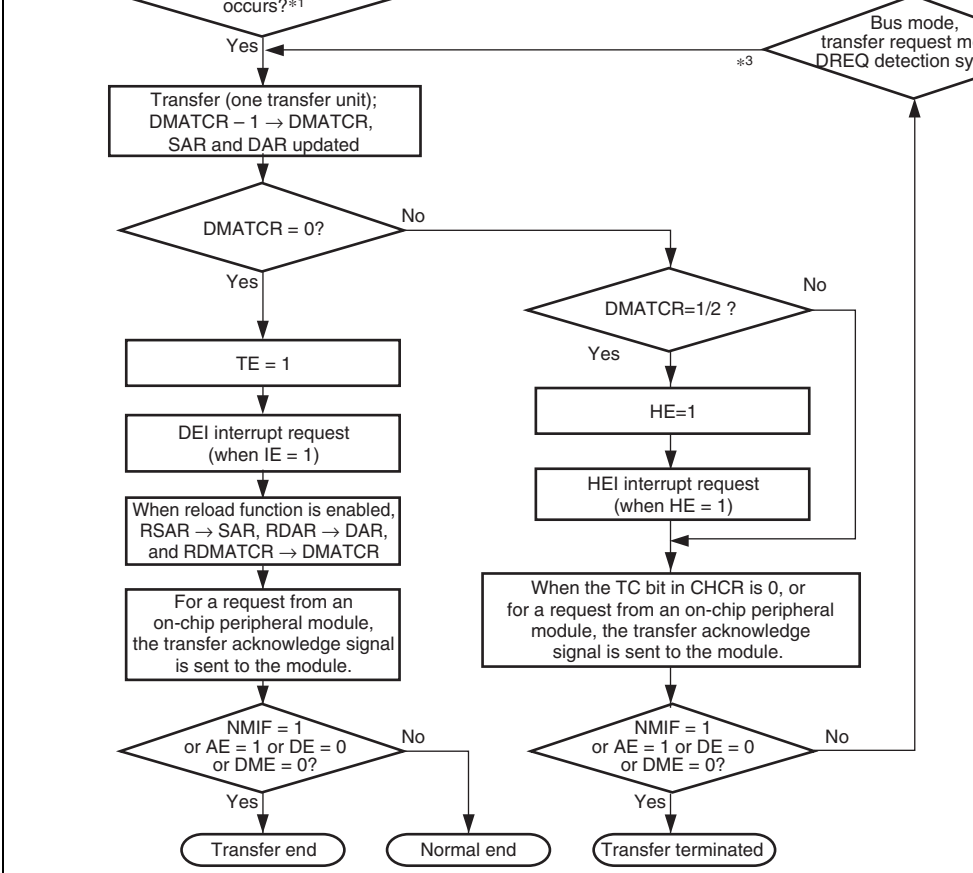
SCIF_2 receiver	H'8A		B'10	Rece
SCIF_3 transmitter	H'8D	B'100011	B'01	Trans
SCIF_3 receiver	H'8E		B'10	Rece
IIC3 transmitter	H'A1	B'101000	B'01	Trans
IIC3 receiver	H'A2		B'10	Rece
A/D converter	H'B3	B'101100	B'11	—
MTU2_0	H'E3	B'111000	B'11	—
MTU2_1	H'E7	B'111001	B'11	—
MTU2_2	H'EB	B'111010	B'11	—
MTU2_3	H'EF	B'111011	B'11	—
MTU2_4	H'F3	B'111100	B'11	—
CMT_0	H'FB	B'111110	B'11	—
CMT_1	H'FF	B'111111	B'11	—

When MID or RID other than the values listed in table 9.4 is set, the operation of this LSI guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS[3:0]) in CHCR0 to CHCR7 have been set to B'1000. Otherwise, even if DMARS has the transfer request source is not accepted.

transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA ope
register (DMAOR), and DMA extension resource selector (DMARS) are set for the targ
conditions, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request comes and transfer is enabled, the DMAC transfers one tran
   data (depending on the TS0 and TS1 settings). For an auto request, the transfer begi
   automatically when the DE bit and DME bit are set to 1. The DMATCR value will b
   decremented by 1 for each transfer. The actual transfer flows vary by address mode a
   mode.
3. When half of the specified transfer count is exceeded (when DMATCR reaches half
   initial value), an HEI interrupt is sent to the CPU if the HIE bit in CHCR is set to 1.
4. When transfer has been completed for the specified count (when DMATCR reaches
   transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt
   the CPU.
5. When an address error in the DMAC or an NMI interrupt is generated, the transfer is
   terminated. Transfers are also terminated when the DE bit in CHCR or the DME bit
   DMAOR is cleared to 0.

Figure 9.2 is a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when the NMIF, AE, and TE bits are cleared to 0 and the DE and DME bits are set to 1.
2. DREQ level detection in burst mode (external request) or cycle steal mode.
3. DREQ edge detection in burst mode (external request), or auto request mode in burst mode.

Figure 9.2 DMA Transfer Flowchart

When there is no transfer request signal from an external source, as in a memory-to-mem transfer or a transfer between memory and an on-chip peripheral module unable to reque transfer, auto-request mode allows the DMAC to automatically generate a transfer reque internally. When the DE bits in CHCR_0 to CHCR_7 and the DME bit in DMAOR are the transfer begins so long as the TE bits in CHCR_0 to CHCR_7, and the AE and NMI DMAOR are 0.

(2) External Request Mode

In this mode a transfer is performed at the request signals (DREQ0 to DREQ3) of an ext device. Choose one of the modes shown in table 9.5 according to the application system DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), DMA transfer performed upon a request at the DREQ input.

Table 9.5 Selecting External Request Modes with the RS Bits

RS[3]	RS[2]	RS[1]	RS[0]	Address Mode	Transfer Source	Transfer Destinatio
0	0	0	0	Dual address mode	Any	Any
0	0	1	0	Single address mode	External memory, memory-mapped external device	External de DACK
			1		External device with DACK	External me memory-ma external de

Choose to detect DREQ by either the edge or level of the signal input with the DL and D CHCR_0 to CHCR_3 as shown in table 9.6. The source of the transfer request does not the data transfer source or destination.

period). After issuing acknowledge DACK signal for the accepted DREQ, the DREQ pin enters the request accept enabled state.

When DREQ is used by level detection, there are following two cases by the timing to de next DREQ after outputting DACK.

Overrun 0: Transfer is terminated after the same number of transfer has been performed a requests.

Overrun 1: Transfer is terminated after transfers have been performed for (the number of plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 9.7 Selecting External Request Detection with DO Bit

CHCR	
DO bit	**External Request**
0	Overrun 0
1	Overrun 1

transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, and NMIF = 0), DMA transfer is performed.

When the transmit data empty from the SCIF is selected, specify the transfer destination as the corresponding SCIF transmit data register. Likewise, when the receive data full from the SCIF is selected, specify the transfer source as the corresponding SCIF receive data register. When a transfer request is made by the A/D converter, the transfer source must be the A/D data register (ADDR). When the IIC3 transmit is selected as the transfer request, the transfer destination must be ICDRT; when the IIC3 reception is selected as the transfer request, the transfer source must be ICDRR. Any address can be specified for data transfer source and destination when a transfer request is sent from the CMT or MTU2.

	10	SCIF_2 receive	RXI2 (receive FIFO data full)	SCFRDR_2	Any
100011	01	SCIF_3 transmit	TXI3 (transmit FIFO data empty)	Any	SCFTDR_3
	10	SCIF_3 receive	RXI3 (receive FIFO data full)	SCFRDR_3	Any
101000	01	IIC3 transmit	TXI (transmit data empty)	Any	ICDRT
	10	IIC3 receive	RXI (receive data full)	ICDRR	Any
101100	11	A/D converter	ADI (A/D conversion end)	ADDR	Any
111000	11	MTU2_0	TGI0A	Any	Any
111001	11	MTU2_1	TGI1A	Any	Any
111010	11	MTU2_2	TGI2A	Any	Any
111011	11	MTU2_3	TGI3A	Any	Any
111100	11	MTU2_4	TGI4A	Any	Any
111110	11	CMT_0	Compare match 0	Any	Any
111111	11	CMT_1	Compare match 1	Any	Any

Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7

Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7

These are selected by the PR1 and PR0 bits in the DMA operation register (DMAOR).

(2) Round-Robin Mode

Each time one unit of word, byte, longword, or 16 bytes is transferred on one channel, th order is rotated. The channel on which the transfer was just finished is rotated to the low priority order among the four round-robin channels (channels 0 to 4). The priority of the other than the round-robin channels (channels 0 to 4) does not change even in round-rob The round-robin mode operation is shown in figure 9.3. The priority in round-robin mod > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 immediately after a reset.

When round-robin mode has been specified, do not concurrently specify cycle steal mod burst mode as the bus modes of any two or more channels.

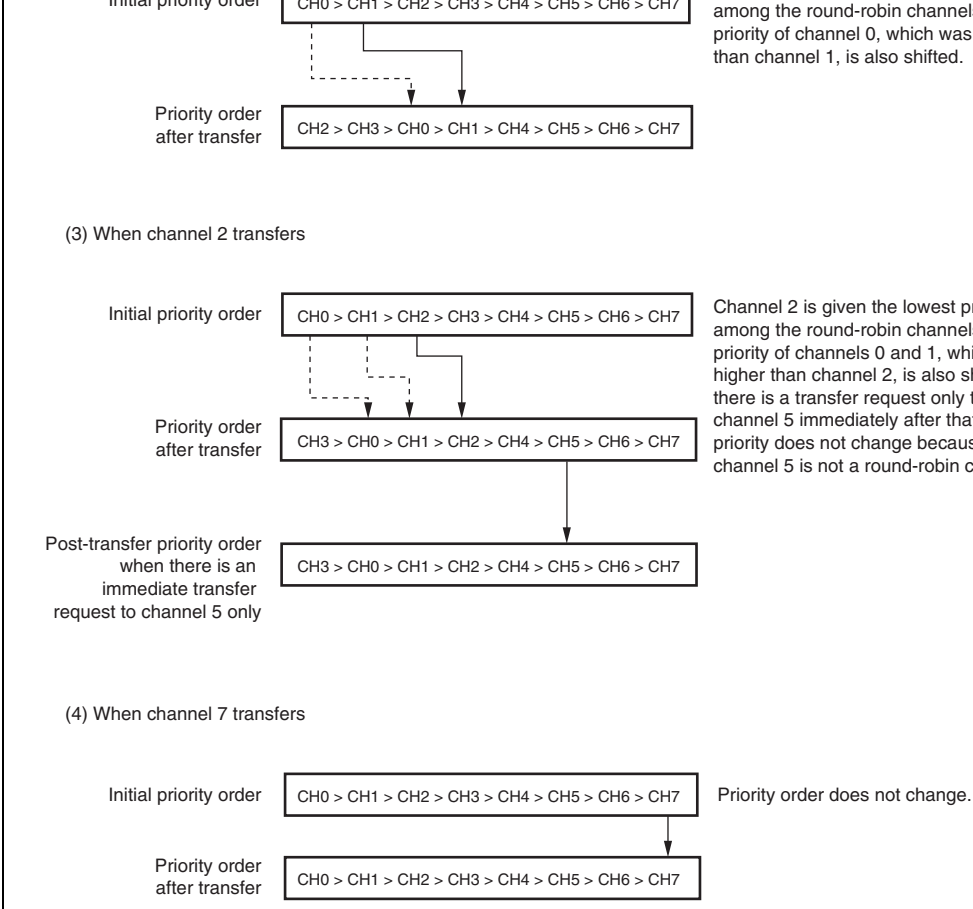


Figure 9.3 Round-Robin Mode

5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfe (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 is given the lowest priority among the r channels.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 are lowered in priority so that ch given the lowest priority among the round-robin channels.

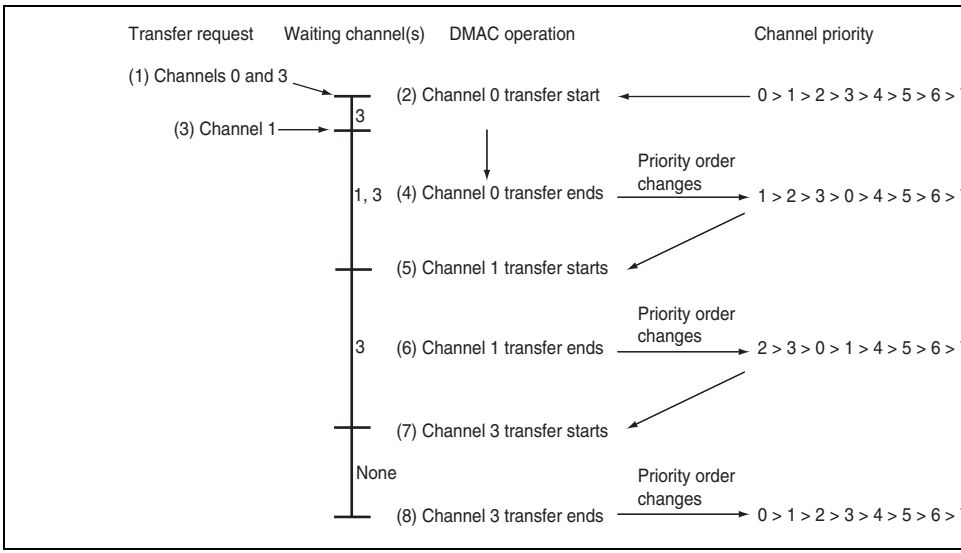


Figure 9.4 Changes in Channel Priority in Round-Robin Mode

Transfer Source	with DACK	Memory	External Device	Peripheral Module	Me
External device with DACK	Not available	Dual, single	Dual, single	Not available	Not
External memory	Dual, single	Dual	Dual	Dual	Dua
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dua
On-chip peripheral module	Not available	Dual	Dual	Dual	Dua
On-chip memory	Not available	Dual	Dual	Dual	Dua

Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. 16-byte transfer is available only for on-chip peripheral modules that support lo access.

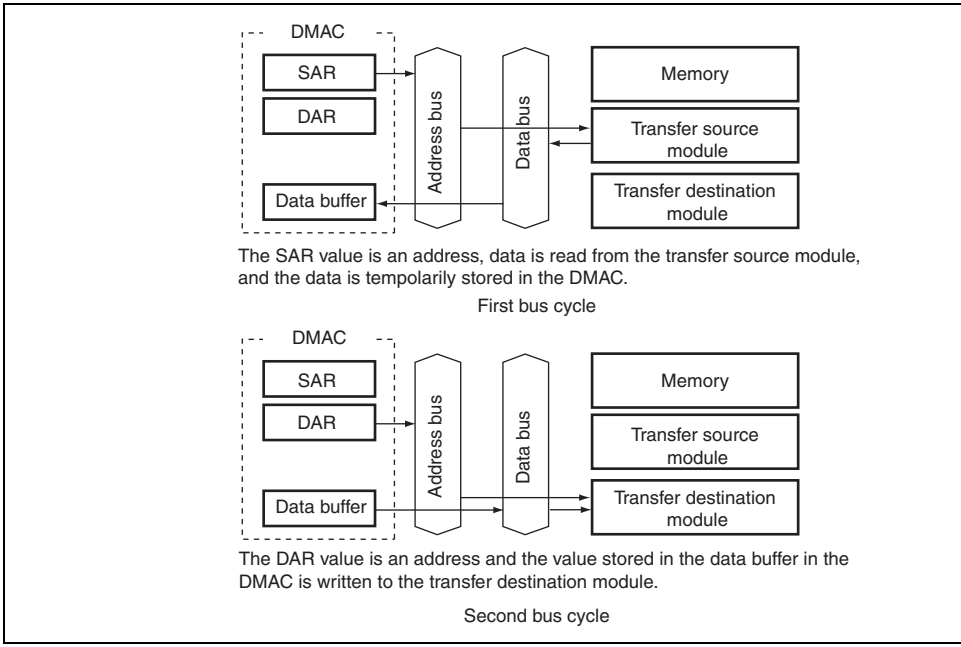
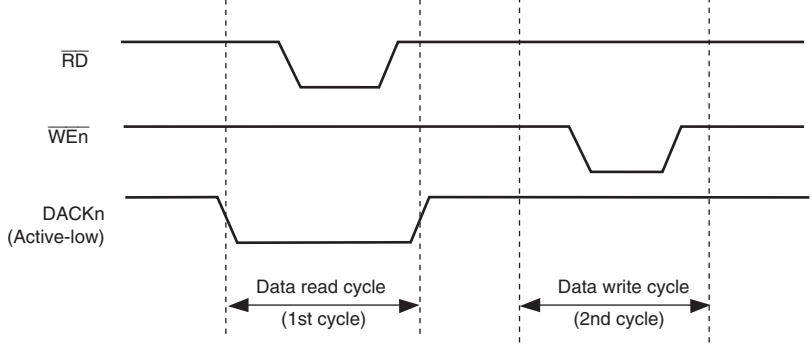


Figure 9.5 Data Flow of Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available for t request. DACK can be output in read cycle or write cycle in dual address mode. The AM channel control register (CHCR) can specify whether the DACK is output in read cycle cycle.

Figure 9.6 shows an example of DMA transfer timing in dual address mode.



Note: In transfer between external memories, with DACK output in the read cycle, DACK output timing is the same as that of CSn.

Figure 9.6 Example of DMA Transfer Timing in Dual Mode (Transfer Source: Normal Memory, Transfer Destination: Normal Memory

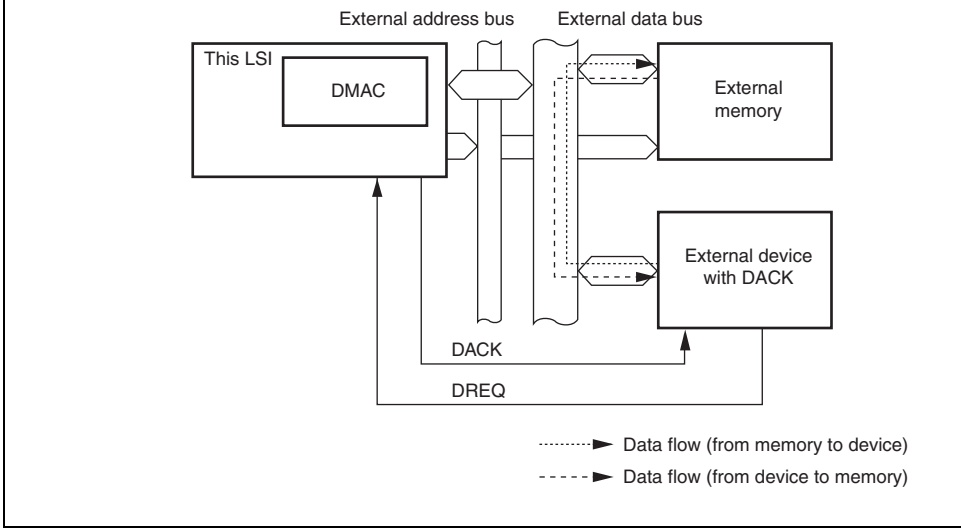
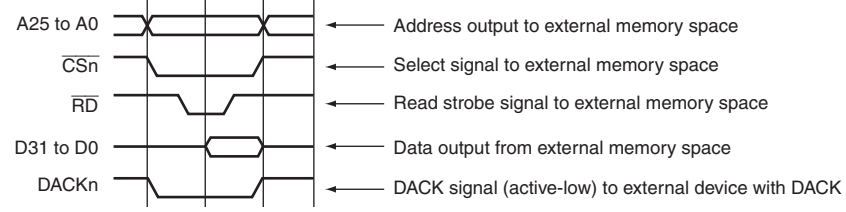


Figure 9.7 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an extern with DACK and a memory-mapped external device, and (2) transfer between an externa with DACK and external memory. In both cases, only the external request signal (DREQ for transfer requests.

Figure 9.8 shows an example of DMA transfer timing in single address mode.



(b) External memory space (normal memory) → External device with DACK

Figure 9.8 Example of DMA Transfer Timing in Single Address Mode

(2) Bus Modes

There are two bus modes; cycle steal and burst. Select the mode by the TB bits in the cha control registers (CHCR).

(a) Cycle Steal Mode

- Normal mode

  In normal mode of cycle steal, the bus mastership is given to another bus master after transfer-unit (byte, word, longword, or 16-byte unit) DMA transfer. When another tra request occurs, the bus mastership is obtained from another bus master and a transfer performed for one transfer unit. When that transfer ends, the bus mastership is passed another bus master. This is repeated until the transfer end conditions are satisfied.

  The cycle-steal normal mode can be used for any transfer section; transfer request sou transfer source, and transfer destination.

  Figure 9.9 shows an example of DMA transfer timing in cycle-steal normal mode. Tra conditions shown in the figure are;

  — Dual address mode

  — DREQ low level detection

whenever a unit of transfer (byte, word, longword, or 16 bytes) is completed. If the t
transfer request occurs after that, DMAC obtains the bus mastership from other bus m
after waiting for 16 or 64 cycles of Bϕ clock. DMAC then transfers data of one unit a
the bus mastership to other bus master. These operations are repeated until the transf
condition is satisfied. It is thus possible to make lower the ratio of bus occupation by
transfer than normal mode of cycle steal.

The cycle-steal intermittent mode can be used for any transfer section; transfer reque
transfer source, and transfer destination. The bus modes, however, must be cycle stea
all channels.

Figure 9.10 shows an example of DMA transfer timing in cycle-steal intermittent mo
Transfer conditions shown in the figure are;

— Dual address mode

— DREQ low level detection

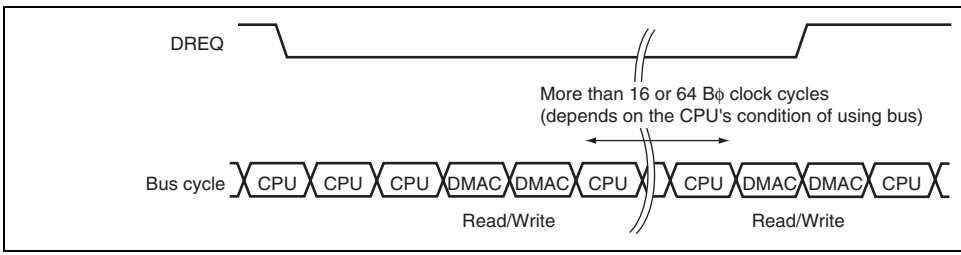


Figure 9.10 Example of DMA Transfer in Cycle-Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

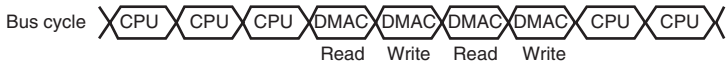


Figure 9.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Categ

Table 9.10 shows the relationship between request modes and bus modes by DMA transf category.

	External memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2
	Memory-mapped external device and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2
	On-chip peripheral module and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2
	On-chip memory and on-chip memory	All*4	B/C	8/16/32/128
	On-chip memory and memory-mapped external device	All*4	B/C	8/16/32/128
	On-chip memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2
	On-chip memory and external memory	All*4	B/C	8/16/32/128
Single	External device with DACK and external memory	External	B/C	8/16/32/128
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128

[Legend]

B: Burst

C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are available. However, along with the exception of CMT and MTU2 as the transf source, the requesting module must be designated as the transfer source or t transfer destination.

2. Access size permitted for the on-chip peripheral module register functioning a transfer source or transfer destination.
3. If the transfer request is an external request, channels 0 to 3 are only availab
4. External requests, auto requests, and on-chip peripheral module requests are available. In the case of on-chip peripheral module requests, however, the CM MTU2 are only available.
5. Only cycle steal except for the MTU2 and CMT as the transfer request source

replaced with a burst-mode transfer cycle (priority execution of burst-mode cycle). An ex
this is shown in figure 9.12.

When multiple channels are in burst mode, data transfer on the channel that has the highe
priority is given precedence. When DMA transfer is being performed on multiple channe
bus mastership is not released to another bus-master device until all of the competing bur
transfers have been completed.

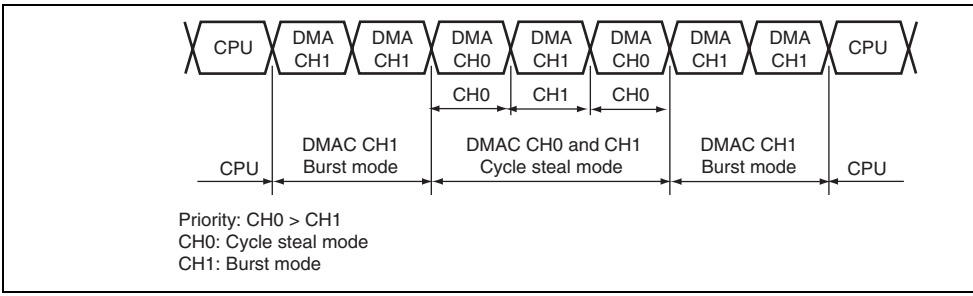


Figure 9.12 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes as shown in figure 9.3. Note that channels in cy
and burst modes must not be mixed.

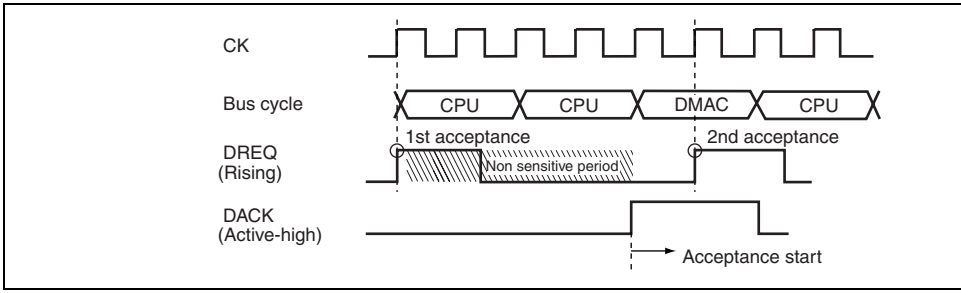


Figure 9.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Det

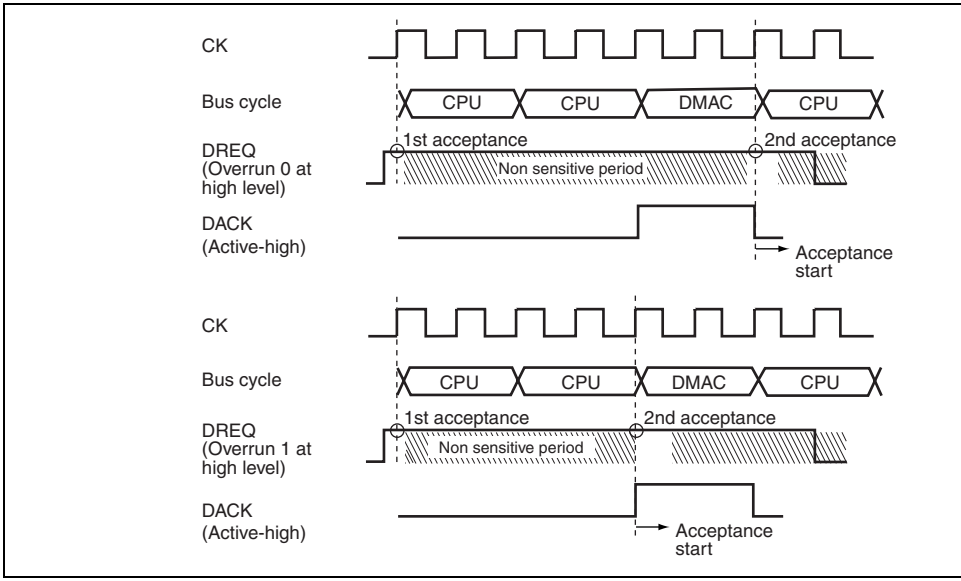


Figure 9.14 Example of DREQ Input Detection in Cycle Steal Mode Level Det

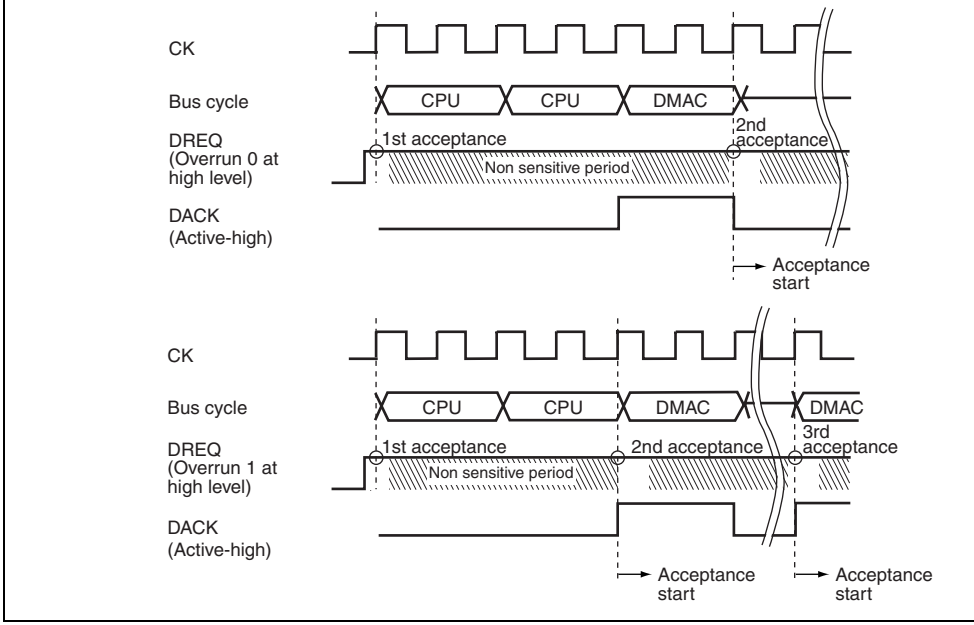
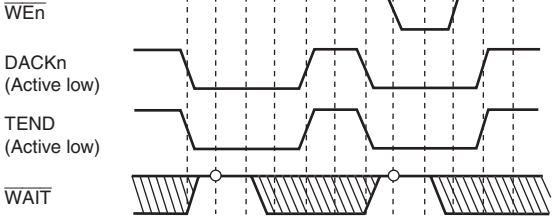


Figure 9.16 Example of DREQ Input Detection in Burst Mode Level Detecti

(Cycle Steal Mode Level Detection)

The unit of the DMA transfer is divided into multiple bus cycles when 16-byte transfer i
performed for an 8-bit or 16-bit external device, when longword access is performed for
or 16-bit external device, or when word access is performed for an 8-bit external device.
setting is made so that the DMA transfer size is divided into multiple bus cycles and the
is negated between bus cycles, note that DACK and TEND are divided like the $\overline{CS}$ signa
alignment. Also, if the DREQ detection is set to level-detection mode (DS bit in CHCR
DREQ sampling may not be detected correctly with divided DACK, and one extra overr
occur at maximum.

Use a setting that does not divide DACK or specify a transfer size smaller than the exter
bus width if DACK is divided. Figure 9.18 shows this example.



Note: TEND is asserted for the last unit of DMA transfer. If a transfer unit is divided into multiple bus cycles and the $\overline{CS}$ is negated between the bus cycles, TEND is also divided.

Figure 9.18 BSC Normal Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

If the first setting of DMAPCR differs from the RDMAPCR setting used in the second
following DMA transfer, the half-end flag setting timing may be earlier than half of the
count or the half-end flag may not be set. The same is true for the half-end interrupt.

— Waveform output at compare match
— Input capture function
— Counter clear operation
— Multiple timer counters (TCNT) can be written to simultaneously
— Simultaneous clearing by compare match and input capture is possible
— Register simultaneous input/output is possible by synchronous counter operation
— A maximum 12-phase PWM output is possible in combination with synchronous
However, waveform output by compare match for channel 5 is not possible.

- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 28 interrupt sources
- Automatic transfer of register data
- A/D converter start trigger can be generated
- Module standby mode can be settable
- A total of six-phase waveform output, which includes complementary PWM output, positive and negative phases of reset PWM output by interlocking operation of chan 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PW and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, a selection of two types of waveform outputs (chopping and level) is possible.
- Dead time compensation counter available in channel 5
- In complementary PWM mode, interrupts at the crest and trough of the counter valu converter start triggers can be skipped.

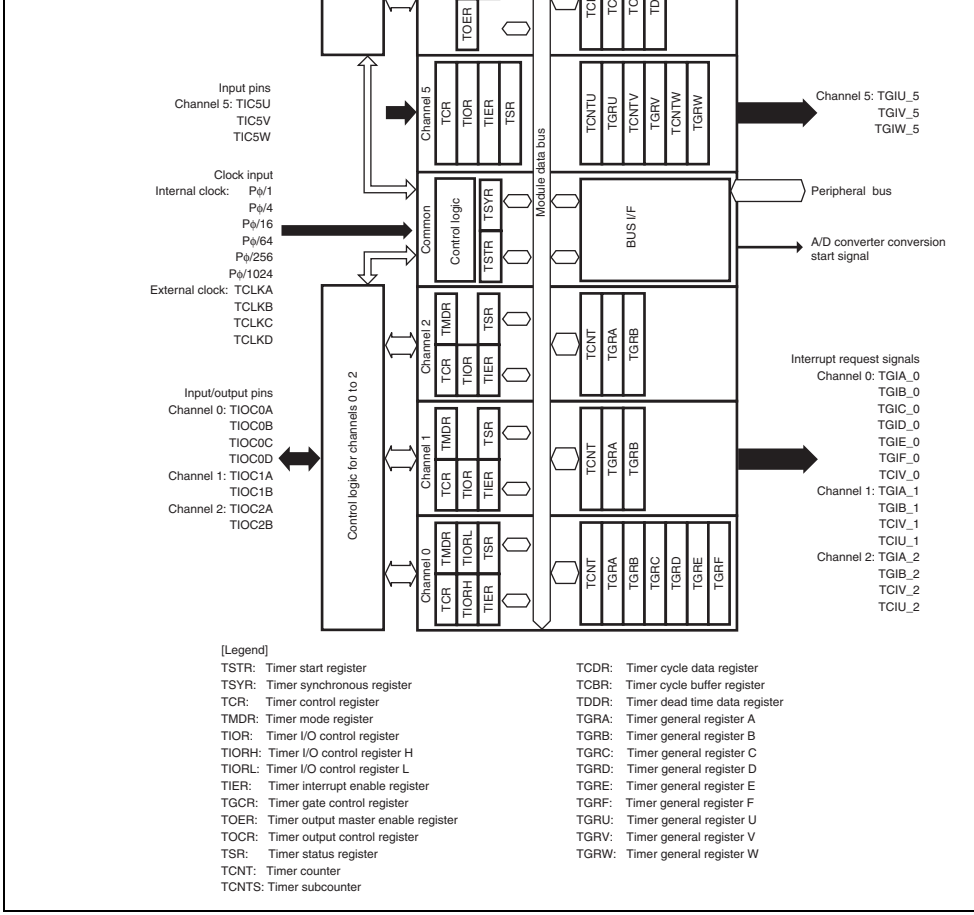
		TGRB_0 TGRE_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TG TG
General registers/ buffer registers		TGRC_0 TGRD_0 TGRF_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4	—
I/O pins		TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D	Inp TIC TIC TIC
Counter clear function		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 con ma inp
Compare match output	0 output	√	√	√	√	√	—
	1 output	√	√	√	√	√	—
	Toggle output	√	√	√	√	√	—
Input capture function		√	√	√	√	√	√
Synchronous operation		√	√	√	√	√	—
PWM mode 1		√	√	√	√	√	—
PWM mode 2		√	√	√	—	—	—
Complementary PWM mode		—	—	—	√	√	—
Reset PWM mode		—	—	—	√	√	—
AC synchronous motor drive mode		√	—	—	√	√	—

A/D converter start trigger	TGRA_0 compare match or input capture TGRE_0 compare match	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture TCNT_4 underflow (trough) in complementary PWM mode	—

capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	capture 1B • Overflow • Underflow	capture 2B • Overflow • Underflow	capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow

					• A/D converter start request at a match between TADCORB_4 and TCNT_4	
Interrupt skipping function	—	—	—	• Skips TGRA_3 compare match interrupts	• Skips TCIV_4 interrupts	—

[Legend]
√: Possible
—: Not possible



[Legend]

TSTR: Timer start register
TSYR: Timer synchronous register
TCR: Timer control register
TMDR: Timer mode register
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIER: Timer interrupt enable register
TGCR: Timer gate control register
TOER: Timer output master enable register
TOCR: Timer output control register
TSR: Timer status register
TCNT: Timer counter
TCNTS: Timer subcounter
TCDR: Timer cycle data register
TCBR: Timer cycle buffer register
TDDR: Timer dead time data register
TGRA: Timer general register A
TGRB: Timer general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TGRU: Timer general register U
TGRV: Timer general register V
TGRW: Timer general register W

Figure 10.1 Block Diagram of MTU2

	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOC0A	I/O	TGRA_0 input capture input/output compare output/PWM
	TIOC0B	I/O	TGRB_0 input capture input/output compare output/PWM
	TIOC0C	I/O	TGRC_0 input capture input/output compare output/PWM
	TIOC0D	I/O	TGRD_0 input capture input/output compare output/PWM
1	TIOC1A	I/O	TGRA_1 input capture input/output compare output/PWM
	TIOC1B	I/O	TGRB_1 input capture input/output compare output/PWM
2	TIOC2A	I/O	TGRA_2 input capture input/output compare output/PWM
	TIOC2B	I/O	TGRB_2 input capture input/output compare output/PWM
3	TIOC3A	I/O	TGRA_3 input capture input/output compare output/PWM
	TIOC3B	I/O	TGRB_3 input capture input/output compare output/PWM
	TIOC3C	I/O	TGRC_3 input capture input/output compare output/PWM
	TIOC3D	I/O	TGRD_3 input capture input/output compare output/PWM
4	TIOC4A	I/O	TGRA_4 input capture input/output compare output/PWM
	TIOC4B	I/O	TGRB_4 input capture input/output compare output/PWM
	TIOC4C	I/O	TGRC_4 input capture input/output compare output/PWM
	TIOC4D	I/O	TGRD_4 input capture input/output compare output/PWM
5	TIC5U	Input	TGRU_5 input capture input/external pulse input pin
	TIC5V	Input	TGRV_5 input capture input/external pulse input pin
	TIC5W	Input	TGRW_5 input capture input/external pulse input pin

Note: For the pin configuration in complementary PWM mode, see table 10.54 in sectio
Complementary PWM Mode.

0	Timer control register_0	TCR_0	R/W	H'00	H'FFFE4300
	Timer mode register_0	TMDR_0	R/W	H'00	H'FFFE4301
	Timer I/O control register H_0	TIORH_0	R/W	H'00	H'FFFE4302
	Timer I/O control register L_0	TIORL_0	R/W	H'00	H'FFFE4303
	Timer interrupt enable register_0	TIER_0	R/W	H'00	H'FFFE4304
	Timer status register_0	TSR_0	R/W	H'C0	H'FFFE4305
	Timer counter_0	TCNT_0	R/W	H'0000	H'FFFE4306
	Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FFFE4308
	Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FFFE430A
	Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FFFE430C
	Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FFFE430E
	Timer general register E_0	TGRE_0	R/W	H'FFFF	H'FFFE4320
	Timer general register F_0	TGRF_0	R/W	H'FFFF	H'FFFE4322
	Timer interrupt enable register2_0	TIER2_0	R/W	H'00	H'FFFE4324
	Timer status register2_0	TSR2_0	R/W	H'C0	H'FFFE4325
	Timer buffer operation transfer mode register_0	TBTM_0	R/W	H'00	H'FFFE4326
1	Timer control register_1	TCR_1	R/W	H'00	H'FFFE4380
	Timer mode register_1	TMDR_1	R/W	H'00	H'FFFE4381
	Timer I/O control register_1	TIOR_1	R/W	H'00	H'FFFE4382
	Timer interrupt enable register_1	TIER_1	R/W	H'00	H'FFFE4384
	Timer status register_1	TSR_1	R/W	H'C0	H'FFFE4385

	Timer interrupt enable register_2	TIER_2	R/W	H'00	H'FFFE4004
	Timer status register_2	TSR_2	R/W	H'C0	H'FFFE4005
	Timer counter_2	TCNT_2	R/W	H'0000	H'FFFE4006
	Timer general register A_2	TGRA_2	R/W	H'FFFF	H'FFFE4008
	Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FFFE400A
3	Timer control register_3	TCR_3	R/W	H'00	H'FFFE4200
	Timer mode register_3	TMDR_3	R/W	H'00	H'FFFE4202
	Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FFFE4204
	Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FFFE4205
	Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FFFE4208
	Timer status register_3	TSR_3	R/W	H'C0	H'FFFE422C
	Timer counter_3	TCNT_3	R/W	H'0000	H'FFFE4210
	Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FFFE4218
	Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FFFE421A
	Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FFFE4224
	Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FFFE4226
	Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FFFE4238
4	Timer control register_4	TCR_4	R/W	H'00	H'FFFE4201
	Timer mode register_4	TMDR_4	R/W	H'00	H'FFFE4203
	Timer I/O control register H_4	TIORH_4	R/W	H'00	H'FFFE4206
	Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FFFE4207

	Timer buffer operation transfer mode register_4	TBTM_4	R/W	H'00	H'FFFE4239
	Timer A/D converter start request control register	TADCR	R/W	H'0000	H'FFFE4240
	Timer A/D converter start request cycle set register A_4	TADCORA_4	R/W	H'FFFF	H'FFFE4244
	Timer A/D converter start request cycle set register B_4	TADCORB_4	R/W	H'FFFF	H'FFFE4246
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFE4248
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE424A
5	Timer control register U_5	TCRU_5	R/W	H'00	H'FFFE4084
	Timer control register V_5	TCRV_5	R/W	H'00	H'FFFE4094
	Timer control register W_5	TCRW_5	R/W	H'00	H'FFFE40A4
	Timer I/O control register U_5	TIORU_5	R/W	H'00	H'FFFE4086
	Timer I/O control register V_5	TIORV_5	R/W	H'00	H'FFFE4096
	Timer I/O control register W_5	TIORW_5	R/W	H'00	H'FFFE40A6
	Timer interrupt enable register_5	TIER_5	R/W	H'00	H'FFFE40B2
	Timer status register_5	TSR_5	R/W	H'00	H'FFFE40B0
	Timer start register_5	TSTR_5	R/W	H'00	H'FFFE40B4
	Timer counter U_5	TCNTU_5	R/W	H'0000	H'FFFE4080
	Timer counter V_5	TCNTV_5	R/W	H'0000	H'FFFE4090

	Timer counter synchronous start register	TCSYSTR	R/W	H'00	H'FFFE4282
	Timer read/write enable register	TRWER	R/W	H'01	H'FFFE4284
Common to 3 and 4	Timer output master enable register	TOER	R/W	H'C0	H'FFFE420A
	Timer output control register 1	TOCR1	R/W	H'00	H'FFFE420E
	Timer output control register 2	TOCR2	R/W	H'00	H'FFFE420F
	Timer gate control register	TGCR	R/W	H80	H'FFFE420D
	Timer cycle control register	TCDR	R/W	H'FFFF	H'FFFE4214
	Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFE4216
	Timer subcounter	TCNTS	R	H'0000	H'FFFE4220
	Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FFFE4222
	Timer interrupt skipping set register	TITCR	R/W	H'00	H'FFFE4230
	Timer interrupt skipping counter	TITCNT	R	H'00	H'FFFE4231
	Timer buffer transfer set register	TBTER	R/W	H'00	H'FFFE4232
	Timer dead time enable register	TDER	R/W	H'01	H'FFFE4234
	Timer synchronous clear register	TSYCR	R/W	H'00	H'FFFE4250
	Timer waveform control register	TWCR	R/W	H'00	H'FFFE4260
	Timer output level buffer register	TOLBR	R/W	H'00	H'FFFE4236

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	CCLR[2:0]	000	R/W	Counter Clear 0 to 2 These bits select the TCNT counter clearing so See tables 10.4 and 10.5 for details.
4, 3	CKEG[1:0]	00	R/W	Clock Edge 0 and 1 These bits select the input clock edge. When th clock is counted using both edges, the input clo period is halved (e.g. Pϕ/4 both edges = Pϕ/2 r edge). If phase counting mode is used on chan and 2, this setting is ignored and the phase cou mode setting has priority. Internal clock edge s is valid when the input clock is Pϕ/4 or slower. Pϕ/1 or the overflow/underflow of another chan selected for the input clock, although values ca written, counter operation compiles with the ini 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges
2 to 0	TPSC[2:0]	000	R/W	Time Prescaler 0 to 2 These bits select the TCNT counter clock. The source can be selected independently for each See tables 10.6 to 10.10 for details.

[Legend]

x: Don't care

				synchronous operation*[1]
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare mat capture*[2]
		1	0	TCNT cleared by TGRD compare mat capture*[2]
			1	TCNT cleared by counter clearing for a channel performing synchronous clear synchronous operation*[1]

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

Table 10.5 CCLR0 to CCLR2 (Channels 1 and 2)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare ma capture
		1	0	TCNT cleared by TGRB compare ma capture
			1	TCNT cleared by counter clearing for channel performing synchronous clea 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 and 2. It is always read as 0 and cannot be mo

		1	0	External clock: counts on TCLKC pin in
			1	External clock: counts on TCLKD pin in

Table 10.7 TPSC0 to TPSC2 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin in
			1	External clock: counts on TCLKB pin in
		1	0	Internal clock: counts on Pϕ/256
			1	Counts on TCNT_2 overflow/underflow

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

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

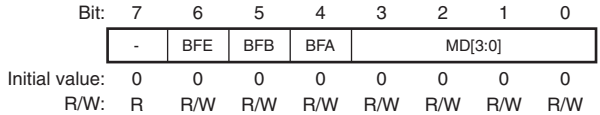
Note: This setting is ignored when channel 2 is in phase counting mode.

Table 10.9 TPSC0 to TPSC2 (Channels 3 and 4)

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

10.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating each channel. The MTU2 has five TMDR registers, one each for channels 0 to 4. TMDR settings should be changed only when TCNT operation is stopped.



Bit:	7	6	5	4	3	2	1	0
	-	BFE	BFB	BFA	MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	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	Reserved This bit is always read as 0. The write value sh always be 0.
6	BFE	0	R/W	Buffer Operation E Specifies whether TGRE_0 and TGRF_0 are t in the normal way or to be used together for bu operation. TGRF compare match is generated when TGR used as the buffer register. In channels 1 to 4, this bit is reserved. It is alwa as 0 and the write value should always be 0. 0: TGRE_0 and TGRF_0 operate normally 1: TGRE_0 and TGRF_0 used together for buf operation

				enable register 3/4 (TIER_3/4) to 0. In channels 1 and 2, which have no TGRD, bi reserved. It is always read as 0 and cannot be 0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the r way, or TGRA and TGRC are to be used toge buffer operation. When TGRC is used as a bu register, TGRC input capture/output compare generated in a mode other than complementa TGRC compare match is generated when in complementary PWM mode. When compare channel 4 occurs during the Tb period in complementary PWM mode, TGFC is set. Th set the TGIEC bit in the timer interrupt enable (TIER_4) to 0. In channels 1 and 2, which have no TGRC, bi reserved. It is always read as 0 and cannot be 0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer
3 to 0	MD[3:0]	0000	R/W	Modes 0 to 3 These bits are used to set the timer operating See table 10.11 for details.

		1	0	Phase counting mode 3*2
			1	Phase counting mode 4*2
1	0	0	0	Reset synchronous PWM mode*3
			1	Setting prohibited
		1	X	Setting prohibited
	1	0	0	Setting prohibited
			1	Complementary PWM mode 1 (transmit at crest)*3
		1	0	Complementary PWM mode 2 (transmit at trough)*3
			1	Complementary PWM mode 2 (transmit at crest and trough)*3

[Legend]

X: Don't care

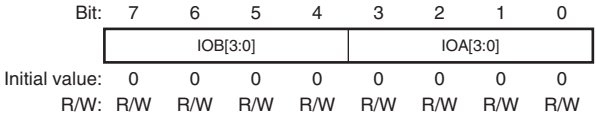
Notes: 1. PWM mode 2 cannot be set for channels 3 and 4.

2. Phase counting mode cannot be set for channels 0, 3, and 4.

3. Reset synchronous PWM mode, complementary PWM mode can only be set f channel 3. When channel 3 is set to reset synchronous PWM mode or comple PWM mode, the channel 4 settings become ineffective and automatically conf channel 3 settings. However, do not set channel 4 to reset synchronous PWM complementary PWM mode. Reset synchronous PWM mode and complement mode cannot be set for channels 0, 1, and 2.

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

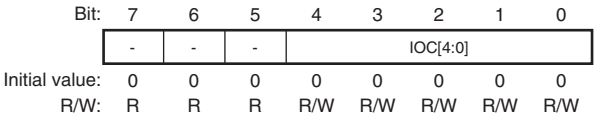
- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4



Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOB[3:0]	0000	R/W	I/O Control B0 to B3 Specify the function of TGRB. See the following tables. TIORH_0: Table 10.12 TIOR_1: Table 10.14 TIOR_2: Table 10.15 TIORH_3: Table 10.16 TIORH_4: Table 10.18
3 to 0	IOA[3:0]	0000	R/W	I/O Control A0 to A3 Specify the function of TGRA. See the following tables. TIORH_0: Table 10.20 TIOR_1: Table 10.22 TIOR_2: Table 10.23 TIORH_3: Table 10.24 TIORH_4: Table 10.26

				See the following tables. TIORL_0: Table 10.13 TIORL_3: Table 10.17 TIORL_4: Table 10.19
3 to 0	IOC[3:0]	0000	R/W	I/O Control C0 to C3 Specify the function of TGRC. See the following tables. TIORL_0: Table 10.21 TIORL_3: Table 10.25 TIORL_4: Table 10.27

- TIORU_5, TIORV_5, TIORW_5



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
4 to 0	IOC[4:0]	00000	R/W	I/O Control C0 to C4 Specify the function of TGRU_5, TGRV_5, and TGRW_5. For details, see table 10.28.

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

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	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. After power-on reset, 0 is output until TIOR is set.

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.

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

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

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.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_4 is set to 1 and TGRD_4 is used as a buffer regis setting is invalid and input capture/output compare is not generated.

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

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0
					Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1
					0 output at compare match
		1	0		Initial output is 1
					1 output at compare match
			1		Initial output is 1
					Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	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. After power-on reset, 0 is output until TIOR is set.

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.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of channel 0 compare match/input capture

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

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.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer regis setting is invalid and input capture/output compare is not generated.

	1	X	X	X		Setting prohibited
1	0	0	0	0	Input capture register	Setting prohibited
				1		Input capture at rising edge
			1	0		Input capture at falling edge
				1		Input capture at both edges
		1	X	X		Setting prohibited
	1	0	0	0		Setting prohibited
				1		Measurement of low pulse width of external i
						Capture at trough in complementary PWM m
			1	0		Measurement of low pulse width of external i
						Capture at crest in complementary PWM mo
				1		Measurement of low pulse width of external i
						Capture at crest and trough in complementar mode
		1	0	0		Setting prohibited
				1		Measurement of high pulse width of external
						Capture at trough in complementary PWM m
			1	0		Measurement of high pulse width of external
						Capture at crest in complementary PWM mo
				1		Measurement of high pulse width of external
						Capture at crest and trough in complementar mode

[Legend]

X: Don't care

Bit	Bit Name	Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
2	CMPCLR5U	0	R/W	TCNT Compare Clear 5U Enables or disables requests to clear TCNTU_ TGRU_5 compare match or input capture. 0: Disables TCNTU_5 to be cleared to H'0000 TCNTU_5 and TGRU_5 compare match or i capture 1: Enables TCNTU_5 to be cleared to H'0000 a TCNTU_5 and TGRU_5 compare match or i capture
1	CMPCLR5V	0	R/W	TCNT Compare Clear 5V Enables or disables requests to clear TCNTV_ TGRV_5 compare match or input capture. 0: Disables TCNTV_5 to be cleared to H'0000 TCNTV_5 and TGRV_5 compare match or i capture 1: Enables TCNTV_5 to be cleared to H'0000 a TCNTV_5 and TGRV_5 compare match or i capture

10.3.5 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disablin interrupt requests for each channel. The MTU2 has seven TIER registers, two for chann one each for channels 1 to 5.

- TIER_0, TIER_1, TIER_2, TIER_3, TIER_4

Bit:	7	6	5	4	3	2	1	0
	TTGE	TTGE2	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE	0	R/W	A/D Converter Start Request Enable Enables or disables generation of A/D conver requests by TGRA input capture/compare ma 0: A/D converter start request generation disa 1: A/D converter start request generation ena

				underflow (trough) enabled
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) b TCFU flag when the TCFU flag in TSR is set to channels 1 and 2. In channels 0, 3, and 4, bit 5 is reserved. It is a read as 0 and the write value should always be 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) b TCFV flag when the TCFV flag in TSR is set to 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) b TGFD bit when the TGFD bit in TSR is set to 1 channels 0, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is alwa as 0 and the write value should always be 0. 0: Interrupt requests (TGID) by TGFD bit disab 1: Interrupt requests (TGID) by TGFD bit enabl

1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) TGFB bit when the TGFB bit in TSR is set to 0: Interrupt requests (TGIB) by TGFB bit disa 1: Interrupt requests (TGIB) by TGFB bit enab
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) TGFA bit when the TGFA bit in TSR is set to 0: Interrupt requests (TGIA) by TGFA bit disa 1: Interrupt requests (TGIA) by TGFA bit enab

				TGRE_0. 0: A/D converter start request generation by co match between TCNT_0 and TGRE_0 disab 1: A/D converter start request generation by co match between TCNT_0 and TGRE_0 enab
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
1	TGIEF	0	R/W	TGR Interrupt Enable F Enables or disables interrupt requests by comp match between TCNT_0 and TGRF_0. 0: Interrupt requests (TGIF) by TGFE bit disab 1: Interrupt requests (TGIF) by TGFE bit enabl
0	TGIEE	0	R/W	TGR Interrupt Enable E Enables or disables interrupt requests by comp match between TCNT_0 and TGRE_0. 0: Interrupt requests (TGIE) by TGEE bit disab 1: Interrupt requests (TGIE) by TGEE bit enabl

2	TGIE5U	0	R/W	TGR Interrupt Enable 5U Enables or disables interrupt requests (TGIU_ CMFU5 bit when this bit in TSR_5 is set to 1. 0: Interrupt requests (TGIU_5) disabled 1: Interrupt requests (TGIU_5) enabled
1	TGIE5V	0	R/W	TGR Interrupt Enable 5V Enables or disables interrupt requests (TGIV_ CMFV5 bit when this bit in TSR_5 is set to 1. 0: Interrupt requests (TGIV_5) disabled 1: Interrupt requests (TGIV_5) enabled
0	TGIE5W	0	R/W	TGR Interrupt Enable 5W Enables or disables interrupt requests (TGIW CMFW5 bit when this bit in TSR_5 is set to 1. 0: Interrupt requests (TGIW_5) disabled 1: Interrupt requests (TGIW_5) enabled

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TC counts in channels 1 to 4. In channel 0, bit 7 is reserved. It is always read and the write value should always be 1. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1. The write value sh always be 1.
5	TCFU	0	R/(W)*[1]	Underflow Flag Status flag that indicates that TCNT underflow occurred when channels 1 and 2 are set to pha counting mode. Only 0 can be written, for flag c In channels 0, 3, and 4, bit 5 is reserved. It is a read as 0 and the write value should always be [Clearing condition] • When 0 is written to TCFU after reading TC [Setting condition] • When the TCNT value underflows (change H'0000 to H'FFFF)

				H'FFFF to H'0000) In channel 4, when the TCNT_4 value un (changes from H'0001 to H'0000) in comp PWM mode, this flag is also set.
3	TGFD	0	R/(W)*1	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of T capture or compare match in channels 0, 3, a Only 0 can be written, for flag clearing. In cha and 2, bit 3 is reserved. It is always read as 0 write value should always be 0. [Clearing condition] • When 0 is written to TGFD after reading TGFD = 1*2 [Setting conditions] • When TCNT = TGRD and TGRD is functi output compare register • When TCNT value is transferred to TGRD capture signal and TGRD is functioning a capture register

				[Setting conditions] • When TCNT = TGRC and TGRC is functio output compare register • When TCNT value is transferred to TGRC capture signal and TGRC is functioning as capture register
1	TGFB	0	R/(W)*1	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TG capture or compare match. Only 0 can be writ flag clearing. [Clearing condition] • When 0 is written to TGFB after reading TGFB = 1*2 [Setting conditions] • When TCNT = TGRB and TGRB is functio output compare register • When TCNT value is transferred to TGRB capture signal and TGRB is functioning as capture register

[Setting conditions]

- When TCNT = TGRA and TGRA is functi output compare register
- When TCNT value is transferred to TGRA capture signal and TGRA is functioning a capture register

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed
2. When writing to the timer status register (TSR), write 0 to the bit to be clearec reading 1. Write 1 to other bits. But 1 is not actually written and the previous v held.

				These bits are always read as 1. The write va should always be 1.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
1	TGFF	0	R/(W)*1	Compare Match Flag F Status flag that indicates the occurrence of co match between TCNT_0 and TGRF_0. [Clearing condition] • When 0 is written to TGFF after reading TGFF = 1*2 [Setting condition] • When TCNT_0 = TGRF_0 and TGRF_0 is functioning as compare register
0	TGFE	0	R/(W)*1	Compare Match Flag E Status flag that indicates the occurrence of co match between TCNT_0 and TGRE_0. [Clearing condition] • When 0 is written to TGFE after reading TGFE = 1*2 [Setting condition] • When TCNT_0 = TGRE_0 and TGRE_0 is functioning as compare register

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed v
2. When writing to the timer status register (TSR), write 0 to the bit to be cleared reading 1. Write 1 to other bits. But 1 is not actually written and the previous va held.

				These bits are always read as 0. The write valu always be 0.
2	CMFU5	0	R/(W)*1	Compare Match/Input Capture Flag U5 Status flag that indicates the occurrence of TGI input capture or compare match. [Clearing condition] • When 0 is written to CMFU5 after reading C [Setting conditions] • When TCNTU_5 = TGRU_5 and TGRU_5 i functioning as output compare register • When TCNTU_5 value is transferred to TGI input capture signal and TGRU_5 is functio input capture register • When TCNTU_5 value is transferred to TGI TGRU_5 is functioning as a register for mea pulse width of the external input signal. The timing is specified by the IOC bits in timer I/ registers U_5, V_5, and W_5 (TIORU_5, TI and TIORW_5).*2

- When TCNTV_5 value is transferred to TGR
  input capture signal and TGRV_5 is functioni
  input capture register
- When TCNTV_5 value is transferred to TGR
  TGRV_5 is functioning as a register for meas
  pulse width of the external input signal. The t
  timing is specified by the IOC bits in timer I/C
  registers U_5, V_5, and W_5 (TIORU_5, TIC
  and TIORW_5).*2

- When TCNTW_5 = TGRW_5 and TGRW_5
  functioning as output compare register
- When TCNTW_5 value is transferred to TG
  input capture signal and TGRW_5 is functio
  input capture register
- When TCNTW_5 value is transferred to TG
  TGRW_5 is functioning as a register for me
  the pulse width of the external input signal.

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed
2. Timing for transfer is set by the IOC bit in the timer I/O control register U_5/V_
(TIORU_5/V_5/W_5).

10.3.7 Timer Buffer Operation Transfer Mode Register (TBTM)

The TBTM registers are 8-bit readable/writable registers that specify the timing for trans
data from the buffer register to the timer general register in PWM mode. The MTU2 has
TBTM registers, one each for channels 0, 3, and 4.

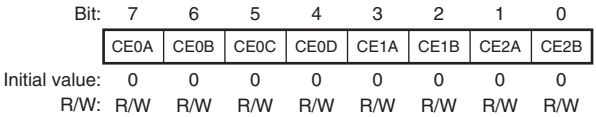
Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TTSE	TTSB	TTSA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

				channel 0 is used in a mode other than PWM m not set this bit to 1. 0: When compare match E occurs in channel 0 1: When TCNT_0 is cleared
1	TTSB	0	R/W	Timing Select B Specifies the timing for transferring data from T TGRB in each channel when they are used tog buffer operation. When the channel is used in a other than PWM mode, do not set this bit to 1. 0: When compare match B occurs in each cha 1: When TCNT is cleared in each channel
0	TTSA	0	R/W	Timing Select A Specifies the timing for transferring data from T TGRA in each channel when they are used tog buffer operation. When the channel is used in a other than PWM mode, do not set this bit to 1. 0: When compare match A occurs in each cha 1: When TCNT is cleared in each channel

Bit	Bit Name	Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
3	I2BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2B pin i TGRB_1 input capture conditions. 0: Does not include the TIOC2B pin in the TG input capture conditions 1: Includes the TIOC2B pin in the TGRB_1 in capture conditions
2	I2AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2A pin i TGRA_1 input capture conditions. 0: Does not include the TIOC2A pin in the TG input capture conditions 1: Includes the TIOC2A pin in the TGRA_1 in capture conditions
1	I1BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1B pin i TGRB_2 input capture conditions. 0: Does not include the TIOC1B pin in the TG input capture conditions 1: Includes the TIOC1B pin in the TGRB_2 in capture conditions

10.3.9 Timer Synchronous Clear Register (TSYCR)

TSYCR is an 8-bit readable/writable register that specifies conditions for clearing TCNT_
TCNT_4 in the MTU2S in synchronization with the MTU2. The MTU2S has one TSYCR
channel 3 but the MTU2 has no TSYCR.



Bit:	7	6	5	4	3	2	1	0
	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CE0A	0	R/W	Clear Enable 0A Enables or disables counter clearing when the flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag 1: Enables counter clearing by the TGFA flag i
6	CE0B	0	R/W	Clear Enable 0B Enables or disables counter clearing when the flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag 1: Enables counter clearing by the TGFB flag i

				0: Disables counter clearing by the TGFD flag 1: Enables counter clearing by the TGFD flag
3	CE1A	0	R/W	Clear Enable 1A Enables or disables counter clearing when the flag of TSR_1 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag 1: Enables counter clearing by the TGFA flag
2	CE1B	0	R/W	Clear Enable 1B Enables or disables counter clearing when the flag of TSR_1 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag 1: Enables counter clearing by the TGFB flag
1	CE2A	0	R/W	Clear Enable 2A Enables or disables counter clearing when the flag of TSR_2 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag 1: Enables counter clearing by the TGFA flag
0	CE2B	0	R/W	Clear Enable 2B Enables or disables counter clearing when the flag of TSR_2 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag 1: Enables counter clearing by the TGFB flag

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BF[1:0]	00	R/W	TADCOBRA_4/TADCOBRB_4 Transfer Timing Select the timing for transferring data from TADCOBRA_4 and TADCOBRB_4 to TADCO and TADCORB_4. For details, see table 10.29.
13 to 8	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
7	UT4AE	0	R/W	Up-Count TRG4AN Enable Enables or disables A/D converter start reques (TRG4AN) during TCNT_4 up-count operation 0: A/D converter start requests (TRG4AN) disa during TCNT_4 up-count operation 1: A/D converter start requests (TRG4AN) ena during TCNT_4 up-count operation
6	DT4AE	0*	R/W	Down-Count TRG4AN Enable Enables or disables A/D converter start reques (TRG4AN) during TCNT_4 down-count operati 0: A/D converter start requests (TRG4AN) disa during TCNT_4 down-count operation 1: A/D converter start requests (TRG4AN) ena during TCNT_4 down-count operation

				Enables or disables A/D converter start requ (TRG4BN) during TCNT_4 down-count opera 0: A/D converter start requests (TRG4BN) dis during TCNT_4 down-count operation 1: A/D converter start requests (TRG4BN) en during TCNT_4 down-count operation
3	ITA3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requ (TRG4AN) with TGIA_3 interrupt skipping ope 0: Does not link with TGIA_3 interrupt skippin 1: Links with TGIA_3 interrupt skipping
2	ITA4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requ (TRG4AN) with TCIV_4 interrupt skipping ope 0: Does not link with TCIV_4 interrupt skippin 1: Links with TCIV_4 interrupt skipping
1	ITB3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requ (TRG4BN) with TGIA_3 interrupt skipping ope 0: Does not link with TGIA_3 interrupt skippin 1: Links with TGIA_3 interrupt skipping

interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE
timer A/D converter start request control register (TADCR) to 0).

3. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.

* Do not set to 1 when complementary PWM mode is not selected.

Table 10.29 Setting of Transfer Timing by Bits BF1 and BF0

Bit 7	Bit 6	
BF1	**BF0**	**Description**
0	0	Does not transfer data from the cycle set buffer register to th set register.
0	1	Transfers data from the cycle set buffer register to the cycle register at the crest of the TCNT_4 count.*[1]
1	0	Transfers data from the cycle set buffer register to the cycle register at the trough of the TCNT_4 count.*[2]
1	1	Transfers data from the cycle set buffer register to the cycle register at the crest and trough of the TCNT_4 count.*[2]

Notes: 1. Data is transferred from the cycle set buffer register to the cycle set register wh crest of the TCNT_4 count is reached in complementary PWM mode, when co match occurs between TCNT_3 and TGRA_3 in reset-synchronized PWM mo when compare match occurs between TCNT_4 and TGRA_4 in PWM mode 1 normal operation mode.

2. These settings are prohibited when complementary PWM mode is not selected

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

Note: TADCORA_4 and TADCORB_4 must not be accessed in eight bits; they should always be accessed in 16

10.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4)

TADCOBRA_4 and TADCOBRB_4 are 16-bit readable/writable registers. When the crest or trough of the TCNT_4 count is reached, these register values are transferred to TADCORA_4 or TADCORB_4, respectively.

TADCOBRA_4 and TADCOBRB_4 are initialized to H'FFFF.

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

Note: TADCOBRA_4 and TADCOBRB_4 must not be accessed in eight bits; they should always be accessed in

Note: The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

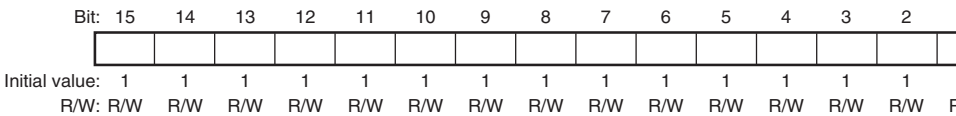
10.3.14 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers. The MTU2 has 21 TGR register
channel 0, two each for channels 1 and 2, four each for channels 3 and 4, and three for ch

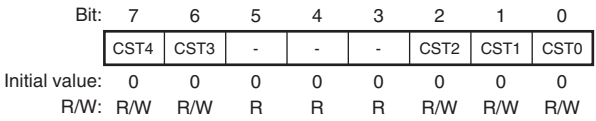
TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture reg
TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer
TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

TGRE_0 and TGRF_0 function as compare registers. When the TCNT_0 count matches t
TGRE_0 value, an A/D converter start request can be issued. TGRF can also be designate
operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

TGRU_5, TGRV_5, and TGRW_5 function as compare match, input capture, or external
width measurement registers.



Note: The TGR registers must not be accessed in eight bits; they should always be accessed in 16 bits.
TGR registers are initialized to H'FFFF.



Bit:	7	6	5	4	3	2	1	0
	CST4	CST3	-	-	-	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CST4	0	R/W	Counter Start 4 and 3
6	CST3	0	R/W	These bits select operation or stoppage for T If 0 is written to the CST bit during operation v TIOC pin designated for output, the counter s the TIOC pin output compare output level is r TIOR is written to when the CST bit is cleared pin output level will be changed to the set initi value. 0: TCNT_4 and TCNT_3 count operation is st 1: TCNT_4 and TCNT_3 performs count oper
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

- TSTR_5

Bit :	7	6	5	4	3	2	1	0
	-	-	-	-	-	CSTU5	CSTV5	CSTW5
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
2	CSTU5	0	R/W	Counter Start U5 Selects operation or stoppage for TCNTU_5. 0: TCNTU_5 count operation is stopped 1: TCNTU_5 performs count operation
1	CSTV5	0	R/W	Counter Start V5 Selects operation or stoppage for TCNTV_5. 0: TCNTV_5 count operation is stopped 1: TCNTV_5 performs count operation
0	CSTW5	0	R/W	Counter Start W5 Selects operation or stoppage for TCNTW_5. 0: TCNTW_5 count operation is stopped 1: TCNTW_5 performs count operation

Bit	Bit Name	Initial Value	R/W	Description
7	SYNC4	0	R/W	Timer Synchronous operation 4 and 3
6	SYNC3	0	R/W	These bits are used to select whether operati independent of or synchronized with other ch When synchronous operation is selected, the synchronous presetting of multiple channels, synchronous clearing by counter clearing on a channel, are possible. To set synchronous operation, the SYNC bits least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC TCNT clearing source must also be set by me bits CCLR0 to CCLR2 in TCR. 0: TCNT_4 and TCNT_3 operate independen presetting/clearing is unrelated to other cha 1: TCNT_4 and TCNT_3 performs synchrono operation TCNT synchronous presetting/synchronous is possible
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

synchronous clearing, in addition to the SYNC
TCNT clearing source must also be set by mea
bits CCLR0 to CCLR2 in TCR.

0: TCNT_2 to TCNT_0 operates independently
presetting /clearing is unrelated to other cha

1: TCNT_2 to TCNT_0 performs synchronous
TCNT synchronous presetting/synchronous
is possible

Bit	Bit Name	Initial Value	R/W	Description
7	SCH0	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_0 in the 0: Does not specify synchronous start for TCN the MTU2 1: Specifies synchronous start for TCNT_0 in [Clearing condition] • When 1 is set to the CST0 bit of TSTR in while SCH0 = 1
6	SCH1	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_1 in the 0: Does not specify synchronous start for TCN the MTU2 1: Specifies synchronous start for TCNT_1 in [Clearing condition] • When 1 is set to the CST1 bit of TSTR in while SCH1 = 1

4	SCH3	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_3 in the M 0: Does not specify synchronous start for TCN the MTU2 1: Specifies synchronous start for TCNT_3 in t [Clearing condition] • When 1 is set to the CST3 bit of TSTR in M while SCH3 = 1
3	SCH4	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_4 in the M 0: Does not specify synchronous start for TCN the MTU2 1: Specifies synchronous start for TCNT_4 in t [Clearing condition] • When 1 is set to the CST4 bit of TSTR in M while SCH4 = 1
2	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

				while SCH3S = 1
0	SCH4S	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_4S in th 0: Does not specify synchronous start for TCN the MTU2S 1: Specifies synchronous start for TCNT_4S i MTU2S [Clearing condition] • When 1 is set to the CST4 bit of TSTRS ir while SCH4S = 1

Note: Only 1 can be written to set the register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	RWE	1	R/W	Read/Write Enable Enables or disables access to the registers wh write-protection capability against accidental modification. 0: Disables read/write access to the registers 1: Enables read/write access to the registers [Clearing condition] • When 0 is written to the RWE bit after read RWE = 1

- Registers and counters having write-protection capability against accidental modificat
  22 registers: TCR_3, TCR_4, TMDR_3, TMDR_4, TIORH_3, TIORH_4, TIORL_3, TIORL_4, TIER_3, TIER_4, TGRA_3, TGRA_4, TGRB_3, TGRB_4, TOER, TOCR TOCR2, TGCR, TCDR, TDDR, TCNT_3, and TCNT4.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write va always be 1.
5	OE4D	0	R/W	Master Enable TIOC4D This bit enables/disables the TIOC4D pin MT 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
4	OE4C	0	R/W	Master Enable TIOC4C This bit enables/disables the TIOC4C pin MT 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
3	OE3D	0	R/W	Master Enable TIOC3D This bit enables/disables the TIOC3D pin MT 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
2	OE4B	0	R/W	Master Enable TIOC4B This bit enables/disables the TIOC4B pin MT 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
1	OE4A	0	R/W	Master Enable TIOC4A This bit enables/disables the TIOC4A pin MT 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled

10.3.20 Timer Output Control Register 1 (TOCR1)

TOCR1 is an 8-bit readable/writable register that enables/disables PWM synchronized to output in complementary PWM mode/reset synchronized PWM mode, and controls outpu inversion of PWM output.

Bit:	7	6	5	4	3	2	1	0
	-	PSYE	-	-	TOCL	TOCS	OLSN	OLSP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/(W)*	R/W	R/W	R/W

Note: * This bit can be set to 1 only once after a power-on reset. After 1 is written, 0 cannot be written to th

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
6	PSYE	0	R/W	PWM Synchronous Output Enable This bit selects the enable/disable of toggle ou synchronized with the PWM period. 0: Toggle output is disabled 1: Toggle output is enabled
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				be used for the output level in complementary mode and reset-synchronized PWM mode. 0: TOCR1 setting is selected 1: TOCR2 setting is selected
1	OLSN	0	R/W	Output Level Select N*[3] This bit selects the reverse phase output leve synchronized PWM mode/complementary PW See table 10.30.
0	OLSP	0	R/W	Output Level Select P*[3] This bit selects the positive phase output leve synchronized PWM mode/complementary PW See table 10.31.

Notes: 1. This bit can be set to 1 only once after a power-on reset. After 1 is written, 0 c written to the bit.
2. Setting the TOCL bit to 1 prevents accidental modification when the CPU goe control.
3. Clearing the TOCS0 bit to 0 makes this bit setting valid.

Table 10.30 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to active level after elap dead time after count start.

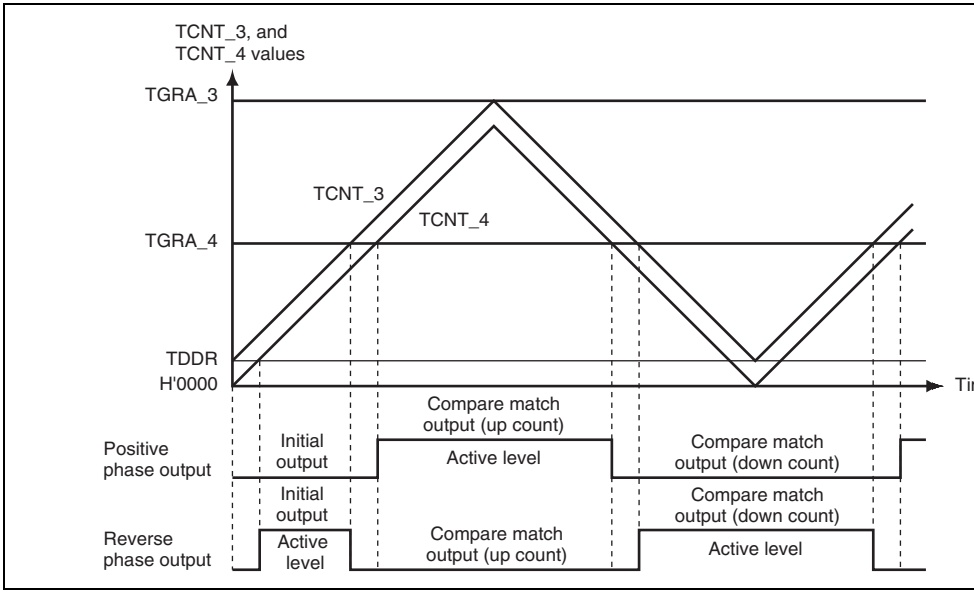


Figure 10.2 Complementary PWM Mode Output Level Example

Bit	Bit Name	value	R/W	Description
7, 6	BF[1:0]	00	R/W	TOLBR Buffer Transfer Timing Select These bits select the timing for transferring da TOLBR to TOCR2. For details, see table 10.32.
5	OLS3N	0	R/W	Output Level Select 3N* This bit selects the output level on TIOC4D in synchronized PWM mode/complementary PW See table 10.33.
4	OLS3P	0	R/W	Output Level Select 3P* This bit selects the output level on TIOC4B in synchronized PWM mode/complementary PW See table 10.34.
3	OLS2N	0	R/W	Output Level Select 2N* This bit selects the output level on TIOC4C in synchronized PWM mode/complementary PW See table 10.35.
2	OLS2P	0	R/W	Output Level Select 2P* This bit selects the output level on TIOC4A in synchronized PWM mode/complementary PW See table 10.36.
1	OLS1N	0	R/W	Output Level Select 1N* This bit selects the output level on TIOC3D in synchronized PWM mode/complementary PW See table 10.37.

BF1	BF0	Complementary PWM Mode	Reset-Synchronized PW
0	0	Does not transfer data from the buffer register (TOLBR) to TOCR2.	Does not transfer data fro buffer register (TOLBR) to
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the TCNT_4 count.	Transfers data from the bu register (TOLBR) to TOCF TCNT_3/TCNT_4 is cleare
1	0	Transfers data from the buffer register (TOLBR) to TOCR2 at the trough of the TCNT_4 count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the TCNT_4 count.	Setting prohibited

Table 10.33 TIOC4D Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to the active level after e the dead time after count start.

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to the active level after the dead time after count start.

Table 10.36 TIOC4A Output Level Select Function

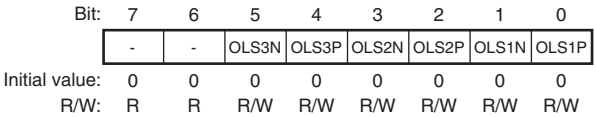
Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 10.37 TIOC3D Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to the active level after the dead time after count start.

TOLBR is an 8-bit readable/writable register that functions as a buffer for TOCR2 and sp
the PWM output level in complementary PWM mode and reset-synchronized PWM mod



Bit:	7	6	5	4	3	2	1	0
	-	-	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
5	OLS3N	0	R/W	Specifies the buffer value to be transferred to t OLS3N bit in TOCR2.
4	OLS3P	0	R/W	Specifies the buffer value to be transferred to t OLS3P bit in TOCR2.
3	OLS2N	0	R/W	Specifies the buffer value to be transferred to t OLS2N bit in TOCR2.
2	OLS2P	0	R/W	Specifies the buffer value to be transferred to t OLS2P bit in TOCR2.
1	OLS1N	0	R/W	Specifies the buffer value to be transferred to t OLS1N bit in TOCR2.
0	OLS1P	0	R/W	Specifies the buffer value to be transferred to t OLS1P bit in TOCR2.

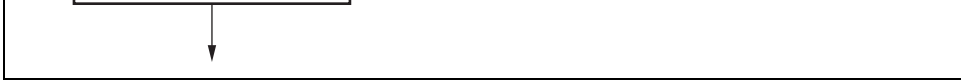


Figure 10.3 PWM Output Level Setting Procedure in Buffer Operation

10.3.23 Timer Gate Control Register (TGCR)

TGCR is an 8-bit readable/writable register that controls the waveform output necessary brushless DC motor control in reset-synchronized PWM mode/complementary PWM m register settings are ineffective for anything other than complementary PWM mode/rese synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	-	BDC	N	P	FB	WF	VF	UF
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.
6	BDC	0	R/W	Brushless DC Motor This bit selects whether to make the functions register (TGCR) effective or ineffective. 0: Ordinary output 1: Functions of this register are made effectiv

				This bit selects whether the level output or the synchronized PWM/complementary PWM outp the positive pin (TIOC3B, TIOC4A, and TIOC4 output. 0: Level output 1: Reset synchronized PWM/complementary P output
3	FB*	0	R/W	External Feedback Signal Enable This bit selects whether the switching of the ou the positive/reverse phase is carried out autom with the MTU2/channel 0 TGRA, TGRB, TGRC capture signals or by writing 0 or 1 to bits 2 to TGCR. 0: Output switching is external input (Input sou channel 0 TGRA, TGRB, TGRC input captu 1: Output switching is carried out by software (values of UF, VF, and WF in TGCR).
2	WF	0	R/W	Output Phase Switch 2 to 0 These bits set the positive phase/negative pha phase on or off state. The setting of these bits only when the FB bit in this register is set to 1. case, the setting of bits 2 to 0 is a substitute fo input. See table 10.39.
1	VF	0	R/W	
0	UF	0	R/W	

Note: * If the BDC bit in the MTU2S is set to 1, the FB bit should not be cleared to 0.

	1	ON	OFF	OFF	OFF	ON
1	0	OFF	OFF	ON	ON	OFF
	1	OFF	OFF	OFF	OFF	OFF

10.3.24 Timer Subcounter (TCNTS)

TCNTS is a 16-bit read-only counter that is used only in complementary PWM mode.

The initial value of TCNTS is H'0000.

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

Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

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

Note: Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

10.3.26 Timer Cycle Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM ca value as the TCDR register value. This register is constantly compared with the TCNTS c complementary PWM mode, and when a match occurs, the TCNTS counter switches dire (decrement to increment).

The initial value of TCDR is H'FFFF.

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

Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

10.3.28 Timer Interrupt Skipping Set Register (TITCR)

TITCR is an 8-bit readable/writable register that enables or disables interrupt skipping a specifies the interrupt skipping count. The MTU2 has one TITCR.

Bit:	7	6	5	4	3	2	1	0
	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	T3AEN	0	R/W	T3AEN Enables or disables TGIA_3 interrupt skipping 0: TGIA_3 interrupt skipping disabled 1: TGIA_3 interrupt skipping enabled
6 to 4	3ACOR[2:0]	000	R/W	These bits specify the TGIA_3 interrupt skipp within the range from 0 to 7.* For details, see table 10.40.
3	T4VEN	0	R/W	T4VEN Enables or disables TCIV_4 interrupt skipping 0: TCIV_4 interrupt skipping disabled 1: TCIV_4 interrupt skipping enabled

Bit 6	Bit 5	Bit 4	
3ACOR2	**3ACOR1**	**3ACOR0**	**Description**
0	0	0	Does not skip TGIA_3 interrupts.
0	0	1	Sets the TGIA_3 interrupt skipping count to 1.
0	1	0	Sets the TGIA_3 interrupt skipping count to 2.
0	1	1	Sets the TGIA_3 interrupt skipping count to 3.
1	0	0	Sets the TGIA_3 interrupt skipping count to 4.
1	0	1	Sets the TGIA_3 interrupt skipping count to 5.
1	1	0	Sets the TGIA_3 interrupt skipping count to 6.
1	1	1	Sets the TGIA_3 interrupt skipping count to 7.

Table 10.41 Setting of Interrupt Skipping Count by Bits 4VCOR2 to 4VCOR0

Bit 2	Bit 1	Bit 0	
4VCOR2	**4VCOR1**	**4VCOR0**	**Description**
0	0	0	Does not skip TCIV_4 interrupts.
0	0	1	Sets the TCIV_4 interrupt skipping count to 1.
0	1	0	Sets the TCIV_4 interrupt skipping count to 2.
0	1	1	Sets the TCIV_4 interrupt skipping count to 3.
1	0	0	Sets the TCIV_4 interrupt skipping count to 4.
1	0	1	Sets the TCIV_4 interrupt skipping count to 5.
1	1	0	Sets the TCIV_4 interrupt skipping count to 6.
1	1	1	Sets the TCIV_4 interrupt skipping count to 7.

Bit	Bit Name	Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.
6 to 4	3ACNT[2:0]	000	R	TGIA_3 Interrupt Counter While the T3AEN bit in TITCR is set to 1, the these bits is incremented every time a TGIA_ occurs. [Clearing conditions] • When the 3ACNT2 to 3ACNT0 value in TI matches the 3ACOR2 to 3ACOR0 value i • When the T3AEN bit in TITCR is cleared t • When the 3ACOR2 to 3ACOR0 bits in TIT cleared to 0
3	—	0	R	Reserved This bit is always read as 0.
2 to 0	4VCNT[2:0]	000	R	TCIV_4 Interrupt Counter While the T4VEN bit in TITCR is set to 1, the these bits is incremented every time a TCIV_ occurs. [Clearing conditions] • When the 4VCNT2 to 4VCNT0 value in TI matches the 4VCOR2 to 4VCOR2 value i • When the T4VEN bit in TITCR is cleared t • When the 4VCOR2 to 4VCOR2 bits in TIT cleared to 0

Note: To clear the TITCNT, clear the bits T3AEN and T4VEN in TITCR to 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
1, 0	BTE[1:0]	00	R/W	These bits enable or disable transfer from the I registers* used in complementary PWM mode temporary registers and specify whether to link transfer with interrupt skipping operation. For details, see table 10.42.

Note: * Applicable buffer registers:
TGRC_3, TGRD_3, TGRC_4, TGRD_4, and TCBR

Note: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For det
to section 10.4.8, Complementary PWM Mode.

2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared t
timer interrupt skipping set register (TITCR) or the skipping count set bits (3A
4VCOR) in TITCR are cleared to 0)), be sure to disable link of buffer transfer
interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (T
0). If link with interrupt skipping is enabled while interrupt skipping is disabled
transfer will not be performed.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	TDER	1	R/(W)	Dead Time Enable Specifies whether to generate dead time. 0: Does not generate dead time 1: Generates dead time* [Clearing condition] • When 0 is written to TDER after reading TD

Note: * TDDR must be set to 1 or a larger value.

Note: * Do not set to 1 when complementary PWM mode is not selected.

Bit	Bit Name	Initial Value	R/W	Description
7	CCE	0*	R/(W)	Compare Match Clear Enable Specifies whether to clear counters at TGRA_ compare match in complementary PWM mod 0: Does not clear counters at TGRA_3 compa 1: Clears counters at TGRA_3 compare matc [Setting condition] • When 1 is written to CCE after reading C0
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

Counter clearing synchronized with the MTU2
disabled by the SCC bit setting only when sync
clearing occurs outside the Tb interval at the tr
When synchronous clearing occurs in the Tb in
the trough including the period immediately aft
TCNT_3 and TCNT_4 start operation, TCNT_3
TCNT_4 in the MTU2S are cleared.

For the Tb interval at the trough in complemen
PWM mode, see figure 10.40.

In the MTU2, this bit is reserved. It is always re
and the write value should always be 0.

0: Enables clearing of TCNT_3 and TCNT_4 in
MTU2S by MTU2–MTU2S synchronous clea
operation

1: Disables clearing of TCNT_3 and TCNT_4 in
MTU2S by MTU2–MTU2S synchronous clea
operation

[Setting condition]

- When 1 is written to SCC after reading SC

When synchronous clearing occurs in the Tb i
the trough immediately after TCNT_3 and TC
operation.

For the Tb interval at the trough in compleme
PWM mode, see figure 10.40.

0: Outputs the initial value specified in TOCR

1: Retains the waveform output immediately b
synchronous clearing

[Setting condition]

- When 1 is written to WRE after reading W

Note: * Do not set to 1 when complementary PWM mode is not selected.

10.3.33 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer
buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register
timer A/D converter start request control register (TADCR), timer A/D converter start re
cycle set registers (TADCOR), and timer A/D converter start request cycle set buffer reg
(TADCOBR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read
bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. These are connected to the
16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

(1) Counter Operation

When one of bits CST0 to CST4 in TSTR or bits CSTU5, CSTV5, and CSTW5 in TSTR to 1, the TCNT counter for the corresponding channel begins counting. TCNT can operat free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 10.4 shows an example of the count operation setting procedure.

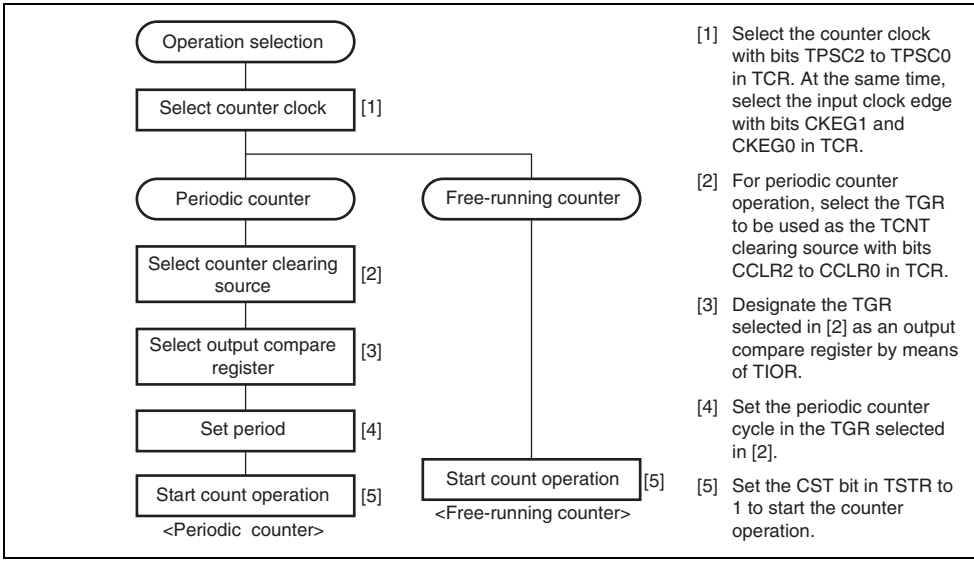


Figure 10.4 Example of Counter Operation Setting Procedure

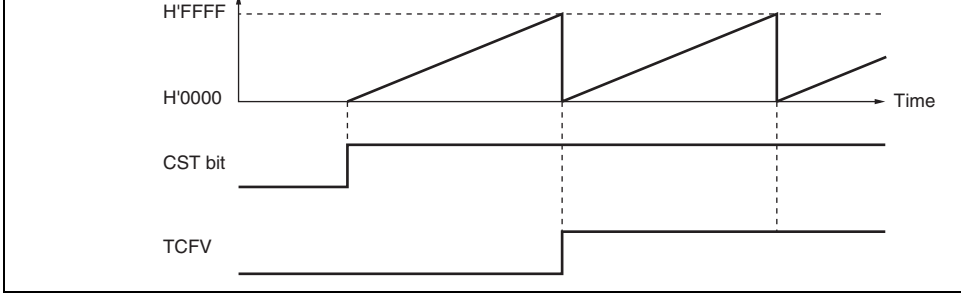


Figure 10.5 Free-Running Counter Operation

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 CCLR0 to CCLR2 in TCR. After the settings have been made, TCNT starts up-count 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 MTU2 requests interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU2 can perform 0, 1, or toggle output from the corresponding output pin using co
match.

(a) Example of Setting Procedure for Waveform Output by Compare Match

Figure 10.7 shows an example of the setting procedure for waveform output by compare

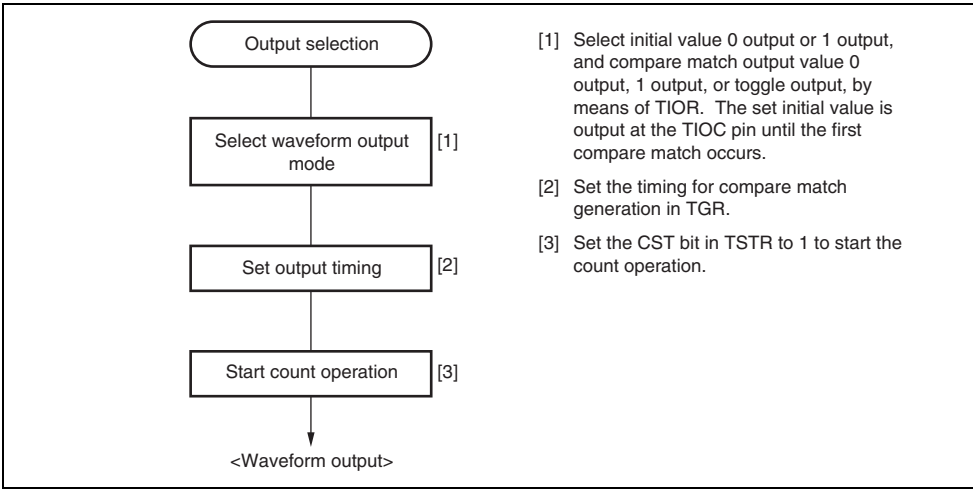


Figure 10.7 Example of Setting Procedure for Waveform Output by Compare M

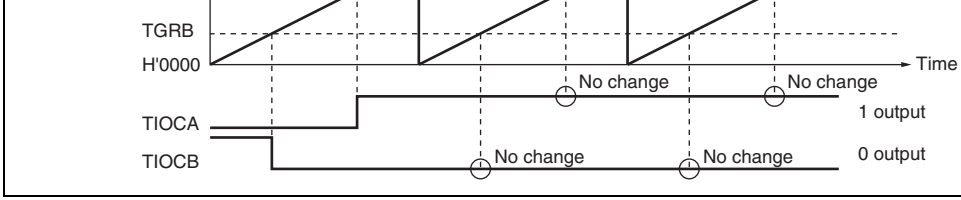


Figure 10.8 Example of 0 Output/1 Output Operation

Figure 10.9 shows an example of toggle output.

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

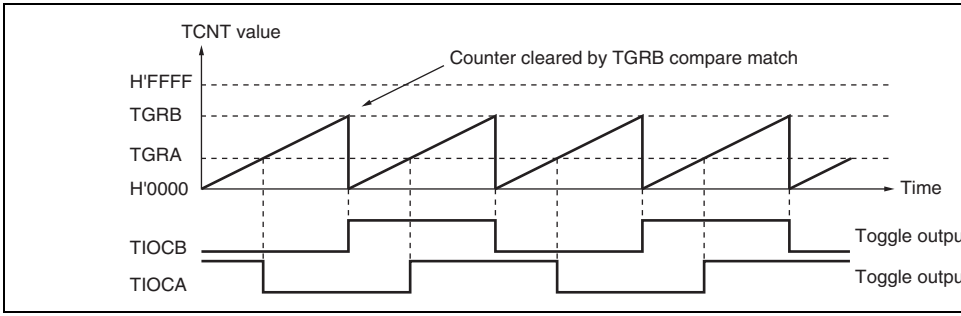


Figure 10.9 Example of Toggle Output Operation

(a) Example of Input Capture Operation Setting Procedure

Figure 10.10 shows an example of the input capture operation setting procedure.

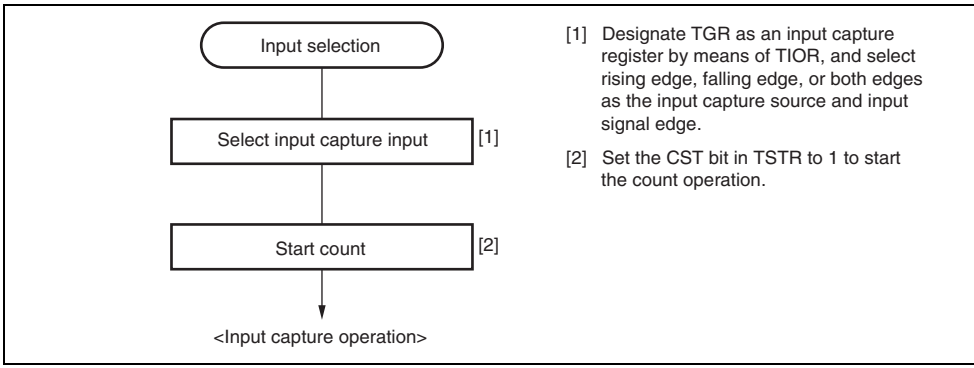


Figure 10.10 Example of Input Capture Operation Setting Procedure

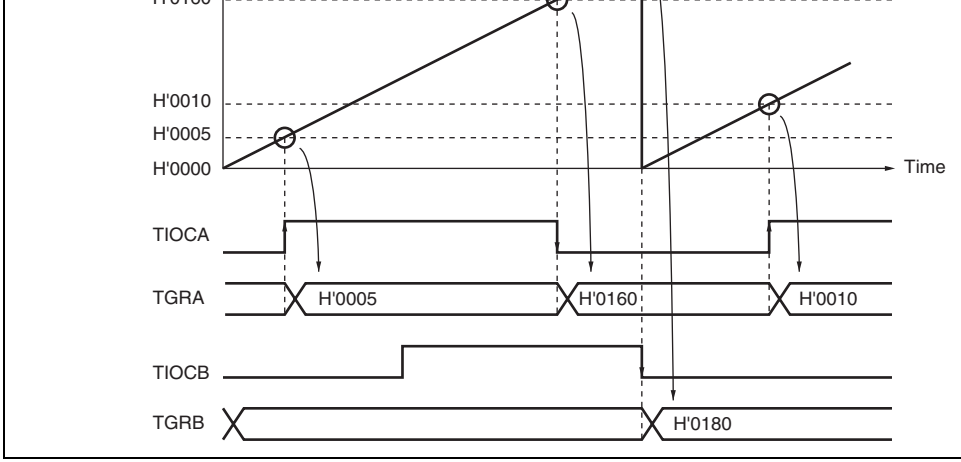


Figure 10.11 Example of Input Capture Operation

(1) Example of Synchronous Operation Setting Procedure

Figure 10.12 shows an example of the synchronous operation setting procedure.

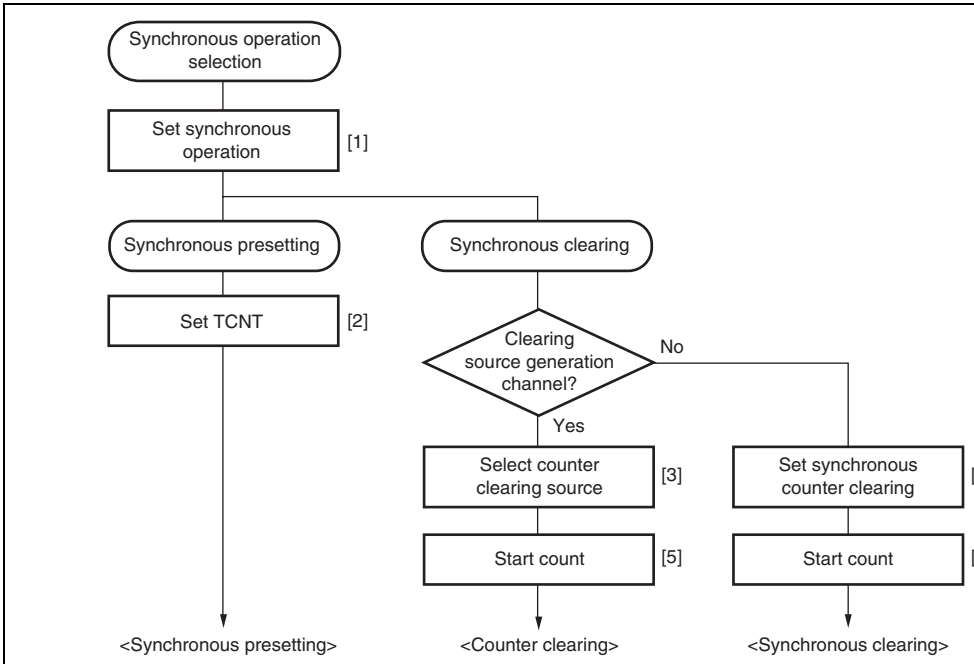


Figure 10.12 Example of Synchronous Operation Setting Procedure

cycle.

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

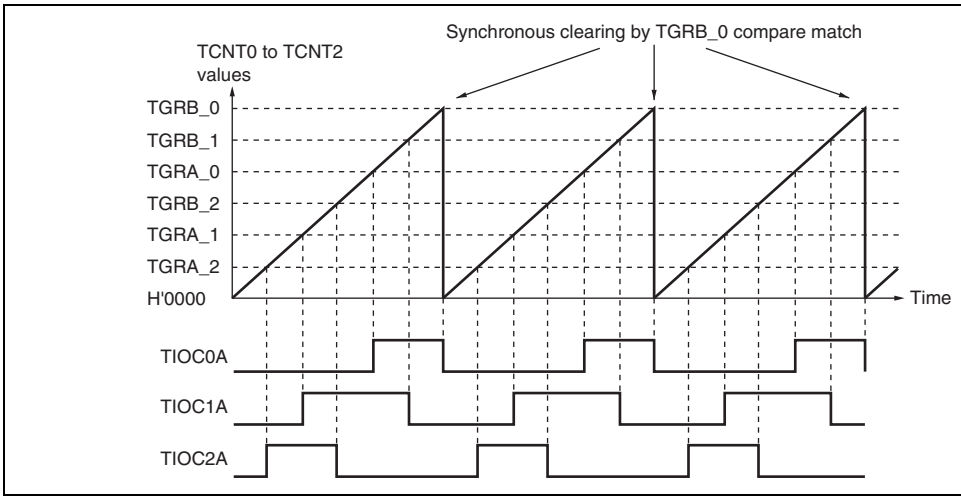


Figure 10.13 Example of Synchronous Operation

Table 10.43 shows the register combinations used in buffer operation.

Table 10.43 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
	TGRE_0	TGRF_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
4	TGRA_4	TGRC_4
	TGRB_4	TGRD_4

- When TGR is an output compare register

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

  This operation is illustrated in figure 10.14.

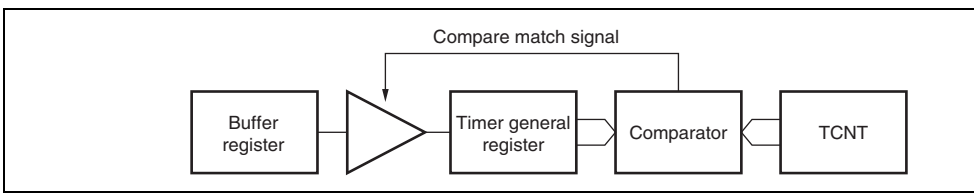


Figure 10.14 Compare Match Buffer Operation

Figure 10.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 10.16 shows an example of the buffer operation setting procedure.

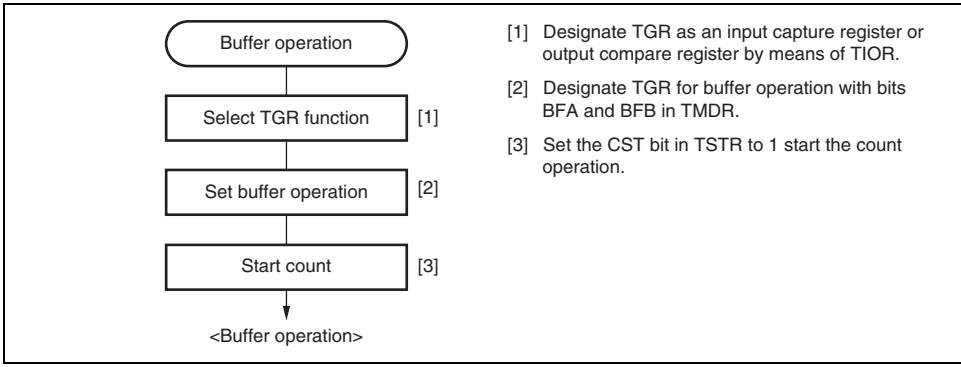


Figure 10.16 Example of Buffer Operation Setting Procedure

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

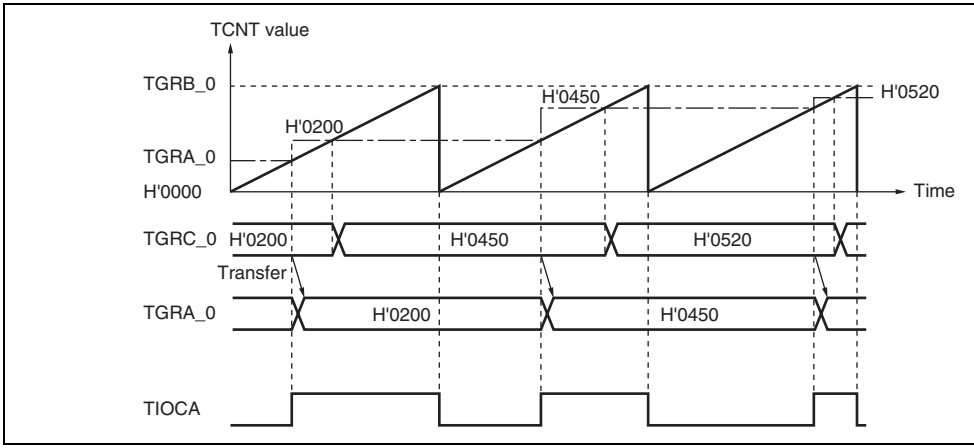


Figure 10.17 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 10.18 shows an operation example in which TGRA has been designated as an inpu
register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falli
have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon the occu
input capture A, the value previously stored in TGRA is simultaneously transferred to TC

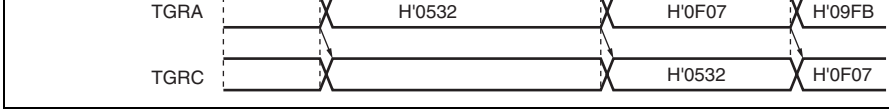


Figure 10.18 Example of Buffer Operation (2)

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Regist Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in mode 1 or 2 for channel 0 or in PWM mode 1 for channels 3 and 4 by setting the buffer transfer mode registers (TBTM_0, TBTM_3, and TBTM_4). Either compare match (init setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing as tran timing is one of the following cases.

- When TCNT overflows (H'FFFF to H'0000)
- When H'0000 is written to TCNT during counting
- When TCNT is cleared to H'0000 under the condition specified in the CCLR2 to CC in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 10.19 shows an operation example in which PWM mode 1 is designated for chan buffer operation is designated for TGRA_0 and TGRC_0. The settings used in this exam TCNT_0 clearing by compare match B, 1 output at compare match A, and 0 output at co match B. The TTSA bit in TBTM_0 is set to 1.

**Figure 10.19 Example of Buffer Operation When TCNT_0 Clearing is Selected
TGRC_0 to TGRA_0 Transfer Timing**

10.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32
counter.

This function works by counting the channel 1 counter clock upon overflow/underflow of
TCNT_2 as set in bits TPSC0 to TPSC2 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase counting mode.

Table 10.44 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1, the counter clock setting is inval
counters operates independently in phase counting mode.

Table 10.44 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2

For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, add
input capture input pins can be specified by the input capture control register (TICCR). F
capture in cascade connection, refer to section 10.7.22, Simultaneous Capture of TCNT_
TCNT_2 in Cascade Connection.

	I1AE bit = 1	TIOC2A, TIOC1A
Input capture from TCNT_2 to TGRB_2	I1BE bit = 0 (initial value)	TIOC2B
	I1BE bit = 1	TIOC2B, TIOC1B

(1) Example of Cascaded Operation Setting Procedure

Figure 10.20 shows an example of the setting procedure for cascaded operation.

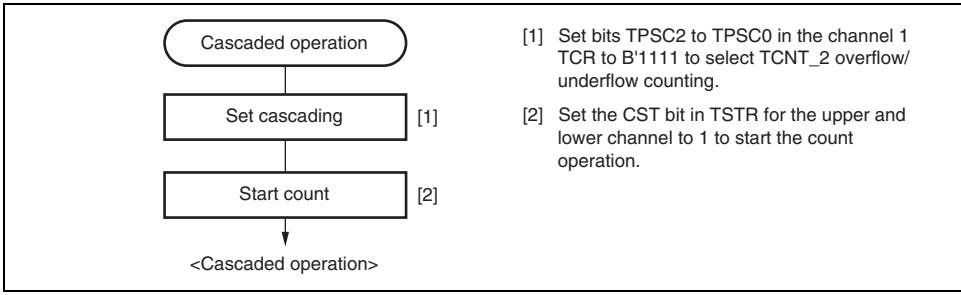


Figure 10.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 10.21 illustrates the operation when TCNT_2 overflow/underflow counting has b
TCNT_1 and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow

Figure 10.22 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded ar
I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input cap
conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected the TIOC1A
edge for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected t
TIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both TIOC1A and TIOC2A is used for the TGI
input capture condition. For the TGRA_2 input capture condition, the TIOC2A rising edg

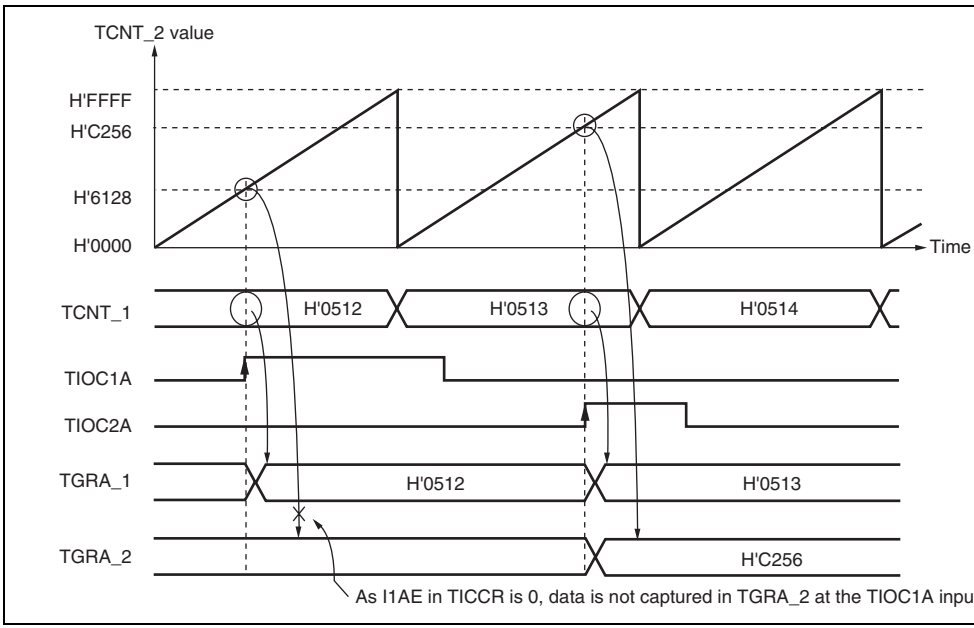


Figure 10.22 Cascaded Operation Example (b)

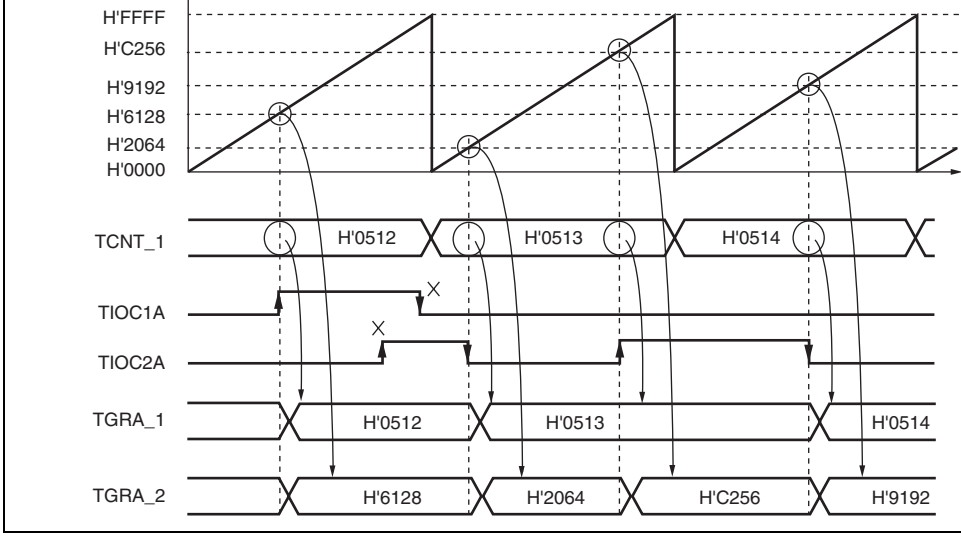


Figure 10.23 Cascaded Operation Example (c)

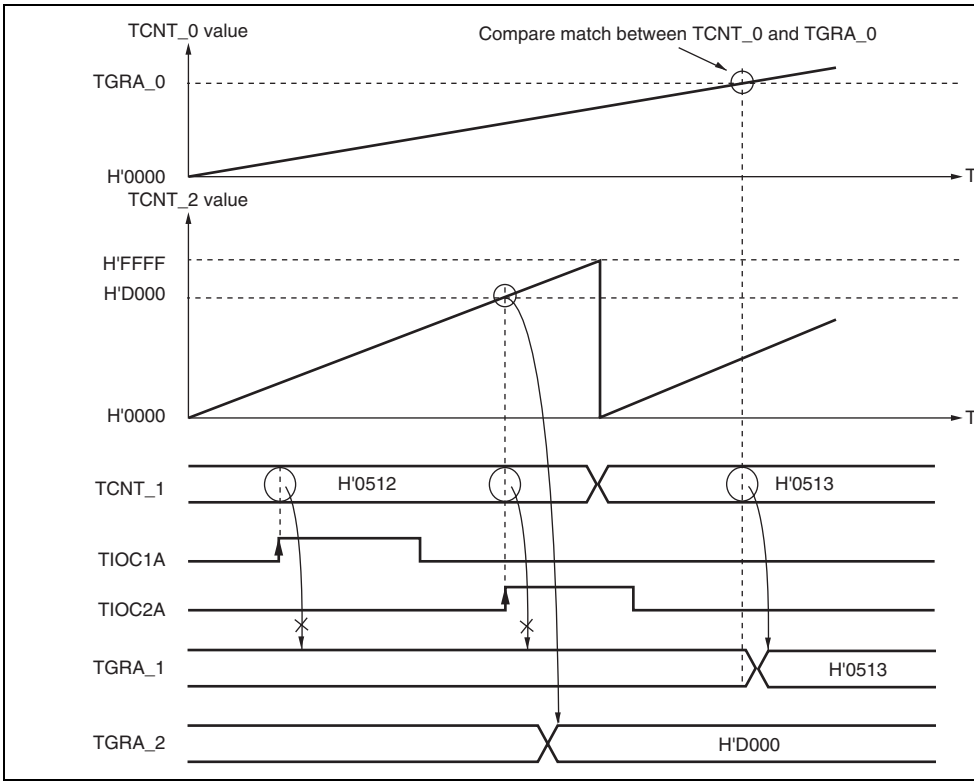


Figure 10.24 Cascaded Operation Example (d)

There are two PWM modes, as described below.

- PWM mode 1

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

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

- PWM mode 2

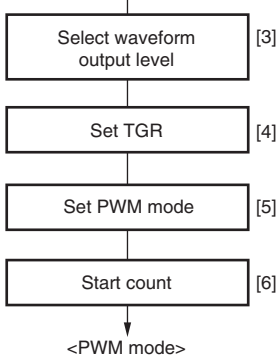
  PWM output is generated using one TGR as the cycle register and the others as duty
  The output specified in TIOR is performed by means of compare matches. Upon cou
  clearing by a synchronization register compare match, the output value of each pin is
  value set in TIOR. If the set values of the cycle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

  In PWM mode 2, a maximum 8-phase PWM output is possible in combination use w
  synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.46.

2	TGRA_2	TIOC2A	TIOC2A
	TGRB_2		TIOC2B
3	TGRA_3	TIOC3A	Cannot be set
	TGRB_3		Cannot be set
	TGRC_3	TIOC3C	Cannot be set
	TGRD_3		Cannot be set
4	TGRA_4	TIOC4A	Cannot be set
	TGRB_4		Cannot be set
	TGRC_4	TIOC4C	Cannot be set
	TGRD_4		Cannot be set

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



and output value.

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

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

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

Figure 10.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 10.26 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for tl initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB are used as the duty levels.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 com
match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 fo
output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting a 5
PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other T
used as the duty levels.

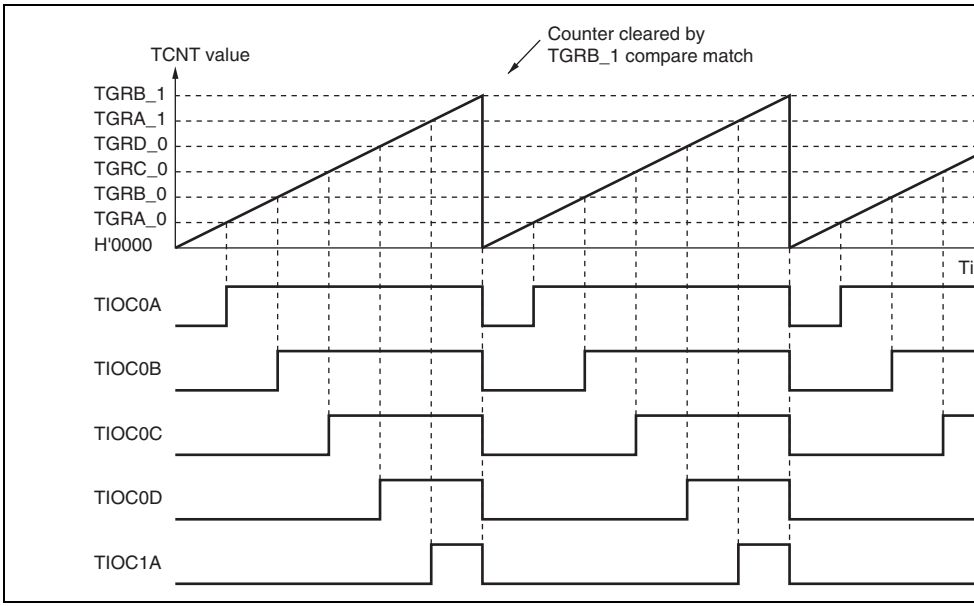


Figure 10.27 Example of PWM Mode Operation (2)

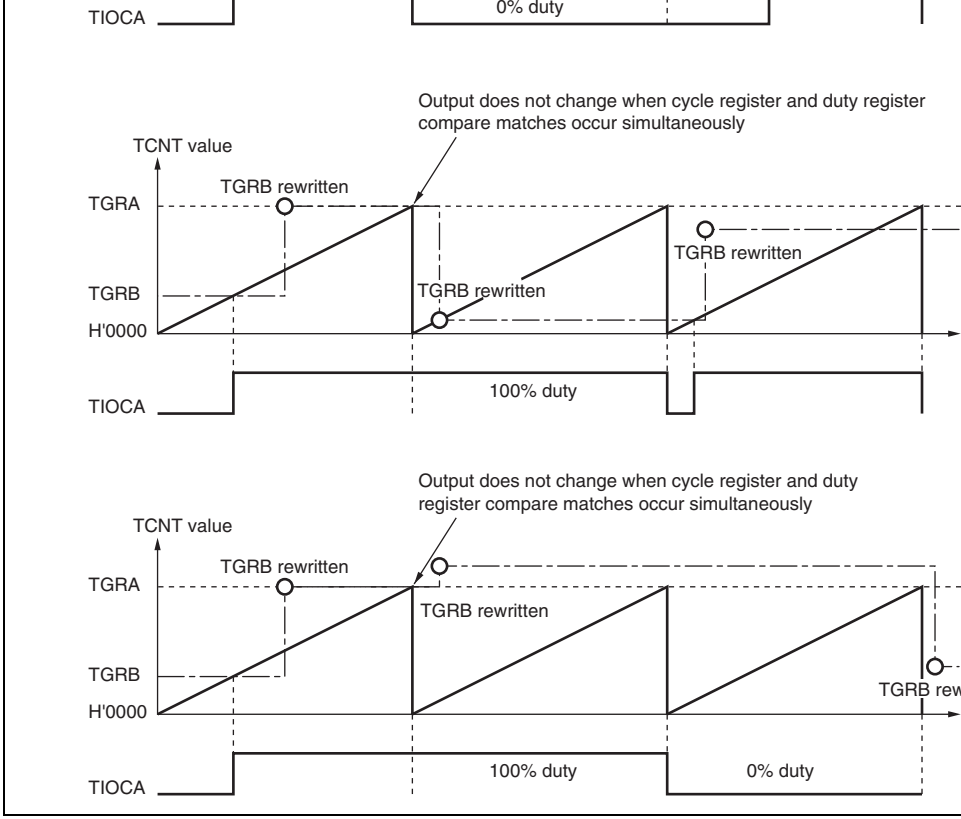


Figure 10.28 Example of PWM Mode Operation (3)

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

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether counting up or down.

Table 10.47 shows the correspondence between external clock pins and channels.

Table 10.47 Phase Counting Mode Clock Input Pins

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

(1) Example of Phase Counting Mode Setting Procedure

Figure 10.29 shows an example of the phase counting mode setting procedure.

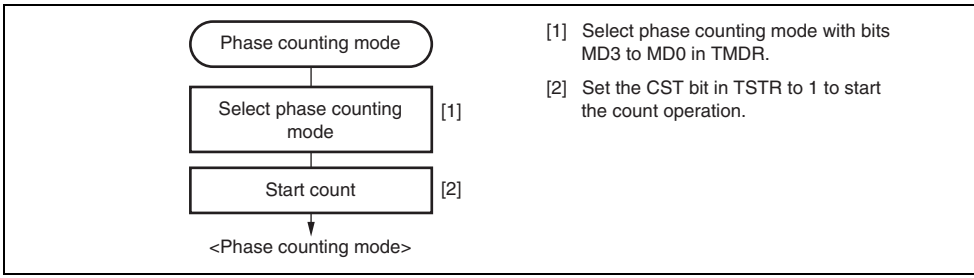


Figure 10.29 Example of Phase Counting Mode Setting Procedure

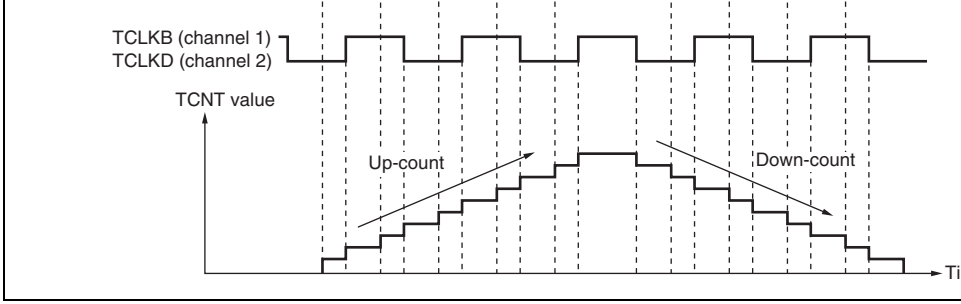


Figure 10.30 Example of Phase Counting Mode 1 Operation

Table 10.48 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

[Legend]

↑: Rising edge

↓: Falling edge

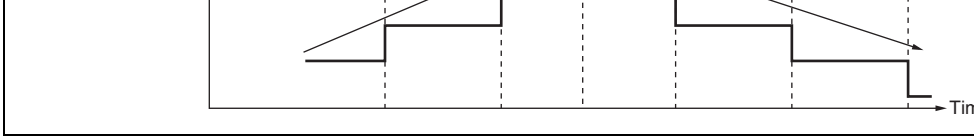


Figure 10.31 Example of Phase Counting Mode 2 Operation

Table 10.49 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⎍	Don't care
Low level	⎎	Don't care
⎍	Low level	Don't care
⎎	High level	Up-count
High level	⎎	Don't care
Low level	⎍	Don't care
⎍	High level	Don't care
⎎	Low level	Down-count

[Legend]

⎍: Rising edge

⎎: Falling edge

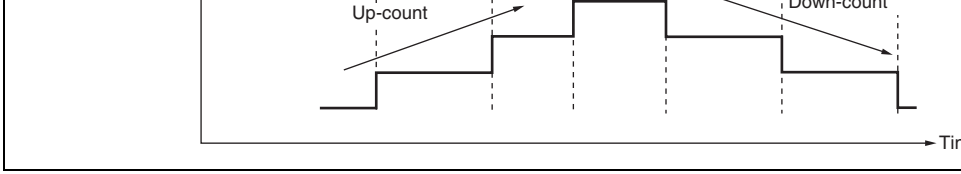


Figure 10.32 Example of Phase Counting Mode 3 Operation

Table 10.50 Up/Down-Count Conditions in Phase Counting Mode 3

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

[Legend]
↑: Rising edge
↓: Falling edge

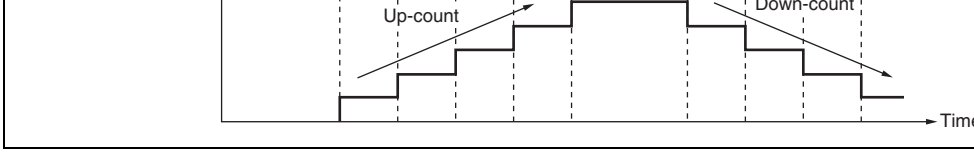


Figure 10.33 Example of Phase Counting Mode 4 Operation

Table 10.51 Up/Down-Count Conditions in Phase Counting Mode 4

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

[Legend]
↑: Rising edge
↓: Falling edge

position control period. TGRB_0 is used for input capture, with TGRB_0 and TGRB_0
in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input ca
source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TG
TGRC_0 compare matches are selected as the input capture source and store the up/dow
values for the control periods.

This procedure enables the accurate detection of position and speed.

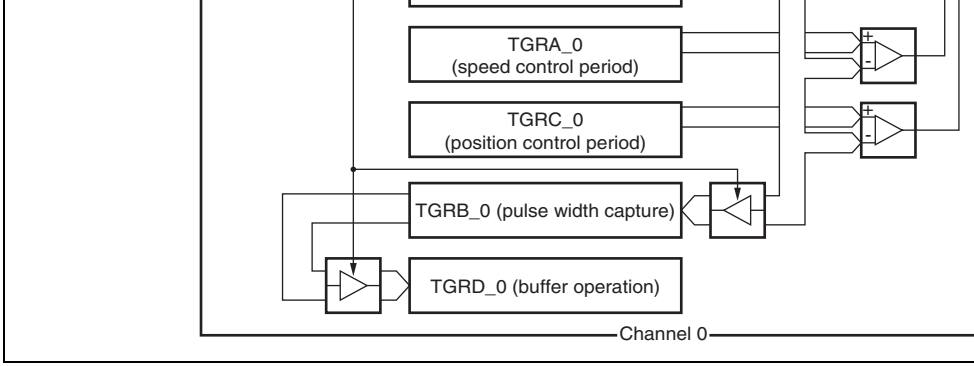
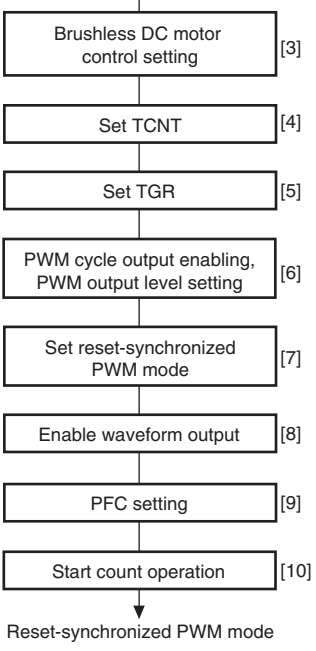


Figure 10.34 Phase Counting Mode Application Example

Channel	Output Pin	Description
3	TIOC3B	PWM output pin 1
	TIOC3D	PWM output pin 1' (negative-phase waveform of PWM ou
4	TIOC4A	PWM output pin 2
	TIOC4C	PWM output pin 2' (negative-phase waveform of PWM ou
	TIOC4B	PWM output pin 3
	TIOC4D	PWM output pin 3' (negative-phase waveform of PWM ou

Table 10.53 Register Settings for Reset-Synchronized PWM Mode

Register	Description of Setting
TCNT_3	Initial setting of H'0000
TCNT_4	Initial setting of H'0000
TGRA_3	Set count cycle for TCNT_3
TGRB_3	Sets the turning point for PWM waveform output by the TIOC3B and TIOC3
TGRA_4	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4
TGRB_4	Sets the turning point for PWM waveform output by the TIOC4B and TIOC4



[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Reset TCNT_3 and TCNT_4 to H'0000.

[5] TGRA_3 is the period register. Set the waveform period value in TGRA_3. Set the transition timing of the PWM output waveforms in TGRB_3, TGRA_4, and TGRB_4. Set times within the compare-match range of TCNT_3.
$X \leq$ TGRA_3 (X: set value).

[6] Select enabling/disabling of toggle output synchronized with the PMW cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 10.3.

[7] Set bits MD3-MD0 in TMDR_3 to B'1000 to select the reset-synchronized PWM mode. Do not set to TMDR_4.

[8] Set the enabling/disabling of the PWM waveform output pin in TOER.

[9] Set the port control register and the port I/O register.

[10] Set the CST3 bit in the TSTR to 1 to start the count operation.

Note: The output waveform starts to toggle operation at the point of TCNT_3 = TGRA_3 = X by setting X = TGRA, i.e., cycle = duty.

Figure 10.35 Procedure for Selecting Reset-Synchronized PWM Mode

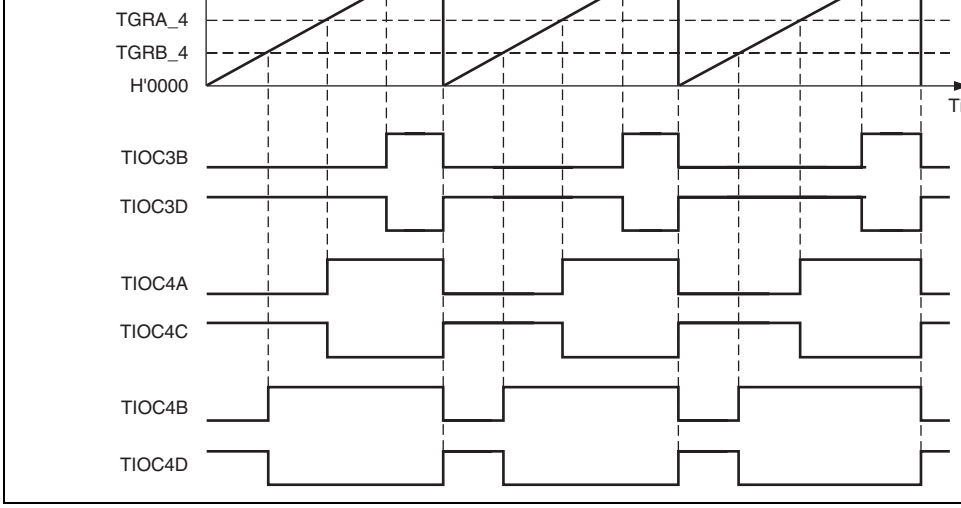


Figure 10.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

used.

A function to directly cut off the PWM output by using an external signal is supported as function.

Table 10.54 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3A	Toggle output synchronized with PWM period (or I/O
	TIOC3B	PWM output pin 1
	TIOC3C	I/O port*
	TIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM PWM output without non-overlapping interval is also a
4	TIOC4A	PWM output pin 2
	TIOC4B	PWM output pin 3
	TIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM PWM output without non-overlapping interval is also a
	TIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM PWM output without non-overlapping interval is also a

Note: * Avoid setting the TIOC3C pin as a timer I/O pin in complementary PWM mode

		register	
4	TCNT_4	Up-count start, initialized to H'0000	Maskable by TRW setting*
	TGRA_4	PWM output 2 compare register	Maskable by TRW setting*
	TGRB_4	PWM output 3 compare register	Maskable by TRW setting*
	TGRC_4	PWM output 2/TGRA_4 buffer register	Always readable/
	TGRD_4	PWM output 3/TGRB_4 buffer register	Always readable/
Timer dead time data register (TDDR)		Set TCNT_4 and TCNT_3 offset value (dead time value)	Maskable by TRW setting*
Timer cycle data register (TCDR)		Set TCNT_4 upper limit value (1/2 carrier cycle)	Maskable by TRW setting*
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/TGRB_3 temporary register	Not readable/writ
Temporary register 2 (TEMP2)		PWM output 2/TGRA_4 temporary register	Not readable/writ
Temporary register 3 (TEMP3)		PWM output 3/TGRB_4 temporary register	Not readable/writ

Note: * Access can be enabled or disabled according to the setting of bit 0 (RWE) in (timer read/write enable register).

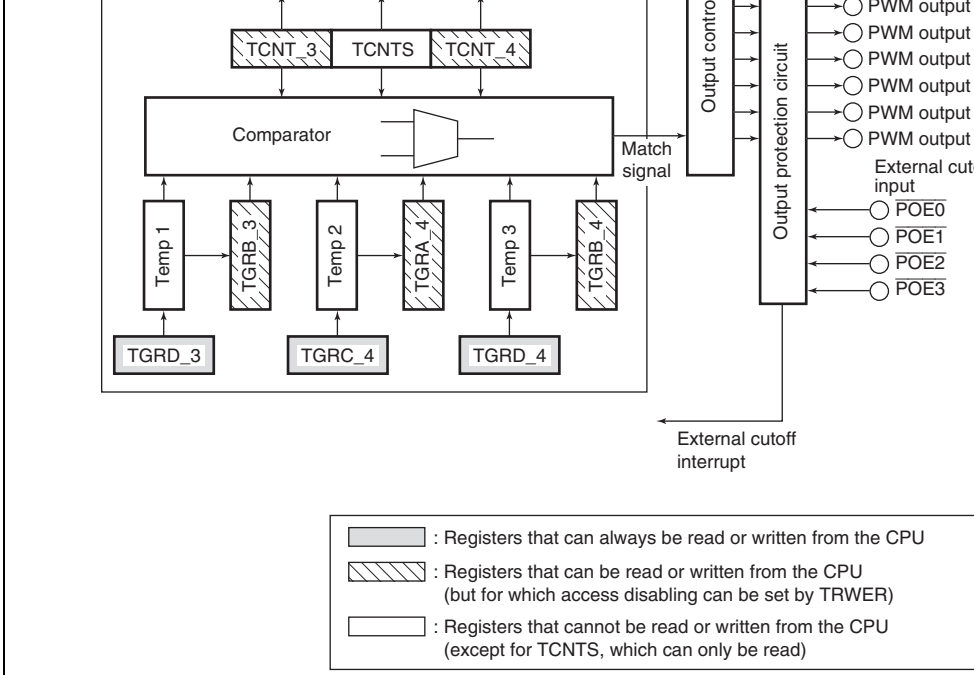
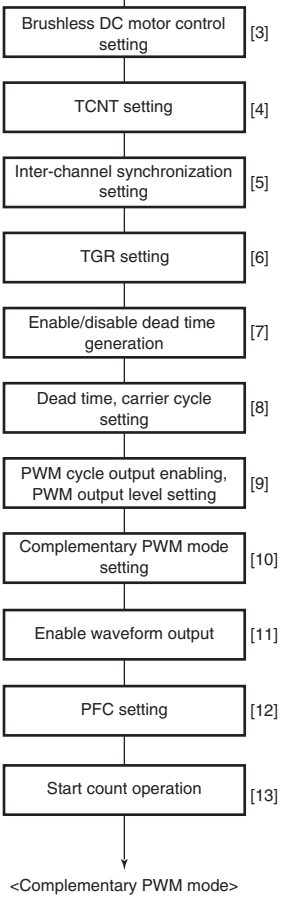


Figure 10.37 Block Diagram of Channels 3 and 4 in Complementary PWM M



[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set the dead time in TCNT_3. Set TCNT_4 to H'0000.

[5] Set only when restarting by a synchronous clear from another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with channels 3 and 4 using the timer synchro register (TSYR).

[6] Set the output PWM duty in the duty registers (TGRB_3, TGRA_4, TGRB_4) and buffer registers (TGRD_3, TGRC_4, TGRD_4). Set the same initial value in each corresponding TGR.

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the timer dead time enable register (TDER) so that no dead time is generated.

[8] Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the carrier cycle data register (TCDR) and carrier cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in TGRA_3 and TGRC_3. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in TGRA_3 and TGRC_3.

[9] Select enabling/disabling of toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 10.3.

[10] Select complementary PWM mode in timer mode register 3 (TMDR_3). Do not set in TMDR_4.

[11] Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).

[12] Set the port control register and the port I/O register.

[13] Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

Figure 10.38 Example of Complementary PWM Mode Setting Procedur

is selected and the CST bit in TSTR is 0.

When the CST bit is set to 1, TCNT_3 counts up to the value set in TGRA_3, then switch down-counting when it matches TGRA_3. When the TCNT3 value matches TDDR, the c switches to up-counting, and the operation is repeated in this way.

TCNT_4 is initialized to H'0000.

When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT_3, and sw down-counting when it matches TCDR. On reaching H'0000, TCNT4 switches to up-cou and the operation is repeated in this way.

TCNTS is a read-only counter. It need not be initialized.

When TCNT_3 matches TCDR during TCNT_3 and TCNT_4 up/down-counting, down- is started, and when TCNTS matches TCDR, the operation switches to up-counting. Whe TCNTS matches TGRA_3, it is cleared to H'0000.

When TCNT_4 matches TDDR during TCNT_3 and TCNT_4 down-counting, up-counti started, and when TCNTS matches TDDR, the operation switches to down-counting. Wh TCNTS reaches H'0000, it is set with the value in TGRA_3.

TCNTS is compared with the compare register and temporary register in which the PWM set during the count operation only.

Figure 10.39 Complementary PWM Mode Counter Operation

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, b registers, and temporary registers. Figure 10.40 shows an example of complementary PW operation.

The registers which are constantly compared with the counters to perform PWM output TGRB_3, TGRA_4, and TGRB_4. When these registers match the counter, the value se OLSN and OLSP in the timer output control register (TOCR) is output.

The buffer registers for these compare registers are TGRD_3, TGRC_4, and TGRD_4.

Between a buffer register and compare register there is a temporary register. The tempor registers cannot be accessed by the CPU.

Data in a compare register is changed by writing the new data to the corresponding buff The buffer registers can be read or written at any time.

The data written to a buffer register is constantly transferred to the temporary register in interval. Data is not transferred to the temporary register in the Tb interval. Data written buffer register in this interval is transferred to the temporary register at the end of the Tb

The value transferred to a temporary register is transferred to the compare register when for which the Tb interval ends matches TGRA_3 when counting up, or H'0000 when cou down. The timing for transfer from the temporary register to the compare register can be with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.40 shows an exam which the mode is selected in which the change is made in the trough.

In the tb interval (tb1 in figure 10.40) in which data transfer to the temporary register is performed, the temporary register has the same function as the compare register, and is c

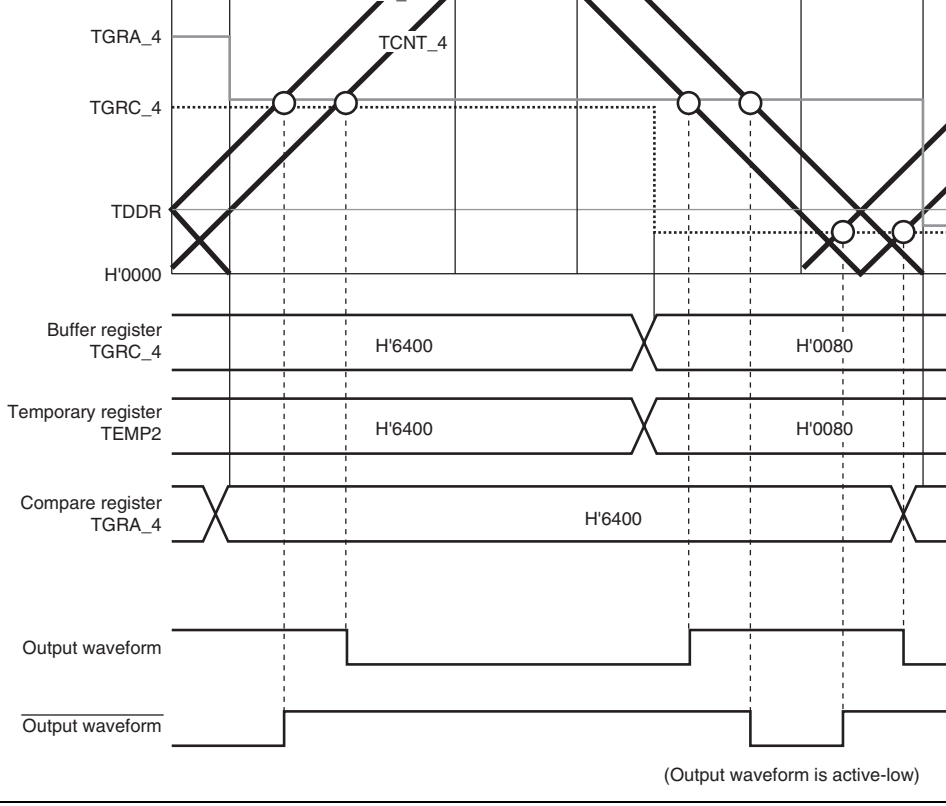


Figure 10.40 Example of Complementary PWM Mode Operation

time Td in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDE be cleared to 0, TGRC_3 and TGRA_3 should be set to 1/2 the PWM carrier cycle + 1, a should be set to 1.

Set the respective initial PWM duty values in buffer registers TGRD_3, TGRC_4, and TG

The values set in the five buffer registers excluding TDDR are transferred simultaneously corresponding compare registers when complementary PWM mode is set.

Set TCNT_4 to H'0000 before setting complementary PWM mode.

Table 10.56 Registers and Counters Requiring Initialization

Register/Counter	Set Value
TGRC_3	1/2 PWM carrier cycle + dead time Td (1/2 PWM carrier cycle + 1 when dead time ge is disabled by TDER)
TDDR	Dead time Td (1 when dead time generation is disabled by TDER)
TCBR	1/2 PWM carrier cycle
TGRD_3, TGRC_4, TGRD_4	Initial PWM duty value for each phase
TCNT_4	H'0000

Note: The TGRC_3 set value must be the sum of 1/2 the PWM carrier cycle set in TCBR dead time Td set in TDDR. When dead time generation is disabled by TDER, TGR must be set to 1/2 the PWM carrier cycle + 1.

In complementary PWM mode, PWM pulses are output with a non-overlapping relation between the positive and negative phases. This non-overlap time is called the dead time.

The non-overlap time is set in the timer dead time data register (TDDR). The value set i used as the TCNT_3 counter start value, and creates non-overlap between TCNT_3 and Complementary PWM mode should be cleared before changing the contents of TDDR.

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time ena register (TDER) to 0. TDER can be cleared to 0 only when 0 is written to it after readin 1.

TGRA_3 and TGRC_3 should be set to 1/2 PWM carrier cycle + 1 and the timer dead ti register (TDDR) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 10.4 an example of operation without dead time.

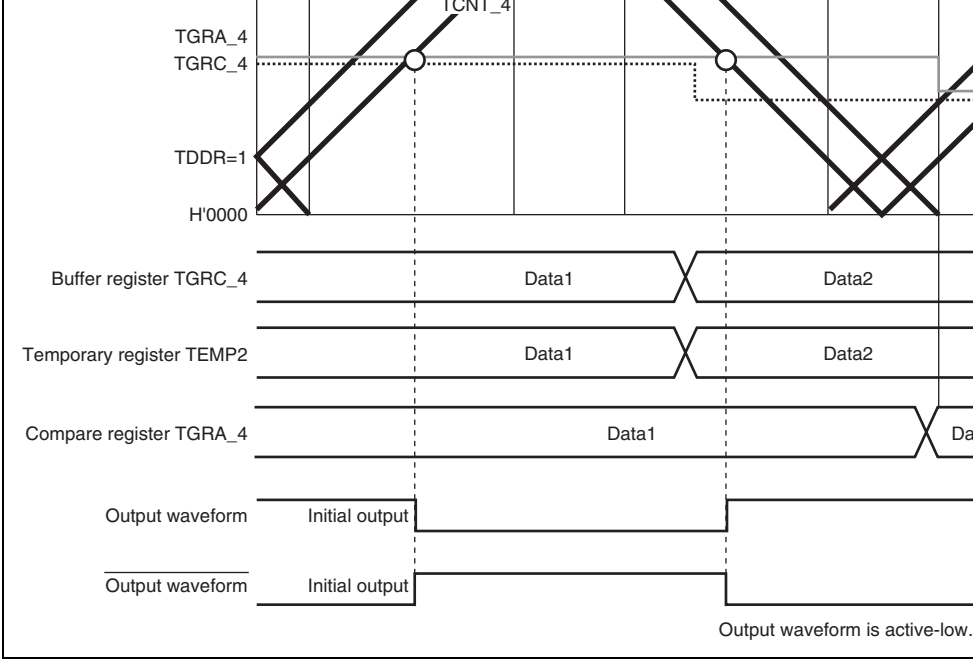


Figure 10.41 Example of Operation without Dead Time

TCDR in accordance with the transfer timing selected with bits MD3 to MD0 in the tim
register (TMDR).

The updated PWM cycle is reflected from the next cycle when the data update is perfor
crest, and from the current cycle when performed in the trough. Figure 10.42 illustrates
operation when the PWM cycle is updated at the crest.

See the following section, Register Data Updating, for the method of updating the data i
buffer register.

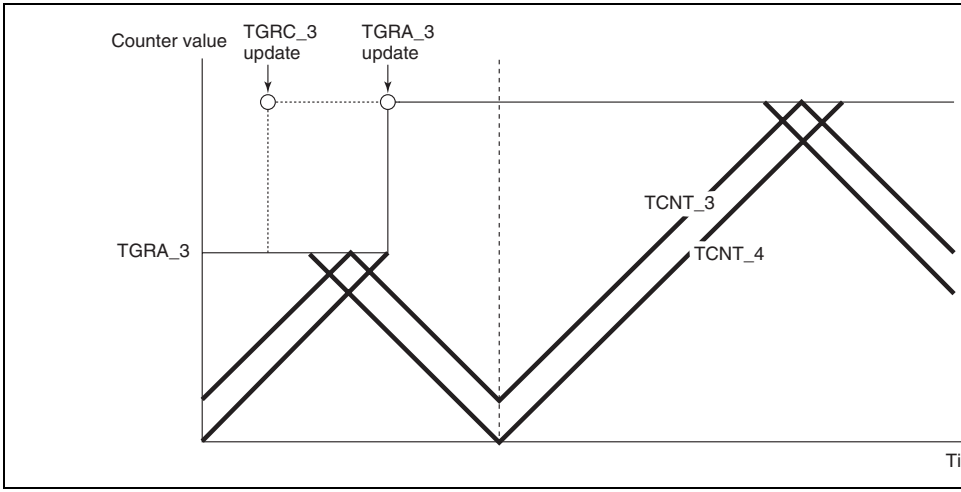


Figure 10.42 Example of PWM Cycle Updating

The temporary register value is transferred to the compare register at the data update timi with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.43 shows an exam data updating in complementary PWM mode. This example shows the mode in which dat updating is performed at both the counter crest and trough.

When rewriting buffer register data, a write to TGRD_4 must be performed at the end of update. Data transfer from the buffer registers to the temporary registers is performed simultaneously for all five registers after the write to TGRD_4.

A write to TGRD_4 must be performed after writing data to the registers to be updated, e not updating all five registers, or when updating the TGRD_4 data. In this case, the data v TGRD_4 should be the same as the data prior to the write operation.

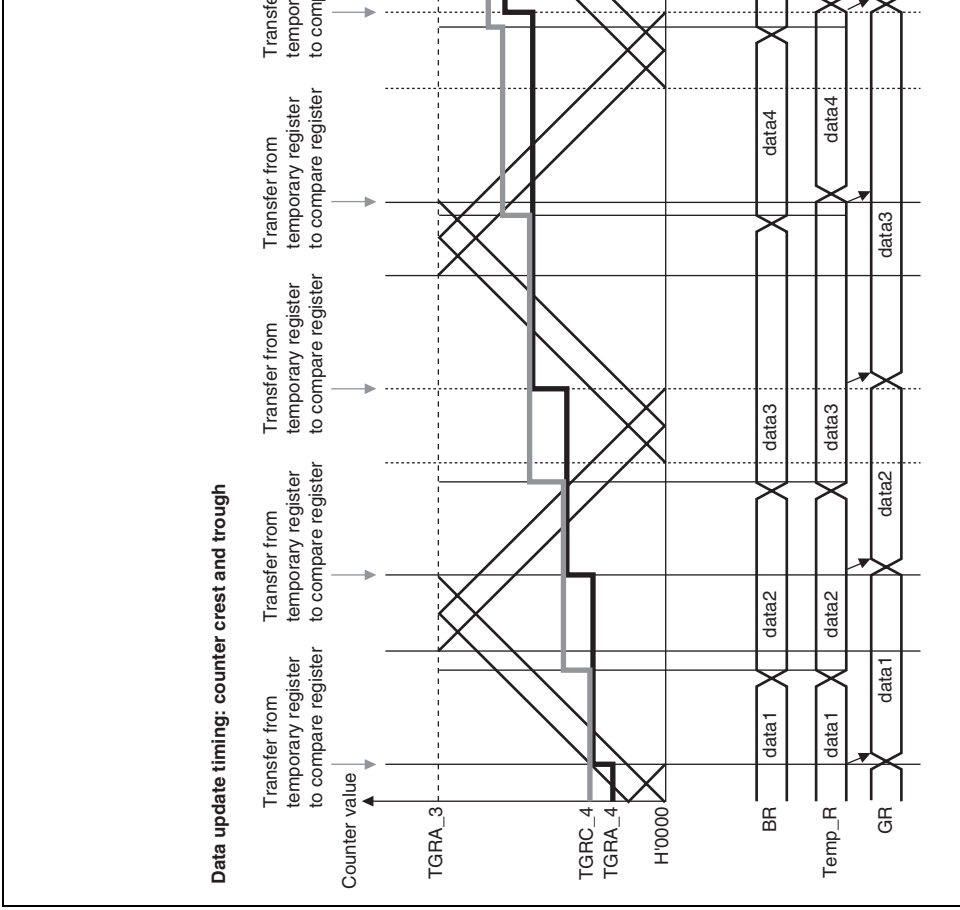


Figure 10.43 Example of Data Update in Complementary PWM Mode

shown in figure 10.45.

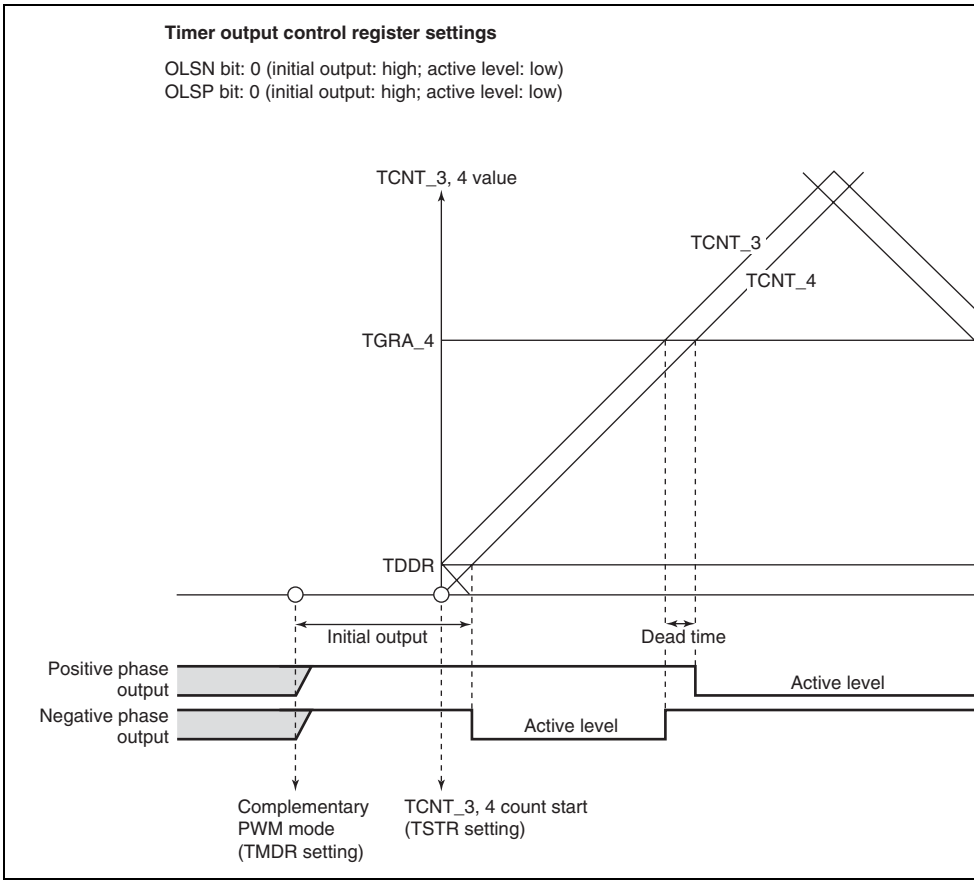


Figure 10.44 Example of Initial Output in Complementary PWM Mode (1

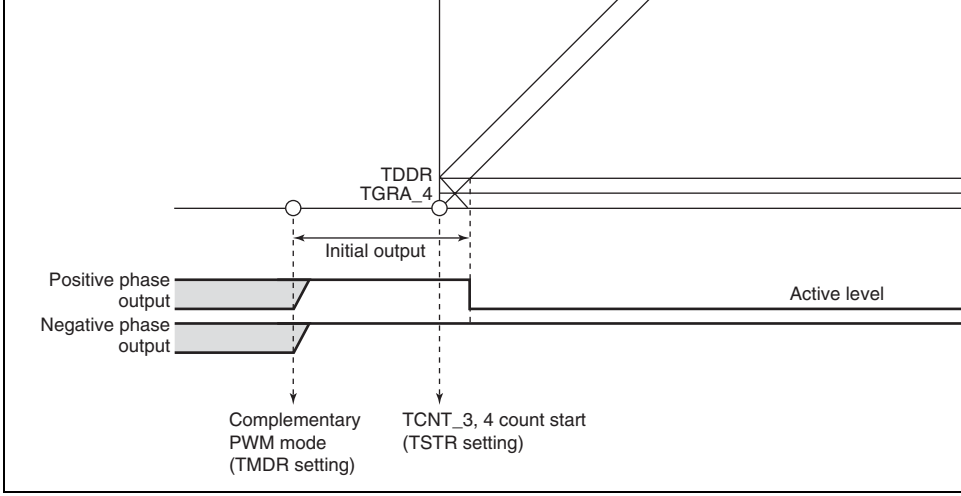


Figure 10.45 Example of Initial Output in Complementary PWM Mode (

the dead time and ensure that the positive phase and negative phase on-times do not overl
Figures 10.46 to 10.48 show examples of waveform generation in complementary PWM

The positive phase/negative phase off timing is generated by a compare-match with the s
counter, and the on timing by a compare-match with the dotted-line counter operating wi
of the dead time behind the solid-line counter. In the T1 period, compare-match **a** that tu
negative phase has the highest priority, and compare-matches occurring prior to **a** are ign
the T2 period, compare-match **c** that turns off the positive phase has the highest priority,
compare-matches occurring prior to **c** are ignored.

In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a'** →
shown in figure 10.46.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which the
phase is off is less than twice the dead time, the figure shows the positive phase is not bei
on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since the time for whic
positive phase is off is less than twice the dead time, the figure shows the negative phase
being turned on.

If compare-match **c** occurs first following compare-match **a**, as shown in figure 10.47, co
match **b** is ignored, and the negative phase is turned off by compare-match **d**. This is bec
turning off of the positive phase has priority due to the occurrence of compare-match **c** (p
phase off timing) before compare-match **b** (positive phase on timing) (consequently, the
does not change since the positive phase goes from off to off).

Similarly, in the example in figure 10.48, compare-match **a'** with the new data in the tem
register occurs before compare-match **c**, but other compare-matches occurring up to **c**, wh
off the positive phase, are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare-matches at turn-off timings take precede
turn-on timing compare-matches that occur before a turn-off timing compare-match are i

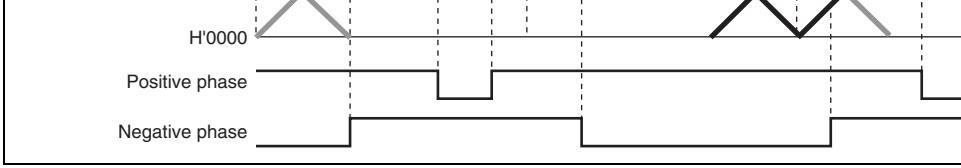


Figure 10.46 Example of Complementary PWM Mode Waveform Output

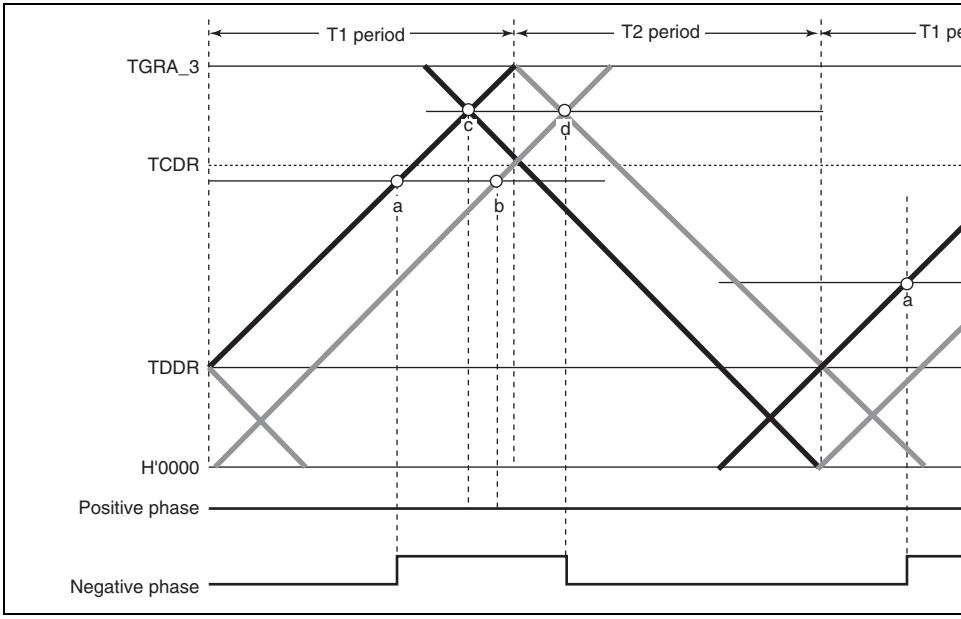


Figure 10.47 Example of Complementary PWM Mode Waveform Output

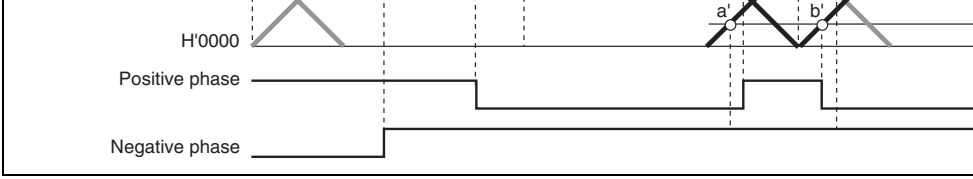


Figure 10.48 Example of Complementary PWM Mode Waveform Output (

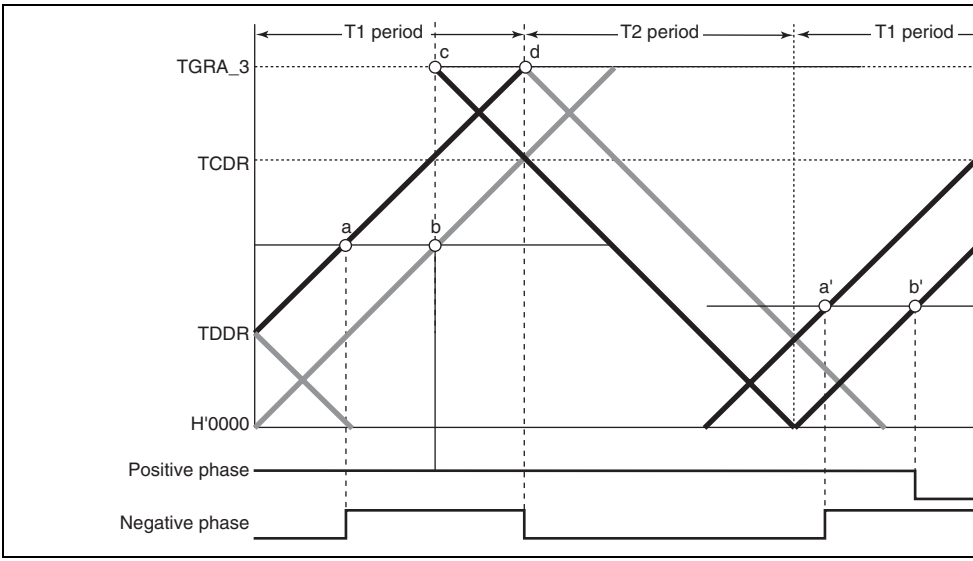


Figure 10.49 Example of Complementary PWM Mode 0% and 100% Waveform O

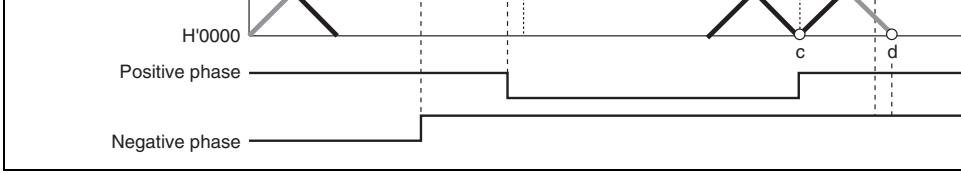


Figure 10.50 Example of Complementary PWM Mode 0% and 100% Waveform

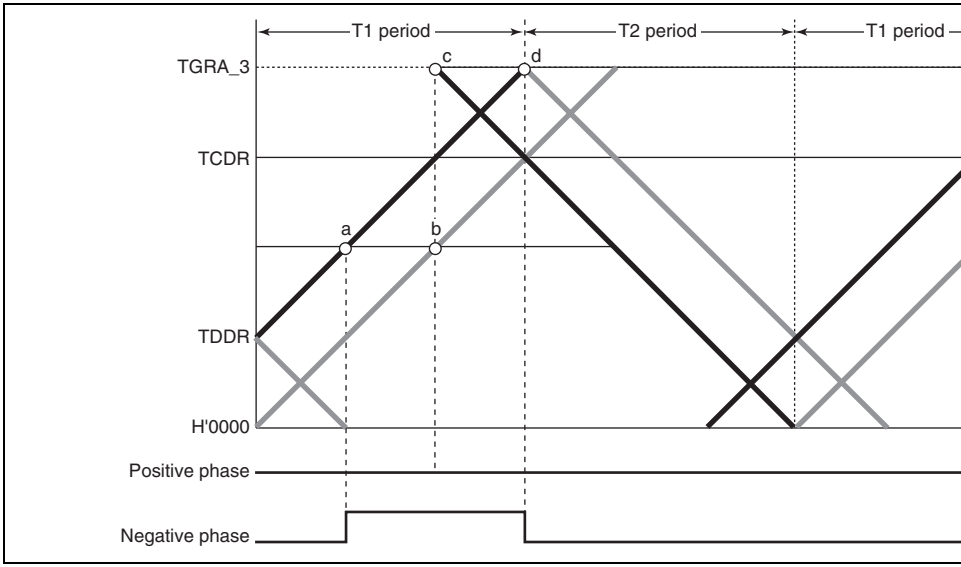


Figure 10.51 Example of Complementary PWM Mode 0% and 100% Waveform

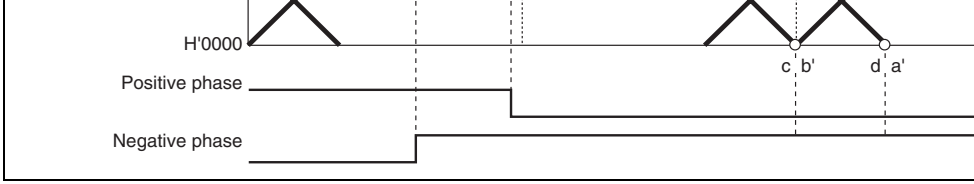


Figure 10.52 Example of Complementary PWM Mode 0% and 100% Waveform O

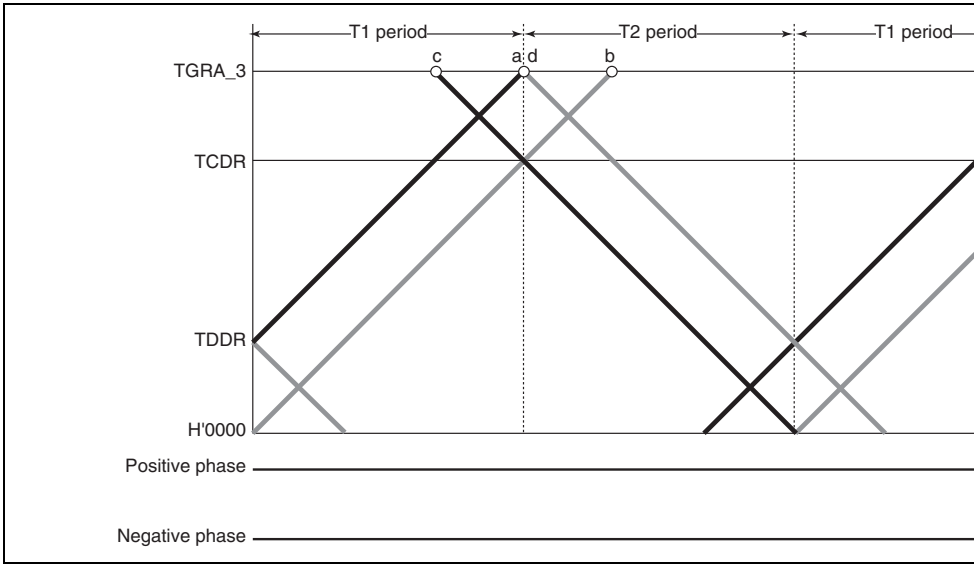


Figure 10.53 Example of Complementary PWM Mode 0% and 100% Waveform O

the waveform does not change.

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output can be performed in synchronization with carrier cycle by setting the PSYE bit to 1 in the timer output control register (TOCR). A of a toggle output waveform is shown in figure 10.54.

This output is toggled by a compare-match between TCNT_3 and TGRA_3 and a comp between TCNT4 and H'0000.

The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

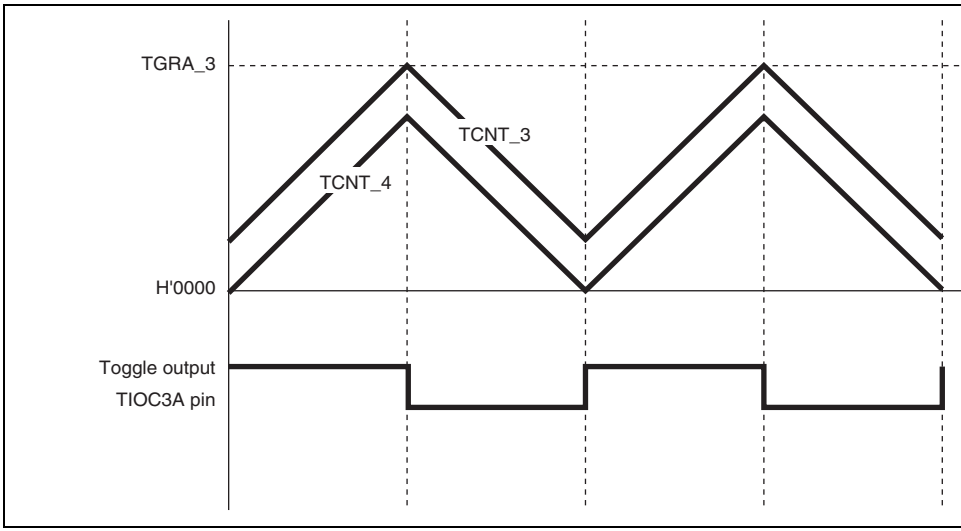


Figure 10.54 Example of Toggle Output Waveform Synchronized with PWM C

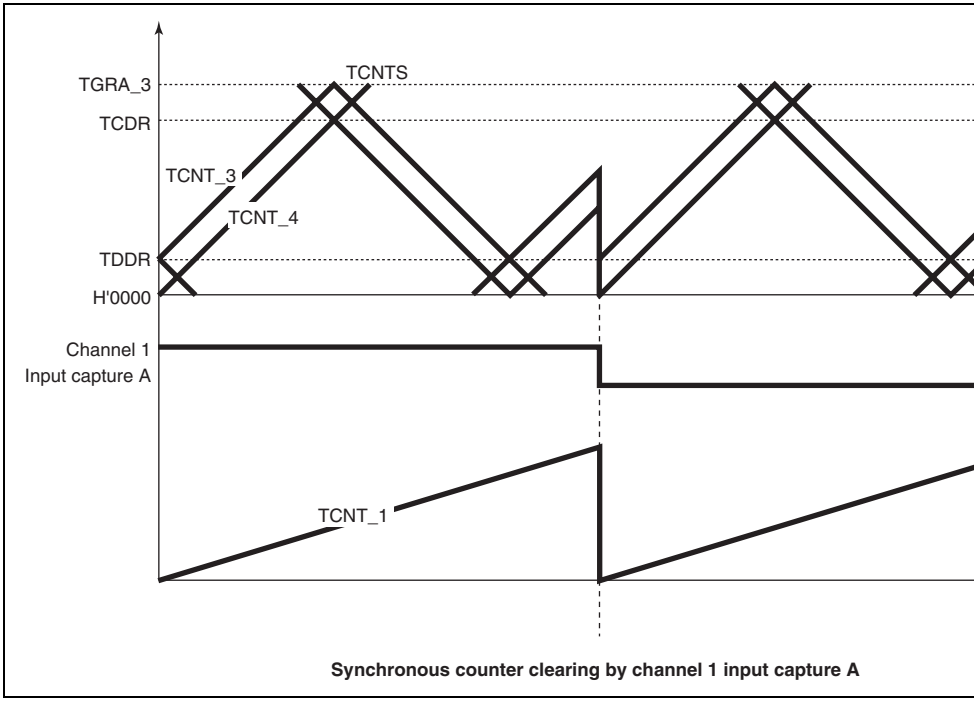


Figure 10.55 Counter Clearing Synchronized with Another Channel

by (1) in figure 10.56) immediately after the counter's start operation, initial value outpu
suppressed.

This function can be used in both the MTU2 and MTU2S. In the MTU2, synchronous cl
generated in channels 0 to 2 in the MTU2 can cause counter clearing in complementary
mode; in the MTU2S, compare match or input capture flag setting in channels 0 to 2 in t
can cause counter clearing.

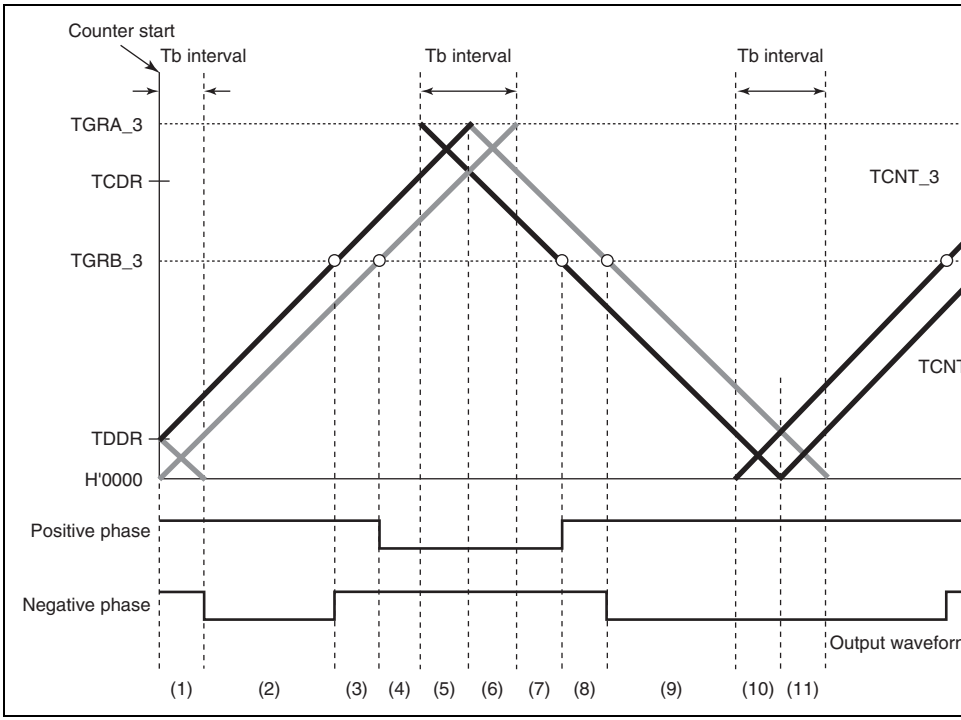
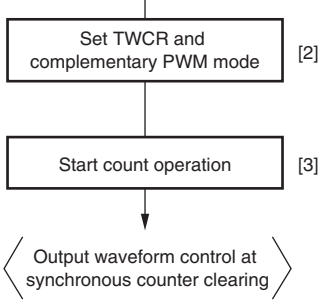


Figure 10.56 Timing for Synchronous Counter Clearing



to it to suppress initial value output at counter clearing.

[3] Set bits CST3 and CST4 in TSTR to 1 to start count operation.

Figure 10.57 Example of Procedure for Setting Output Waveform Control at Sync Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Compler PWM Mode

  Figures 10.58 to 10.61 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generate the WRE bit in TWCR is set to 1. In the examples shown in figures 10.58 to 10.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 1 respectively.

  In the MTU2S, these examples are equivalent to the cases when the MTU2S operates complementary PWM mode and synchronous counter clearing is generated while the is cleared to 0 and the WRE bit is set to 1 in TWCR.

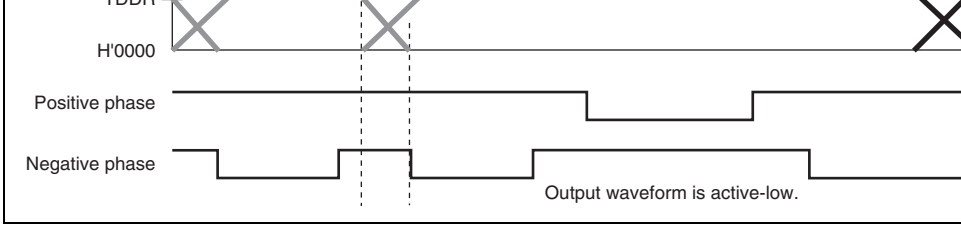


Figure 10.58 Example of Synchronous Clearing in Dead Time during Up-Cou
(Timing (3) in Figure 10.56; Bit WRE of TWCR in MTU2 is 1)

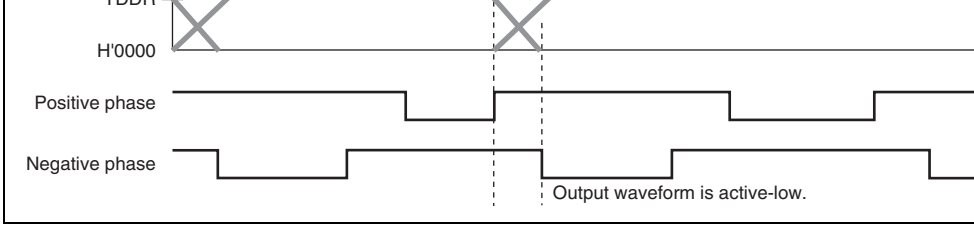


Figure 10.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 10.56; Bit WRE of TWCR in MTU2 is 1)

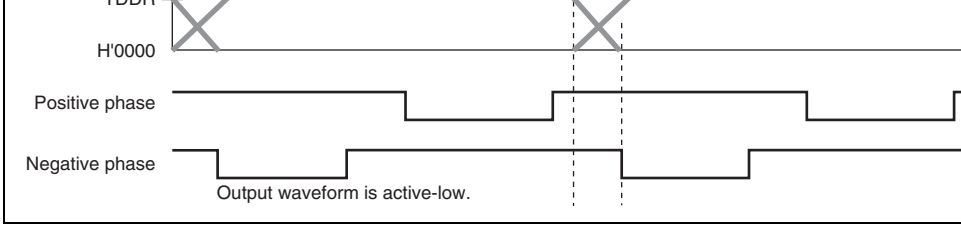


Figure 10.60 Example of Synchronous Clearing in Dead Time during Down-Co (Timing (8) in Figure 10.56; Bit WRE of TWCR is 1)

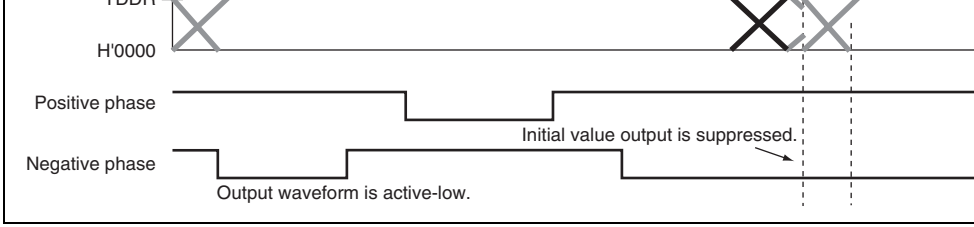


Figure 10.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 10.56; Bit WRE of TWCR is 1)

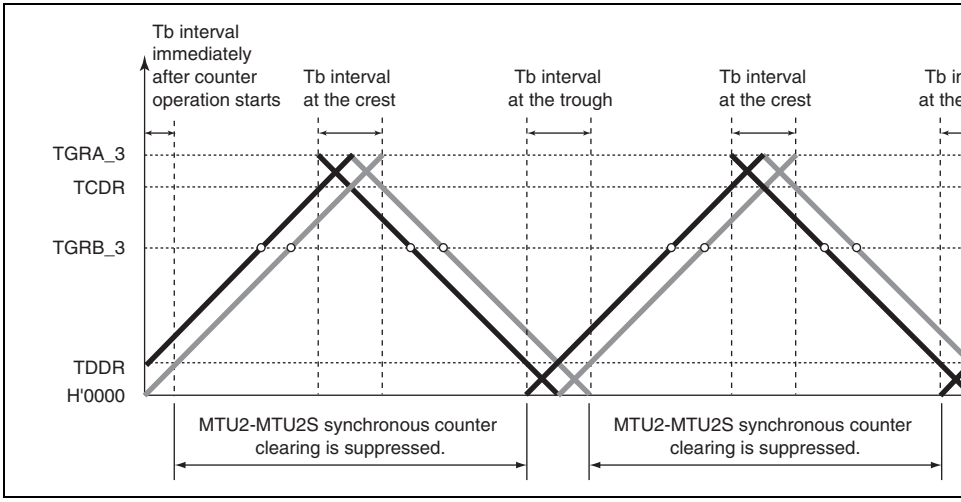


Figure 10.62 MTU2–MTU2S Synchronous Clearing-Suppressed Interval Specifie Bit in TWCR

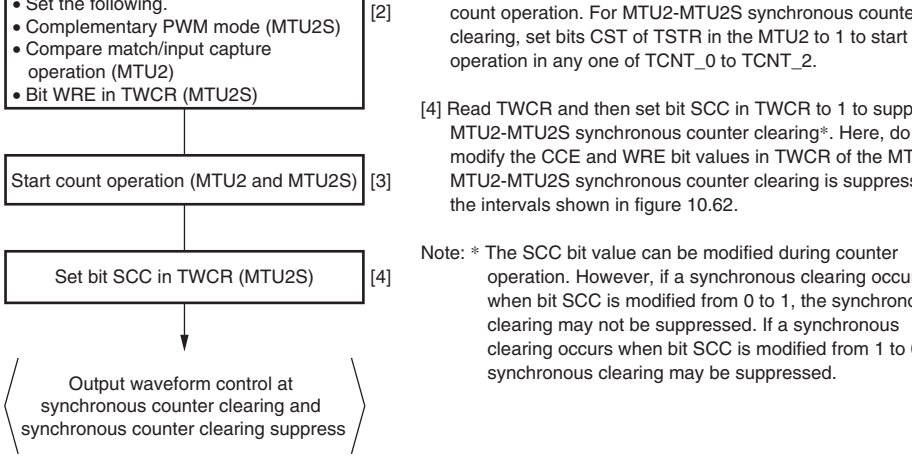


Figure 10.63 Example of Procedure for Suppressing MTU2–MTU2S Synchronous Clearing

- Examples of Suppression of MTU2–MTU2S Synchronous Counter Clearing

  Figures 10.64 to 10.67 show examples of operation in which the MTU2S operates in complementary PWM mode and MTU2–MTU2S synchronous counter clearing is sup by setting the SCC bit in TWCR in the MTU2S to 1. In the examples shown in figure 10.67, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in f 10.56, respectively.

  In these examples, the WRE bit in TWCR of the MTU2S is set to 1.

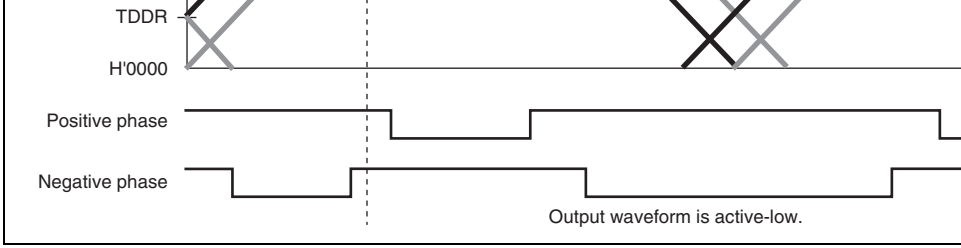


Figure 10.64 Example of Synchronous Clearing in Dead Time during Up-Cou
(Timing (3) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU

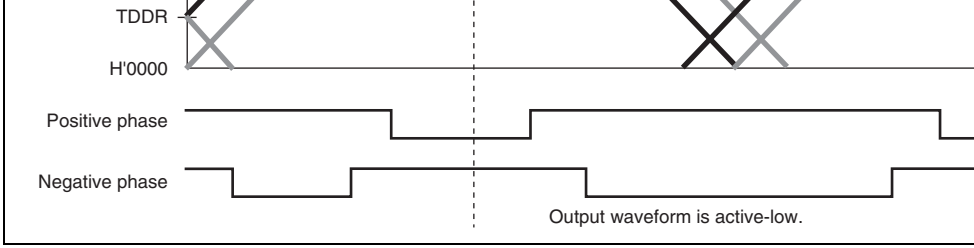


Figure 10.65 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU

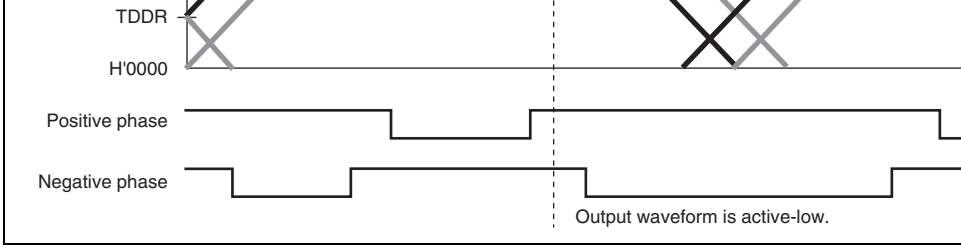


Figure 10.66 Example of Synchronous Clearing in Dead Time during Down-Co
(Timing (8) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU

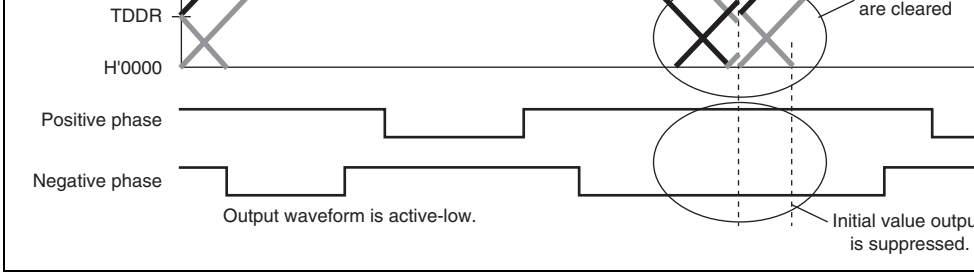


Figure 10.67 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU

(TSYCR) to 1).

3. Do not set the PWM duty value to H'0000.
4. Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

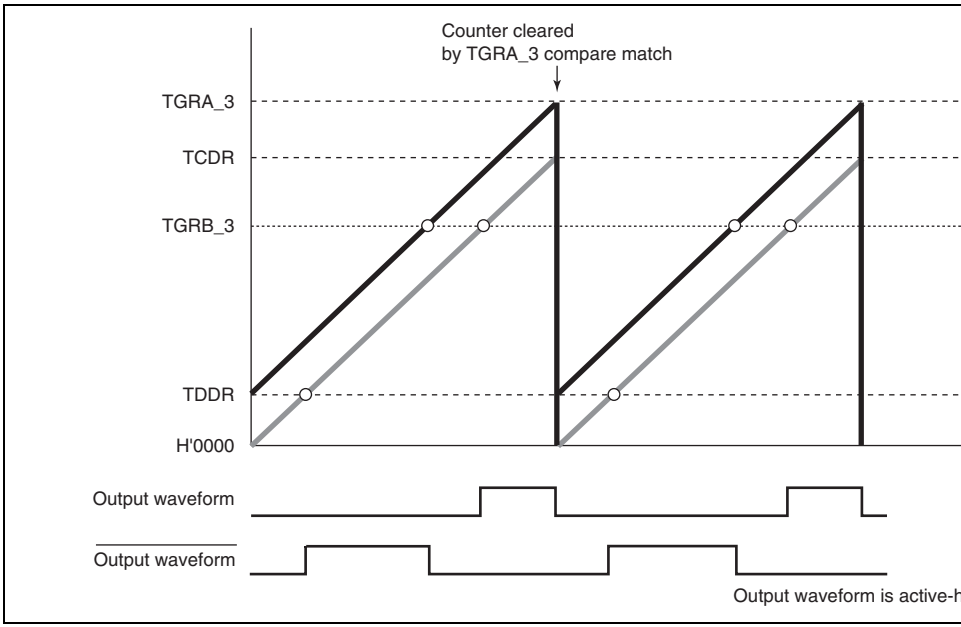


Figure 10.68 Example of Counter Clearing Operation by TGRA_3 Compare I

When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in T
cleared to 0 or set to 1.

The drive waveforms are output from the complementary PWM mode 6-phase output pin
this 6-phase output, in the case of on output, it is possible to use complementary PWM m
output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P
level output is selected.

The 6-phase output active level (on output level) can be set with the OLSN and OLSP bit
timer output control register (TOCR) regardless of the setting of the N and P bits.

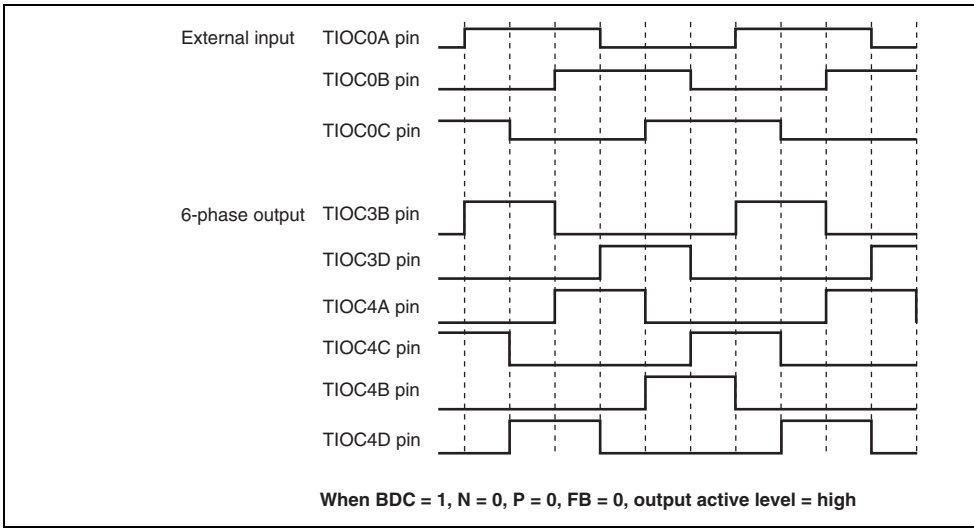


Figure 10.69 Example of Output Phase Switching by External Input (1)

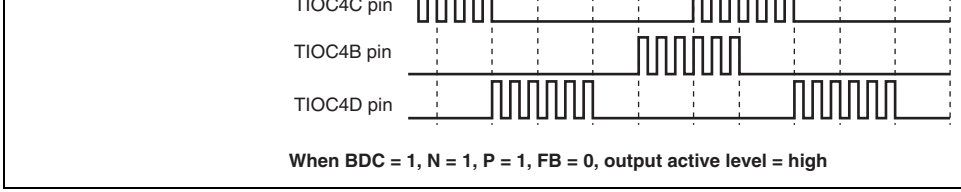


Figure 10.70 Example of Output Phase Switching by External Input (2)

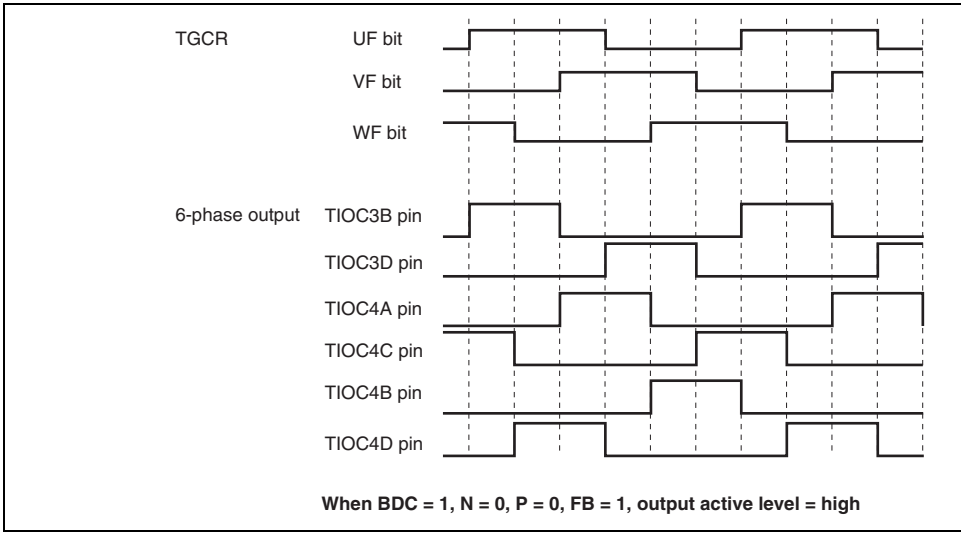


Figure 10.71 Example of Output Phase Switching by Means of UF, VF, WF Bit S

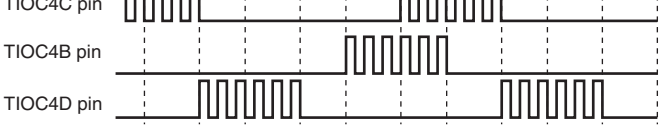


Figure 10.72 Example of Output Phase Switching by Means of UF, VF, WF Bit Set

(r) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using a TGR compare-match, TCNT_4 underflow (trough), or compare-match on a channel other than 3 and 4.

When start requests using a TGRA_3 compare-match are specified, A/D conversion can b at the crest of the TCNT_3 count.

A/D converter start requests can be set by setting the TTGE bit to 1 in the timer interrupt register (TIER). To issue an A/D converter start request at a TCNT_4 underflow (trough) TTGE2 bit in TIER_4 to 1.

converter request control register (TADCR). For the linkage with the A/D converter star
delaying function, refer to section 10.4.9, A/D Converter Start Request Delaying Functi

The setting of the timer interrupt skipping setting register (TITCR) must be done while t
TGIA_3 and TCIV_4 interrupt requests are disabled by the settings of TIER_3 and TIE
with under the conditions in which TGFA_3 and TCFV_4 flag settings by compare mat
occur. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits
clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 10.73 shows an example of the interrupt skipping operation setting procedure. Fi
shows the periods during which interrupt skipping count can be changed.

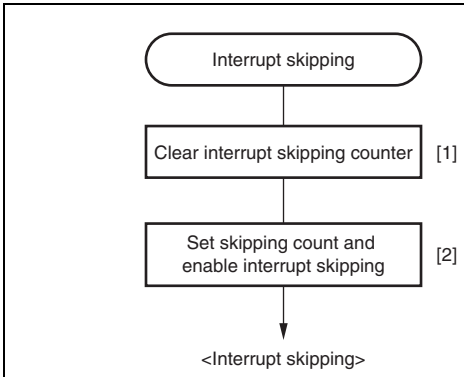


Figure 10.73 Example of Interrupt Skipping Operation Setting Procedur

Figure 10.74 Periods during which Interrupt Skipping Count can be Chang

(b) Example of Interrupt Skipping Operation

Figure 10.75 shows an example of TGIA_3 interrupt skipping in which the interrupt skip count is set to three by the 3ACOR bit and the T3AEN bit is set to 1 in the timer interrupt set register (TITCR).

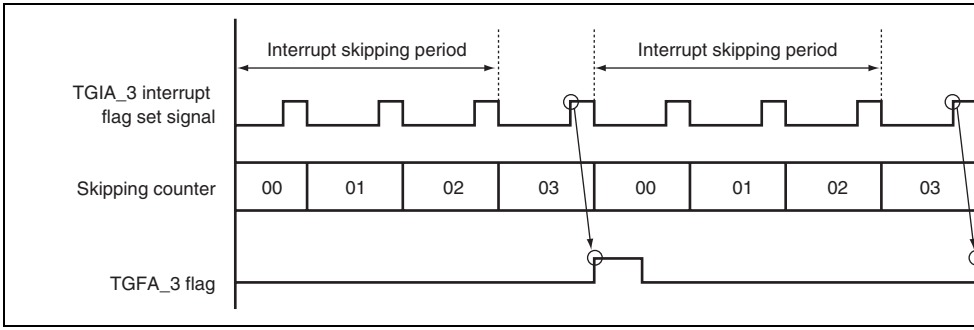


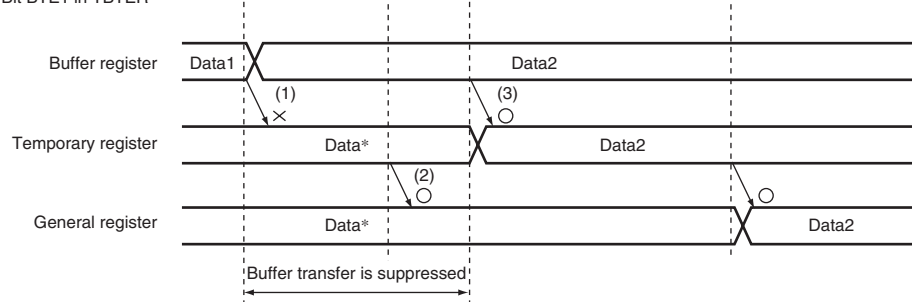
Figure 10.75 Example of Interrupt Skipping Operation

register outside the buffer transfer-enabled period.

There are two types of timing in which data is transferred from the buffer register to the register or to general register, depending on the buffer register modification timing after interrupt occurrence.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit sett timer interrupt skipping set register (TITCR). Figure 10.78 shows the relationship betwe T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer in skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3A 4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TI 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is buffer transfer is never performed.

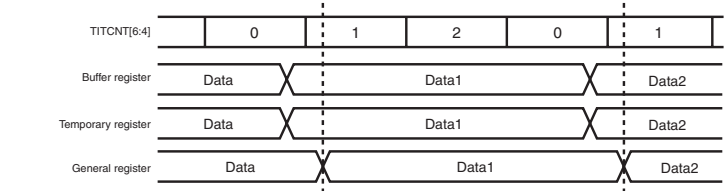


[Legend]

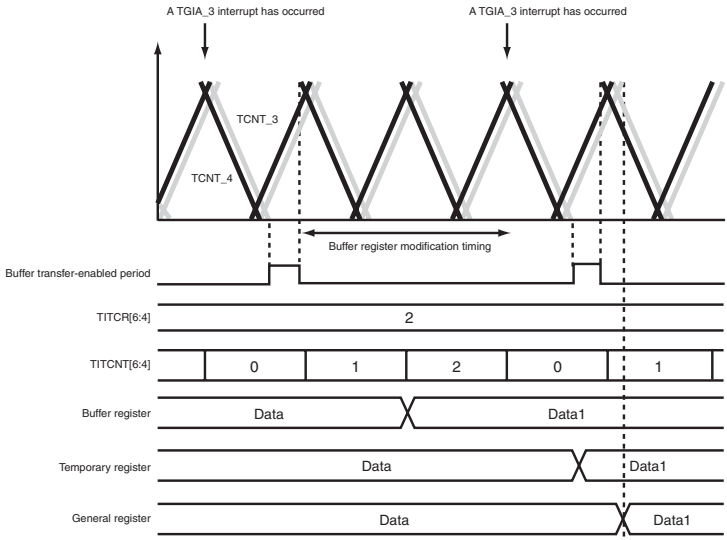
(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).
(2) Data is transferred from the temporary register to the general register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note: * When buffer transfer at the crest is selected.

Figure 10.76 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)



(2) When the buffer register is modified after one carrier cycle has been passed from a TGIA_3 interrupt occurrence



Note: MD[3:0] in TMDR_3 = 1101
Buffer transfer at the crest is selected.
The skipping count is set to two.
T3AEN and T4VEN are set to 1 and cleared to 0, respectively.

Figure 10.77 Example of Operation when Buffer Transfer is Linked with Inte Skipping (BTE1 = 1 and BTE0 = 0)

Note: MD[3:0] in TMDR_3 = 1111
Buffer transfer at the crest and trough is selected.
The skipping count is set to three.
T3AEN and T4VEN are set to 1.

Figure 10.78 Relationship between Bits T3AEN and T4VEN in TITCR and Bu Transfer-Enabled Period

- TCR_3 and TCR_4, TMDR_3 and TMDR_4, TIORH_3 and TIORH_4, TIORL_3 a
  TIORL_4, TIER_3 and TIER_4, TCNT_3 and TCNT_4, TGRA_3 and TGRA_4, TC
  and TGRB_4, TOER, TOCR, TGCR, TCDR, and TDDR.

This function enables miswriting due to CPU runaway to be prevented by disabling CPU
the mode registers, control registers, and counters. When the applicable registers are rea
access-disabled state, undefined values are returned. Writing to these registers is ignored

(b) Halting of PWM output by external signal

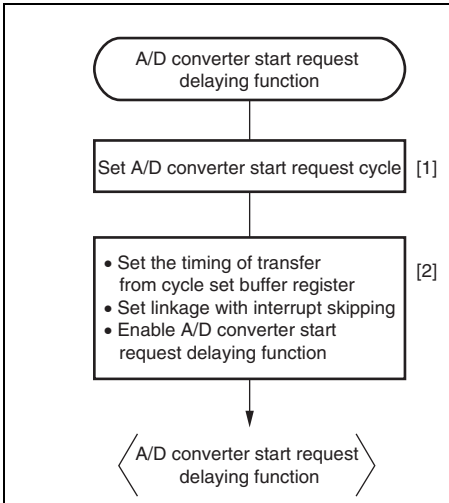
The 6-phase PWM output pins can be set automatically to the high-impedance state by i
specified external signals. There are four external signal input pins.

See section 12, Port Output Enable 2 (POE2), for details.

A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination wi
interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bi
TADCR.

- Example of Procedure for Specifying A/D Converter Start Request Delaying Functio

  Figure 10.79 shows an example of procedure for specifying the A/D converter start re
  delaying function.



[1] Set the cycle in the timer A/D converter start request cycl
buffer register (TADCOBRA_4 or TADCOBRB_4) and tim
A/D converter start request cycle register (TADCORA_4
TADCORB_4). (The same initial value must be specified
the cycle buffer register and cycle register.)

[2] Use bits BF1 and BF2 in the timer A/D converter start request control register (TADCR) to specify the timing of transfer from the timer A/D converter start request cycle buffer register to A/D converter start request cycle registe
- Specify whether to link with interrupt skipping through b ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits TU4AE, DT4AE, UT4BE, and DT4BE to enabl A/D conversion start requests (TRG4AN or TRG4BN).

Notes: 1. Perform TADCR setting while TCNT_4 is stopped.
2. Do not set BF1 to 1 when complementary PWM mo is not selected.
3. Do not set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, or DT4BE to 1 when complementary PWM mode is not selected.

Figure 10.79 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

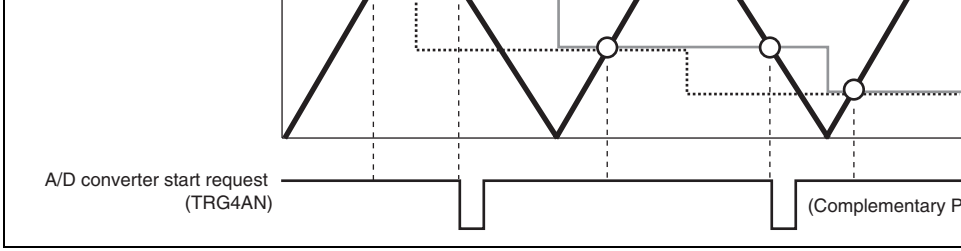


Figure 10.80 Basic Example of A/D Converter Start Request Signal (TRG4AN) C

- Buffer Transfer

  The data in the timer A/D converter start request cycle set registers (TADCORA_4 a TADCORB_4) is updated by writing data to the timer A/D converter start request cy buffer registers (TADCOBRA_4 and TADCOBRB_4). Data is transferred from the registers to the respective cycle set registers at the timing selected with the BF1 and in the timer A/D converter start request control register (TADCR_4).

- A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

  A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4V the timer A/D converter start request control register (TADCR).

  Figure 10.81 shows an example of A/D converter start request signal (TRG4AN) op when TRG4AN output is enabled during TCNT_4 up-counting and down-counting a converter start requests are linked with interrupt skipping.

  Figure 10.82 shows another example of A/D converter start request signal (TRG4AN operation when TRG4AN output is enabled during TCNT_4 up-counting and A/D c start requests are linked with interrupt skipping.

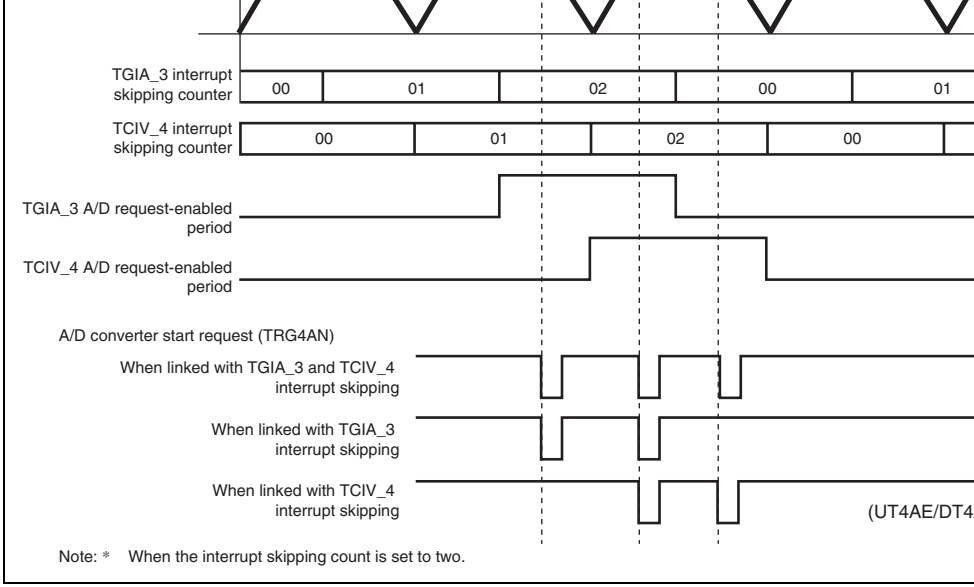


Figure 10.81 Example of A/D Converter Start Request Signal (TRG4AN) Operatio with Interrupt Skipping

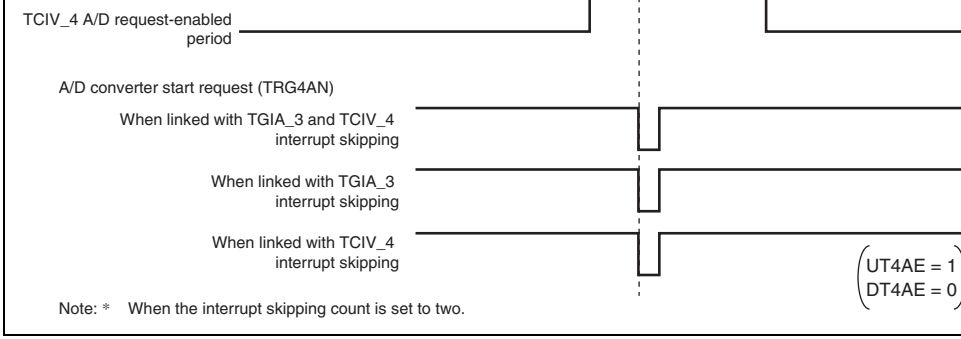
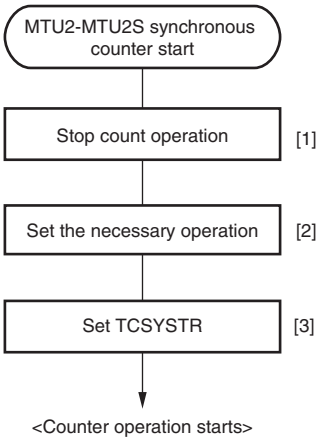


Figure 10.82 Example of A/D Converter Start Request Signal (TRG4AN) Operati with Interrupt Skipping



[1] Use TSTR registers in the MTU2 and MTU2S and halt the counters used for synchronous start operation.

[2] Specify necessary operation with appropriate registers such TCR and TMDR.

[3] In TCSYSTR in the MTU2, set the bits corresponding to the counters to be started synchronously to 1. The TSTRs are automatically set appropriately and the counters start synchronously.

Notes: 1. Even if a bit in TCSYSTR corresponding to an operati counter is cleared to 0, the counter will not stop. To st the counter, clear the corresponding bit in TSTR to 0 directly.

2. To start channels 3 and 4 in reset-synchronized PWM mode or complementary PWM mode, make appropria settings in TCYSTR according to the TSTR setting for the respective mode. For details, refer to section 10.4 Reset-Synchronized PWM Mode, and section 10.4.8, Complementary PWM Mode.

Figure 10.83 Example of Synchronous Counter Start Setting Procedure

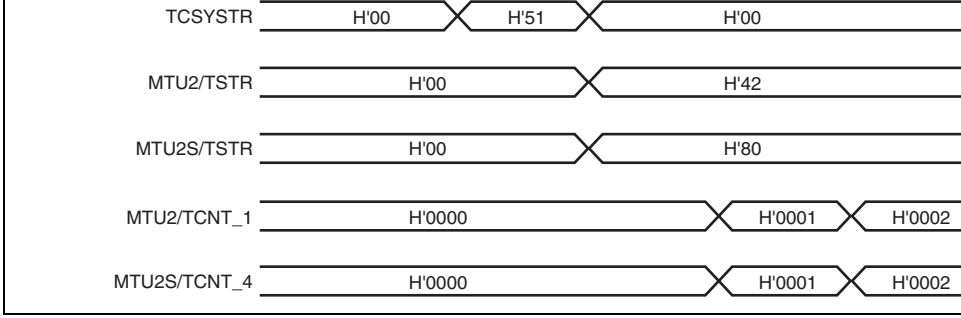


Figure 10.84 (1) Example of Synchronous Counter Start Operation (MTU2-to-M Clock Frequency Ratio = 1:1)

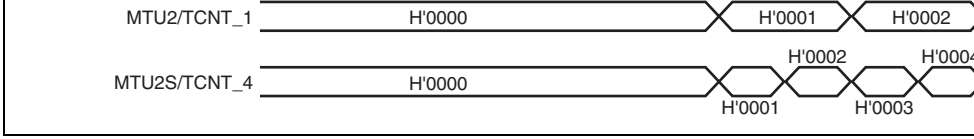


Figure 10.84 (2) Example of Synchronous Counter Start Operation (MTU2-to-M Clock Frequency Ratio = 1:2)

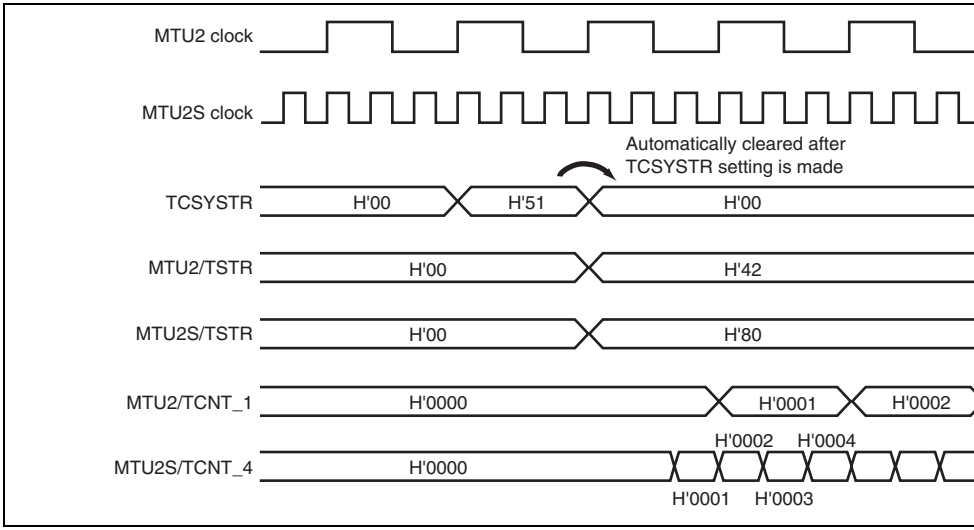


Figure 10.84 (3) Example of Synchronous Counter Start Operation (MTU2-to-M Clock Frequency Ratio = 1:3)

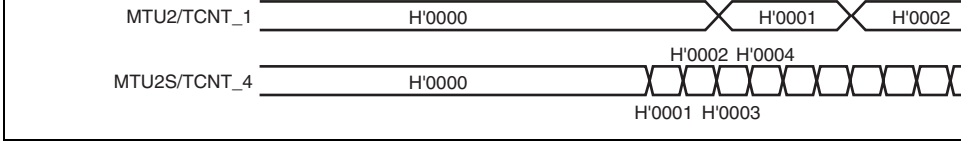
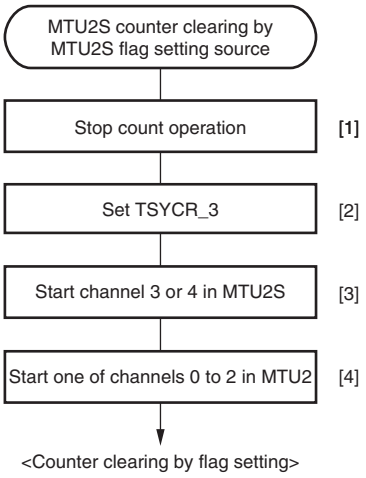


Figure 10.84 (4) Example of Synchronous Counter Start Operation (MTU2-to-M Clock Frequency Ratio = 1:4)



[1] Use TSTR registers in the MTU2 and MTU2S and halt the counters used for this function.

[2] Use TSYCR_3 in the MTU2S to specify the flag setting sou to be used for the TCNT_3 and TCNT_4 clearing source.

[3] Start TCNT_3 or TCNT_4 in the MTU2S.

[4] Start TCNT_0, TCNT_1, or TCNT_2 in the MTU2.

Note: The TSYCR_3 setting is ignored while the counter is stopped. The setting becomes valid after TCNT_3 or TCNT4 is started.

Figure 10.85 Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

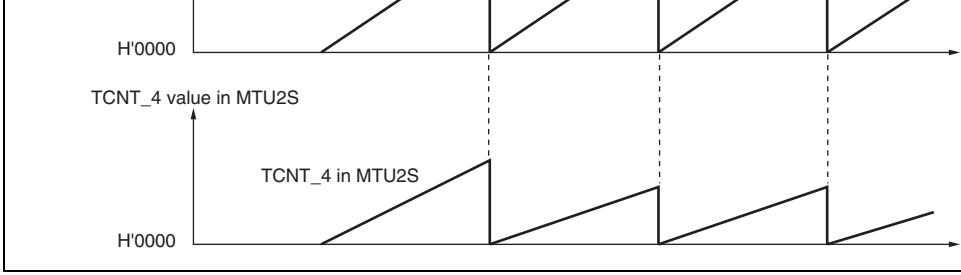


Figure 10.86 (1) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (1)

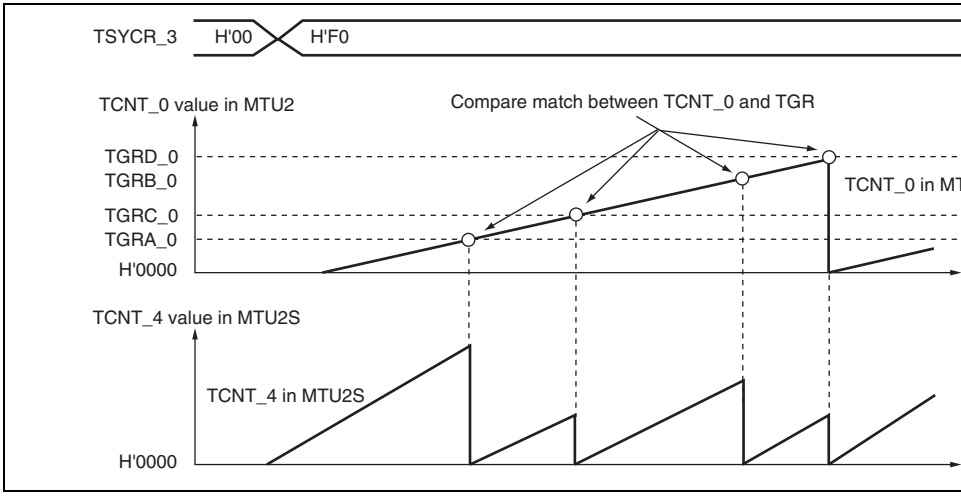
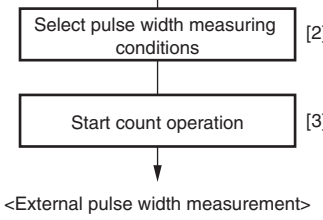


Figure 10.86 (2) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (2)



Notes: 1. Do not set bits CMPCLR5U, CMPCLR5V, or CMPCLR5W in TCNTCMPCLR to 1.
2. Do not set bits TGIE5U, TGIE5V, or TGIE5W in TIER_5 to 1.
3. The value in TCNT is not captured in TGR.

Figure 10.87 Example of External Pulse Width Measurement Setting Proced

(2) Example of External Pulse Width Measurement

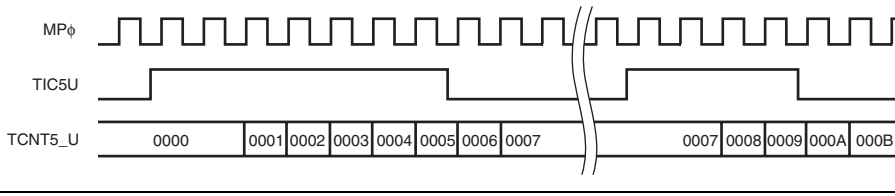


Figure 10.88 Example of External Pulse Width Measurement (Measuring High Pulse Width)

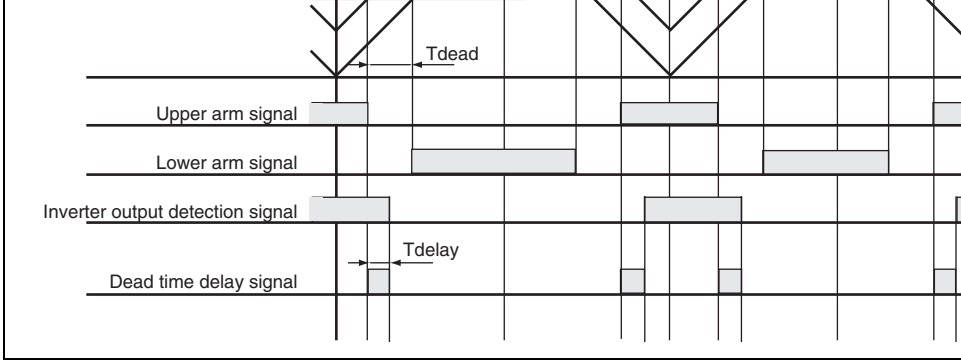
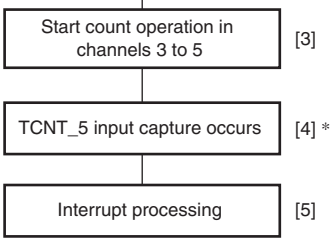


Figure 10.89 Delay in Dead Time in Complementary PWM Operation



and CST5W in TSTR2 to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the TCNT_5 value is captured in TGR_5.

[5] For U-phase dead time compensation, when an interrupt generated at the crest (TGIA_3) or trough (TCIV_4) in complementary PWM mode, read the TGRU_5 value, calculate the difference in time in TGRB_3, and write the corrected value to TGRD_3 in the interrupt processing. For the V phase and W phase, read the TGRV_5 and TGRW_5 values and write the corrected values to TGRC and TGRD_4, respectively, in the same way as for U-pha compensation.
The TCNT_5 value should be cleared through the TCNTCMPCLR setting or by software.

Notes: The PFC settings must be completed in advance.
* As an interrupt flag is set under the capture conditio specified in TIOR, do not enable interrupt requests i TIER_5.

Figure 10.90 Example of Dead Time Compensation Setting Procedure

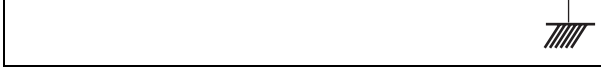


Figure 10.91 Example of Motor Control Circuit Configuration

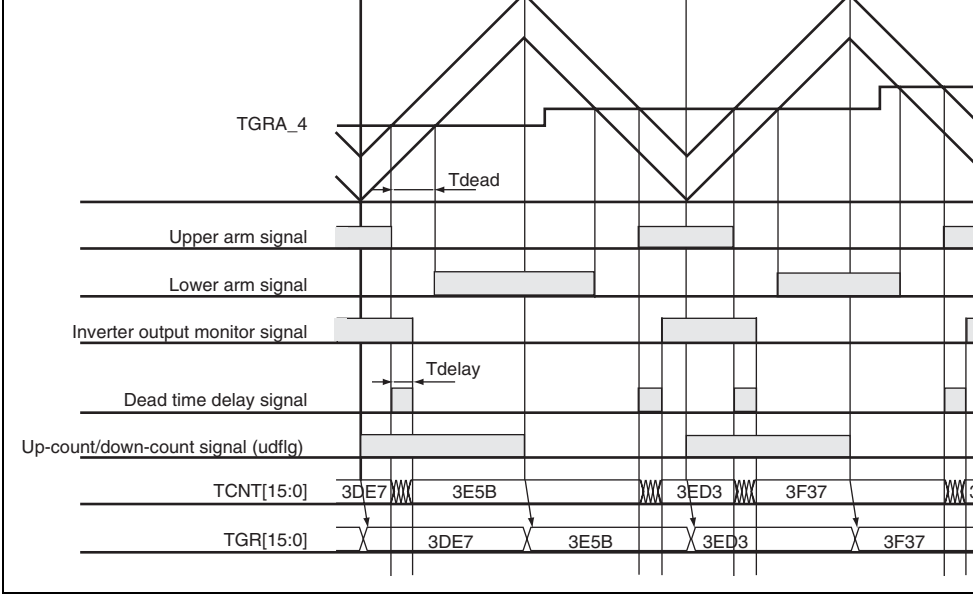


Figure 10.92 TCNT Capturing at Crest and/or Trough in Complementary PWM C

interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the prior
within a channel is fixed. For details, see section 6, Interrupt Controller (INTC).

Table 10.57 lists the MTU2 interrupt sources.

	TGIF_0	TGRF_0 compare match	TGFF_0	Not possible
1	TGIA_1	TGRA_1 input capture/compare match	TGFA_1	Possible
	TGIB_1	TGRB_1 input capture/compare match	TGFB_1	Not possible
	TCIV_1	TCNT_1 overflow	TCFV_1	Not possible
	TCIU_1	TCNT_1 underflow	TCFU_1	Not possible
2	TGIA_2	TGRA_2 input capture/compare match	TGFA_2	Possible
	TGIB_2	TGRB_2 input capture/compare match	TGFB_2	Not possible
	TCIV_2	TCNT_2 overflow	TCFV_2	Not possible
	TCIU_2	TCNT_2 underflow	TCFU_2	Not possible
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible
	TGIB_3	TGRB_3 input capture/compare match	TGFB_3	Not possible
	TGIC_3	TGRC_3 input capture/compare match	TGFC_3	Not possible
	TGID_3	TGRD_3 input capture/compare match	TGFD_3	Not possible
	TCIV_3	TCNT_3 overflow	TCFV_3	Not possible
4	TGIA_4	TGRA_4 input capture/compare match	TGFA_4	Possible
	TGIB_4	TGRB_4 input capture/compare match	TGFB_4	Not possible
	TGIC_4	TGRC_4 input capture/compare match	TGFC_4	Not possible
	TGID_4	TGRD_4 input capture/compare match	TGFD_4	Not possible
	TCIV_4	TCNT_4 overflow/underflow	TCFV_4	Not possible
5	TGIU_5	TGRU_5 input capture/compare match	TGFU_5	Not possible
	TGIV_5	TGRV_5 input capture/compare match	TGFV_5	Not possible
	TGIW_5	TGRW_5 input capture/compare match	TGFW_5	Not possible

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

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TS
1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by
the TCFV flag to 0. The MTU2 has five overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TS
1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared b
the TCFU flag to 0. The MTU2 has two underflow interrupts, one each for channels 1 a

10.5.2 DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in eac
For details, see section 9, Direct Memory Access Controller (DMAC).

In the MTU2, a total of five TGRA input capture/compare match interrupts can be used
activation sources, one each for channels 0 to 4.

H'0000).

A/D converter start request signal TRGAN is issued to the A/D converter under either on
following conditions.

- When the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/co
  match on a particular channel while the TTGE bit in TIER is set to 1
- When the TCNT_4 count reaches the trough (TCNT_4 = H'0000) during complement
  PWM operation while the TTGE2 bit in TIER_4 is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU2
selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Converter Activation by Compare Match between TCNT_0 and TGRE_0

The A/D converter can be activated by generating A/D converter start request signal TRG
when a compare match occurs between TCNT_0 and TGRE_0 in channel 0.

When the TGFE flag in TSR2_0 is set to 1 by the occurrence of a compare match betwee
TCNT_0 and TGRE_0 in channel 0 while the TTGE2 bit in TIER2_0 is set to 1, A/D con
start request TGR0N is issued to the A/D converter. If A/D converter start signal TGR0N
MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

Table 16.38 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	Interrupt Source	A/D Converter Start Signal
TGRA_0 and TCNT_0	Input capture/compare match	TRGAN
TGRA_1 and TCNT_1		
TGRA_2 and TCNT_2		
TGRA_3 and TCNT_3		
TGRA_4 and TCNT_4		
TCNT_4	TCNT_4 Trough in complementary PWM mode	
TGRE_0 and TCNT_0	Compare match	TRG0N
TADCORA and TCNT_4		TRG4AN
TADCORB and TCNT_4		TRG4BN

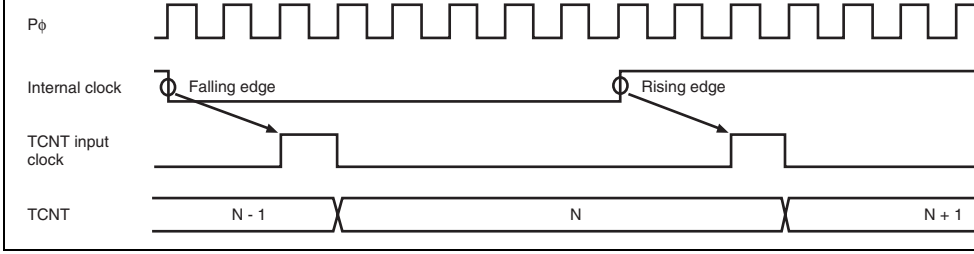


Figure 10.93 Count Timing in Internal Clock Operation (Channels 0 to 4)

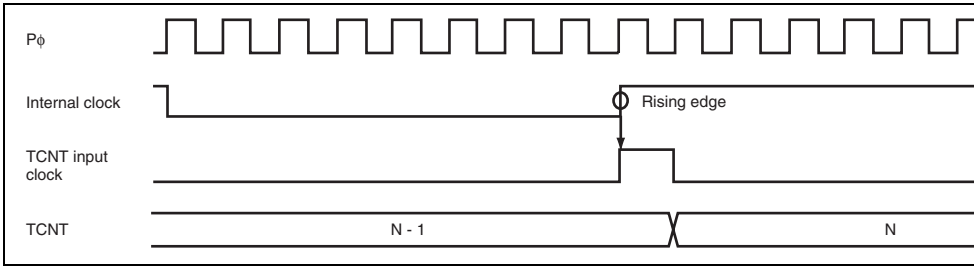


Figure 10.94 Count Timing in Internal Clock Operation (Channel 5)

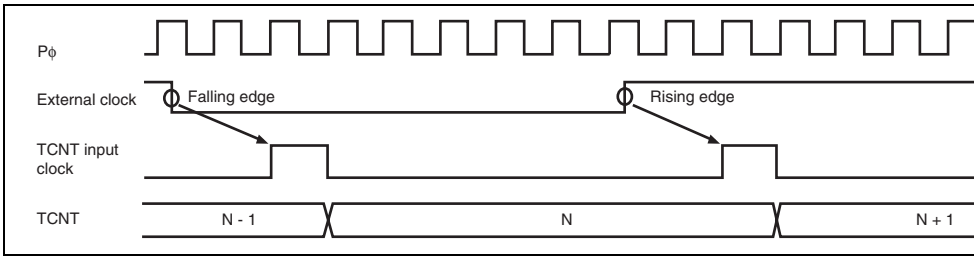


Figure 10.95 Count Timing in External Clock Operation (Channels 0 to 4)

A compare match signal is generated in the final state in which TCNT and TGR match (
at which the count value matched by TCNT is updated). When a compare match signal i
generated, the output value set in TIOR is output at the output compare output pin (TIO
After a match between TCNT and TGR, the compare match signal is not generated until
TCNT input clock is generated.

Figure 10.97 shows output compare output timing (normal mode and PWM mode) and f
10.98 shows output compare output timing (complementary PWM mode and reset syncl
PWM mode).

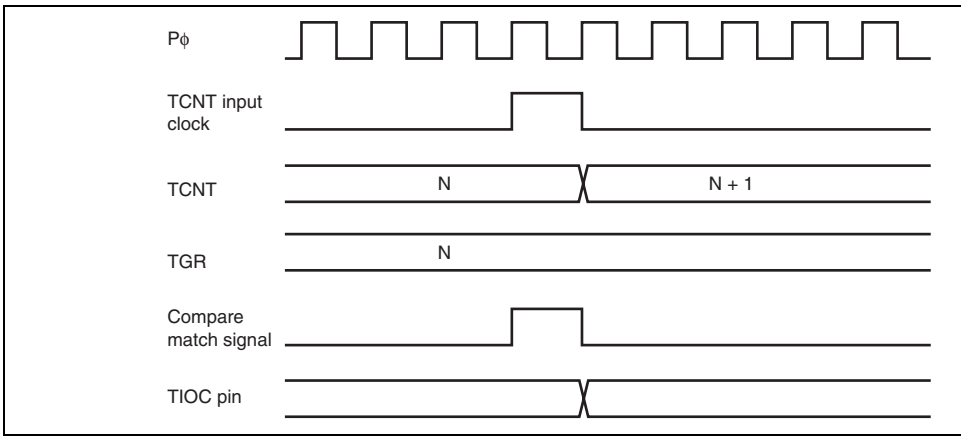


Figure 10.97 Output Compare Output Timing (Normal Mode/PWM Mod

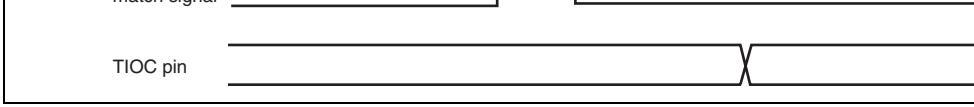


Figure 10.98 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 10.99 shows input capture signal timing.

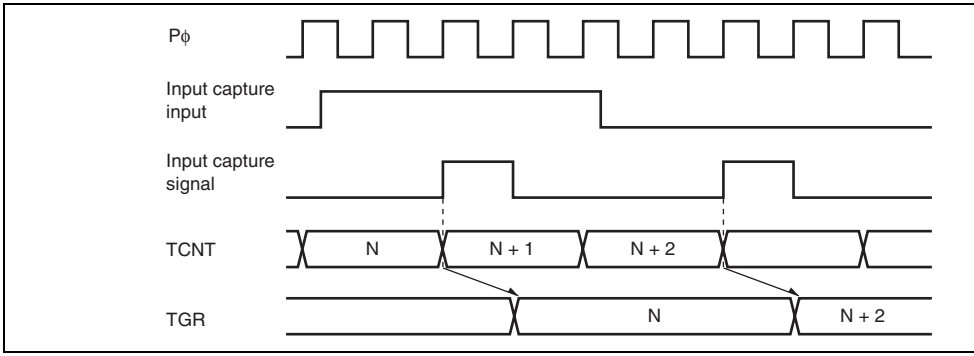


Figure 10.99 Input Capture Input Signal Timing

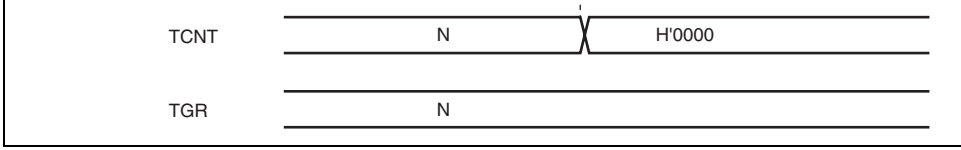


Figure 10.100 Counter Clear Timing (Compare Match) (Channels 0 to 4

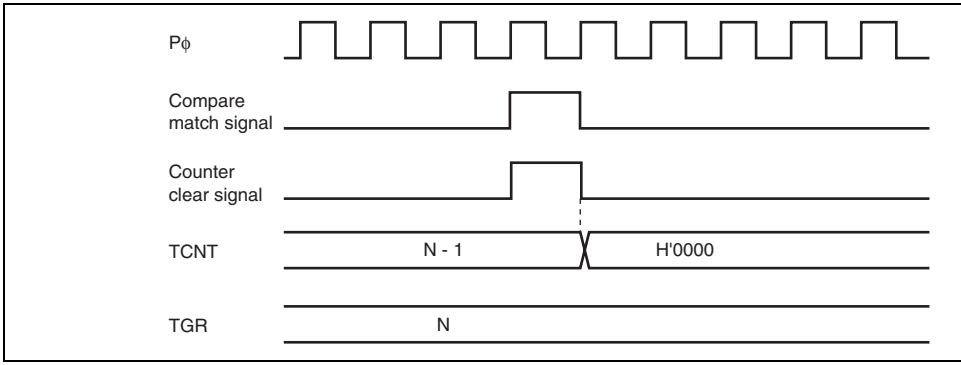


Figure 10.101 Counter Clear Timing (Compare Match) (Channel 5)

Figure 10.102 Counter Clear Timing (Input Capture) (Channels 0 to 5)

(5) Buffer Operation Timing

Figures 10.103 to 10.105 show the timing in buffer operation.

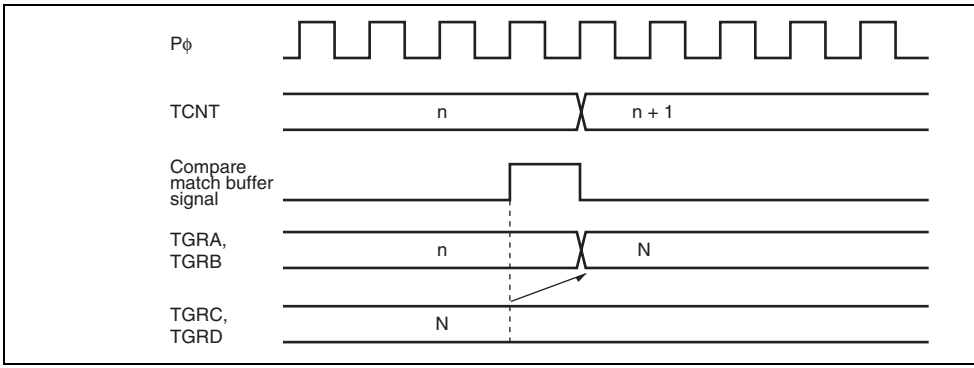


Figure 10.103 Buffer Operation Timing (Compare Match)

Figure 10.104 Buffer Operation Timing (Input Capture)

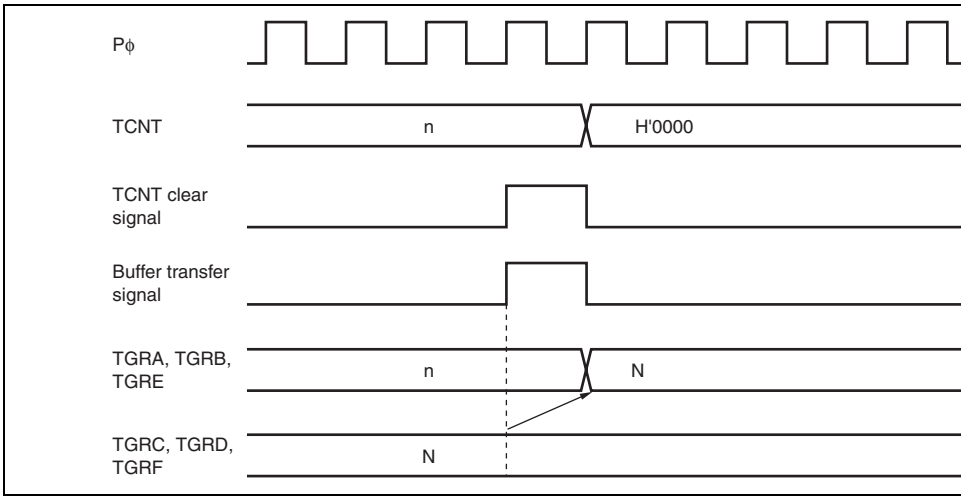


Figure 10.105 Buffer Transfer Timing (when TCNT Cleared)

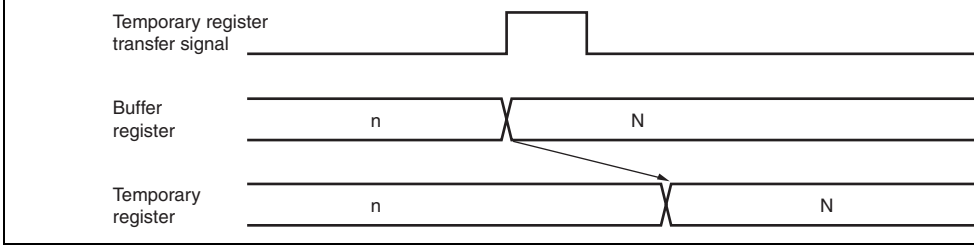


Figure 10.106 Transfer Timing from Buffer Register to Temporary Register (TCN

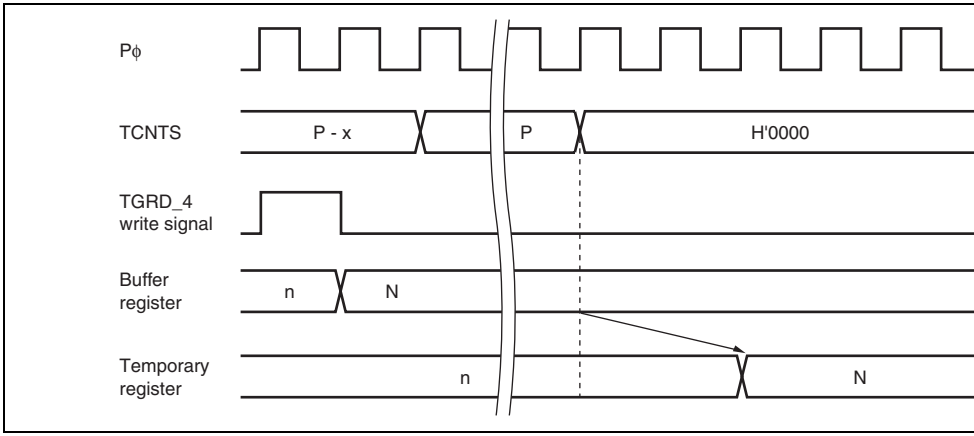


Figure 10.107 Transfer Timing from Buffer Register to Temporary Registe (TCNTS Operating)

Figure 10.108 Transfer Timing from Temporary Register to Compare Regi

10.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figures 10.109 and 110 show the timing for setting of the TGF flag in TSR on compare
and TGI interrupt request signal timing.

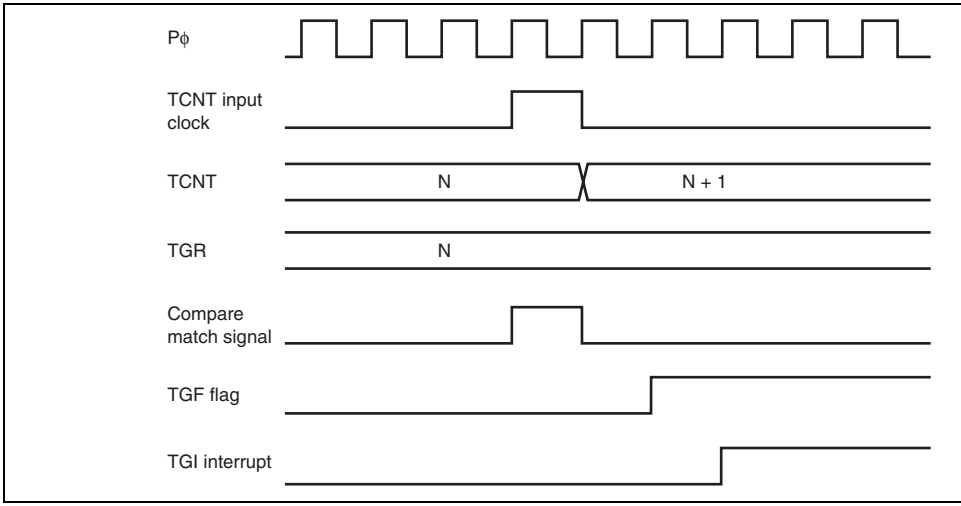


Figure 10.109 TGI Interrupt Timing (Compare Match)

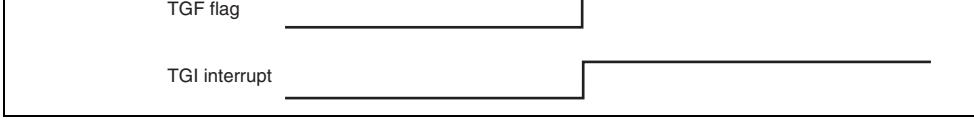


Figure 10.110 TGI Interrupt Timing (Compare Match) (Channel 5)

(2) TGF Flag Setting Timing in Case of Input Capture

Figures 10.111 and 112 show the timing for setting of the TGF flag in TSR on input captu
TGI interrupt request signal timing.

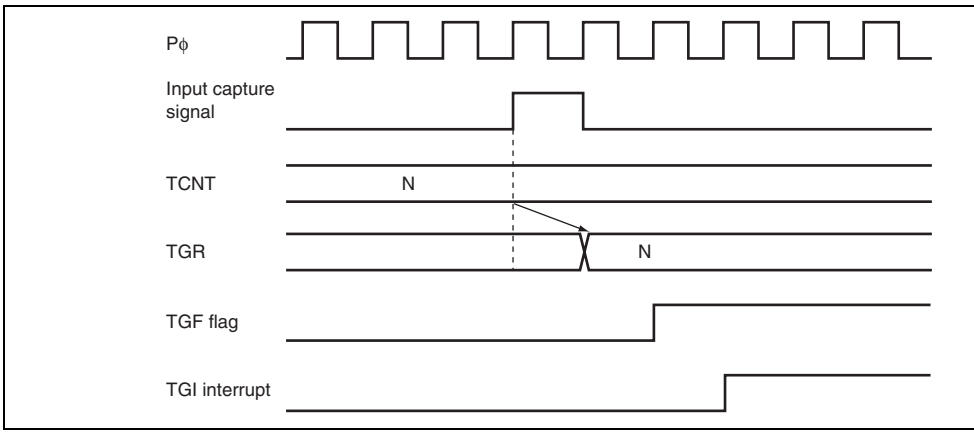


Figure 10.111 TGI Interrupt Timing (Input Capture) (Channels 0 to 4)

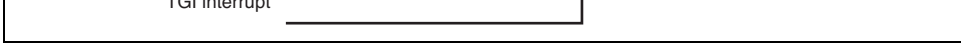


Figure 10.112 TGI Interrupt Timing (Input Capture) (Channel 5)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 10.113 shows the timing for setting of the TCFV flag in TSR on overflow, and T
interrupt request signal timing.

Figure 10.114 shows the timing for setting of the TCFU flag in TSR on underflow, and
interrupt request signal timing.

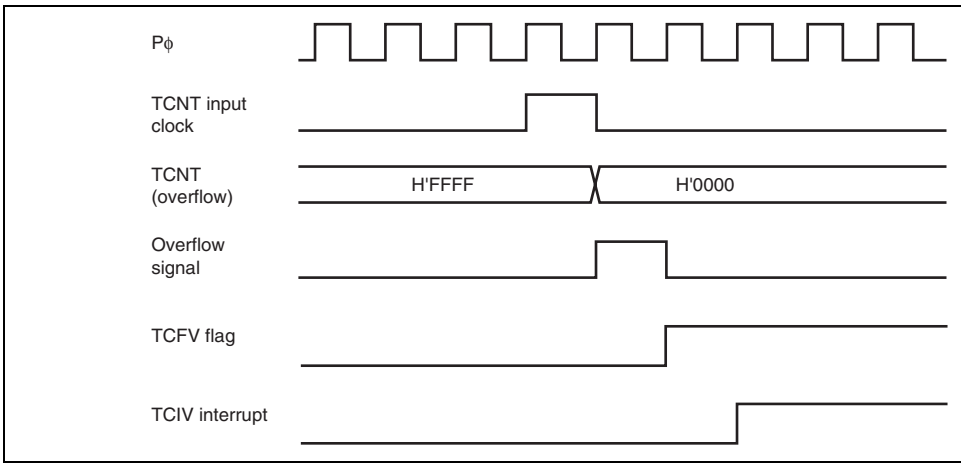


Figure 10.113 TCIV Interrupt Setting Timing

Figure 10.114 TCIU Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DMA activated, the flag is cleared automatically. Figures 10.115 and 116 show the timing for st clearing by the CPU, and figure 10.117 shows the timing for status flag clearing by the D

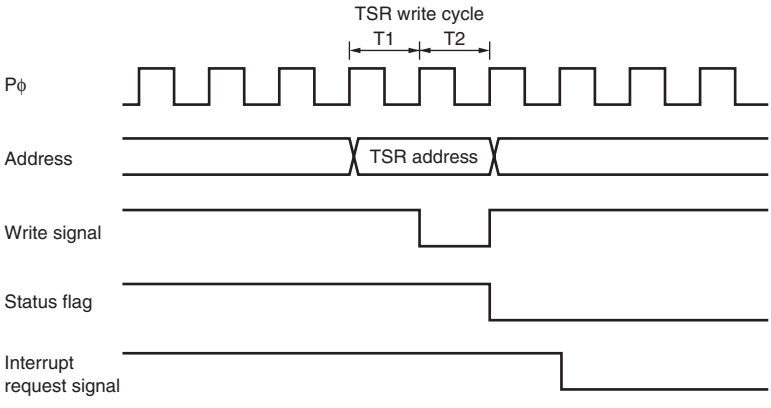


Figure 10.115 Timing for Status Flag Clearing by CPU (Channels 0 to 4)

request signal

Figure 10.116 Timing for Status Flag Clearing by CPU (Channel 5)

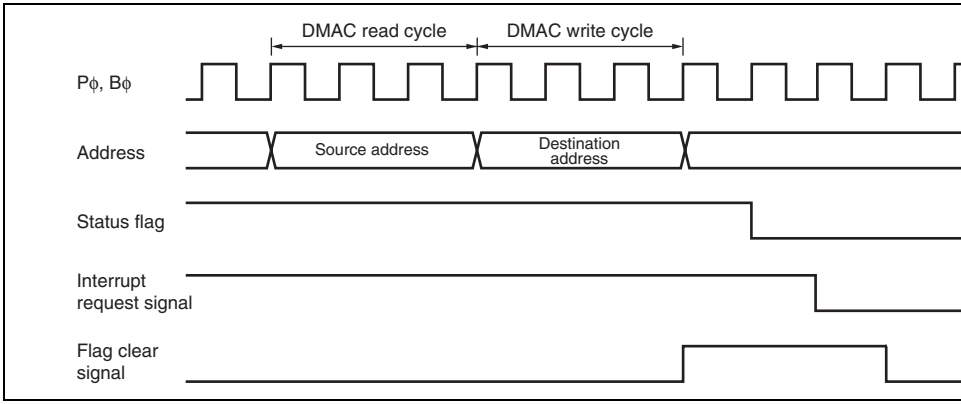


Figure 10.117 Timing for Status Flag Clearing by DTC Activation (Channels

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 MTU2 will not operate properly at pulse widths.

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 10.118 shows the in conditions in phase counting mode.

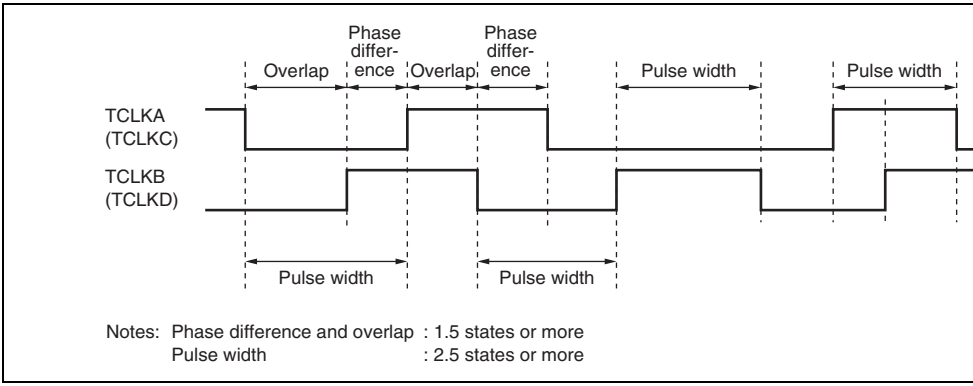


Figure 10.118 Phase Difference, Overlap, and Pulse Width in Phase Counting M

- Channel 5

$$f = \frac{P\phi}{N}$$

Where f: Counter frequency
Pϕ: Peripheral clock operating frequency
N: TGR set value

10.7.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clea
precedence and the TCNT write is not performed.

Figure 10.119 shows the timing in this case.

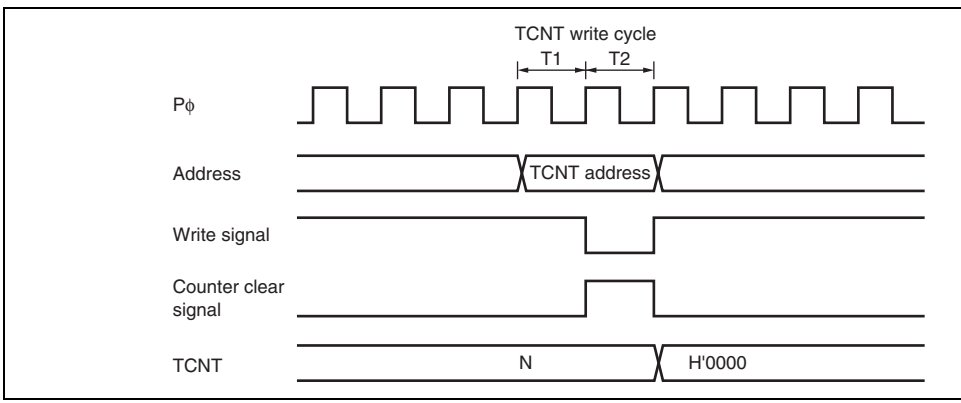


Figure 10.119 Contention between TCNT Write and Clear Operations

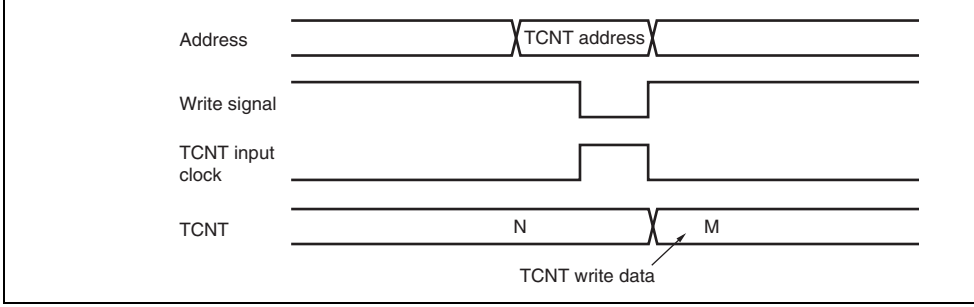


Figure 10.120 Contention between TCNT Write and Increment Operations

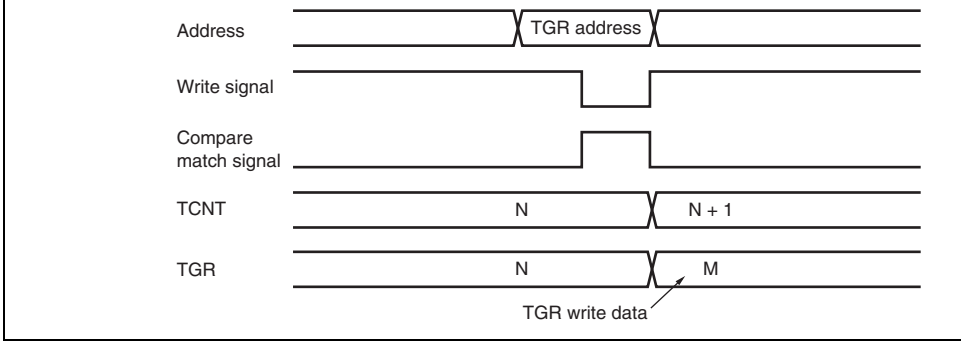


Figure 10.121 Contention between TGR Write and Compare Match

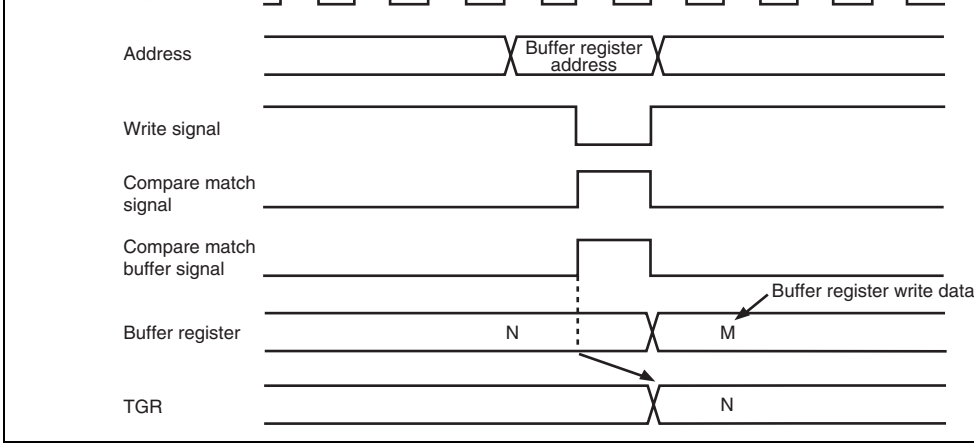


Figure 10.122 Contention between Buffer Register Write and Compare Mat

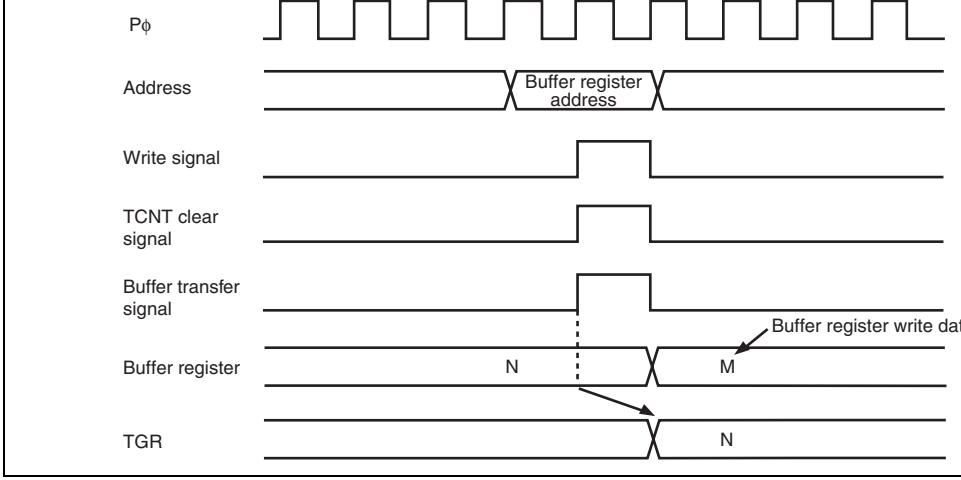


Figure 10.123 Contention between Buffer Register Write and TCNT Clea

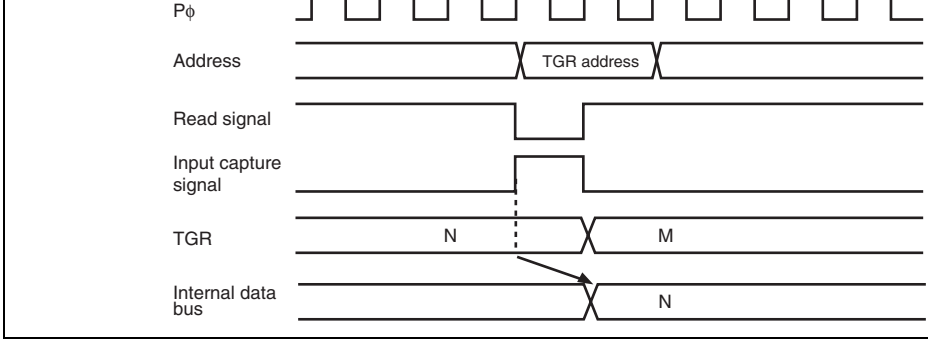


Figure 10.124 Contention between TGR Read and Input Capture (Channels 0

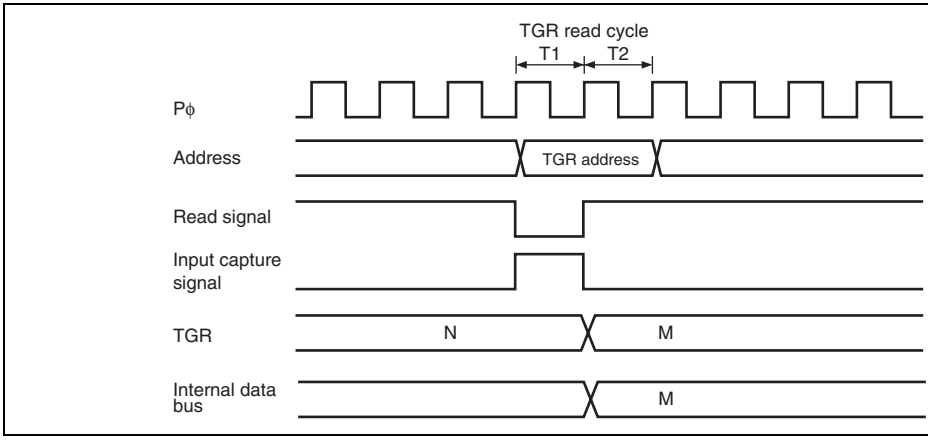


Figure 10.125 Contention between TGR Read and Input Capture (Channel

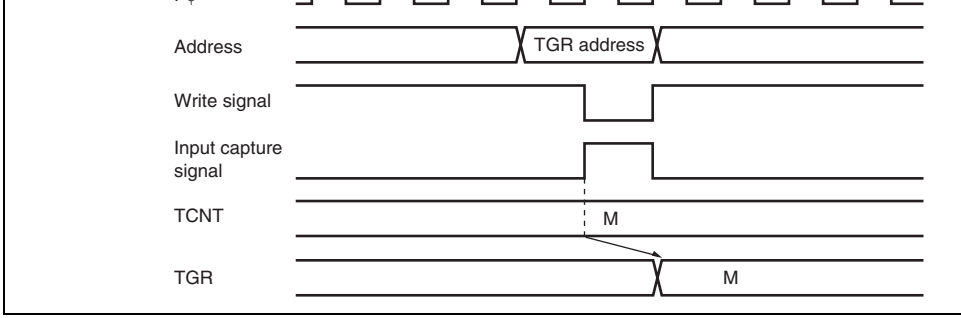


Figure 10.126 Contention between TGR Write and Input Capture (Channels

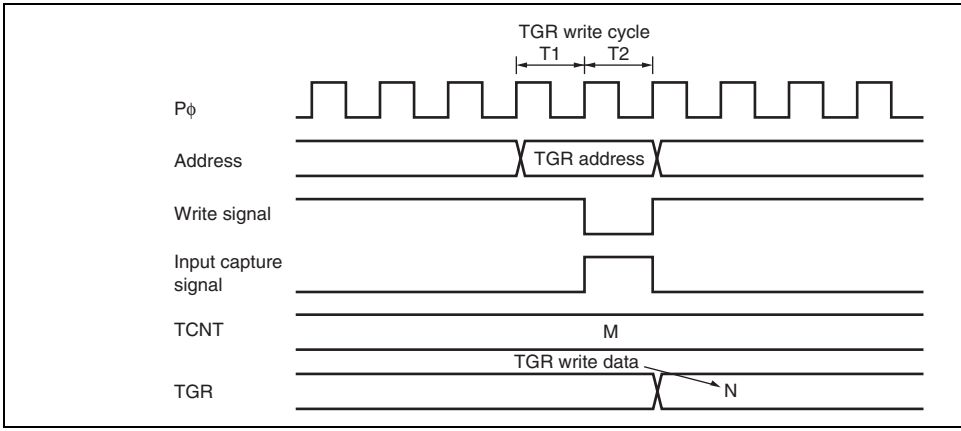


Figure 10.127 Contention between TGR Write and Input Capture (Channe

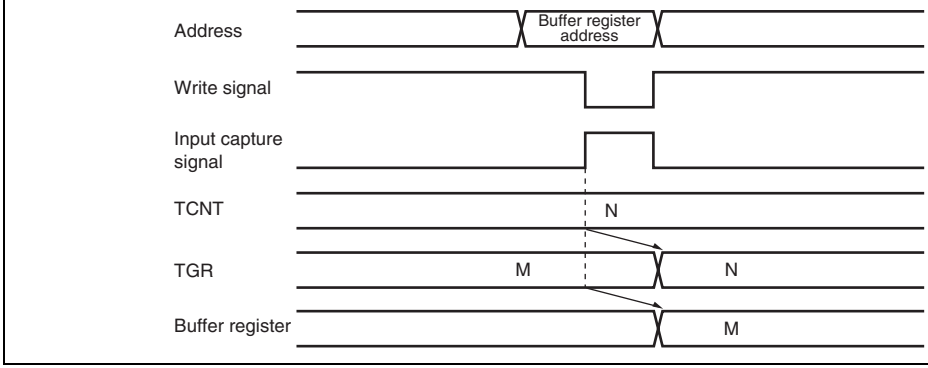


Figure 10.128 Contention between Buffer Register Write and Input Captur

10.7.12 TCNT2 Write and Overflow/Underflow Contention in Cascade Connectio

With timer counters TCNT1 and TCNT2 in a cascade connection, when a contention occ
during TCNT_1 count (during a TCNT_2 overflow/underflow) in the T_2 state of the TCN
write cycle, the write to TCNT_2 is conducted, and the TCNT_1 count signal is disabled.
point, if there is match with TGRA_1 and the TCNT_1 value, a compare signal is issued.
Furthermore, when the TCNT_1 count clock is selected as the input capture source of cha
TGRA_0 to D_0 carry out the input capture operation. In addition, when the compare ma
capture is selected as the input capture source of TGRB_1, TGRB_1 carries out input cap
operation. The timing is shown in figure 10.129.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when settin
clearing.

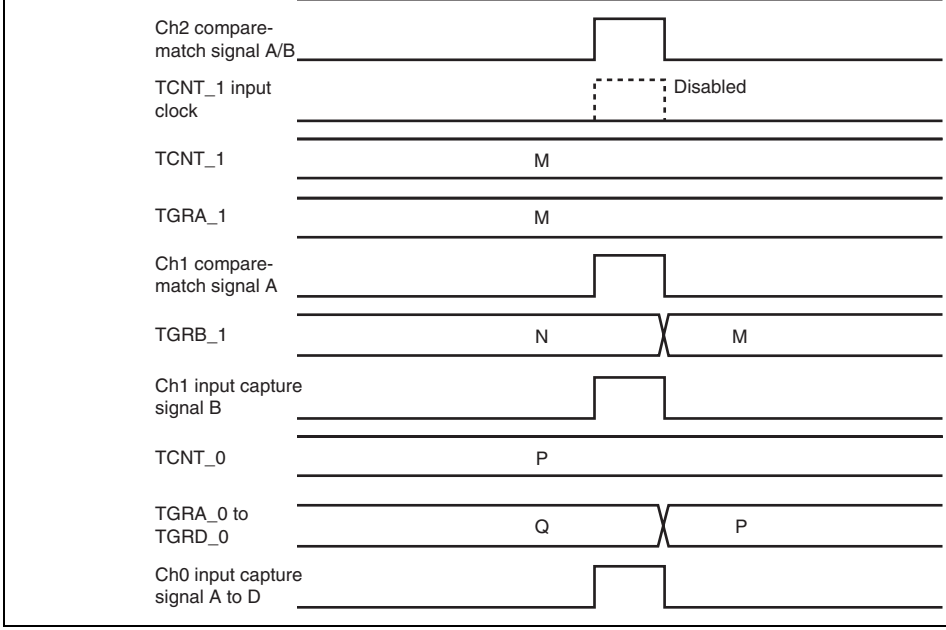


Figure 10.129 TCNT_2 Write and Overflow/Underflow Contention with Cas Connection

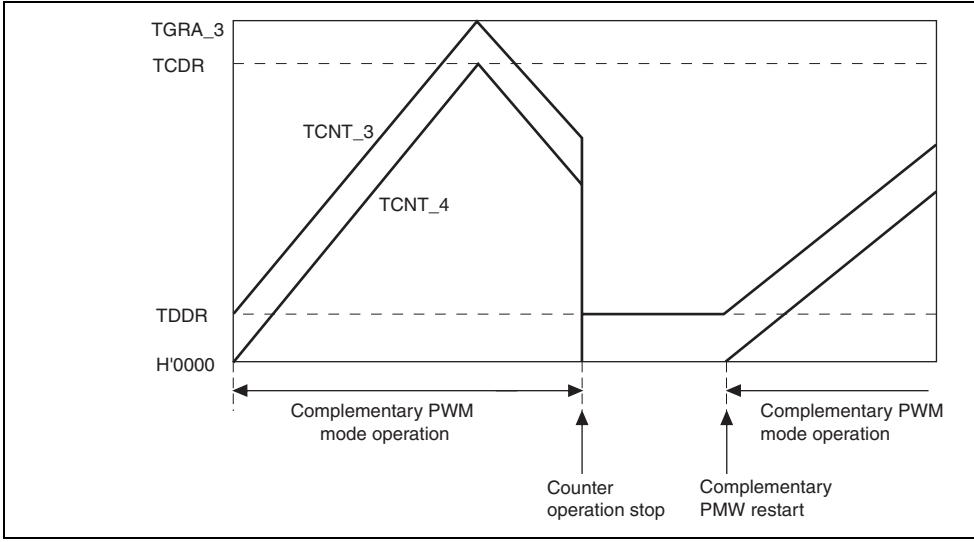


Figure 10.130 Counter Value during Complementary PWM Mode Stop

10.7.14 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle register (TGRA_3), timer cycle data register (TCDR), and duty setting registers (TGRB_ TGRA_4, and TGRB_4).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance wi settings BFA and BFB of TMDR_3. When TMDR_3's BFA bit is set to 1, TGRC_3 func buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register f TGRA_4, and TCBR functions as the TCDR's buffer register.

The TGFC bit and TGFD bit of TSR_3 and TSR_4 are not set when TGRC_3 and TGR operating as buffer registers.

Figure 10.131 shows an example of operations for TGR_3, TGR_4, TIOC3, and TIOC4 TMDR_3's BFA and BFB bits set to 1, and TMDR_4's BFA and BFB bits set to 0.

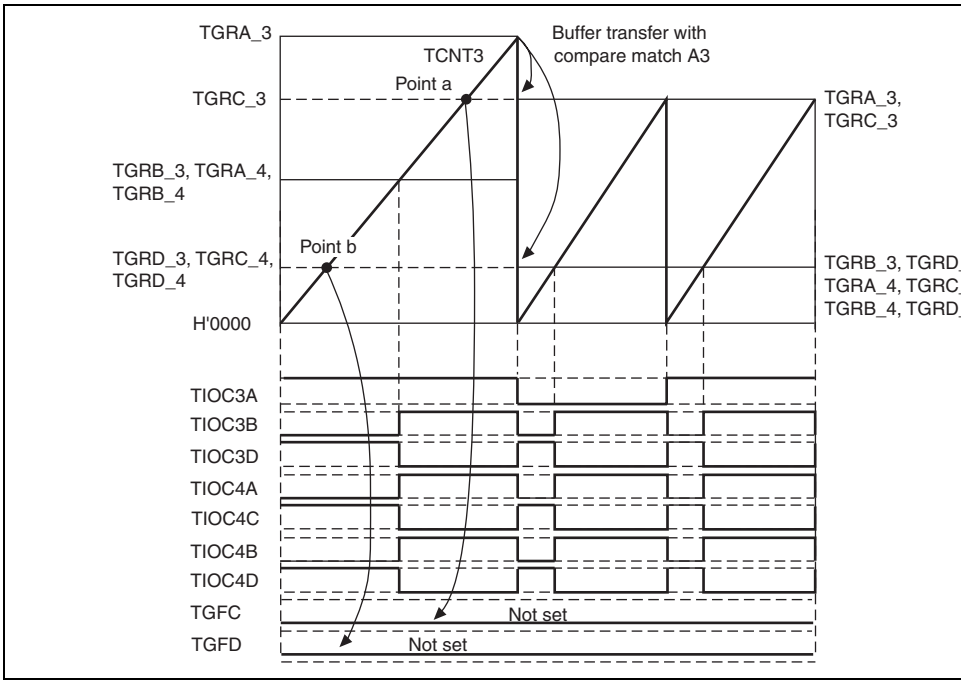


Figure 10.131 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

Figure 10.132 shows a TCFV bit operation example in reset synchronous PWM mode wi
value for cycle register TGRA_3 of H'FFFF, when a TGRA_3 compare-match has been s
without synchronous setting for the counter clear source.

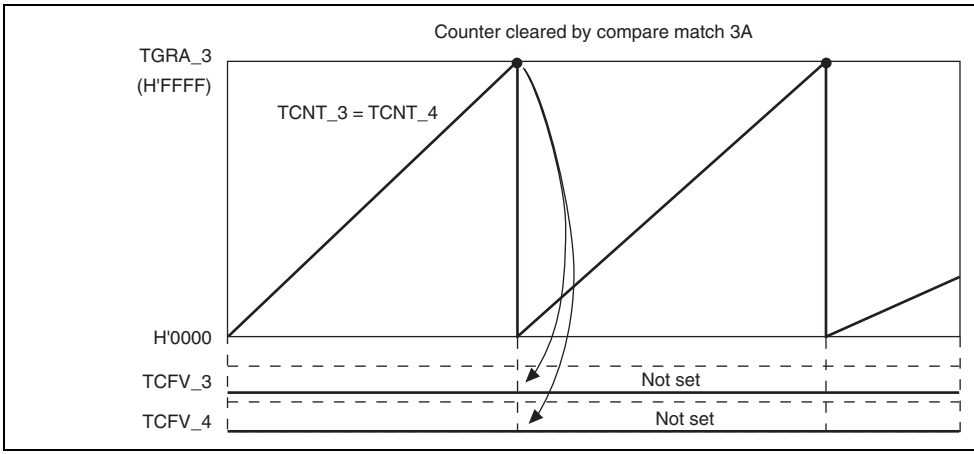


Figure 10.132 Reset Synchronous PWM Mode Overflow Flag

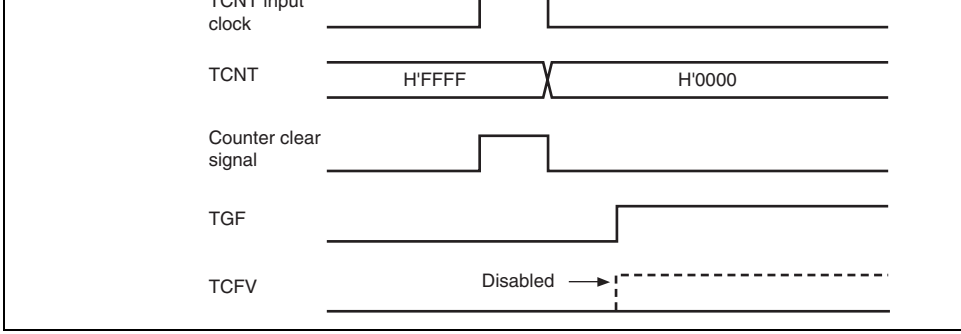


Figure 10.133 Contention between Overflow and Counter Clearing

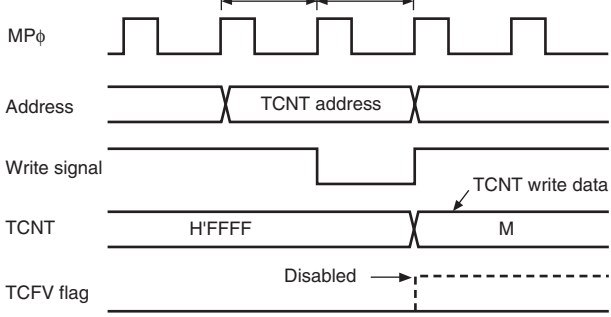


Figure 10.134 Contention between TCNT Write and Overflow

10.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset synchronized PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3I TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-level state, followed by the transition synchronized PWM mode and operation in that mode, the initial pin output will not be co

When making a transition from normal operation to reset-synchronized PWM mode, writ registers TIORH_3, TIORL_3, TIORH_4, and TIORL_4 to initialize the output pins to lo output, then set an initial register value of H'00 before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, first sw normal operation, then initialize the output pins to low level output and set an initial regis of H'00 before making the transition to reset-synchronized PWM mode.

clear the CPU interrupt source or the DMAC activation source. Interrupts should therefo
disabled before entering module standby mode.

10.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection

When timer counters 1 and 2 (TCNT_1 and TCNT_2) are operated as a 32-bit counter in
connection, the cascade counter value cannot be captured successfully even if input-capt
is simultaneously done to TIOC1A and TIOC2A or to TIOC1B and TIOC2B. This is be
input timing of TIOC1A and TIOC2A or of TIOC1B and TIOC2B may not be the same
external input-capture signals to be input into TCNT_1 and TCNT_2 are taken in synchr
with the internal clock. For example, TCNT_1 (the counter for upper 16 bits) does not c
count-up value by overflow from TCNT_2 (the counter for lower 16 bits) but captures th
value before the count-up. In this case, the values of TCNT_1 = H'FFF1 and TCNT_2 =
should be transferred to TGRA_1 and TGRA_2 or to TGRB_1 and TGRB_2, but the va
TCNT_1 = H'FFF0 and TCNT_2 = H'0000 are erroneously transferred.

The MTU2 additionally supports the function that can capture TCNT_1 and TCNT_2
simultaneously via a single input capture input. This function allows 32-bit counter fetch
without TCNT_1 and TCNT_2 capture timing deviation. For details, see section 10.3.8,
Input Capture Control Register (TICCR).

- Phase counting modes 1 to 4 (channels 1 and 2)
- Complementary PWM mode (channels 3 and 4)
- Reset-synchronized PWM mode (channels 3 and 4)

The MTU2 output pin initialization method for each of these modes is described in this se

10.8.2 Reset Start Operation

The MTU2 output pins (TIOC*) are initialized low by a reset and in standby mode. Since pin function selection is performed by the pin function controller (PFC), when the PFC is MTU2 pin states at that point are output to the ports. When MTU2 output is selected by t immediately after a reset, the MTU2 output initial level, low, is output directly at the port the active level is low, the system will operate at this point, and therefore the PFC setting be made after initialization of the MTU2 output pins is completed.

Note: Channel number and port notation are substituted for *.

Table 10.59 Mode Transition Combinations

	After					
Before	**Normal**	**PWM1**	**PWM2**	**PCM**	**CPWM**	**RP**
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12
PWM2	(13)	(14)	(15)	(16)	None	No
PCM	(17)	(18)	(19)	(20)	None	No
CPWM	(21)	(22)	None	None	(23) (24)	(25
RPWM	(26)	(27)	None	None	(28)	(29

[Legend]
Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

not initialize the pins. If initialization is required, carry it out in normal mode, then sw
PWM mode 2.

- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, set
TIOR will not initialize the buffer register pins. If initialization is required, clear buffe
carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR
initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initi
then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level i
selected by the timer output control register (TOCR) setting, switch to normal mode a
perform initialization with TIOR, then restore TIOR to its initial value, and temporari
channel 3 and 4 output with the timer output master enable register (TOER). Then op
unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, T
setting).

Note: Channel number is substituted for * indicated in this article.

Pin initialization procedures are described below for the numbered combinations in table
The active level is assumed to be low.

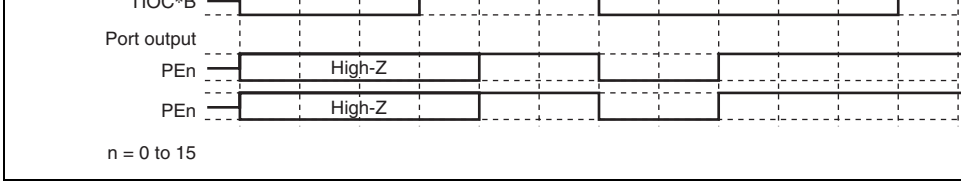


Figure 10.135 Error Occurrence in Normal Mode, Recovery in Normal Mo

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIO
4. Initialize the pins with TIOR. (The example shows initial high output, with low outp compare-match occurrence.)
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

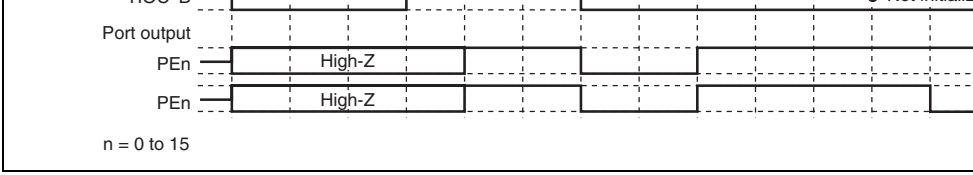


Figure 10.136 Error Occurrence in Normal Mode, Recovery in PWM Mode

1 to 10 are the same as in figure 10.135.

11. Set PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized. I initialization is required, initialize in normal mode, then switch to PWM mode 1.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

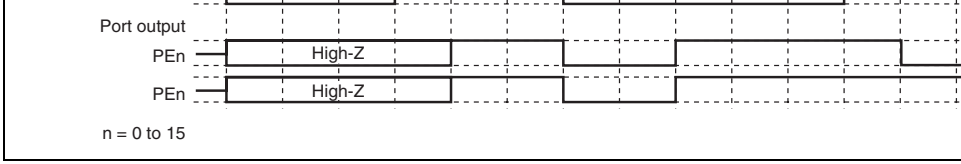


Figure 10.137 Error Occurrence in Normal Mode, Recovery in PWM Mod

1 to 10 are the same as in figure 10.135.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initia initialization is required, initialize in normal mode, then switch to PWM mode 2.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is necessary.

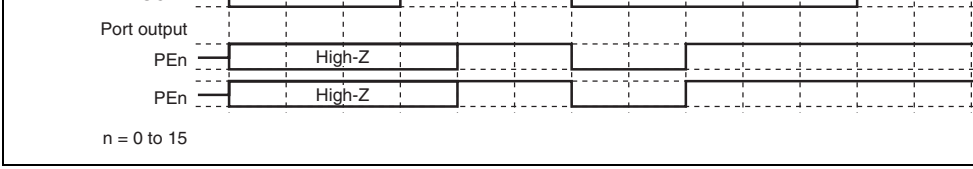


Figure 10.138 Error Occurrence in Normal Mode, Recovery in Phase Counting

1 to 10 are the same as in figure 10.135.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER s not necessary.

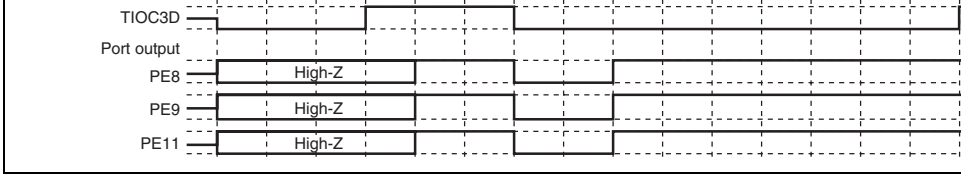


Figure 10.139 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.135.

11. Initialize the normal mode waveform generation section with TIOR.
12. Disable operation of the normal mode waveform generation section with TIOR.
13. Disable channel 3 and 4 output with TOER.
14. Select the complementary PWM output level and cyclic output enabling/disabling w TOCR.
15. Set complementary PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU2 output with the PFC.
18. Operation is restarted by TSTR.

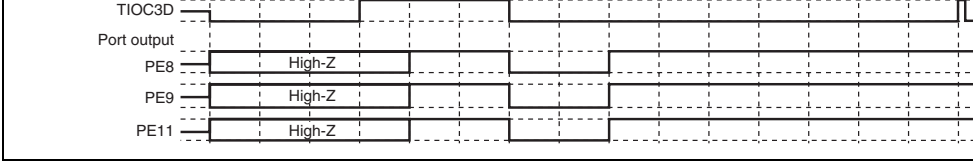


Figure 10.140 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 10.135.

14. Select the reset-synchronized PWM output level and cyclic output enabling/disablin TOCR.
15. Set reset-synchronized PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU2 output with the PFC.
18. Operation is restarted by TSTR.

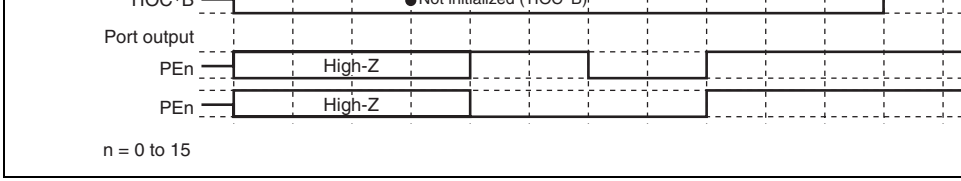


Figure 10.141 Error Occurrence in PWM Mode 1, Recovery in Normal Mo

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set PWM mode 1.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIO
4. Initialize the pins with TIOR. (The example shows initial high output, with low out compare-match occurrence. In PWM mode 1, the TIOC*B side is not initialized.)
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Set normal mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

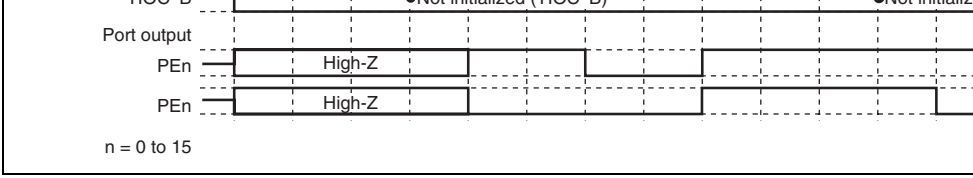


Figure 10.142 Error Occurrence in PWM Mode 1, Recovery in PWM Mode

1 to 10 are the same as in figure 10.141.

11. Not necessary when restarting in PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

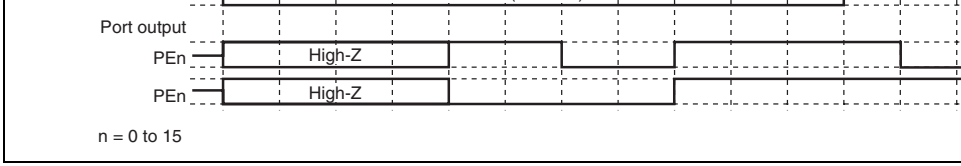


Figure 10.143 Error Occurrence in PWM Mode 1, Recovery in PWM Mod

1 to 10 are the same as in figure 10.141.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initia
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is necessary.

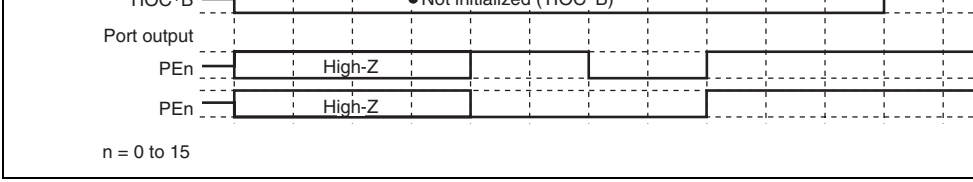


Figure 10.144 Error Occurrence in PWM Mode 1, Recovery in Phase Counting

1 to 10 are the same as in figure 10.141.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER s not necessary.

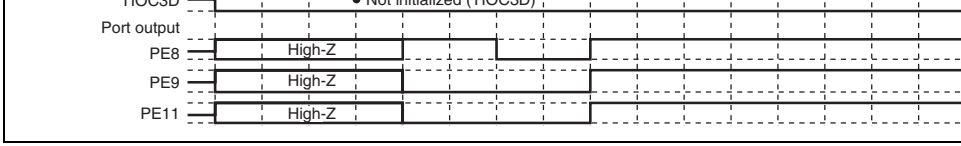


Figure 10.145 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.141.

11. Set normal mode for initialization of the normal mode waveform generation section
12. Initialize the PWM mode 1 waveform generation section with TIOR.
13. Disable operation of the PWM mode 1 waveform generation section with TIOR.
14. Disable channel 3 and 4 output with TOER.
15. Select the complementary PWM output level and cyclic output enabling/disabling w TOCR.
16. Set complementary PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU2 output with the PFC.
19. Operation is restarted by TSTR.

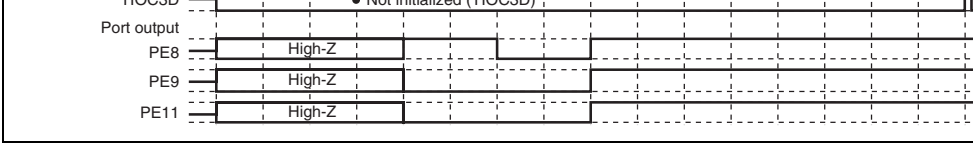


Figure 10.146 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 10.145.

15. Select the reset-synchronized PWM output level and cyclic output enabling/disabling TOCR.
16. Set reset-synchronized PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU2 output with the PFC.
19. Operation is restarted by TSTR.

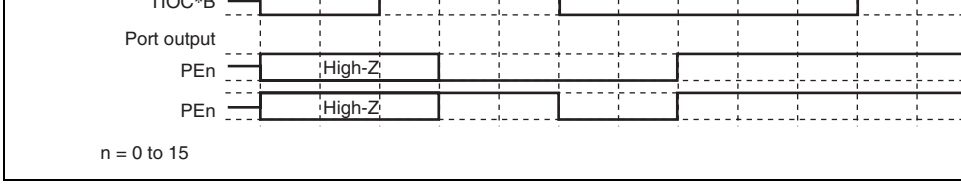


Figure 10.147 Error Occurrence in PWM Mode 2, Recovery in Normal Mo

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low out compare-match occurrence. In PWM mode 2, the cycle register pins are not initializ example, TIOC*A is the cycle register.)
4. Set MTU2 output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

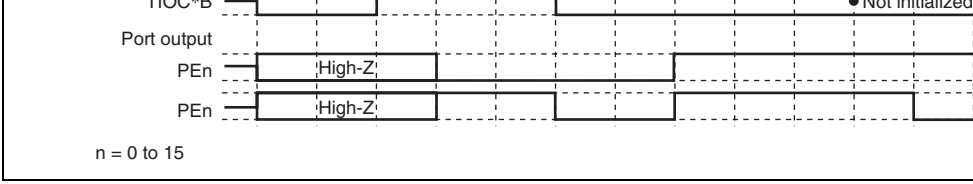


Figure 10.148 Error Occurrence in PWM Mode 2, Recovery in PWM Mode

1 to 9 are the same as in figure 10.147.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

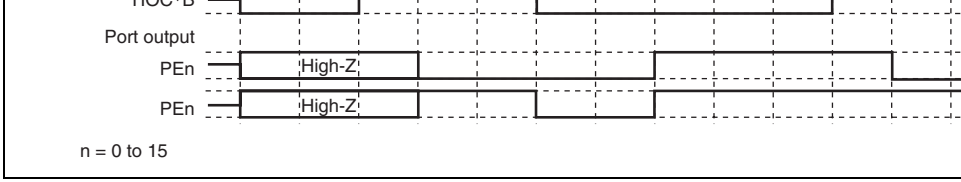


Figure 10.149 Error Occurrence in PWM Mode 2, Recovery in PWM Mod

1 to 9 are the same as in figure 10.147.

10. Not necessary when restarting in PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initia
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

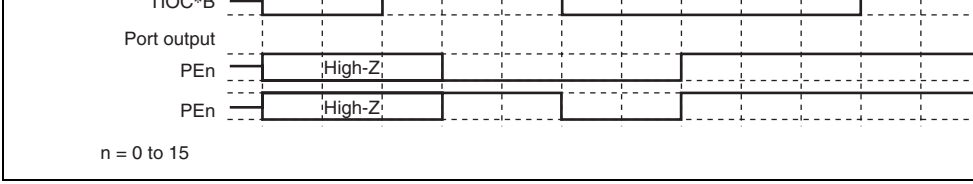


Figure 10.150 Error Occurrence in PWM Mode 2, Recovery in Phase Counting

1 to 9 are the same as in figure 10.147.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

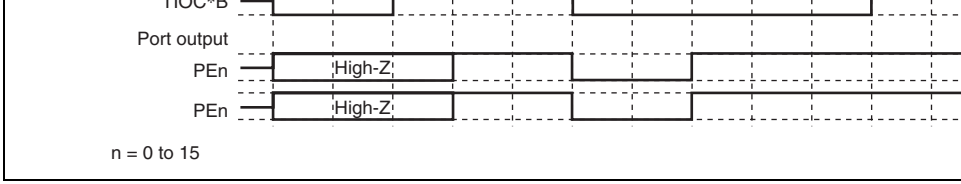


Figure 10.151 Error Occurrence in Phase Counting Mode, Recovery in Norma

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low ou compare-match occurrence.)
4. Set MTU2 output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

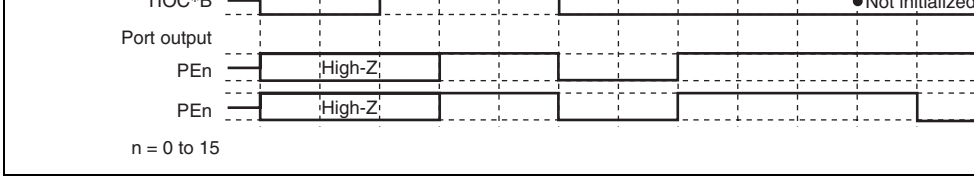


Figure 10.152 Error Occurrence in Phase Counting Mode, Recovery in PWM M

1 to 9 are the same as in figure 10.151.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

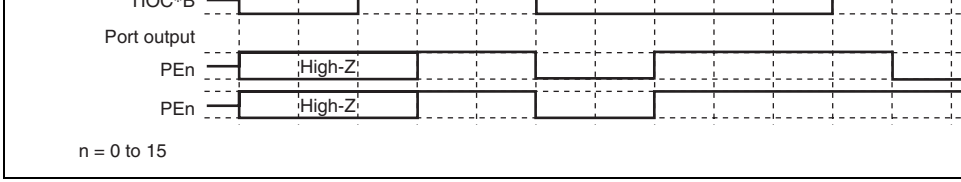


Figure 10.153 Error Occurrence in Phase Counting Mode, Recovery in PWM

1 to 9 are the same as in figure 10.151.

10. Set PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initia
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

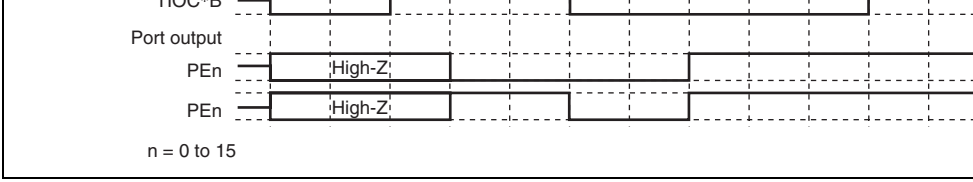


Figure 10.154 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 10.151.

10. Not necessary when restarting in phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

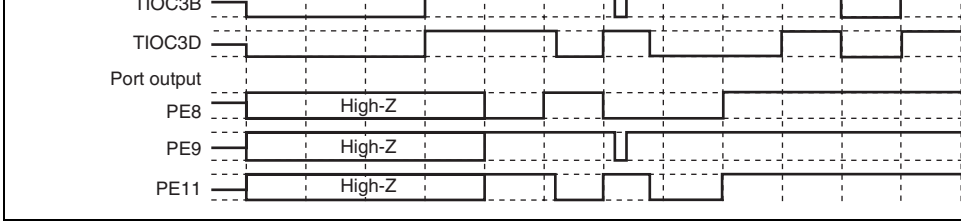


Figure 10.155 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling v TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the complementa output initial value.)
11. Set normal mode. (MTU2 output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

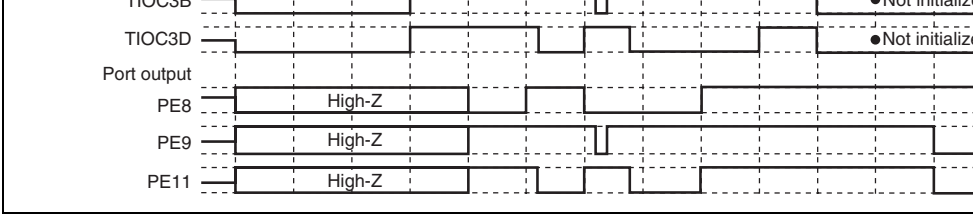


Figure 10.156 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 10.155.

11. Set PWM mode 1. (MTU2 output goes low.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

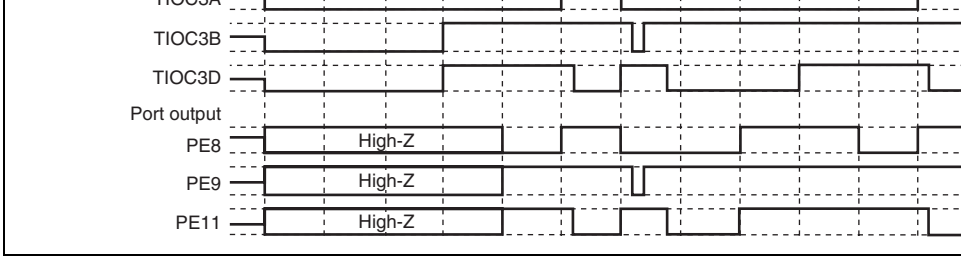


Figure 10.157 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.155.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The complementary PWM waveform is output on compare-match occurrence.

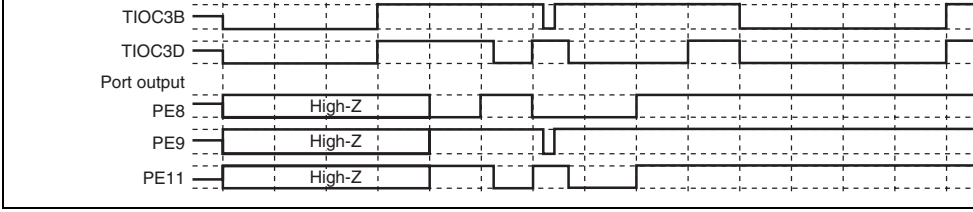


Figure 10.158 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.155.

11. Set normal mode and make new settings. (MTU2 output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the complementary PWM mode output level and cyclic output enabling/disabl TOCR.
14. Set complementary PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU2 output with the PFC.
17. Operation is restarted by TSTR.

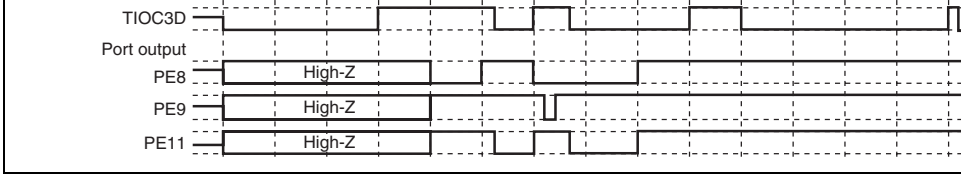


Figure 10.159 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 10.155.

11. Set normal mode. (MTU2 output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the reset-synchronized PWM mode output level and cyclic output enabling/d with TOCR.
14. Set reset-synchronized PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU2 output with the PFC.
17. Operation is restarted by TSTR.

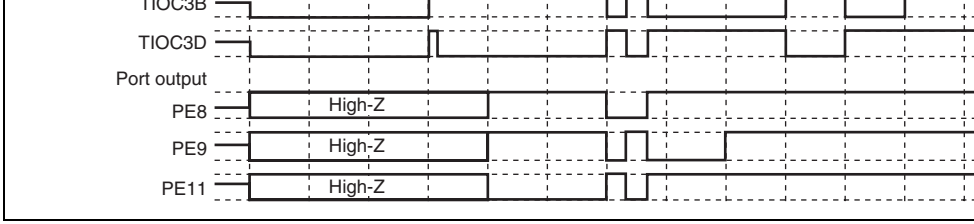


Figure 10.160 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the reset-synchronized PWM output level and cyclic output enabling/disabling TOCR.
3. Set reset-synchronized PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The reset-synchronized PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the reset-synchron PWM output initial value.)
11. Set normal mode. (MTU2 positive phase output is low, and negative phase output is
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

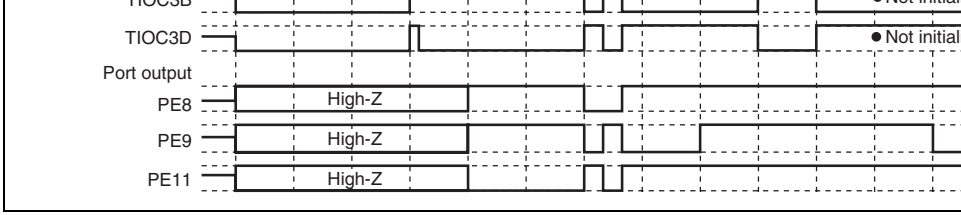


Figure 10.161 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 10.160.

11. Set PWM mode 1. (MTU2 positive phase output is low, and negative phase output i
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

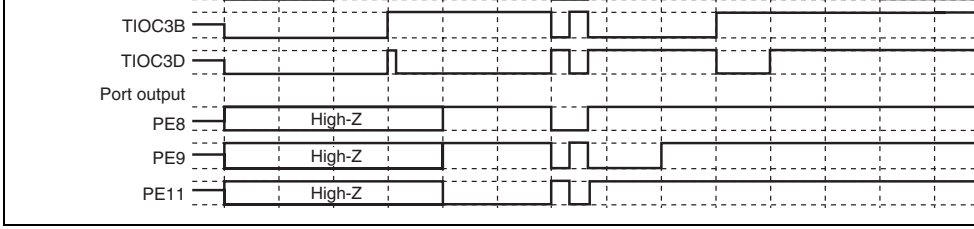


Figure 10.162 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.160.

11. Disable channel 3 and 4 output with TOER.
12. Select the complementary PWM output level and cyclic output enabling/disabling wi TOCR.
13. Set complementary PWM. (The MTU2 cyclic output pin goes low.)
14. Enable channel 3 and 4 output with TOER.
15. Set MTU2 output with the PFC.
16. Operation is restarted by TSTR.

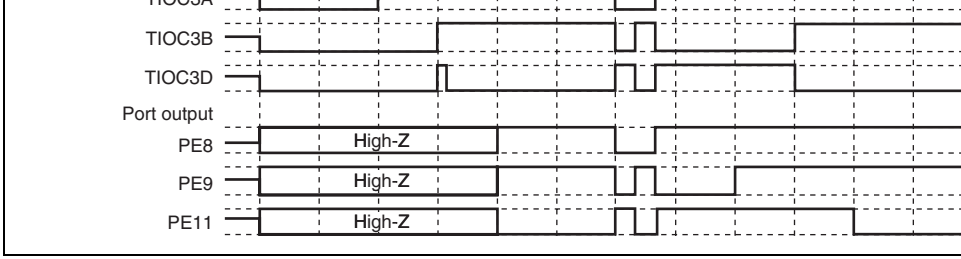


Figure 10.163 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 10.160.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The reset-synchronized PWM waveform is output on compare-match occurrence.

The MTU2S can operate at 80 MHz max. for complementary PWM output functions or
max. for the other functions.

General registers/ buffer registers		TGRC_3S TGRD_3S	TGRC_4S TGRD_4S	—
I/O pins		TIOC3AS TIOC3BS TIOC3CS TIOC3DS	TIOC4AS TIOC4BS TIOC4CS TIOC4DS	Input pins TIC5US TIC5VS TIC5WS
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare r input capture
Compare match output	0 output	√	√	—
	1 output	√	√	—
	Toggle output	√	√	—
Input capture function		√	√	√
Synchronous operation		√	√	—
PWM mode 1		√	√	—
PWM mode 2		—	—	—
Complementary PWM mode		√	√	—
Reset PWM mode		√	√	—
AC synchronous motor drive mode		—	—	—
Phase counting mode		—	—	—
Buffer operation		√	√	—

Interrupt sources	5 sources • Compare match or input capture 3AS • Compare match or input capture 3BS • Compare match or input capture 3CS • Compare match or input capture 3DS • Overflow	5 sources • Compare match or input capture 4AS • Compare match or input capture 4BS • Compare match or input capture 4CS • Compare match or input capture 4DS • Overflow or underflow	3 sources • Compare m input captu • Compare m input captu • Compare m input captu
A/D converter start request delaying function	—	• A/D converter start request at a match between TADCORA_4S and TCNT_4S • A/D converter start request at a match between TADCORB_4S and TCNT_4S	—

	TIOC4BS	I/O	TGRB_4S input capture input/output compare output/PWM
	TIOC4CS	I/O	TGRC_4S input capture input/output compare output/PWM
	TIOC4DS	I/O	TGRD_4S input capture input/output compare output/PWM
5	TIC5US	Input	TGRU_5S input capture input/external pulse input pin
	TIC5VS	Input	TGRV_5S input capture input/external pulse input pin
	TIC5WS	Input	TGRW_5S input capture input/external pulse input pin

Note: For the pin configuration in complementary PWM mode, see table 10.54.

3	Timer control register_3S	TCR_3S	R/W	H'00	H'FFFE4A00
	Timer mode register_3S	TMDR_3S	R/W	H'00	H'FFFE4A02
	Timer I/O control register H_3S	TIORH_3S	R/W	H'00	H'FFFE4A04
	Timer I/O control register L_3S	TIORL_3S	R/W	H'00	H'FFFE4A05
	Timer interrupt enable register_3S	TIER_3S	R/W	H'00	H'FFFE4A08
	Timer status register_3S	TSR_3S	R/W	H'C0	H'FFFE4A2C
	Timer counter_3S	TCNT_3S	R/W	H'0000	H'FFFE4A10
	Timer general register A_3S	TGRA_3S	R/W	H'FFFF	H'FFFE4A18
	Timer general register B_3S	TGRB_3S	R/W	H'FFFF	H'FFFE4A1A
	Timer general register C_3S	TGRC_3S	R/W	H'FFFF	H'FFFE4A24
	Timer general register D_3S	TGRD_3S	R/W	H'FFFF	H'FFFE4A26
	Timer buffer operation transfer mode register_3S	TBTM_3S	R/W	H'00	H'FFFE4A38
4	Timer control register_4S	TCR_4S	R/W	H'00	H'FFFE4A01
	Timer mode register_4S	TMDR_4S	R/W	H'00	H'FFFE4A03
	Timer I/O control register H_4S	TIORH_4S	R/W	H'00	H'FFFE4A06
	Timer I/O control register L_4S	TIORL_4S	R/W	H'00	H'FFFE4A07
	Timer interrupt enable register_4S	TIER_4S	R/W	H'00	H'FFFE4A09
	Timer status register_4S	TSR_4S	R/W	H'C0	H'FFFE4A2D
	Timer counter_4S	TCNT_4S	R/W	H'0000	H'FFFE4A12
	Timer general register A_4S	TGRA_4S	R/W	H'FFFF	H'FFFE4A1C
	Timer general register B_4S	TGRB_4S	R/W	H'FFFF	H'FFFE4A1E

	Timer A/D converter start request cycle set register B_4S	TADCORB_4S	R/W	H'FFFF	H'FFFE4A46
	Timer A/D converter start request cycle set buffer register A_4S	TADCOBRA_4S	R/W	H'FFFF	H'FFFE4A48
	Timer A/D converter start request cycle set buffer register B_4S	TADCOBRB_4S	R/W	H'FFFF	H'FFFE4A4A
5	Timer control register U_5S	TCRU_5S	R/W	H'00	H'FFFE4884
	Timer control register V_5S	TCRV_5S	R/W	H'00	H'FFFE4894
	Timer control register W_5S	TCRW_5S	R/W	H'00	H'FFFE48A4
	Timer I/O control register U_5S	TIORU_5S	R/W	H'00	H'FFFE4886
	Timer I/O control register V_5S	TIORV_5S	R/W	H'00	H'FFFE4896
	Timer I/O control register W_5S	TIORW_5S	R/W	H'00	H'FFFE48A6
	Timer interrupt enable register_5S	TIER_5S	R/W	H'00	H'FFFE48B2
	Timer status register_5S	TSR_5S	R/W	H'00	H'FFFE48B0
	Timer start register_5S	TSTR_5S	R/W	H'00	H'FFFE48B4
	Timer counter U_5S	TCNTU_5S	R/W	H'0000	H'FFFE4880
	Timer counter V_5S	TCNTV_5S	R/W	H'0000	H'FFFE4890
	Timer counter W_5S	TCNTW_5S	R/W	H'0000	H'FFFE48A0
	Timer general register U_5S	TGRU_5S	R/W	H'FFFF	H'FFFE4882
	Timer general register V_5S	TGRV_5S	R/W	H'FFFF	H'FFFE4892
	Timer general register W_5S	TGRW_5S	R/W	H'FFFF	H'FFFE48A2
	Timer compare match clear register S	TCNTCMPCLRS	R/W	H'00	H'FFFE48B6

Timer output control register 1S	TOCR1S	R/W	H'00	H'FFFE4A0E
Timer output control register 2S	TOCR2S	R/W	H'00	H'FFFE4A0F
Timer gate control register S	TGCRS	R/W	H80	H'FFFE4A0D
Timer cycle control register S	TCDRS	R/W	H'FFFF	H'FFFE4A14
Timer dead time data register S	TDDRS	R/W	H'FFFF	H'FFFE4A16
Timer subcounter S	TCNTSS	R	H'0000	H'FFFE4A20
Timer cycle buffer register S	TCBRS	R/W	H'FFFF	H'FFFE4A22
Timer interrupt skipping set register S	TITCRS	R/W	H'00	H'FFFE4A30
Timer interrupt skipping counter S	TITCNTS	R	H'00	H'FFFE4A31
Timer buffer transfer set register S	TBTERS	R/W	H'00	H'FFFE4A32
Timer dead time enable register S	TDERS	R/W	H'01	H'FFFE4A34
Timer synchronous clear register S	TSYCRS	R/W	H'00	H'FFFE4A50
Timer waveform control register S	TWCRS	R/W	H'00	H'FFFE4A60
Timer output level buffer register S	TOLBRS	R/W	H'00	H'FFFE4A36

- Interrupts can be generated by input-level sampling or output-level comparison result

The POE2 has input level detection circuits, output level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in the block diagram of f 12.1.

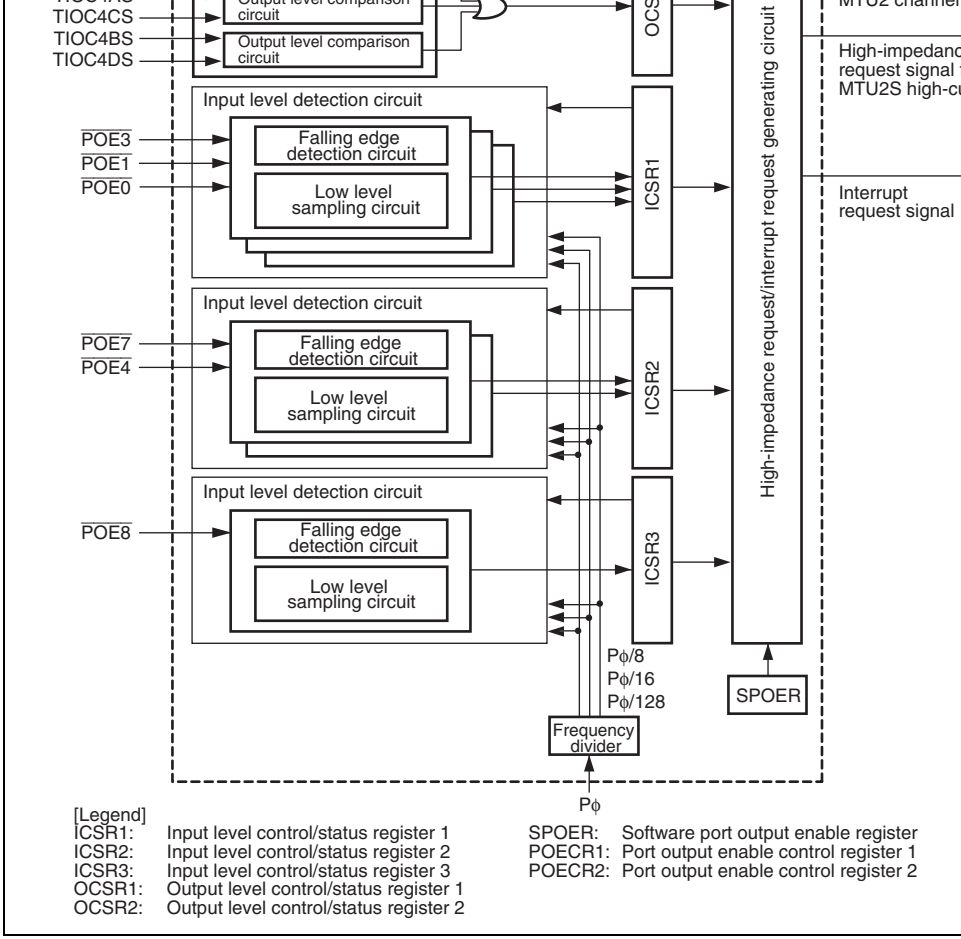


Figure 12.1 Block Diagram of POE2

Port output enable input pins 4 and 7	POE4 and POE7	Input	Input request signals to plac current pins (PB21/TIOC3BS PB20/TIOC3DS, PB12/TIOC PB13/TIOC4BS, PB10/TIOC and PB11/TIOC4DS) for MT high-impedance state
Port output enable input pin 8	$\overline{\text{POE8}}$	Input	Inputs a request signal to pl (PA22/TIOC0A, PA23/TIOC PA24/TIOC0C, and PA25/T for channel 0 in MTU2 in hig impedance state

		the OLSP bit is 1. In the case of TOCS = level when the OLS3N, OLS3P, OLS2N, OLS1N, and OLS1P bits are 0 in TOCR2 level when these bits are 1.) This active level comparison is done whe MTU2 output function or general output f selected in the pin function controller. If a function is selected, the output level is no checked. Pin combinations for output comparison impedance control can be selected by PC registers.
PB21/TIOC3BS and PB20/TIOC3DS PB12/TIOC4AS and PB10/TIOC4CS PB13/TIOC4BS and PB11/TIOC4DS	Output	The high-current pins for the MTU2S are high-impedance state when the pins simultaneously output an active level for more cycles of the peripheral clock (Pϕ). case of TOCS = 0 in timer output control 1S (TOCR1S) in the MTU2S, low level w output level select P (OLSP) bit is 0, or h when the OLSP bit is 1. In the case of TC low level when the OLS3N, OLS3P, OLS OLS2P, OLS1N, and OLS1P bits are 0 in TOCR2S, or high level when these bits a This active level comparison is done whe MTU2S output function or general output is selected in the pin function controller. I function is selected, the output level is no checked. Pin combinations for output comparison impedance control can be selected by PC registers.

Output level control/status register 1	OCSR1	R/W	H'0000	H'FFFE5002
Input level control/status register 2	ICSR2	R/W	H'0000	H'FFFE5004
Output level control/status register 2	OCSR2	R/W	H'0000	H'FFFE5006
Input level control/status register 3	ICSR3	R/W	H'0000	H'FFFE5008
Software port output enable register	SPOER	R/W	H'00	H'FFFE500A
Port output enable control register 1	POECR1	R/W	H'00	H'FFFE500B
Port output enable control register 2	POECR2	R/W	H'7700	H'FFFE500C

2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	POE3F	0	R/(W)*1	POE3 Flag Indicates that a high impedance request has be to the $\overline{\text{POE3}}$ pin. [Clearing conditions] • By writing 0 to POE3F after reading POE3F (when the falling edge is selected by bits 7 a ICSR1) • By writing 0 to POE3F after reading POE3F a high level input to $\overline{\text{POE3}}$ is sampled at Pϕ or Pϕ/128 clock (when low-level sampling is by bits 7 and 6 in ICSR1) [Setting condition] • When the input set by bits 7 and 6 in ICSR1 the $\overline{\text{POE3}}$ pin

(when the falling edge is selected by bits 3 a
ICSR1)

- By writing 0 to POE1F after reading POE1F =
a high level input to $\overline{\text{POE1}}$ is sampled at Pϕ/8
or Pϕ/128 clock (when low-level sampling is
by bits 3 and 2 in ICSR1)

[Setting condition]

- When the input set by bits 3 and 2 in ICSR1
the $\overline{\text{POE1}}$ pin

				a high level input to POE0 is sampled at Pϕ or Pϕ/128 clock (when low-level sampling is by bits 1 and 0 in ICSR1) [Set condition] • When the input set by bits 1 and 0 in ICSR1 the $\overline{\text{POE0}}$ pin
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
8	PIE1	0	R/W	Port Interrupt Enable 1 Enables or disables interrupt requests when an the POE0F, POE1F, and POE3F bits of the ICS to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7, 6	POE3M[1:0]	00	R/W*2	POE3 Mode These bits select the input mode of the $\overline{\text{POE3}}$ p 00: Accept request on falling edge of $\overline{\text{POE3}}$ inp 01: Accept request when $\overline{\text{POE3}}$ input has been for 16 Pϕ/8 clock pulses and all are low leve 10: Accept request when $\overline{\text{POE3}}$ input has been for 16 Pϕ/16 clock pulses and all are low lev 11: Accept request when $\overline{\text{POE3}}$ input has been for 16 Pϕ/128 clock pulses and all are low le

				10: Accept request when POE1 input has been s for 16 Pϕ/16 clock pulses and all are low leve
				11: Accept request when POE1 input has been s for 16 Pϕ/128 clock pulses and all are low le
1, 0	POE0M[1:0]	00	R/W*2	POE0 Mode
				These bits select the input mode of the POE0 pi
				00: Accept request on falling edge of POE0 inpu
				01: Accept request when POE0 input has been s for 16 Pϕ/8 clock pulses and all are low level
				10: Accept request when POE0 input has been s for 16 Pϕ/16 clock pulses and all are low leve
				11: Accept request when POE0 input has been s for 16 Pϕ/128 clock pulses and all are low le

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	OSF1	0	R/(W)*1	Output Short Flag 1 Indicates that any one of the three pairs of MTU phase outputs to be compared has simultaneou become an active level. [Clearing condition] • By writing 0 to OSF1 after reading OSF1 = [Setting condition] • When any one of the three pairs of 2-phase has simultaneously become an active level
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
9	OCE1	0	R/W*2	Output Short High-Impedance Enable 1 Specifies whether to place the pins in high-impe state when the OSF1 bit in OCSR1 is set to 1. 0: Does not place the pins in high-impedance s 1: Places the pins in high-impedance state
8	OIE1	0	R/W	Output Short Interrupt Enable 1 Enables or disables interrupt requests when the in OCSR is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled

ICSR2 is a 16-bit readable/writable register that selects the POE4 and POE7 pin input mo
controls the enable/disable of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	POE7F	-	-	POE4F	-	-	-	PIE2	POE7M[1:0]		-	-	-	-	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/(W)*1	R	R	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	R	R	R	R	R

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	POE7F	0	R/(W)*1	POE7 Flag Indicates that a high impedance request has bee to the $\overline{\text{POE7}}$ pin. [Clearing conditions] • By writing 0 to POE7F after reading POE7F = (when the falling edge is selected by bits 7 a ICSR2) • By writing 0 to POE7F after reading POE7F = a high level input to $\overline{\text{POE7}}$ is sampled at Pϕ/8 or Pϕ/128 clock (when low-level sampling is s by bits 7 and 6 in ICSR2) [Setting condition] • When the input condition set by bits 7 and 6 occurs at the $\overline{\text{POE7}}$ pin

				(when the falling edge is selected by bits 1 a ICSR2) • By writing 0 to POE4F after reading POE4F a high level input to $\overline{\text{POE4}}$ is sampled at Pϕ or Pϕ/128 clock (when low-level sampling is by bits 1 and 0 in ICSR2) [Setting condition] • When the input condition set by bits 1 and 0 occurs at the $\overline{\text{POE4}}$ pin
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
8	PIE2	0	R/W	Port Interrupt Enable 2 Enables or disables interrupt requests when an the POE4F and POE7F bits of the ICSR2 is set 0: Interrupt requests disabled 1: Interrupt requests enabled
7, 6	POE7M[1:0]	00	R/W*2	POE7 Mode These bits select the input mode of the $\overline{\text{POE7}}$ p 00: Accept request on falling edge of $\overline{\text{POE7}}$ inp 01: Accept request when $\overline{\text{POE7}}$ input has been for 16 Pϕ/8 clock pulses and all are at a low 10: Accept request when $\overline{\text{POE7}}$ input has been for 16 Pϕ/16 clock pulses and all are at a lo 11: Accept request when $\overline{\text{POE7}}$ input has been for 16 Pϕ/128 clock pulses and all are at a l

10: Accept request when POE4 input has been s
for 16 Pϕ/16 clock pulses and all are at a low

11: Accept request when $\overline{\text{POE4}}$ input has been s
for 16 Pϕ/128 clock pulses and all are at a lo

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	OSF2	0	R/(W)*1	Output Short Flag 2 Indicates that any one of the three pairs of MTU phase outputs to be compared has simultaneou become an active level. [Clearing condition] • By writing 0 to OSF2 after reading OSF2 = [Setting condition] • When any one of the three pairs of 2-phase has simultaneously become an active level
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
9	OCE2	0	R/W*2	Output Short High-Impedance Enable 2 Specifies whether to place the pins in high-impe state when the OSF2 bit in OCSR2 is set to 1. 0: Does not place the pins in high-impedance s 1: Places the pins in high-impedance state

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

12.3.5 Input Level Control/Status Register 3 (ICSR3)

ICSR3 is a 16-bit readable/writable register that selects the $\overline{\text{POE8}}$ pin input mode, control enable/disable of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	-	-	-	POE8F	-	-	POE8E	PIE3	-	-	-	-	-	-	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R/(W)*1	R	R	R/W*2	R/W	R	R	R	R	R	R	R

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				a high level input to POE8 is sampled at P or Pϕ/128 clock (when low-level sampling i by bits 1 and 0 in ICSR3) [Setting condition] • When the input condition set by bits 1 and ICSR3 occurs at the $\overline{\text{POE8}}$ pin
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write val always be 0.
9	POE8E	0	R/W*2	POE8 High-Impedance Enable Specifies whether to place the pins in high-imp state when the POE8F bit in ICSR3 is set to 1. 0: Does not place the pins in high-impedance 1: Places the pins in high-impedance state
8	PIE3	0	R/W	Port Interrupt Enable 3 Enables or disables interrupt requests when th bit in ICSR3 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write val always be 0.

for 16 ϕ/128 clock pulses and all are low le

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

12.3.6 Software Port Output Enable Register (SPOER)

SPOER is an 8-bit readable/writable register that controls high-impedance state of the pin

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	MTU2S HIZ	MTU2 CH0HIZ	MTU2 CH34HIZ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2SHIZ
1	MTU2CH0HIZ	0	R/W	MTU2 Channel 0 Output High-Impedance Specifies whether to place the pins for chan the MTU2 in high-impedance state. 0: Does not place the pins in high-impedanc [Clearing conditions] • Power-on reset • By writing 0 to MTU2CH0HIZ after readi MTU2CH0HIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2CH0HIZ
0	MTU2CH34HIZ	0	R/W	MTU2 Channel 3 and 4 Output High-Impeda Specifies whether to place the high-current the MTU2 in high-impedance state. 0: Does not place the pins in high-impedanc [Clearing conditions] • Power-on reset • By writing 0 to MTU2CH34HIZ after rea MTU2CH34HIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2CH34HIZ

Bit	Bit Name	Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
3	MTU2PA25ZE	0	R/W*	MTU2PA25 High-Impedance Enable Specifies whether to place the PA25/TIOC0D channel 0 in the MTU2 in high-impedance sta either POE8F or MTU2CH0HIZ bit is set to 1 0: Does not place the pin in high-impedance 1: Places the pin in high-impedance state
2	MTU2PA24ZE	0	R/W*	MTU2PA24 High-Impedance Enable Specifies whether to place the PA24/TIOC0C channel 0 in the MTU2 in high-impedance sta either POE8F or MTU2CH0HIZ bit is set to 1 0: Does not place the pin in high-impedance 1: Places the pin in high-impedance state
1	MTU2PA23ZE	0	R/W*	MTU2PA23 High-Impedance Enable Specifies whether to place the PA23/TIOC0B channel 0 in the MTU2 in high-impedance sta either POE8F or MTU2CH0HIZ bit is set to 1 0: Does not place the pin in high-impedance 1: Places the pin in high-impedance state

POECR2 is a 16-bit readable/writable register that controls high-impedance state of the

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	-	MTU2S P1CZE	MTU2S P2CZE	MTU2S P3CZE	-	-	-	-	-	-
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	0	0
R/W:	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*	R	R	R	R	R	R

Note: * Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
14	MTU2P1CZE	1	R/W*	MTU2 Port 1 Output Comparison/High-Impe Enable Specifies whether to compare output levels MTU2 high-current PB18/TIOC3B and PB1 pins and to place them in high-impedance s the OSF1 bit is set to 1 while the OCE1 bit i when any one of the POE0F, POE1F, POE MTU2CH34HIZ bits is set to 1. 0: Does not compare output levels or place high-impedance state 1: Compares output levels and places the p high-impedance state

				1: Compares output levels and places the pin high-impedance state
12	MTU2P3CZE	1	R/W*	MTU2 Port 3 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for MTU2 high-current PB5/TIOC4B and PB7/TIOC4D pins and to place them in high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F, POE1F, POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not compare output levels or place the high-impedance state 1: Compares output levels and places the pin high-impedance state
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	MTU2SP1CZE	1	R/W*	MTU2S Port 1 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for MTU2S high-current PB21/TIOC3BS and PB20/TIOC3DS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when any one of the POE4F, POE7F, and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the high-impedance state. 1: Compares output levels and places the pin high-impedance state.

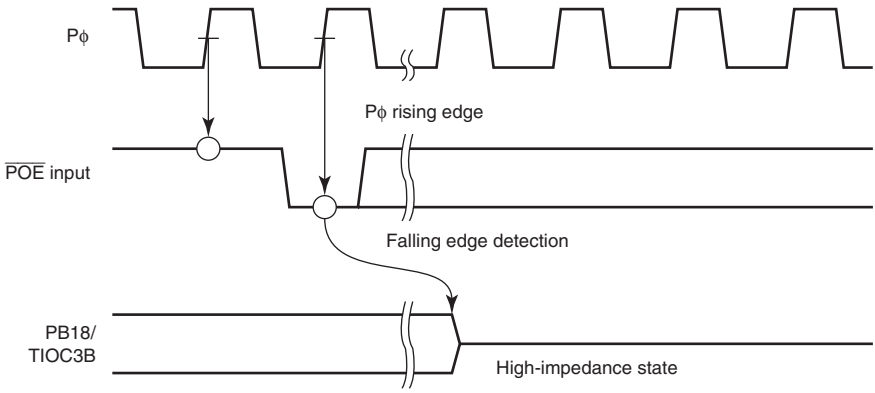
				1: Compares output levels and places the p high-impedance state.
8	MTU2SP3CZE	1	R/W*	MTU2S Port 3 Output Comparison/High-Imp Enable Specifies whether to compare output levels MTU2S high-current PB13/TIOC4BS and PB11/TIOC4DS pins and to place them in h impedance state when the OSF2 bit is set t the OCE2 bit is 1 or when any one of the P POE7F, and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place high-impedance state. 1: Compares output levels and places the p high-impedance state.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

Note: * Can be modified only once after a power-on reset.

MTU2 high-current pins (PB4/TIOC4A and PB6/TIOC4C)	Input level detection, output level comparison, or SPOER setting	MTU2P2CZE • ((POE3F+POE1F+POE0F) + (OCE1) + (MTU2CH34HIZ))
MTU2 high-current pins (PB5/TIOC4B and PB7/TIOC4D)	Input level detection, output level comparison, or SPOER setting	MTU2P3CZE • ((POE3F+POE1F+POE0F) + (OCE1) + (MTU2CH34HIZ))
MTU2S high-current pins (PB21/TIOC3BS and PB20/TIOC3DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP1CZE • ((POE4F+POE7F) + (OSF2 • ((MTU2SHIZ))
MTU2S high-current pins (PB12/TIOC4AS and PB10/TIOC4CS)	Input level detection, output level comparison, or SPOER setting	MTU2SP2CZE • ((POE4F+POE7F) + (OSF2 • ((MTU2SHIZ))
MTU2S high-current pins (PB13/TIOC4BS and PB11/TIOC4DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP3CZE • ((POE4F+POE7F) + (OSF2 • ((MTU2SHIZ))
MTU2 channel 0 pins (PA22/TIOC0A, PA23/TIOC0B, PA24/TIOC0C, and PA25/TIOC0D)	Input level detection or SPOER setting	((POE8F • POE8E) + (MTU2C

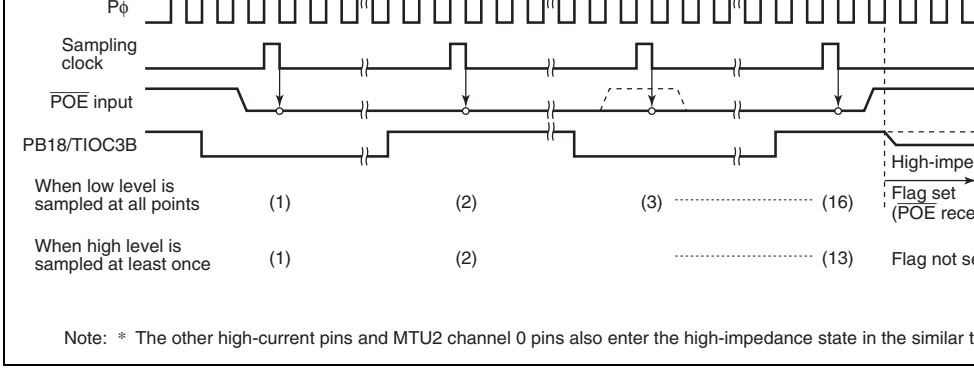
$\overline{\text{POE8}}$ pins, the high-current pins and the pins for channel 0 of the MTU2 are placed in h
impedance state.

Figure 12.2 shows the sample timing after the level changes in input to the $\overline{\text{POE0}}$, $\overline{\text{POE1}}$
$\overline{\text{POE4}}$, $\overline{\text{POE7}}$, and $\overline{\text{POE8}}$ pins until the respective pins enter high-impedance state.



Note: The other high-current pins and MTU2 channel 0 pins also enter the high-impedance state in the simila

Figure 12.2 Falling Edge Detection



Note: * The other high-current pins and MTU2 channel 0 pins also enter the high-impedance state in the similar t

Figure 12.3 Low-Level Detection Operation

12.4.2 Output-Level Compare Operation

Figure 12.4 shows an example of the output-level compare operation for the combination TIOC3B and TIOC3D. The operation is the same for the other pin combinations.

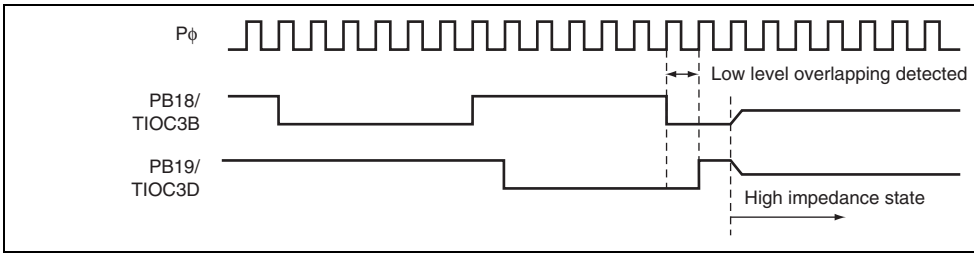


Figure 12.4 Output-Level Compare Operation

released either by returning them to their initial state with a power-on reset, or by clearin
in bit 15 (OCF1 and OCF2) in OCSR1 and OCSR2. However, note that just writing 0 to
ignored (the flag is not cleared); flags can be cleared only after an inactive level is outpu
high-current pins. Inactive-level outputs can be achieved by setting the MTU2 and MTU
internal registers.

OEI2	Output enable interrupt 2	POE8F	PIE3 • POE8F
OEI3	Output enable interrupt 3	POE4F, POE7F, and OSF2	PIE2 • (POE4F + POE OIE2 • OSF2

If a power-on reset is issued by the WDT during high-impedance processing by MTU2 c
short detection, the I/O port pins are placed in the same status as described above.

Figure 12.5 shows the I/O port pin status when a power-on reset is issued by the WDT d
high-impedance processing by the POE input while the timer output is selected.

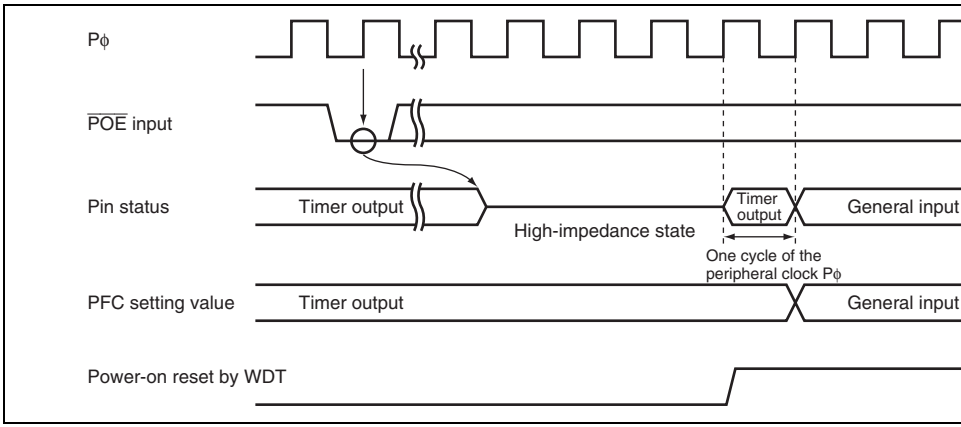


Figure 12.5 Pin Status When Power-on Reset is Issued from Watchdog Tim

DMAC setting

- When not in use, the CMT can be stopped by halting its clock supply to reduce powe consumption.

Figure 13.1 shows a block diagram of CMT.

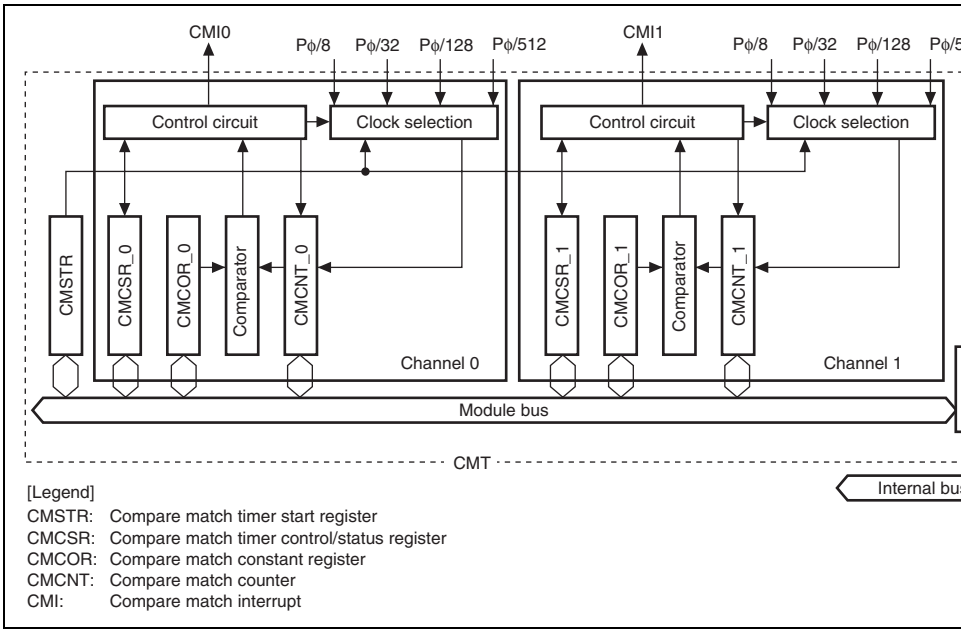


Figure 13.1 Block Diagram of CMT

	Compare match counter_0	CMCNT_0	R/W	H'0000	H'FFFEC004
	Compare match constant register_0	CMCOR_0	R/W	H'FFFF	H'FFFEC006
1	Compare match timer control/ status register_1	CMCSR_1	R/(W)*	H'0000	H'FFFEC008
	Compare match counter_1	CMCNT_1	R/W	H'0000	H'FFFEC00A
	Compare match constant register_1	CMCOR_1	R/W	H'FFFF	H'FFFEC00C

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
1	STR1	0	R/W	Count Start 1 Specifies whether compare match counter_1 or is stopped. 0: CMCNT_1 count is stopped 1: CMCNT_1 count is started
0	STR0	0	R/W	Count Start 0 Specifies whether compare match counter_0 or is stopped. 0: CMCNT_0 count is stopped 1: CMCNT_0 count is started

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

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
7	CMF	0	R/(W)*	Compare Match Flag Indicates whether or not the values of CMCNT CMCOR match. 0: CMCNT and CMCOR values do not match [Clearing condition] • When 0 is written to CMF after reading CM 1: CMCNT and CMCOR values match
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (generation when CMCNT and CMCOR values (CMF = 1). 0: Compare match interrupt (CMI) disabled 1: Compare match interrupt (CMI) enabled
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

11: Pϕ/512

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

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

13.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset or in software standby mode, but re
previous value in module standby mode.

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

Figure 13.2 shows the operation of the compare match counter.

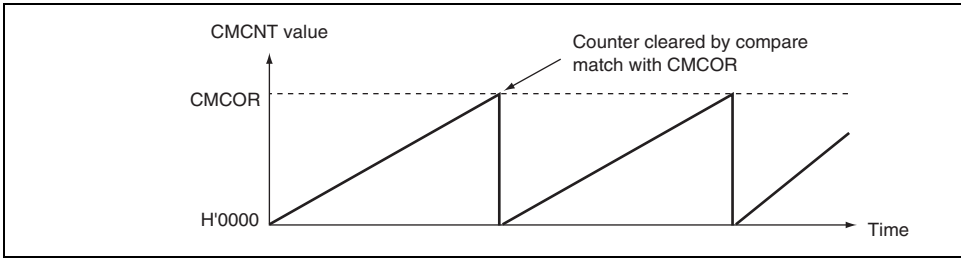


Figure 13.2 Counter Operation

13.3.2 CMCNT Count Timing

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the peripher (Pϕ) can be selected with the CKS[1:0] bits in CMCSR. Figure 13.3 shows the timing.

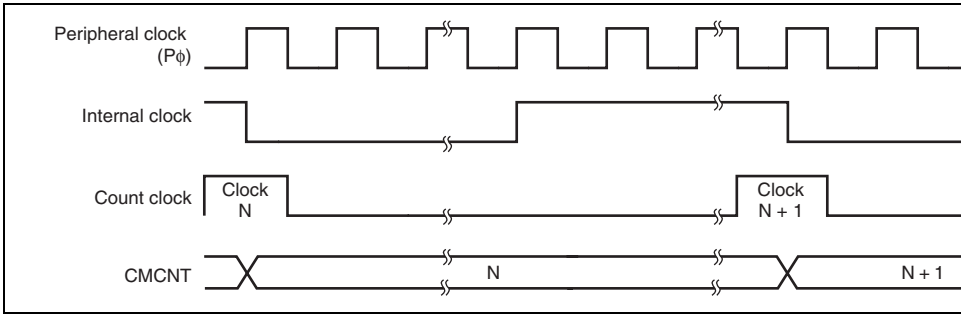


Figure 13.3 Count Timing

Clear the CMF bit to 0 by the user exception handling routine. If this operation is not carr
another interrupt will be generated. The direct memory access controller (DMAC) can be
activated when a compare match interrupt is requested. In this case, an interrupt is not iss
the CPU. If the setting to activate the DMAC has not been made, an interrupt request is s
CPU. The CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

13.4.2 Timing of Compare Match Flag Setting

When CMCOR and CMCNT match, a compare match signal is generated at the last state
the values match (the timing when the CMCNT value is updated to H'0000) and the CMF
CMCSR is set to 1. That is, after a match between CMCOR and CMCNT, the compare m
signal is not generated until the next CMCNT counter clock input. Figure 13.4 shows the
CMF bit setting.

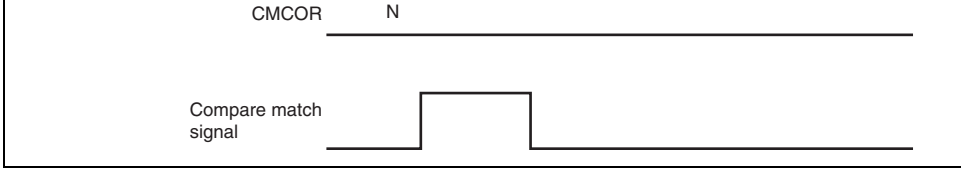


Figure 13.4 Timing of CMF Setting

13.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by first, reading as 1 then writing to 0. However, in t
the DMAC being activated, the CMF bit is automatically cleared to 0 when data is trans
the DMAC.

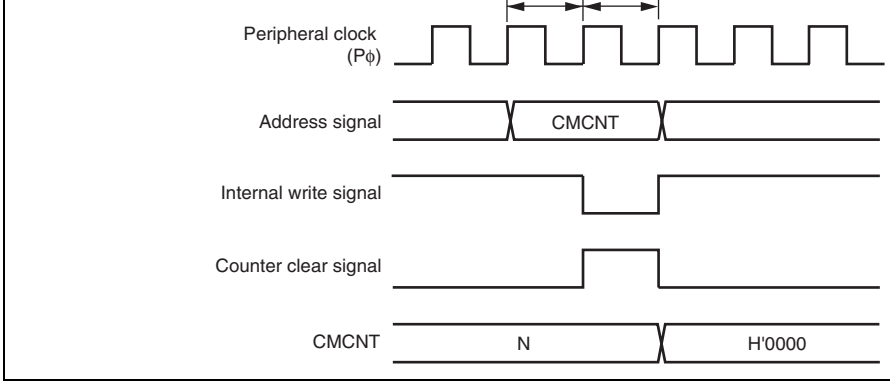


Figure 13.5 Conflict between Write and Compare Match Processes of CMCN

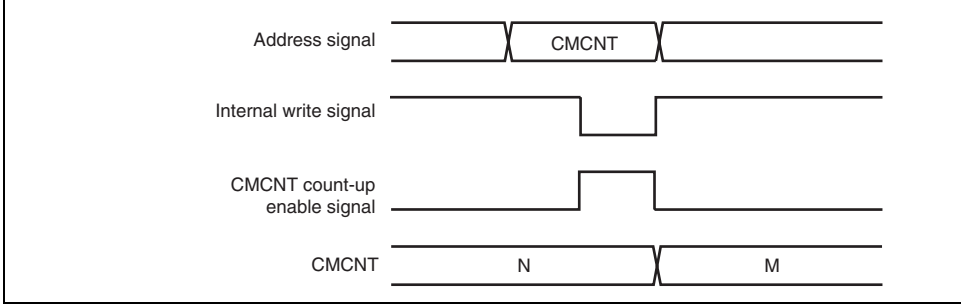


Figure 13.6 Conflict between Word-Write and Count-Up Processes of CMC

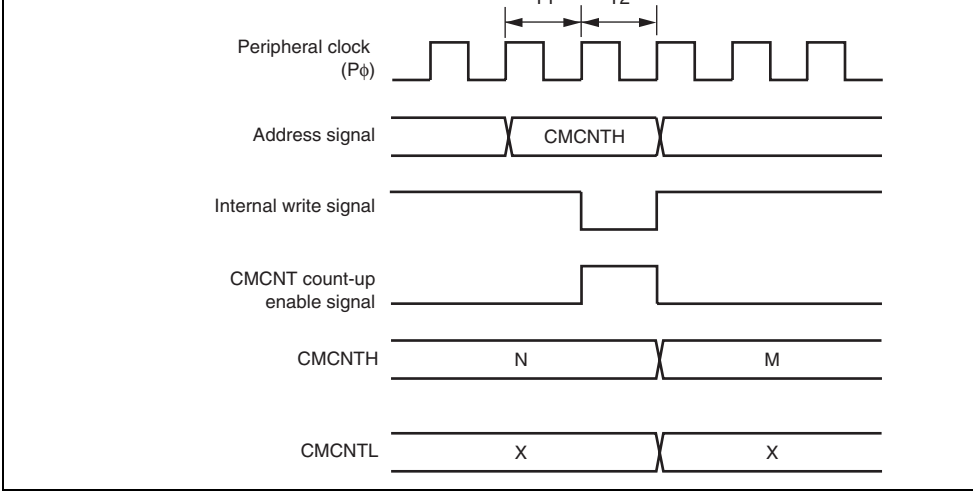


Figure 13.7 Conflict between Byte-Write and Count-Up Processes of CMCN

13.5.4 Compare Match Between CMCNT and CMCOR

Do not set a same value to CMCNT and CMCOR while the count operation of CMCNT i
stopped.

14.1 Features

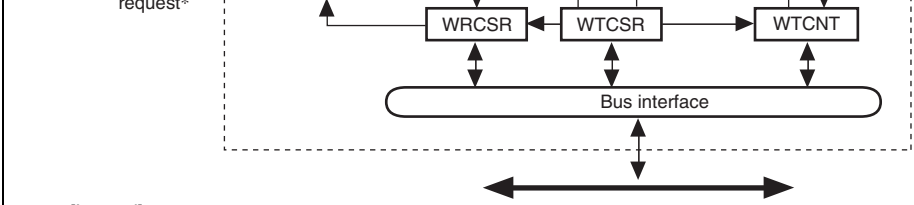
- Can be used to ensure the clock oscillation settling time

  The WDT is used in leaving software standby mode or the temporary standby period occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Outputs $\overline{\text{WDTOVF}}$ signal in watchdog timer mode

  When the counter overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. It is possible to select whether to reset the LSI internally when this happens the power-on reset or manual reset signal can be selected as the internal reset type.
- Interrupt generation in interval timer mode

  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks

  Eight clocks (Pϕ × 1 to Pϕ × 1/16384) that are obtained by dividing the peripheral clock selected.



[Legend]
WTCSR: Watchdog timer control/status register
WTCNT: Watchdog timer counter
WRCSR: Watchdog reset control/status register

Note: * The internal reset signal can be generated by making a register setting.

Figure 14.1 Block Diagram of WDT

Watchdog reset control/status register	WRCSR	R/W	H'1F	H'FFFE0004	1

Note: * For the access size, see section 14.3.4, Notes on Register Access.

14.3.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that is incremented by cycles of the selecte signal. When an overflow occurs, it generates a watchdog timer overflow signal ($\overline{\text{WDTO}}$ watchdog timer mode and an interrupt in interval timer mode. WTCNT is initialized to H power-on reset caused by the $\overline{\text{RES}}$ pin or in software standby mode.

Use word access to write to WTCNT, writing H'5A in the upper byte. Use byte access to from WTCNT.

Note: The method for writing to WTCNT differs from that for other registers to prevent erroneous writes. See section 14.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The method for writing to WTCSR differs from that for other registers to prevent erroneous writes. See section 14.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{IT}$	TME	-	-	CKS[2:0]		
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IOVF	0	R/(W)	Interval Timer Overflow Indicates that WTCNT has overflowed in int mode. This flag is not set in watchdog timer 0: No overflow 1: WTCNT overflow in interval timer mode [Clearing condition] • When 0 is written to IOVF after reading
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a watch or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: When the WTCNT overflows in watc mode, the $\overline{WDTOVF}$ signal is output If this bit is modified when the WDT i the up-count may not be performed c

				These bits are always read as 1. The write va should always be 1.
2 to 0	CKS[2:0]	000	R/W	Clock Select These bits select the clock to be used for the count from the eight types obtainable by divid peripheral clock (Pϕ). The overflow period tha shown in the table is the value when the peri clock (Pϕ) is 40 MHz.

Bits 2 to 0	Clock Ratio	Overflo
000:	1 × Pϕ	6.4 μs
001:	1/64 × Pϕ	409.6 μs
010:	1/128 × Pϕ	819.2 ms
011:	1/256 × Pϕ	1.64 ms
100:	1/512 × Pϕ	3.3 ms
101:	1/1024 × Pϕ	6.6 ms
110:	1/4096 × Pϕ	26.2 ms
111:	1/16384 × Pϕ	104.9 ms

Note: If bits CKS[2:0] are modified when the running, the up-count may not be perf correctly. Ensure that these bits are m only when the WDT is not running.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	-	-	-	-	-
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)	Watchdog Timer Overflow Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in i timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog time [Clearing condition] • When 0 is written to WOVF after reading
6	RSTE	0	R/W	Reset Enable Selects whether to generate a signal to rese internally if WTCNT overflows in watchdog t mode. In interval timer mode, this setting is 0: Not reset when WTCNT overflows* 1: Reset when WTCNT overflows Note: * LSI not reset internally, but WTCI WTCSR reset within WDT.

14.3.4 Notes on Register Access

The watchdog timer counter (WTCNT), watchdog timer control/status register (WTCSR) watchdog reset control/status register (WRCSR) are more difficult to write to than other r The procedures for reading or writing to these registers are given below.

(1) Writing to WTCNT and WTCSR

These registers must be written by a word transfer instruction. They cannot be written by longword transfer instruction.

When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the w as shown in figure 14.2. When writing to WTCSR, set the upper byte to H'A5 and transfe lower byte as the write data. This transfer procedure writes the lower byte data to WTCN WTCSR.

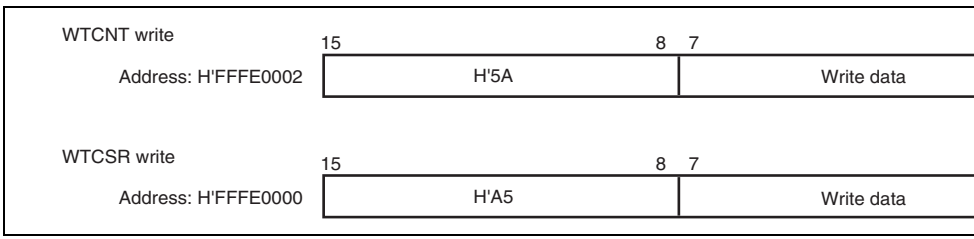


Figure 14.2 Writing to WTCNT and WTCSR

The WOVF bit is not affected.

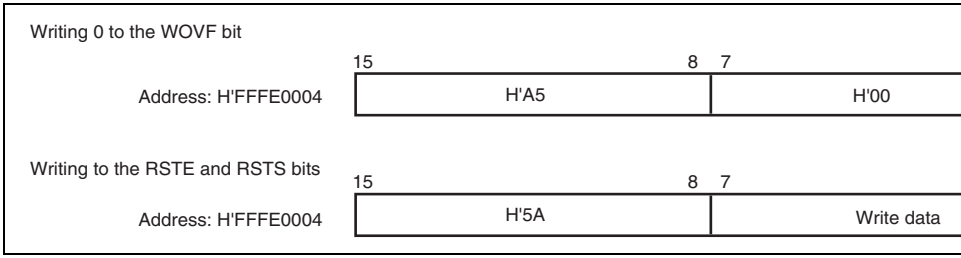


Figure 14.3 Writing to WRCSR

(3) Reading from WTCNT, WTCSR, and WRCSR

WTCNT, WTCSR, and WRCSR are read in a method similar to other registers. WTCSR allocated to address H'FFFE0000, WTCNT to address H'FFFE0002, and WRCSR to add H'FFFE0004. Byte transfer instructions must be used for reading from these registers.

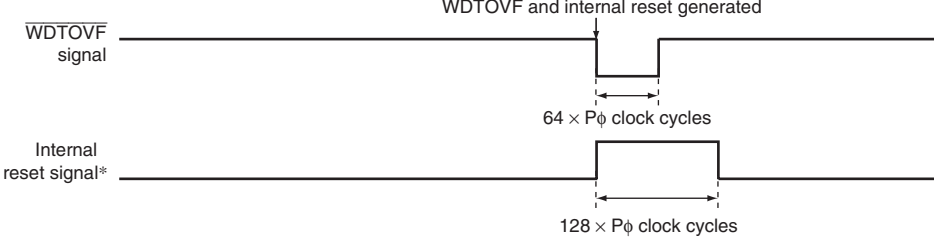
when the count overflows.

2. Set the type of count clock used in the CKS[2:0] bits in WTCSR and the initial value counter in WTCNT. These values should ensure that the time till count overflow is lo the clock oscillation settling time.
3. After setting the STBY bit of the standby control register (STBCR: see section 23, Po Down Modes) to 1, the execution of a SLEEP instruction puts the system in software mode and clock operation then stops.
4. The WDT starts counting by detecting the edge change of the NMI signal.
5. When the WDT count overflows, the CPG starts supplying the clock and this LSI resu operation. The WOVF flag in WRCSR is not set when this happens.

5. When the frequency control register (FRQCR) is written to, this LSI stops temporari
   WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and this LSI
   operation. The WOVF flag in WRCSR is not set when this happens.
5. The counter stops at the value of H'00.
6. Before changing WTCNT after execution of the frequency change instruction, alway
   that the value of WTCNT is H'00 by reading from WTCNT.

the system. The WDTOVF signal is output for 64 × Pϕ clock cycles.

5. If the RSTE bit in WRCSR is set to 1, a signal to reset the inside of this LSI can be ge simultaneously with the $\overline{\text{WDTOVF}}$ signal. Either power-on reset or manual reset can selected for this interrupt by the RSTS bit in WRCSR. The internal reset signal is outp 128 × Pϕ clock cycles.
6. When a WDT overflow reset is generated simultaneously with a reset input on the $\overline{\text{RE}}$ the $\overline{\text{RES}}$ pin reset takes priority, and the WOVF bit in WRCSR is cleared to 0.



[Legend]
WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit

Note: * Internal reset signal occurs only when the RSTE bit is set to 1.

Figure 14.4 Operation in Watchdog Timer Mode

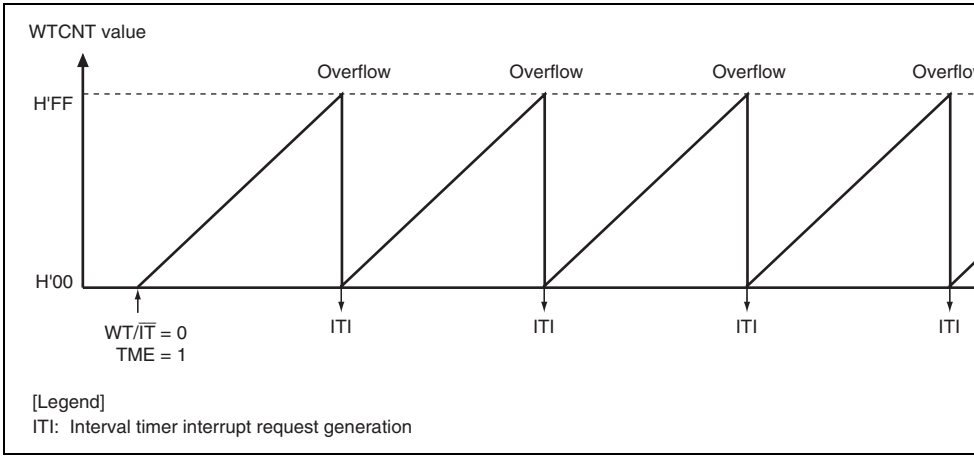


Figure 14.5 Operation in Interval Timer Mode

result of frequency division according to the value in the CKS[2:0] bits. The timing of s
incrementation is in accord with the selected frequency division ratio. Accordingly, this
difference is referred to as timer variation.

This also applies to the timing of the first incrementation after WTCNT has been writte
timer operation.

14.5.2 Prohibition against Setting H'FF to WTCNT

When the value in WTCNT reaches H'FF, the WDT assumes that an overflow has occu
Accordingly, when H'FF is set in WTCNT, an interval timer interrupt or WDT reset wil
immediately, regardless of the current clock selection by the CKS[2:0] bits.

14.5.3 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin of this LSI, this LSI cannot be initialized

Avoid input of the $\overline{\text{WDTOVF}}$ signal to the $\overline{\text{RES}}$ pin of this LSI through glue logic circui
reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 14.6.

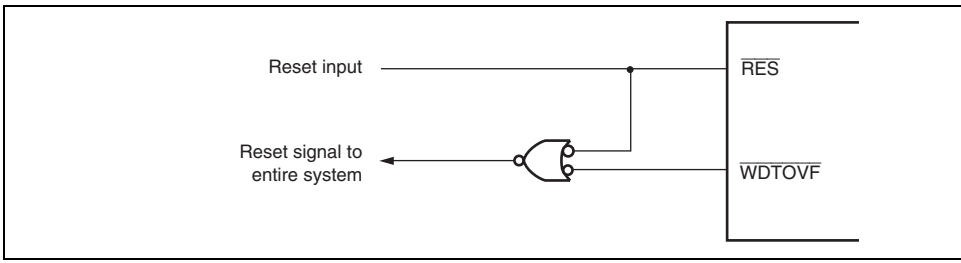
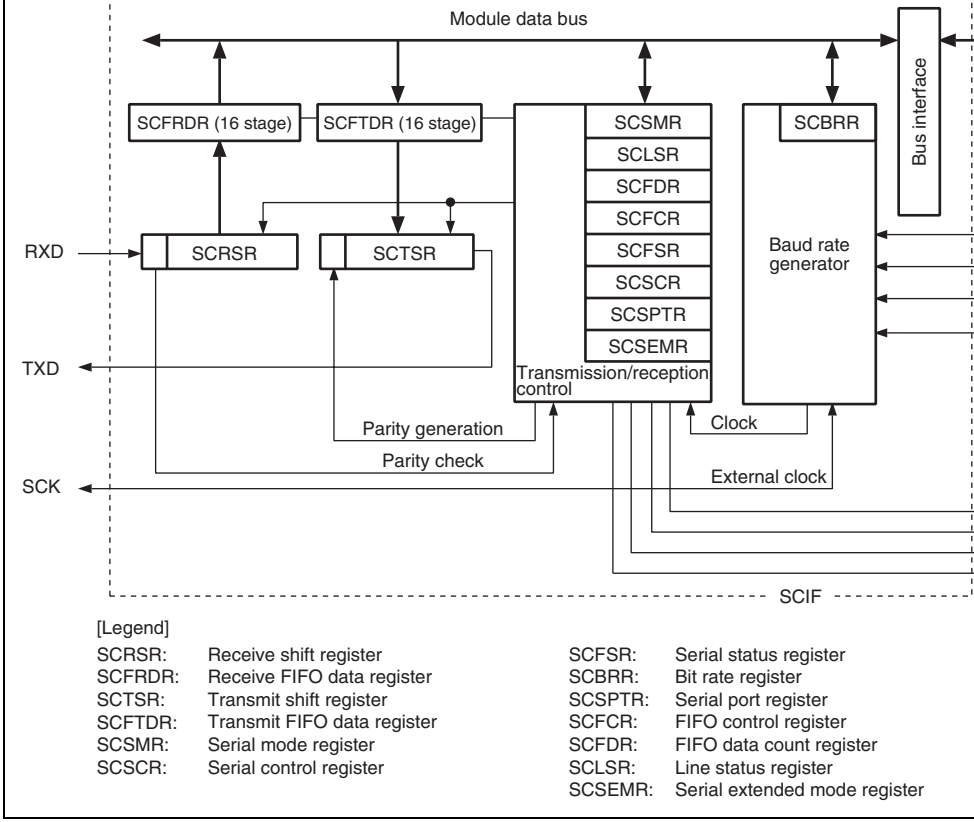


Figure 14.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCI communicate with a universal asynchronous receiver/transmitter (UART), an asy communication interface adapter (ACIA), or any other communications chip that a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least on the space 0 level (low level). It is also detected by reading the RXD level directly serial port register when a framing error occurs.
- Clocked synchronous serial communication:
  - Serial data communication is synchronized with a clock signal. The SCIF can co with other chips having a clocked synchronous communication function. There is data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buff high-speed continuous data transfer is possible in both the transmit and receive direc
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (i SCK pin (external)



[Legend]

SCRSR:	Receive shift register	SCFSR:	Serial status register
SCFRDR:	Receive FIFO data register	SCBRR:	Bit rate register
SCTSR:	Transmit shift register	SCSPTR:	Serial port register
SCFTDR:	Transmit FIFO data register	SCFCR:	FIFO control register
SCSMR:	Serial mode register	SCFDR:	FIFO data count register
SCSCR:	Serial control register	SCLSR:	Line status register
		SCSEMR:	Serial extended mode register

Figure 15.1 Block Diagram of SCIF

	Transmit FIFO data register_0	SCFTDR_0	W	Undefined	H'FFFE800C
	Serial status register_0	SCFSR_0	R/(W)*1	H'0060	H'FFFE8010
	Receive FIFO data register_0	SCFRDR_0	R	Undefined	H'FFFE8014
	FIFO control register_0	SCFCR_0	R/W	H'0000	H'FFFE8018
	FIFO data count register_0	SCFDR_0	R	H'0000	H'FFFE801C
	Serial port register_0	SCSPTR_0	R/W	H'0050	H'FFFE8020
	Line status register_0	SCLSR_0	R/(W)*2	H'0000	H'FFFE8024
1	Serial mode register_1	SCSMR_1	R/W	H'0000	H'FFFE8800
	Bit rate register_1	SCBRR_1	R/W	H'FF	H'FFFE8804
	Serial control register_1	SCSCR_1	R/W	H'0000	H'FFFE8808
	Transmit FIFO data register_1	SCFTDR_1	W	Undefined	H'FFFE880C
	Serial status register_1	SCFSR_1	R/(W)*1	H'0060	H'FFFE8810
	Receive FIFO data register_1	SCFRDR_1	R	Undefined	H'FFFE8814
	FIFO control register_1	SCFCR_1	R/W	H'0000	H'FFFE8818
	FIFO data count register_1	SCFDR_1	R	H'0000	H'FFFE881C
	Serial port register_1	SCSPTR_1	R/W	H'0050	H'FFFE8820
	Line status register_1	SCLSR_1	R/(W)*2	H'0000	H'FFFE8824
	Serial extended mode register_1	SCSEMR_1	R/W	H'00	H'FFFE8900

	FIFO data count register_2	SCFDR_2	R	H'0000	H'FFFE9018
	Serial port register_2	SCSPTR_2	R/W	H'0050	H'FFFE9020
	Line status register_2	SCLSR_2	R/(W)*2	H'0000	H'FFFE9024
	Serial extended mode register_2	SCSEMR_2	R/W	H'00	H'FFFE9100
3	Serial mode register_3	SCSMR_3	R/W	H'0000	H'FFFE9800
	Bit rate register_3	SCBRR_3	R/W	H'FF	H'FFFE9804
	Serial control register_3	SCSCR_3	R/W	H'0000	H'FFFE9808
	Transmit FIFO data register_3	SCFTDR_3	W	Undefined	H'FFFE980C
	Serial status register_3	SCFSR_3	R/(W)*1	H'0060	H'FFFE9810
	Receive FIFO data register_3	SCFRDR_3	R	Undefined	H'FFFE9814
	FIFO control register_3	SCFCR_3	R/W	H'0000	H'FFFE9818
	FIFO data count register_3	SCFDR_3	R	H'0000	H'FFFE981C
	Serial port register_3	SCSPTR_3	R/W	H'0050	H'FFFE9820
	Line status register_3	SCLSR_3	R/(W)*2	H'0000	H'FFFE9824

Notes: 1. Only 0 can be written to clear the flag. Bits 15 to 8, 3, and 2 are read-only bits cannot be modified.
2. Only 0 can be written to clear the flag. Bits 15 to 1 are read-only bits that can modified.

R/W:	-	-	-	-	-	-	-	-

15.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a register that stores serial receive data. The SCIF completes the reception of of serial data by moving the received data from the receive shift register (SCRSR) into SC for storage. Continuous reception is possible until 16 bytes are stored. The CPU can read write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to an undefined value by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

15.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 16-byte FIFO register that stores data for serial transmission. When the SC
that the transmit shift register (SCTSR) is empty, it moves transmit data written in the S
into SCTSR and starts serial transmission. Continuous serial transmission is performed u
is no transmit data left in SCFTDR. The CPU can write to SCFTDR at all times.

When SCFTDR is full of transmit data (16 bytes), no more data can be written. If writin
data is attempted, the data is ignored.

SCFTDR is initialized to an undefined value by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynchro clocked synchronous mode. 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronou In clocked synchronous mode, the data length i 8 bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB the transmit FIFO data register is not transmitted.

				mode (O/E) setting. Receive data pari checked according to the even/odd (O setting.
4	O/$\overline{E}$	0	R/W	Parity mode Selects even or odd parity when parity bits are and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity (PE) is set to 1 to enable parity addition and ch The O/$\overline{E}$ setting is ignored in clocked synchron mode, or in asynchronous mode when parity a and checking is disabled. 0: Even parity*1 1: Odd parity*2 Notes:1. If even parity is selected, the parity b added to transmit data to make an e number of 1s in the transmitted char parity bit combined. Receive data is to see if it has an even number of 1s received character and parity bit com 2. If odd parity is selected, the parity bi to transmit data to make an odd num in the transmitted character and pari combined. Receive data is checked has an odd number of 1s in the rece character and parity bit combined.

				0: One stop bit When transmitting, a single 1-bit is added at of each transmitted character. 1: Two stop bits When transmitting, two 1 bits are added at t each transmitted character.
2	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select Select the internal clock source of the on-chip generator. For further information on the clock bit rate register settings, and baud rate, see se 15.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-e interrupt (TXI) requested when the serial trans is transferred from the transmit FIFO data reg (SCFTDR) to the transmit shift register (SCTS the quantity of data in the transmit FIFO regis becomes less than the specified number of transmission triggers, and when the TDFE fla serial status register (SCFSR) is set to1. 0: Transmit-FIFO-data-empty interrupt reques disabled 1: Transmit-FIFO-data-empty interrupt reques enabled* Note: * The TXI interrupt request can be clea writing a greater quantity of transmit d the specified transmission trigger nur SCFTDR and by clearing TDFE to 0 a reading 1 from TDFE, or can be clear clearing TIE to 0.

				are disabled 1: Receive FIFO data full interrupt (RXI), receiv interrupt (ERI), and break interrupt (BRI) rec are enabled* Note: * RXI interrupt requests can be cleared reading the DR or RDF flag after it has set to 1, then clearing the flag to 0, or clearing RIE to 0. ERI or BRI interrupt can be cleared by reading the ER, BR ORER flag after it has been set to 1, th clearing the flag to 0, or by clearing RI REIE to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the serial transmitter. 0: Transmitter disabled 1: Transmitter enabled* Note: * Serial transmission starts after writing transmit data into SCFTDR. Select the format in SCSMR and SCFCR and res transmit FIFO before setting TE to 1.

				2. Serial reception starts when a start detected in asynchronous mode, or synchronous clock input is detected clocked synchronous mode. Select receive format in SCSMR and SCF reset the receive FIFO before settin
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables the receive-error (ERI) in and break (BRI) interrupts. The setting of REI valid only when RIE bit is set to 0. 0: Receive-error interrupt (ERI) and break inte (BRI) requests are disabled 1: Receive-error interrupt (ERI) and break inte (BRI) requests are enabled* Note: * ERI or BRI interrupt requests can be reading the ER, BR or ORER flag aft been set to 1, then clearing the flag t clearing RIE and REIE to 0. Even if F to 0, when REIE is set to 1, ERI or BI interrupt requests are enabled. Set s wants to inform INTC of ERI or BRI in requests during DMA transfer.

then set CKE[1:0].

- Asynchronous mode

00: Internal clock, SCK pin used for input pin (i
signal is ignored)

01: Internal clock, SCK pin used for clock outp
(The output clock frequency is 16 times the

10: External clock, SCK pin used for clock inpu
(The input clock frequency is 16 times the

11: Setting prohibited

- Clocked synchronous mode

00: Internal clock, SCK pin used for serial clock

01: Internal clock, SCK pin used for serial clock

10: External clock, SCK pin used for serial cloc

11: Setting prohibited

Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PER[3:0]	0000	R	Number of Parity Errors Indicate the quantity of data including a pari the receive data stored in the receive FIFO register (SCFRDR). The value indicated by 12 after the ER bit in SCFSR is set, represe number of parity errors in SCFRDR. When errors have occurred in all 16-byte receive SCFRDR, PER[3:0] shows 0000.
11 to 8	FER[3:0]	0000	R	Number of Framing Errors Indicate the quantity of data including a fra in the receive data stored in SCFRDR. The indicated by bits 11 to 8 after the ER bit in S set, represents the number of framing error SCFRDR. When framing errors have occur 16-byte receive data in SCFRDR, FER[3:0] 0000.

1: A framing error or parity error has occurred

[Setting conditions]

- ER is set to 1 when the stop bit is 0 after whether or not the last stop bit of the rece data is 1 at the end of one data receive operation*2
- ER is set to 1 when the total number of 1s receive data plus parity bit does not matc even/odd parity specified by the O/$\overline{E}$ bit in

Notes: 1. Clearing the RE bit to 0 in SCSCR not affect the ER bit, which retain previous value. Even if a receive occurs, the receive data is transfe SCFRDR and the receive operati continued. Whether or not the da from SCFRDR includes a receive can be detected by the FER and in SCFSR.

2. In two stop bits mode, only the fir bit is checked; the second stop bi checked.

SCFTDR

1: End of transmission

[Setting conditions]

- TEND is set to 1 when the chip is a pow reset
- TEND is set to 1 when TE is cleared to 0 serial control register (SCSCR)
- TEND is set to 1 when SCFTDR does n receive data when the last bit of a one-b character is transmitted

Note: * Do not use this bit as a transmit when the DMAC writes data to S due to a TXI interrupt request.

number

[Clearing conditions]

- TDFE is cleared to 0 when data exceedin specified transmission trigger number is v SCFTDR after 1 is read from TDFE and t written
- TDFE is cleared to 0 when data exceedin specified transmission trigger number is v SCFTDR by the DMAC.

1: The quantity of transmit data in SCFTDR is to or less than the specified transmission t number*

[Setting conditions]

- TDFE is set to 1 by a power-on reset
- TDFE is set to 1 when the quantity of tran data in SCFTDR becomes equal to or les the specified transmission trigger number result of transmission.

Note: * Since SCFTDR is a 16-byte FIFO re the maximum quantity of data that ca written when TDFE is 1 is "16 minus specified transmission trigger numbe attempt is made to write additional d data is ignored. The quantity of data SCFTDR is indicated by the upper 8 SCFDR.

				after it has been set to 1, then writes 0 t 1: Break signal received* [Setting condition] • BRK is set to 1 when data including a fra error is received, and a framing error oc space 0 in the subsequent receive data Note: * When a break is detected, transfer receive data (H'00) to SCFRDR sto detection. When the break ends an receive signal becomes mark 1, the of receive data resumes.
3	FER	0	R	Framing Error Indication Indicates a framing error in the data read fro next receive FIFO data register (SCFRDR) asynchronous mode. 0: No receive framing error occurred in the read from SCFRDR [Clearing conditions] • FER is cleared to 0 when the chip under power-on reset • FER is cleared to 0 when no framing err present in the next data read from SCF 1: A receive framing error occurred in the ne read from SCFRDR. [Setting condition] • FER is set to 1 when a framing error is p the next data read from SCFRDR

- PER is cleared to 0 when no parity error i
  in the next data read from SCFRDR

1: A receive parity error occurred in the next
  from SCFRDR

[Setting condition]

- PER is set to 1 when a parity error is pres
  the next data read from SCFRDR

- RDF is cleared to 0 by a power-on reset
  mode
- RDF is cleared to 0 when the SCFRDR
  until the quantity of receive data in SCF
  becomes less than the specified receive
  number after 1 is read from RDF and the
  written
- RDF is cleared to 0 when SCFRDR is re
  DMAC until the quantity of receive data
  SCFRDR becomes less than the specifi
  trigger number.

1: The quantity of receive data in SCFRDR
  than the specified receive trigger number

[Setting condition]

- RDF is set to 1 when a quantity of receiv
  more than the specified receive trigger n
  stored in SCFRDR*

Note: * As SCFTDR is a 16-byte FIFO regi
  maximum quantity of data that can
  when RDF is 1 becomes the specif
  receive trigger number. If an attemp
  to read after all the data in SCFRDR
  been read, the data is undefined. T
  quantity of receive data in SCFRDR
  indicated by the lower 8 bits of SCF

[Clearing conditions]

- DR is cleared to 0 when the chip undergo power-on reset
- DR is cleared to 0 when all receive data a after 1 is read from DR and then 0 is writt
- DR is cleared to 0 when all receive data i SCFRDR are read by the DMAC.

1: Next receive data has not been received

[Setting condition]

- DR is set to 1 when SCFRDR contains le than the specified receive trigger number next data has not yet been received after elapse of 15 ETU from the last stop bit.*

Note: * This is equivalent to 1.5 frames with 1-stop-bit format. (ETU: elementary

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

Initial value:	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

The SCBRR setting is calculated as follows:

- Asynchronous mode:

  (1) In normal mode (when the ABCS bit in SCSEMR is 0)

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

  (2) In serial extended mode (when the ABCS bit in SCSEMR is 1)

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$

- Clocked synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

  B: Bit rate (bits/s)
  N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
  (The setting must satisfy the electrical characteristics.)
  $P\phi$: Operating frequency for peripheral modules (MHz)
  n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and value see table 15.3.)

(1) In normal mode (when the ABCS bit in SCSEMR is 0)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

(2) In serial extended mode (when the ABCS bit in SCSEMR is 1)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 15.4 lists examples of SCBRR settings in asynchronous mode, and table 15.5 lists e
of SCBRR settings in clocked synchronous mode.

2400	1	103	0.16	1	116	0.16	1	129
4800	1	51	0.16	1	58	–0.69	1	64
9600	0	103	0.16	0	116	0.16	0	129
19200	0	51	0.16	0	58	–0.69	0	64
31250	0	31	0.00	0	35	0.00	0	39
38400	0	25	0.16	0	28	1.02	0	32

Note: Settings with an error of 1% or less are recommended.

25 k	1	79	1	89	1	97
50 k	1	39	1	44	1	48
100 k	0	79	0	89	0	97
250 k	0	31	0	35	0	38
500 k	0	15	0	17	0	19
1 M	0	7	0	8	0	9
2 M	0	3	—	—	0	4

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

32	1000000	0	0
36	1125000	0	0
40	1250000	0	0

Table 15.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (b
32	8.0000	500000
36	9.0000	562500
40	10.0000	625000

Table 15.8 Maximum Bit Rates with External Clock Input (Clocked Synchronous Mode, $t_{Scyc} = 12t_{pcyc}$)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (b
32	2.6667	2666666.7
36	3.0000	3000000.0
40	3.3333	3333333.3

Note: Confirm that these bit rates meet the electrical characteristics of this LSI and the communication device.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
7, 6	RTRG[1:0]	00	R/W	Receive FIFO Data Trigger Set the quantity of receive data which sets the r data full (RDF) flag in the serial status register (The RDF flag is set to 1 when the quantity of re data stored in the receive FIFO register (SCFR increased more than the set trigger number sho below. • Asynchronous mode 00: 1 01: 4 10: 8 11: 14 • Clocked synchron 00: 1 01: 2 10: 8 11: 14 Note: In clock synchronous mode, to transfer receive data using DMAC, set the recei number to 1. If set to other than 1, CPU read the receive data left in SCFRDR.

				10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the num empty bytes in SCFTDR when the TD set to 1.
3	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIFO register and resets the data to the empty state 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a pow reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in the receive FIFO register and resets the data to the empty state 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a pow reset.
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TX receive input pin (RXD) and internally connect pin and $\overline{\text{CTS}}$ pin and enables loop-back testing 0: Loop back test disabled 1: Loop back test enabled

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
12 to 8	T[4:0]	00000	R	T4 to T0 bits indicate the quantity of non-transm data stored in SCFTDR. H'00 means no transm and H'10 means that SCFTDR is full of transm
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
4 to 0	R[4:0]	00000	R	R4 to R0 bits indicate the quantity of receive da stored in SCFRDR. H'00 means no receive dat H'10 means that SCFRDR full of receive data.

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/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
3	SCKIO	0	R/W	SCK Port Input/Output Indicates input or output of the serial port SCI When the SCK pin is actually used as a port c the SCKDT bit value, the CKE[1:0] bits in SCS should be cleared to 0. 0: SCKDT bit value not output to SCK pin 1: SCKDT bit value output to SCK pin
2	SCKDT	0	R/W	SCK Port Data Indicates the input/output data of the serial po pin. Input/output is specified by the SCKIO bit output, the SCKDT bit value is output to the S The SCK pin status is read from the SCKDT b regardless of the SCKIO bit setting. However, input/output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level

data of the TXD pin used as serial ports. Input/
specified by the SPB2IO bit. When the TXD pi
output, the SPB2DT bit value is output to the T
The RXD pin status is read from the SPB2DT b
regardless of the SPB2IO bit setting. However,
input and TXD output must be set in the PFC.

0: Input/output data is low level

1: Input/output data is high level

15.3.12 Line Status Register (SCLSR)

The CPU can always read or write to SCLSR, but cannot write 1 to the ORER flag. This
be cleared to 0 only if it has first been read (after being set to 1).

SCLSR is initialized to H'0000 by a power-on reset.

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

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

reset

- ORER is cleared to 0 when 0 is written a
  read from ORER.

1: An overrun error has occurred*[2]

[Setting condition]

- ORER is set to 1 when the next serial re
  finished while the receive FIFO is full of
  receive data.

Notes: 1. Clearing the RE bit to 0 in SCSC
not affect the ORER bit, which r
previous value.

2. The receive FIFO data register (
retains the data before an overru
has occurred, and the next rece
is discarded. When the ORER b
1, the SCIF cannot continue the
serial reception.

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

Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ABCS	0	R/W	Asynchronous Basic Clock Select Selects the basic clock for 1-bit period in asynchronous mode. Setting of ABCS is valid when the asynchron mode bit (C/$\overline{A}$ in SCSMR) = 0. 0: Basic clock with a frequency of 16 times th transfer rate 1: Basic clock with a frequency of 8 times the rate
6 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write va should always be 0.

The transmission format is selected in the serial mode register (SCSMR), as shown in ta
The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the
control register (SCSCR), as shown in table 15.10.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the
  selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data f
  overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of on-chip
    generator.
  - When an external clock is selected, the external clock input must have a frequenc
    the bit rate. (The on-chip baud rate generator is not used.)

(2) Clocked Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of the on-c
    rate generator, and outputs this clock to external devices as the synchronous cloc
  - When an external clock is selected, the SCIF operates on the input synchronous c
    using the on-chip baud rate generator.

			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clocked synchronous	8 bits	Not set	None

[Legend]

x: Don't care

Table 15.10 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR	SCSCR		SCIF Transmit/Receive Cloc	
Bit 7	Bit 1	Bit 0		Clock Source	SCK Pin Function
C/$\overline{A}$	CKE1	CKE0	Mode		
0	0	0	Asynchronous	Internal	SCIF does not use the SCK
		1			Outputs a clock with a freq times the bit rate
	1	0		External	Inputs a clock with frequen times the bit rate
		1		Setting prohibited	
1	0	x	Clocked synchronous	Internal	Outputs the serial clock
	1	0		External	Inputs the serial clock
		1		Setting prohibited	

[Legend]

x: Don't care

In asynchronous serial communication, the communication line is normally held in the r
(high) state. The SCIF monitors the line and starts serial communication when the line g
space (low) state, indicating a start bit. One serial character consists of a start bit (low), d
first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIF synchronizes at the falling edge of the
The SCIF samples each data bit on the eighth pulse of a clock with a frequency 16 times
rate. Receive data is latched at the center of each bit.

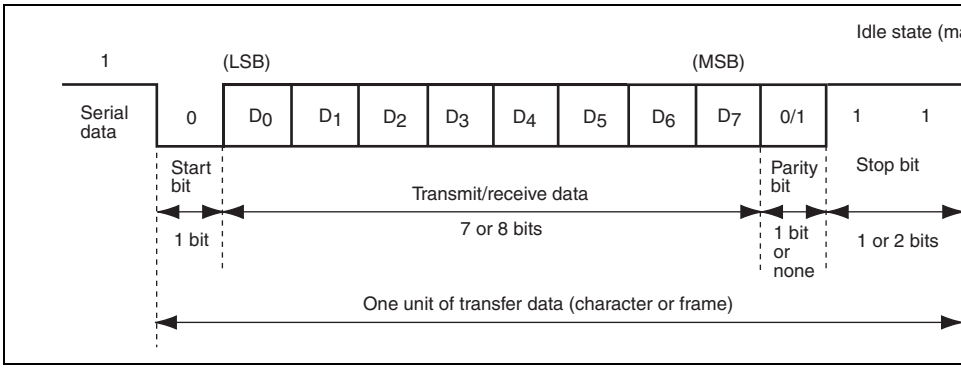


Figure 15.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

Note: * This is an example when ABCS = 0 in SCSEMR. When ABCS = 1, a freque
times the bit rate becomes the basic clock, and receive data is sampled at the
rising edge of the basic clock.

0	0	1	START	8-bit data		STOP	STO
0	1	0	START	8-bit data		P	STO
0	1	1	START	8-bit data		P	STO
1	0	0	START	7-bit data	STOP		
1	0	1	START	7-bit data	STOP	STOP	
1	1	0	START	7-bit data	P	STOP	
1	1	1	START	7-bit data	P	STOP	STO

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit

(2) Clock

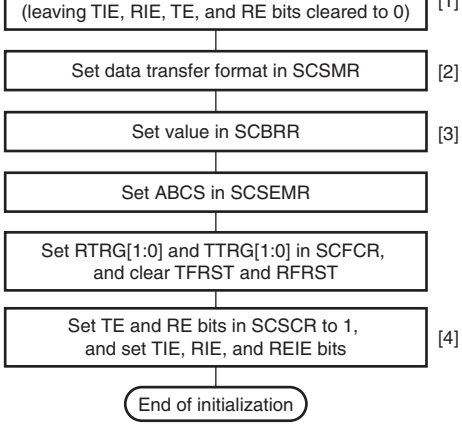
An internal clock generated by the on-chip baud rate generator or an external clock input SCK pin can be selected as the SCIF transmit/receive clock. The clock source is selected C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE[1:0] in the serial control regis (SCSCR). For clock source selection, refer to table 15.10, SCSMR and SCSCR Settings a Clock Source Selection.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 time desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal on the SCK pin frequency of this output clock is 16 times the desired bit rate.

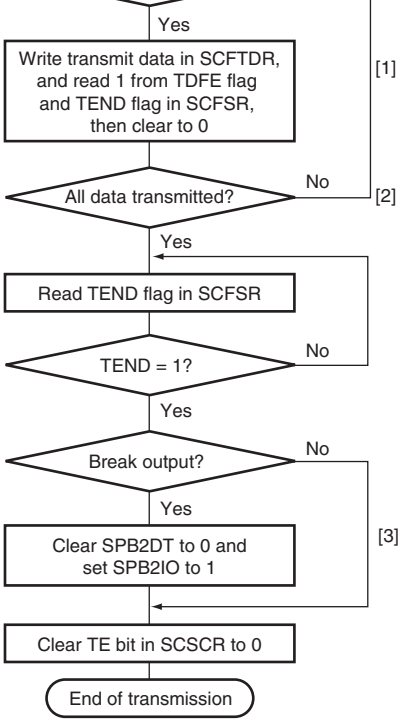
TEND flag in the SCFSR is set. The TE bit can be cleared to 0 during transmission, but transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SC and reset SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or operation. SCIF operation becomes unreliable if the clock is stopped.



[4] Set the TE bit or RE bit in SCSCR to 1. Also set the RIE, REIE, and TIE bits. Setting the TE and RE bits enables the TxD and RxD pins to be used.
When transmitting, the SCIF will go to the mark state; when receiving, it will go to the idle state, waiting for a start bit.

Figure 15.3 Sample Flowchart for SCIF Initialization



The quantity of transmit data that can be written is 16 - (transmit trigger set number).

[2] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

[3] Break output during serial transmission:
To output a break in serial transmission, clear the SPB2DT bit to 0 and set the SPB2IO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.

In [1] and [2], it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of SCFDR.

Figure 15.4 Sample Flowchart for Transmitting Serial Data

generated.

The serial transmit data is sent from the TXD pin in the following order.

A. Start bit: One-bit 0 is output.
B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parit not output can also be selected.)
D. Stop bit(s): One or two 1 bits (stop bits) are output.
E. Mark state: 1 is output continuously until the start bit that starts the next transmiss sent.

3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If d present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then transmission of the next frame is started.

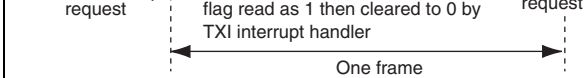
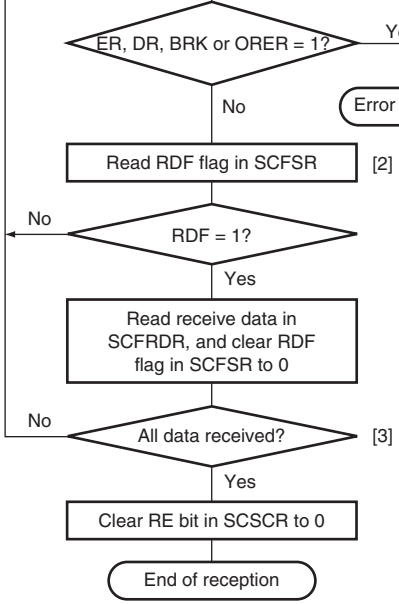


Figure 15.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)



and ORER flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD pin.

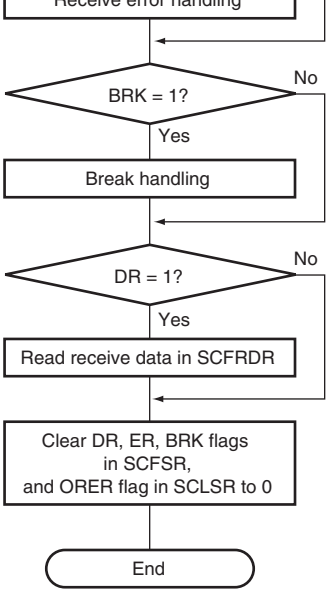
[2] SCIF status check and receive data read:

Read SCFSR and check that RDF flag = 1, then read the receive data in SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can be identified by an RXI interrupt.

[3] Serial reception continuation procedure:

To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading from SCRFDR.

Figure 15.6 Sample Flowchart for Receiving Serial Data



stored.

Figure 15.7 Sample Flowchart for Receiving Serial Data (cont)

(SCRSR) to SCFRDR.

C. Overrun check: The SCIF checks that the ORER flag is 0, indicating that the overr has not occurred.

D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break sta set.

If all the above checks are passed, the receive data is stored in SCFRDR.

Note: When a parity error or a framing error occurs, reception is not suspended.

4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-I data-full interrupt (RXI) request is generated. If the RIE bit or the REIE bit in SCSCR 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated. RIE bit or the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to break reception interrupt (BRI) request is generated.

Figure 15.8 shows an example of the operation for reception.

error

Figure 15.8 Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

15.4.3 Operation in Clocked Synchronous Mode

In clocked synchronous mode, the SCIF transmits and receives data in synchronization v pulses. This mode is suitable for high-speed serial communication.

The SCIF transmitter and receiver are independent, so full-duplex communication is pos while sharing the same clock. The transmitter and receiver are also 16-byte FIFO buffer continuous transmitting or receiving is possible by reading or writing data while transmi receiving is in progress.

Figure 15.9 shows the general format in clocked synchronous serial communication.

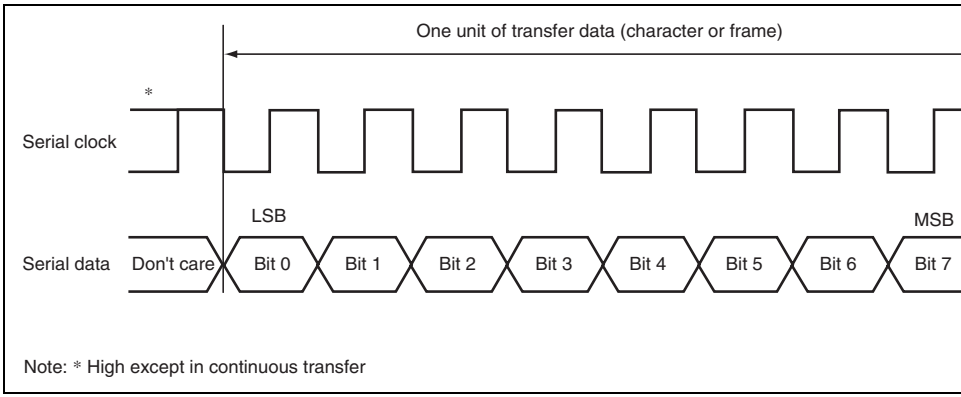


Figure 15.9 Data Format in Clocked Synchronous Communication

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

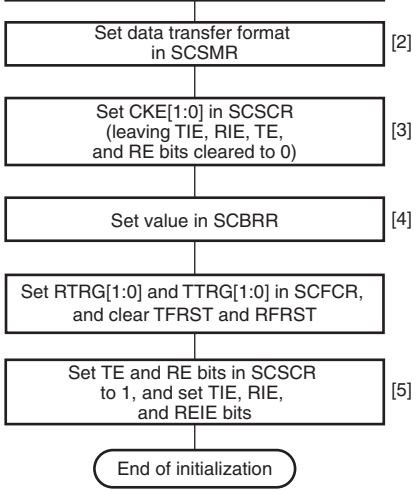
An internal clock generated by the on-chip baud rate generator by the setting of the C/A b
SCSMR and CKE[1:0] in SCSCR, or an external clock input from the SCK pin can be se
the SCIF transmit/receive clock.

When the SCIF operates on an internal clock, it outputs the clock signal at the SCK pin. E
clock pulses are output per transmitted or received character. When the SCIF is not transı
receiving, the clock signal remains in the high state. When only receiving, the clock signa
while the RE bit of SCSCR is 1 and the number of data in receive FIFO is more than the
FIFO data trigger number.

(3) Transmitting and Receiving Data

- **SCIF Initialization (Clocked Synchronous Mode)**

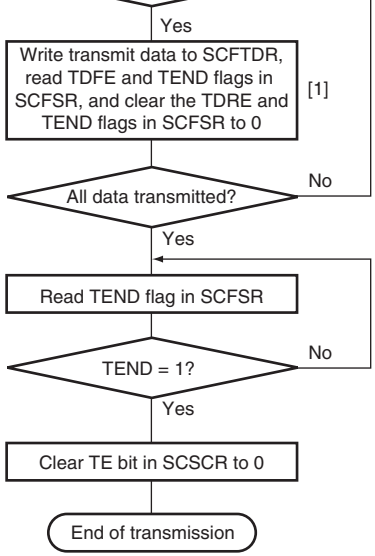
Before transmitting, receiving, or changing the mode or communication format, the softw
clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SC
Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing RE to 0, howeve
not initialize the RDF, PER, FER, and ORER flags and receive data register (SCRDR), w
retain their previous contents.



is not necessary if an external clock is used.

[5] Set the TE or RE bit in SCSCR to 1. Also set the TIE, RIE, and REIE bits to enable the TXD, RXD, and SCK pins to be used. When transmitting, the TXD pin will go to the mark state. When receiving in clocked synchronous mode with the synchronization clock output (clock master) selected, a clock starts to be output from the SCK pin at this point.

Figure 15.10 Sample Flowchart for SCIF Initialization



clear these flags to 0.

[2] Serial transmission continuation procedeure:

To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, them write data to SCFTDR, and then clear the TDFE flag to 0.

Figure 15.11 Sample Flowchart for Transmitting Serial Data

generated.

If clock output mode is selected, the SCIF outputs eight synchronous clock pulses. If external clock source is selected, the SCIF outputs data in synchronization with the i clock. Data is output from the TXD pin in order from the LSB (bit 0) to the MSB (bi

3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit 7 is present, the data is transferred from SCFTDR to SCTSR, and then serial transmiss next frame is started. If there is no data, the TXD pin holds the state after the TEND SCFSR is set to 1 and the MSB (bit 7) is sent.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 15.12 shows an example of SCIF transmit operation.

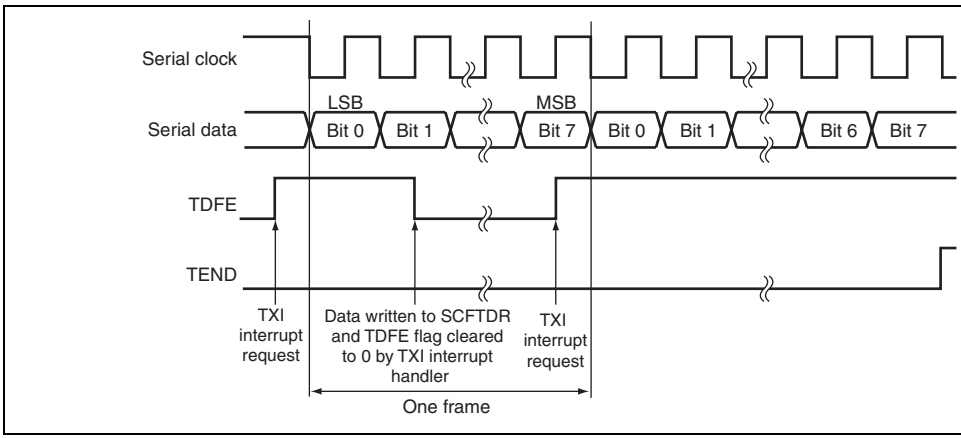
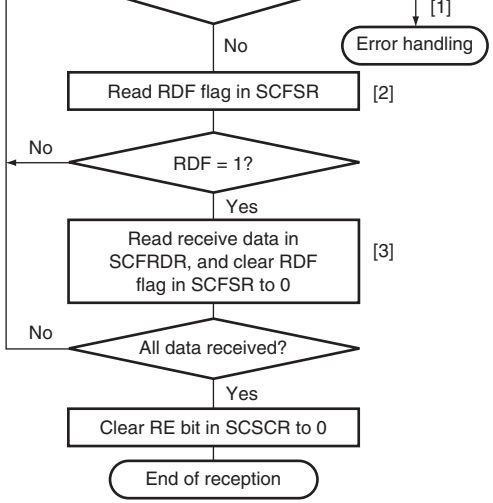


Figure 15.12 Example of SCIF Transmit Operation



[2] SCIF status check and receive data read
Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transitio of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[3] Serial reception continuation procedure:
To continue serial reception, read at leas the receive trigger set number of receive data bytes from SCFRDR, read 1 from th RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading SCFRDR. However, the RDF bit is cleared to 0 automatically when an RXI interrupt activates the DMAC to read the data in SCFRDR.

Figure 15.13 Sample Flowchart for Receiving Serial Data (1)

Figure 15.14 Sample Flowchart for Receiving Serial Data (2)

Figure 15.15 shows an example of SCIF receive operation.

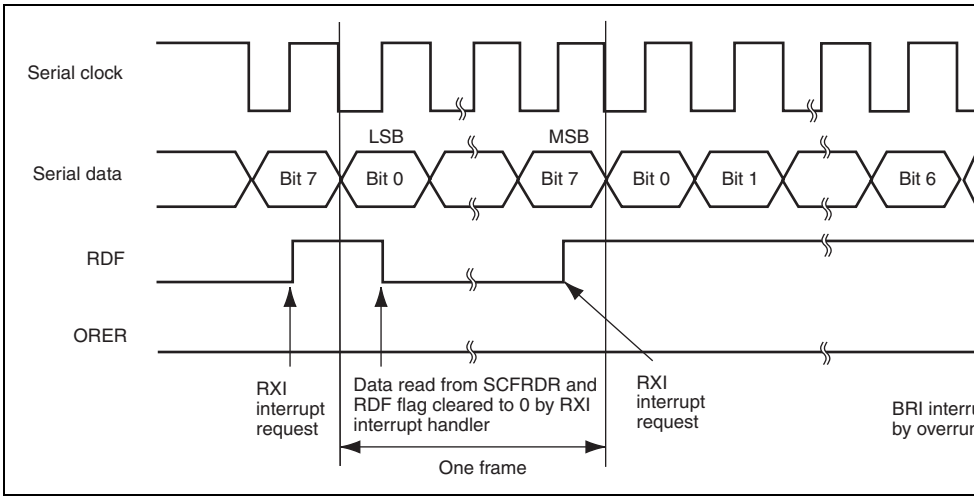
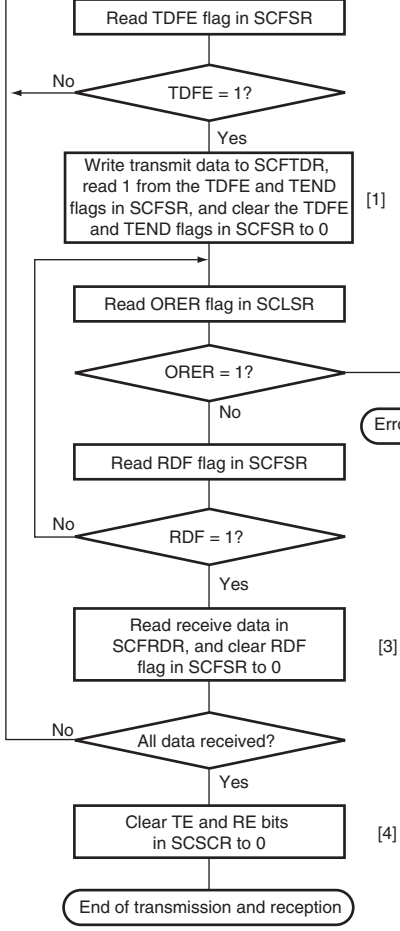


Figure 15.15 Example of SCIF Receive Operation



of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error handling:

Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[3] SCIF status check and receive data read:

Read SCFSR and check that RDF flag = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial transmission and reception continuation procedure:

To continue serial transmission and reception, read 1 from the RDF flag and the receive data in SCFRDR, and clear the RDF flag to 0 before receiving the MSB in the current frame. Similarly, read 1 from the TDFE flag to confirm that writing is possible before transmitting the MSB in the current frame. Then write data to SCFTDR and clear the TDFE flag

Note: When switching from a transmit operatio or receive operation to simultaneous transmission and reception operations, clear the TE and RE bits to 0, and then set them simultaneously to 1.

Figure 15.16 Sample Flowchart for Transmitting/Receiving Serial Data

transfer performed by this TXI interrupt request. At this time, an interrupt request is not s
CPU.

When an RXI request is enabled by the RIE bit and the RDFE flag or the DR flag in SCF
to 1, an RXI interrupt request is generated. The DMAC can be activated and data transfer
performed by this RXI interrupt request. At this time, an interrupt request is not sent to th
The RXI interrupt request caused by the DR flag is generated only in asynchronous mode

When the RIE bit is set to 0 and the REIE bit is set to 1, the SCIF requests only an ERI in
without requesting an RXI interrupt.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indica
there is receive data in SCFRDR.

Table 15.12 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority Reset
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	Not possible	High
ERI	Interrupt initiated by receive error (ER)	Not possible	
RXI	Interrupt initiated by receive FIFO data full (RDF) or data ready (DR)	Possible	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	Possible	Low

However, if the number of data bytes written in SCFTDR is equal to or less than the tra
trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to
flag clearing should therefore be carried out when SCFTDR contains more than the tran
trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of the
data count register (SCFDR).

15.6.2 SCFRDR Reading and RDF Flag

The RDF flag in the serial status register (SCFSR) is set when the number of receive dat
the receive FIFO data register (SCFRDR) has become equal to or greater than the receiv
number set by bits RTRG[1:0] in the FIFO control register (SCFCR). After RDF flag is
receive data equivalent to the trigger number can be read from SCFRDR, allowing effici
continuous reception.

However, if the number of data bytes in SCFRDR exceeds the trigger number, the RDF
be set to 1 again if it is cleared to 0. The RDF flag should therefore be cleared to 0 after
as 1 after reading the number of the received data in the receive FIFO data register (SCF
which is less than the trigger number.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the
count register (SCFDR).

and the parity error flag (PER) may also be set.

Note that, although transfer of receive data to SCFRDR is halted in the break state, the SC
receiver continues to operate.

15.6.5 Sending a Break Signal

The I/O condition and level of the TXD pin are determined by the SPB2IO and SPB2DT
the serial port register (SCSPTR). This feature can be used to send a break signal.

Until TE bit is set to 1 (enabling transmission) after initializing, the TXD pin does not wo
During the period, mark status is performed by the SPB2DT bit. Therefore, the SPB2IO a
SPB2DT bits should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating
level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, th
transmitter is initialized regardless of the current transmission state, and 0 is output from
pin.

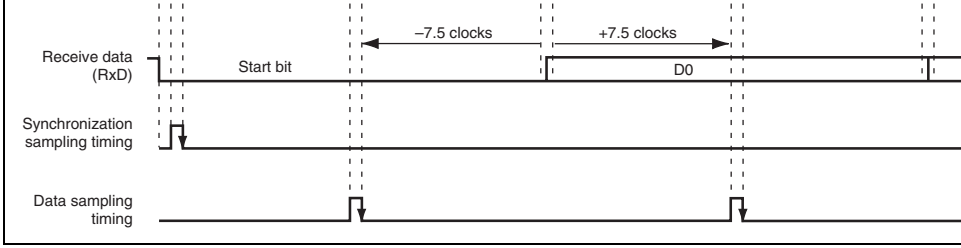


Figure 15.17 Receive Data Sampling Timing in Asynchronous Mode

Note: * This is an example when ABCS = 0 in SCSEMR. When ABCS = 1, a frequency
the bit rate becomes the basic clock, and receive data is sampled at the fourth ris
of the basic clock.

L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0, D = 0.5, and N = 16, the receive margin is 46.875%, as given b equation 2.

Equation 2:

When D = 0.5 and F = 0:

$M = (0.5 - 1/(2 \times 16)) \times 100\%$

$= 46.875\%$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%

15.6.7 FER and PER Flags in the Serial Status Register (SCFSR)

The FER (framing error) and PER (parity error) flags in the serial status register (SCFSR) status flags of the receive FIFO data register (SCFRDR) to be read next. If the CPU or DI reads the receive FIFO data register, the FER (framing error) and PER (parity error) flags current receive data will be lost. To check the framing error and parity error status of the receive data correctly, the serial status register (SCFSR) should be read before the receive data register is read.

Since the shift register, transmit data register, and receive data register are independe each other, the continuous transmission/reception can be performed.

I²C bus format:

- 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/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full slave-address match), arbitration lost, NACK detection, and stop condition detection
- The direct memory access controller (DMAC) can be activated by a transmit-data-er request or receive-data-full request to transfer data.
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus d function is selected.

Clocked synchronous serial format:

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error
- The direct memory access controller (DMAC) can be activated by a transmit-data-er request or receive-data-full request to transfer data.

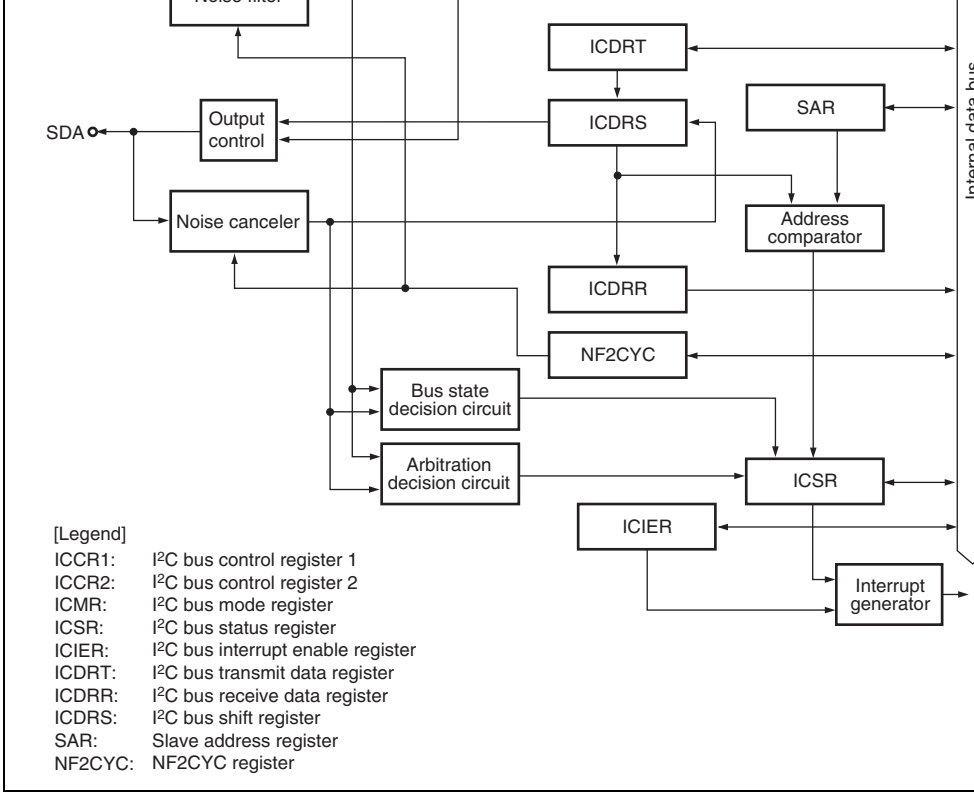
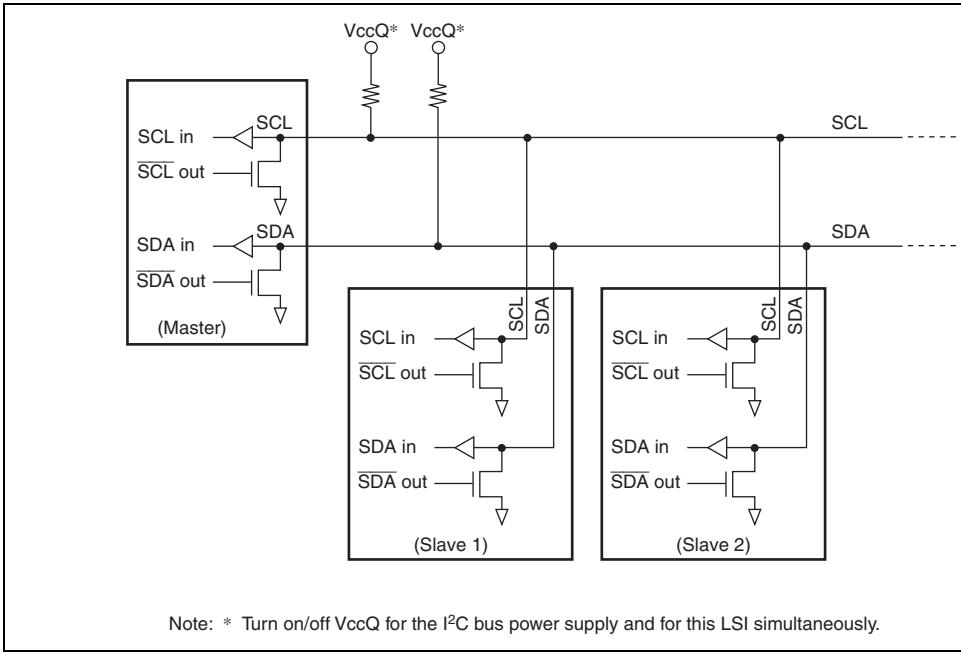


Figure 16.1 Block Diagram of I²C Bus Interface 3

Figure 16.2 shows an example of I/O pin connections to external circuits.



Note: * Turn on/off VccQ for the I^2C bus power supply and for this LSI simultaneously.

Figure 16.2 External Circuit Connections of I/O Pins

I^2C bus interrupt enable register	ICIER	R/W	H'00	H'FFFEE003	8
I^2C bus status register	ICSR	R/W	H'00	H'FFFEE004	8
Slave address register	SAR	R/W	H'00	H'FFFEE005	8
I^2C bus transmit data register	ICDRT	R/W	H'FF	H'FFFEE006	8
I^2C bus receive data register	ICDRR	R/W	H'FF	H'FFFEE007	8
NF2CYC register	NF2CYC	R/W	H'00	H'FFFEE008	8

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I^2C Bus Interface 3 Enable 0: This module is halted. (SCL and SDA pins as ports.) 1: This bit is enabled for transfer operations. SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable Enables or disables continuous reception wh 0 and ICDRR is not read. If ICDRR cannot b the rising of 8th clock cycle of SCL in master mode, reception in byte units should be perfo setting the RCVD bit to 1. 0: Enables continuous reception 1: Disables continuous reception

				SAR and the 8th bit is set to 1, TRS is automa set to 1. If an overrun error occurs in master r mode with the clocked synchronous serial forr is cleared and the mode changes to slave rec mode. Operating modes are described below accord MST and TRS combination. When clocked synchronous serial format is selected and MS clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3 to 0	CKS[3:0]	0000	R/W	Transfer Clock Select These bits should be set according to the nec transfer rate (table 16.3) in master mode.

			1	Pϕ/92	347.8	391.3	434.8
		1	0	Pϕ/100	320.0	360.0	400.0
			1	Pϕ/108	296.3	333.3	370.4
1	0	0	0	Pϕ/176	181.8	204.5	227.3
			1	Pϕ/208	153.8	173.1	192.3
		1	0	Pϕ/256	125.0	140.6	156.3
			1	Pϕ/288	111.1	125.0	138.9
	1	0	0	Pϕ/336	95.2	107.1	119.0
			1	Pϕ/368	87.0	97.8	108.7
		1	0	Pϕ/400	80.0	90.0	100.0
			1	Pϕ/432	74.1	83.3	92.6

Note: The settings should satisfy external specifications.

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy Enables to confirm whether the I^2C bus is occu released and to issue start/stop conditions in m mode. With the clocked synchronous serial for bit is always read as 0. With the I^2C bus format is set to 1 when the SDA level changes from hi under the condition of SCL = high, assuming th start condition has been issued. This bit is clea when the SDA level changes from low to high u condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY an SCP to issue a start condition. Follow this proc when also re-transmitting a start condition. Wri BBSY and 0 in SCP to issue a stop condition.
6	SCP	1	R/W	Start/Stop Issue Condition Disable Controls the issue of start/stop conditions in ma mode. To issue a start condition, write 1 in BBS in SCP. A retransmit start condition is issued in same way. To issue a stop condition, write 0 in and 0 in SCP. This bit is always read as 1. Eve written to this bit, the data will not be stored.

				(outputs high by external pull-up resistance
4	SDAOP	1	R/W	SDAO Write Protect Controls change of output level of the SDA pi modifying the SDAO bit. To change the outpu clear SDAO and SDAOP to 0 or set SDAO to clear SDAOP to 0. This bit is always read as
3	SCLO	1	R	SCL Output Level Monitors SCL output level. When SCLO is 1, outputs high. When SCLO is 0, SCL pin outpu
2	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.
1	IICRST	0	R/W	IIC Control Part Reset Resets the control part except for I^2C register is set to 1 when hang-up occurs because of communication failure during I^2C bus operatio IIC3 registers and the control part can be rese
0	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.

	R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I^2C bus format is use
6	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
5, 4	—	All 1	R	Reserved These bits are always read as 1. The write valu always be 1.
3	BCWP	1	R/W	BC Write Protect Controls the BC[2:0] modifications. When mod BC[2:0] bits, this bit should be cleared to 0. In synchronous serial mode, the BC[2:0] bits sho be modified. 0: When writing, values of the BC[2:0] bits are 1: When reading, 1 is always read. When writing, settings of the BC[2:0] bits are

transfer, including the acknowledge bit. These
cleared by a power-on reset and in software s
mode and module standby mode. These bits
cleared by setting the IICRST bit of ICCR2 to
the clocked synchronous serial format, these
should not be modified.

I^2C Bus Format	Clocked Synchronous Ser
000: 9 bits	000: 8 bits
001: 2 bits	001: 1 bit
010: 3 bits	010: 2 bits
011: 4 bits	011: 3 bits
100: 5 bits	100: 4 bits
101: 6 bits	101: 5 bits
110: 7 bits	110: 6 bits
111: 8 bits	111: 7 bits

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

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 or 0, thi enables or disables the transmit data empty int (TXI). 0: Transmit data empty interrupt request (TXI) disabled. 1: Transmit data empty interrupt request (TXI) enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt the rising of the ninth clock while the TDRE bit is 1. TEI can be canceled by clearing the TEND the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disab 1: Transmit end interrupt request (TEI) is enab
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interru request (RXI) and the overrun error interrupt re (ERI) in the clocked synchronous format when data is transferred from ICDRS to ICDRR and RDRF bit in ICSR is set to 1. RXI can be cance clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) are 1: Receive data full interrupt request (RXI) are

3	STIE	0	R/W	Stop Condition Detection Interrupt Enable Enables or disables the stop condition detecti interrupt request (STPI) when the STOP bit in set. 0: Stop condition detection interrupt request (disabled. 1: Stop condition detection interrupt request (enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continu transfer is halted.
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. This bit can be canceled by setti BBSY bit in ICCR2 to 1. 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	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Clearing conditions] • When 0 is written in TDRE after reading TD • When data is written to ICDRT [Setting conditions] • When data is transferred from ICDRT to IC ICDRT becomes empty • When TRS is set • When the start condition (including retransr is issued • When slave mode is changed from receive transmit mode
6	TEND	0	R/W	Transmit End [Clearing conditions] • When 0 is written in TEND after reading TE • When data is written to ICDRT [Setting conditions] • When the ninth clock of SCL rises with the format while the TDRE flag is 1 • When the final bit of transmit frame is sent clocked synchronous serial format

				• When 0 is written in NACKF after reading = 1 [Setting condition] • When no acknowledge is detected from th device in transmission while the ACKE bit is 1
3	STOP	0	R/W	Stop Condition Detection Flag [Clearing condition] • When 0 is written in STOP after reading S [Setting conditions] • In master mode, when a stop condition is after frame transfer • In slave mode, when the slave address in byte after the general call and detecting st condition matches the address set in SAR the stop condition is detected

				[Clearing condition] • When 0 is written in AL/OVE after reading A = 1 [Setting conditions] • If the internal SDA and SDA pin disagree a of SCL in master transmit mode • When the SDA pin outputs high in master r while a start condition is detected • When the final bit is received with the clock synchronous format while RDRF = 1
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the frame following a start condition matches bits S in SAR. [Clearing condition] • When 0 is written in AAS after reading AAS [Setting conditions] • When the slave address is detected in slav mode • When the general call address is detected receive mode.
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode with the format. [Clearing condition] • When 0 is written in ADZ after reading ADZ [Setting condition] • When the general call address is detected receive mode

Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA[6:0]	0000000	R/W	Slave Address These bits set a unique address in these b differing form the addresses of other slave connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected 1: Clocked synchronous serial format is se

16.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT space in the shift register (ICDRS), it transfers the transmit data which is written in ICD ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. ICDRT is initialized to H'FF

Bit:	7	6	5	4	3	2	1	0
Initial value:	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	R/W	R/W

16.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferre ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred ICDRS to ICDRR after data of one byte is received. This register cannot be read directly CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
0	NF2CYC	0	R/W	Noise Filtering Range Select 0: The noise less than one cycle of the periph can be filtered out 1: The noise less than two cycles of the perip can be filtered out

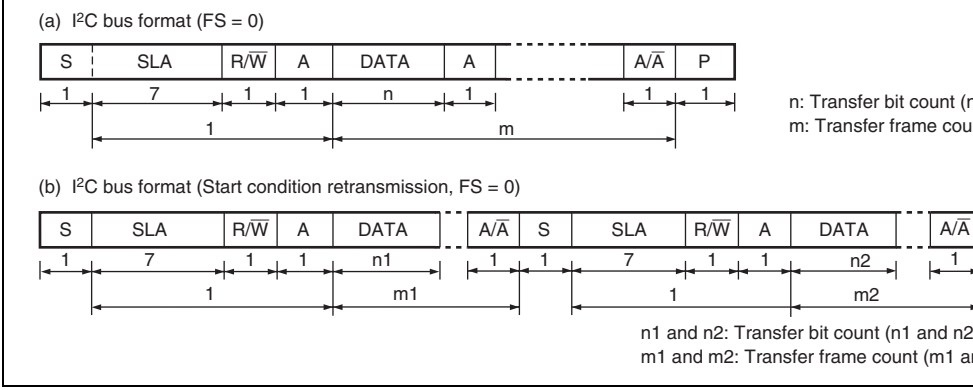


Figure 16.3 I²C Bus Formats

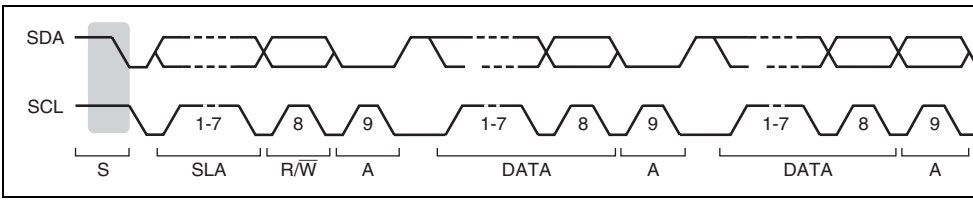


Figure 16.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

issued). This generates the start condition.

3. After confirming that TDRE in ICSR has been set, write the transmit data (the first b
show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cle
and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR i
at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confir
slave device has been selected. Then, write second byte data to ICDRT. When ACK
the slave device has not been acknowledged, so issue the stop condition. To issue the
condition, write 0 to BBSY and SCP. SCL is fixed low until the transmit data is prep
the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the e
byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1)
receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TE
NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to slave receive mode.

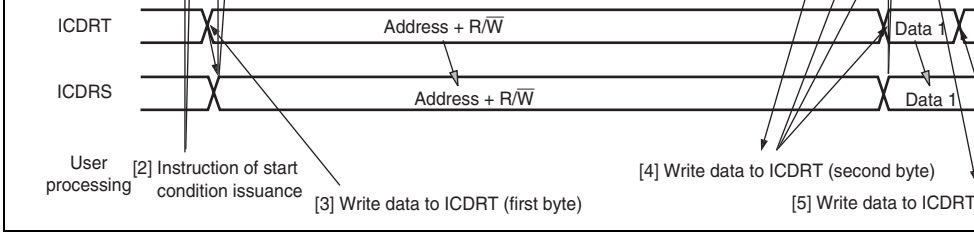


Figure 16.5 Master Transmit Mode Operation Timing (1)

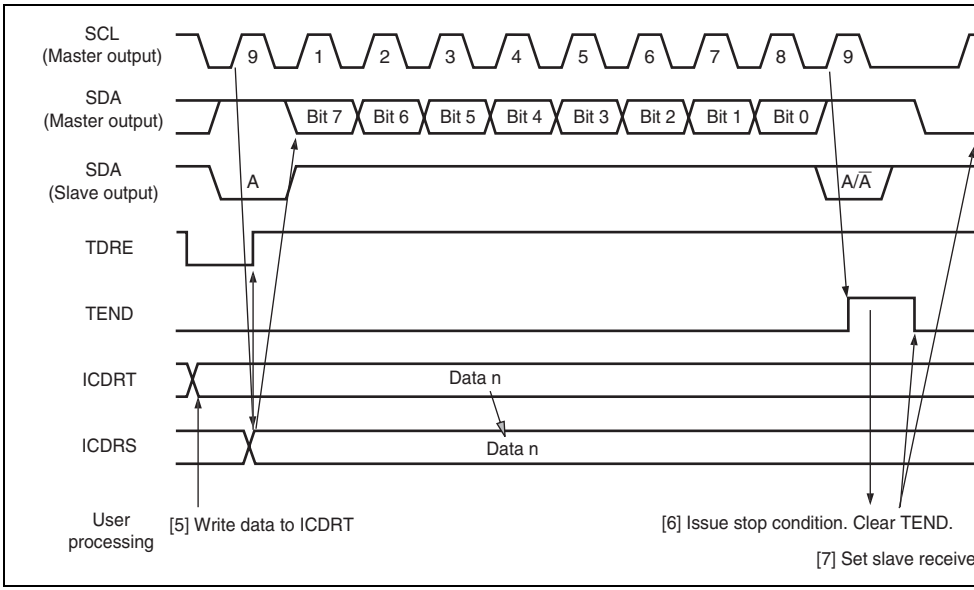


Figure 16.6 Master Transmit Mode Operation Timing (2)

level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.

3. After the reception of first frame data is completed, the RDRF bit in ICSR is set to 1 of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, a is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. I receive clock pulse falls after reading ICDRR by the other processing while RDRF i fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage c
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to slave receive mode.

Note: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in I set.

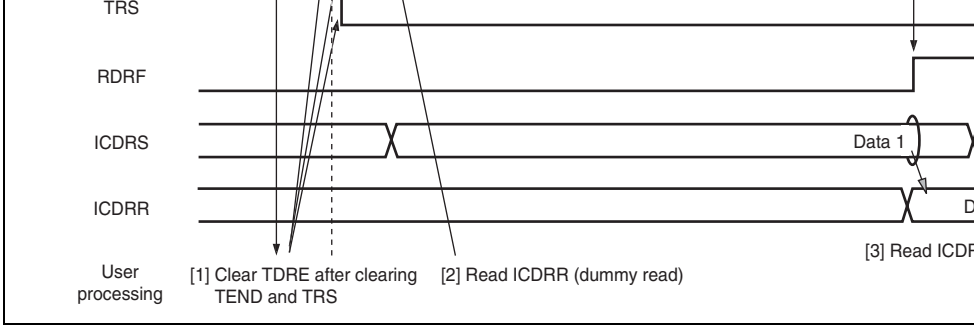


Figure 16.7 Master Receive Mode Operation Timing (1)

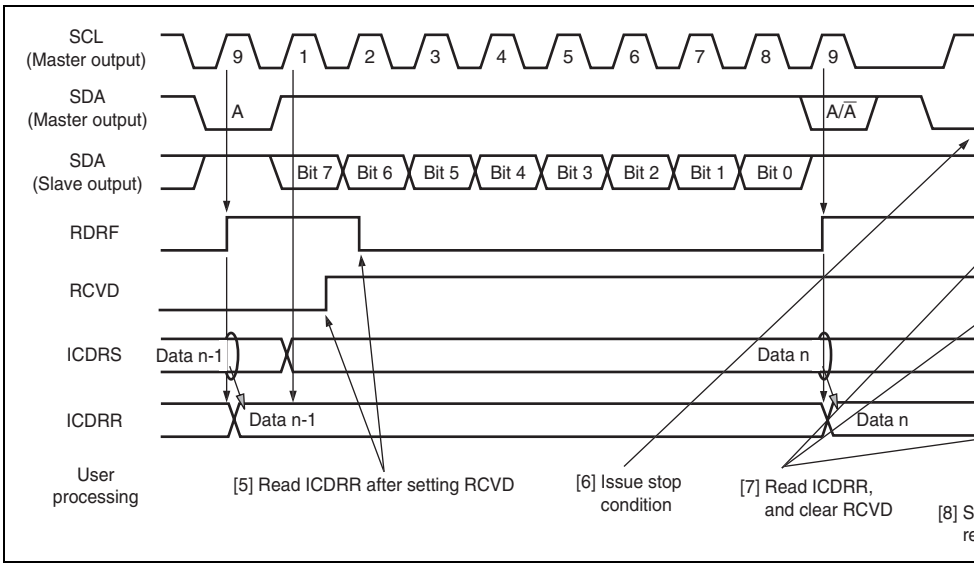


Figure 16.8 Master Receive Mode Operation Timing (2)

the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise
clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS bit in ICCR1 and the
in ICSR are set to 1, and the mode changes to slave transmit mode automatically. Th
continuous transmission is performed by writing transmit data to ICDRT every time
set.

3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is
with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is opened.
5. Clear TDRE.

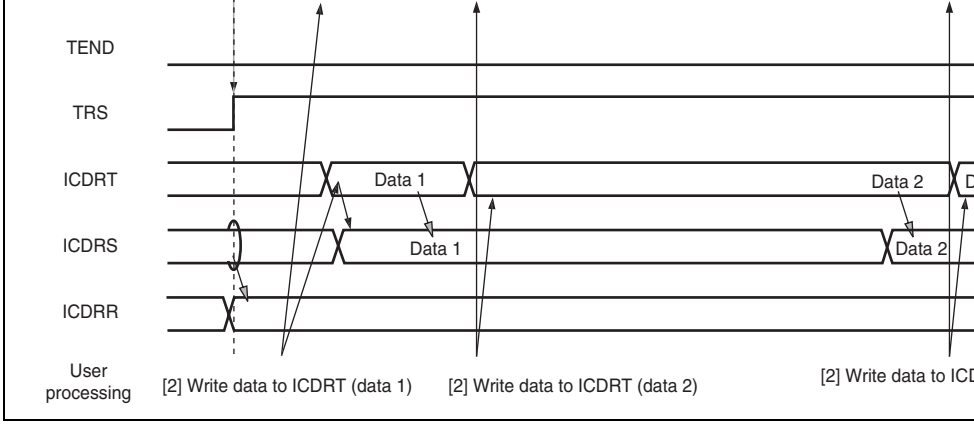


Figure 16.9 Slave Transmit Mode Operation Timing (1)

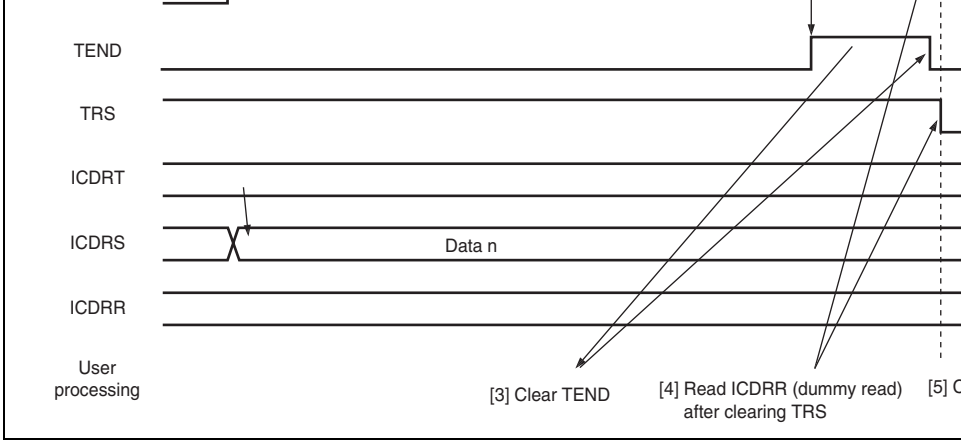


Figure 16.10 Slave Transmit Mode Operation Timing (2)

clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (S
read data show the slave address and R/$\overline{W}$, it is not used.)

3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is
fixed low until ICDRR is read. The change of the acknowledge before reading ICDR
returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

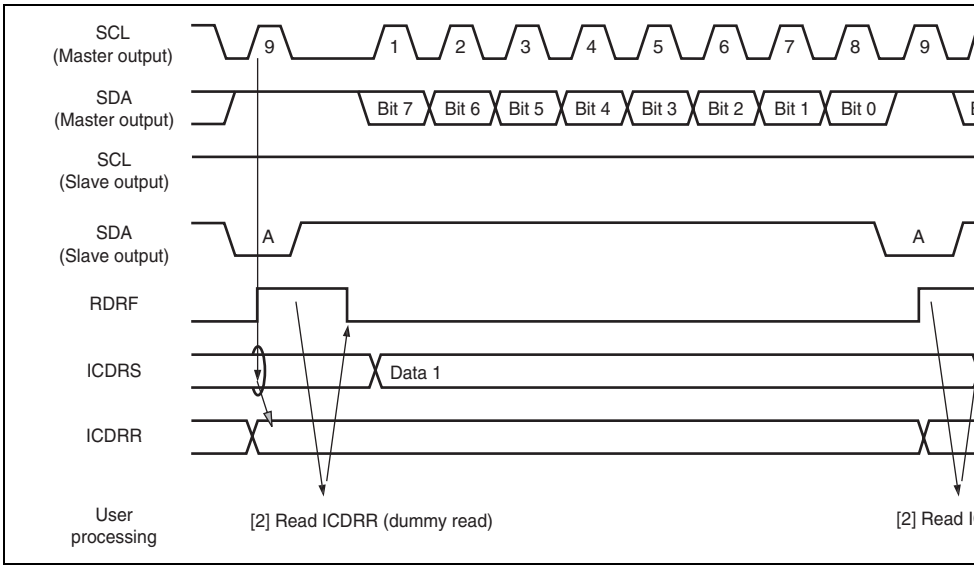


Figure 16.11 Slave Receive Mode Operation Timing (1)

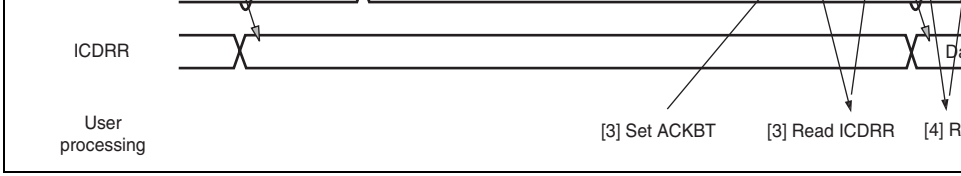


Figure 16.12 Slave Receive Mode Operation Timing (2)

of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in e
MSB first or LSB first. The output level of SDA can be changed during the transfer wait,
SDAO bit in ICCR2.

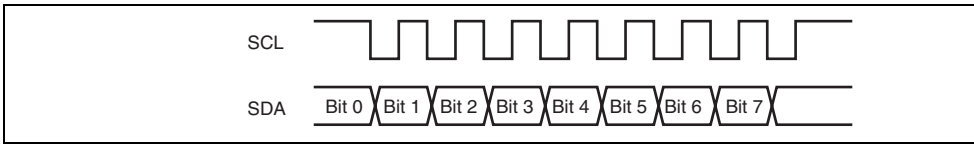


Figure 16.13 Clocked Synchronous Serial Transfer Format

(2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of th
clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0
transmit mode operation timing, refer to figure 16.14. The transmission procedure and op
in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS[3:0] bits in ICCR1. (Initial setti
2. Set the TRS bit in ICCR1 to select transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is
   transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous
   transmission is performed by writing data to ICDRT every time TDRE is set. When c
   from transmit mode to receive mode, clear TRS while TDRE is 1.

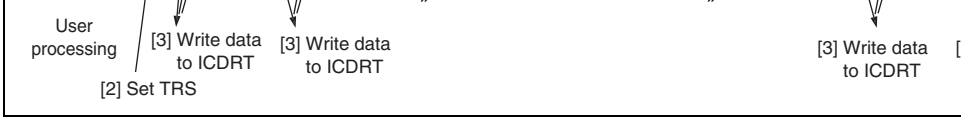


Figure 16.14 Transmit Mode Operation Timing

(3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is out MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, ref figure 16.15. The reception procedure and operations in receive mode are described belo

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected an AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDR
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then fixed high after receiving the next byte data.

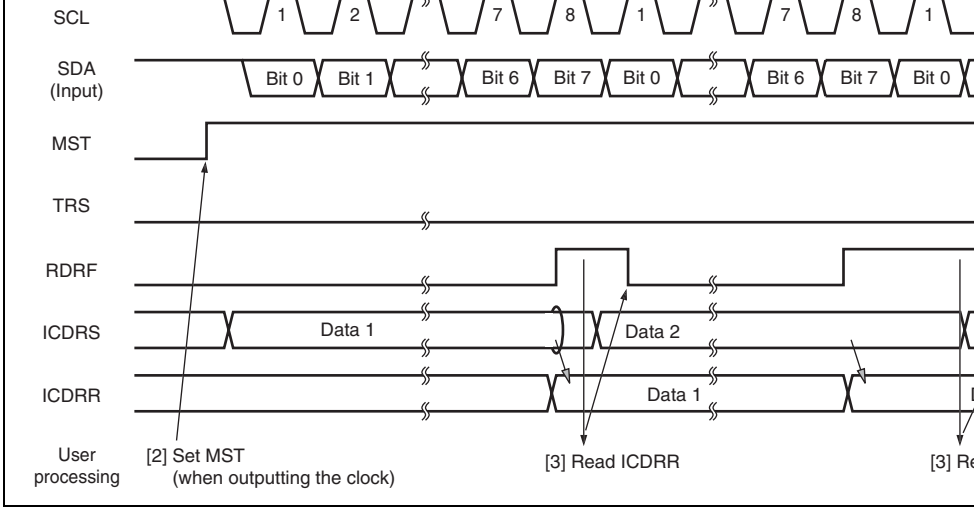


Figure 16.15 Receive Mode Operation Timing

Figure 16.16 Operation Timing For Receiving One Byte (MST = 1)

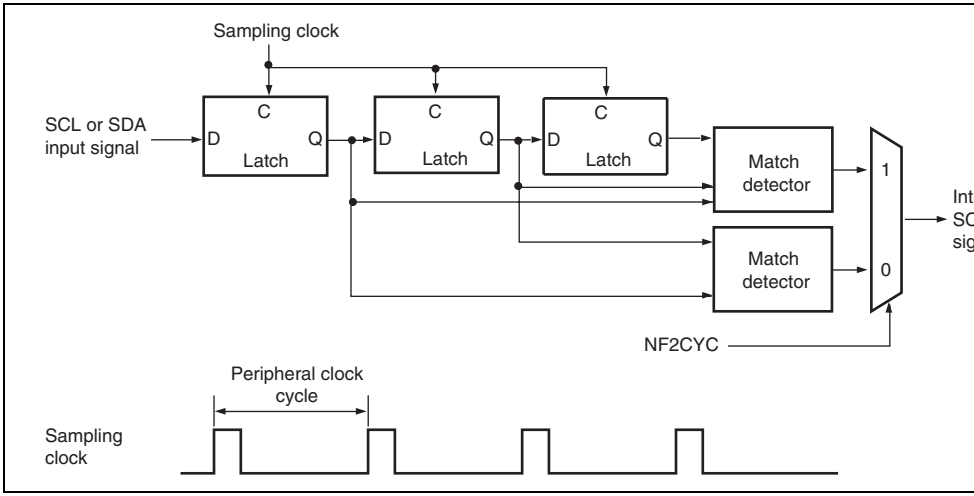
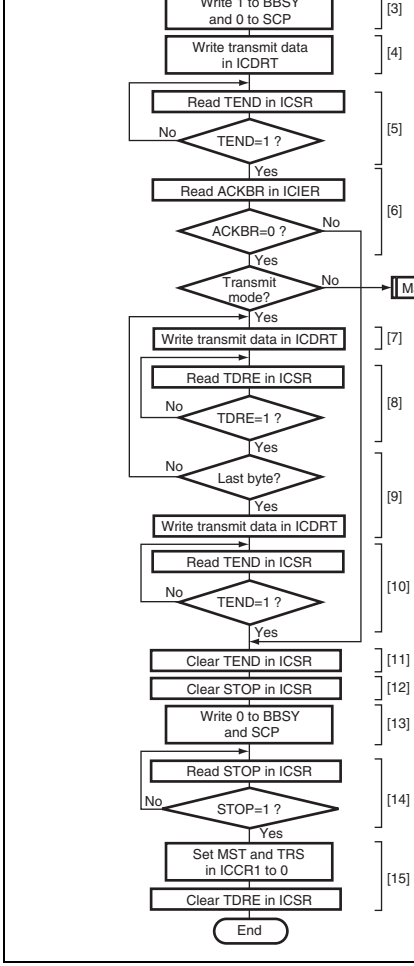


Figure 16.17 Block Diagram of Noise Filter



[6] Test the acknowledge transferred from the specified slave device.

[7] Set the second and subsequent bytes (except for the final byte) of transmit data

[8] Wait for ICDRT empty.

[9] Set the last byte of transmit data.

[10] Wait for last byte to be transmitted.

[11] Clear the TEND flag.

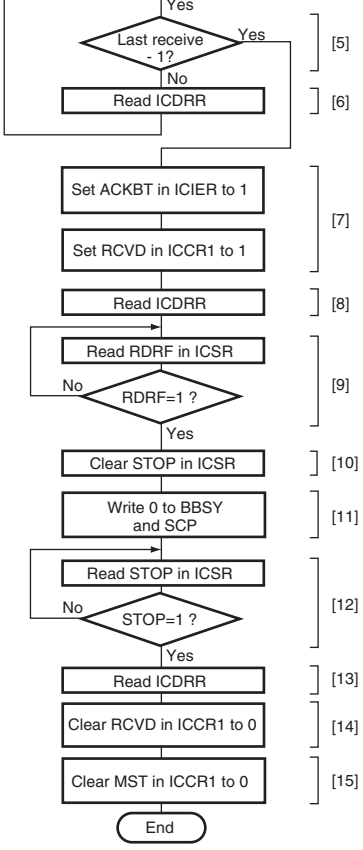
[12] Clear the STOP flag.

[13] Issue the stop condition.

[14] Wait for the creation of stop condition.

[15] Set slave receive mode. Clear TDRE.

Figure 16.18 Sample Flowchart for Master Transmit Mode



[9] Wait for the last byte to be receive.

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

[13] Read the last byte of receive data.

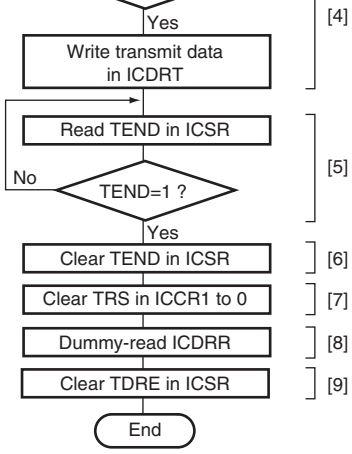
[14] Clear RCVD.

[15] Set slave receive mode.

Notes: * Make sure that no interrupt will be generated during steps [1] to

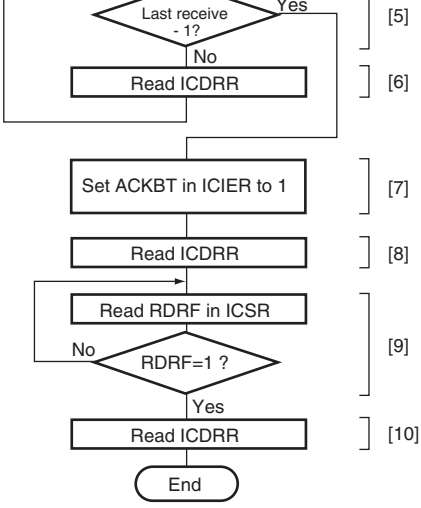
When the size of receive data is only one byte in reception, steps [2] to [6] are skipped after step [1], before jumping to step The step [8] is dummy-read in ICDRR.

Figure 16.19 Sample Flowchart for Master Receive Mode



[9] Clear the TDRE flag.

Figure 16.20 Sample Flowchart for Slave Transmit Mode



[8] Read the (last byte - 1) of receive data.

[9] Wait the last byte to be received.

[10] Read for the last byte of receive data.

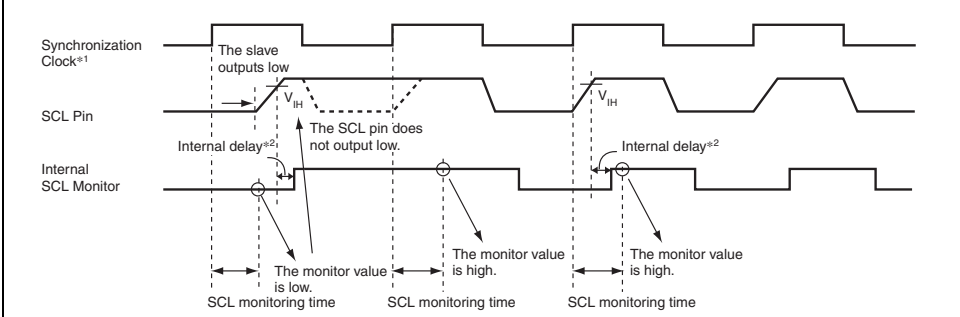
Note: When the size of receive data is only one byt
reception, steps [2] to [6] are skipped after
step [1], before jumping to step [7]. The step [
is dummy-read in ICDRR.

Figure 16.21 Sample Flowchart for Slave Receive Mode

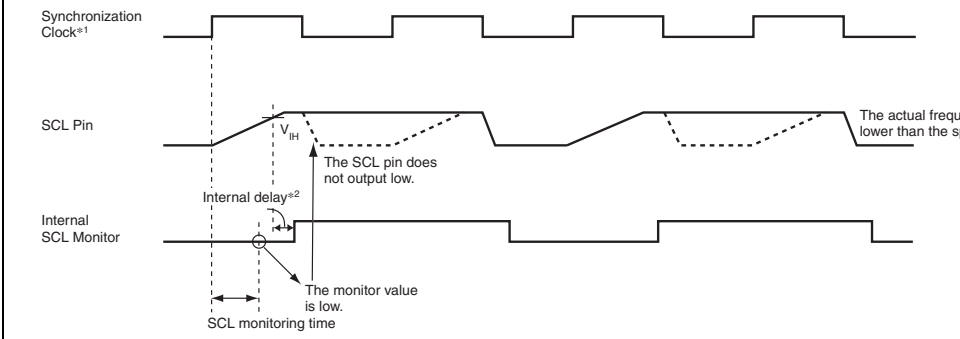
Transmit end	TEI	(TEND = 1) • (TEIE = 1)	√	√
Receive data full	RXI	(RDRF = 1) • (RIE = 1)	√	√
STOP recognition	STPI	(STOP = 1) • (STIE = 1)	√	—
NACK detection	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)	√	—
Arbitration lost/ overrun error			√	√

When the interrupt condition described in table 16.4 is 1, the CPU executes an interrupt handling. Note that a TXI or RXI interrupt can activate the DMAC if the setting for DM activation has been made. In such a case, an interrupt request is not sent to the CPU. Inte sources should be cleared in the exception handling. The TDRE and TEND bits are auto cleared to 0 by writing the transmit data to ICDRT. The RDRF bit is automatically clear reading ICDRR. The TDRE bit is set to 1 again at the same time when the transmit data to ICDRT. Therefore, when the TDRE bit is cleared to 0, then an excessive data of one b be transmitted.

Figure 16.22 shows the timing of the bit synchronous circuit and table 16.5 shows the tim
the SCL output changes from low to Hi-Z then SCL is monitored.



(3) When the SCL pin rises slowly



Notes: 1. A clock whose transfer rate is specified by the CKS[3:0] bits in the I^2C bus control register (ICCR1).
2. 3 to 4 t_{pcyc} when the NF2CYC bit in the NF2CYC register (NF2CYC) is cleared to 0; 4 to 5 t_{pcyc} when the NF2CYC bit in NF2CYC is

Figure 16.22 Bit Synchronous Circuit Timing

16.7.2 Note on Master Receive Mode

If ICDRR is read near the falling edge of 8th clock, the receive data will not be received cases. In addition, if RCVD is set to 1 near the falling edge of 8th clock, a stop condition be issued in some cases. To prevent these errors, one of the following two methods shou selected.

1. In master receive mode, ICDRR should be read before the falling edge of 8th clock.
2. In master receive mode, RCVD should be set to 1 and the processing should be perfo byte units.

16.7.3 Note on Master Receive Mode with ACKBT Setting

In master receive mode operation, ACKBT should be set before the 8th falling edge of S final data transfer during continuous data transfer. Otherwise, the slave device may over

16.7.4 Note on MST and TRS Bit Status When an Arbitration was Lost

If the master transmission is set according to the MST and TRS bit settings while multip are used, the conflicting status in which the AL bit in ICSR is set to 1 in master transmit (MST and TRS are set to 1) depending on the arbitration lost generation timing during T handling instruction execution.

This problem can be avoided by the following methods.

- When multiple masters are used, the MST and TRS bits should be set by a MOV ins
- When an arbitration lost occurs, check if both MST and TRS bits are cleared to 0. If both of MST and TRS bits are not cleared to 0, both the bits should be cleared to 0.

High-speed conversion:

When Aϕ = 40 MHz: Minimum 1.25 μs per channel

AD clock = 40 MHz, 50 conversion states

- Two operating modes
  - — Single-cycle scan mode: Continuous A/D conversion on one to eight channels
  - — Continuous scan mode: Repetitive A/D conversion on one to eight channels
- A/D data registers

  Eight A/D data registers (ADDR) are provided. A/D conversion results are stored in registers (ADDR) that correspond to the input channels.
- Sample-and-hold function

  A sample-and-hold circuit is built into the A/D converter of this LSI, simplifying the configuration of the external analog input circuitry. Multiple channels can be sample simultaneously because sample-and-hold circuits can be dedicated for channels 0 to 10.
  - — Group A (GrA): Analog input pins selected from channels 0, 1, and 2 can be simultaneously sampled.
- Three methods for starting conversion

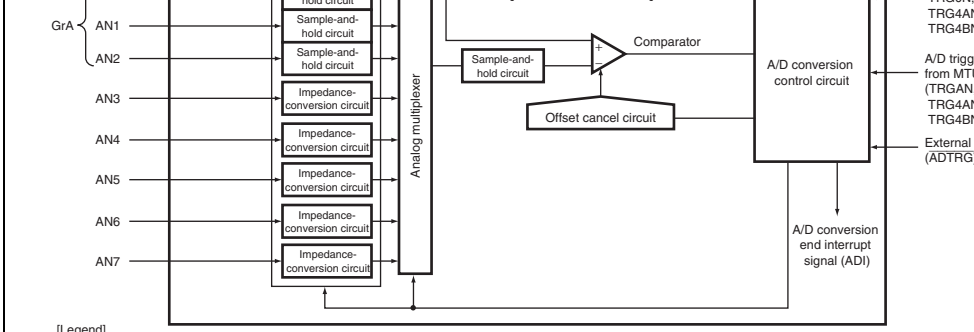
  Software: Setting of the ADST bit in ADCR

  Timer: TRGAN, TRG0N, TRG4AN, and TRG4BN from the MTU2
  TRGAN, TRG4AN, and TRG4BN from the MTU2S

  External trigger: $\overline{\text{ADTRG}}$ (LSI pin)
- Selectable analog input channel

  A/D conversion of a selected channel is accomplished by setting the A/D analog inp select registers (ADANSR).
- A/D conversion end interrupt and DMAC transfer function is supported

  On completion of A/D conversion, A/D conversion end interrupts (ADI) can be gene the DMAC can be activated by ADI.



[Legend]
ADDR: A/D data register
ADCR: A/D control register
ADANSR: A/D analog input channel select register
ADSR: A/D status register
ADSTRGR: A/D start trigger select register
GrA: Group A

Figure 17.1 Block Diagram of A/D Converter

AVREF	Input	Analog block reference power supply pin
AVREFVss	Input	Analog block reference ground pin
$\overline{\text{ADTRG}}$	Input	A/D external trigger input pin
AN0	Input	Analog input pin 0 (Group A)
AN1	Input	Analog input pin 1 (Group A)
AN2	Input	Analog input pin 2 (Group A)
AN3	Input	Analog input pin 3
AN4	Input	Analog input pin 4
AN5	Input	Analog input pin 5
AN6	Input	Analog input pin 6
AN7	Input	Analog input pin 7

register					
A/D analog input channel select register	ADANSR	R/W	H'00	H'FFFFE820	8
A/D data register 0	ADDR0	R	H'0000	H'FFFFE840	16
A/D data register 1	ADDR1	R	H'0000	H'FFFFE842	16
A/D data register 2	ADDR2	R	H'0000	H'FFFFE844	16
A/D data register 3	ADDR3	R	H'0000	H'FFFFE846	16
A/D data register 4	ADDR4	R	H'0000	H'FFFFE848	16
A/D data register 5	ADDR5	R	H'0000	H'FFFFE84A	16
A/D data register 6	ADDR6	R	H'0000	H'FFFFE84C	16
A/D data register 7	ADDR7	R	H'0000	H'FFFFE84E	16

7	ADST	0	R/W	A/D Start When this bit is cleared to 0, A/D conversion is s and the A/D converter enters the idle state. Whe is set to 1, A/D conversion is started. In single-c mode, this bit is automatically cleared to 0 when conversion ends on the selected single channel. continuous scan mode, A/D conversion is contin performed for the selected channels in sequenc bit is cleared by software, a reset, or in software mode, or module standby mode.
6	ADCS	0	R/W	A/D Continuous Scan Selects either a single-cycle or a continuous sca mode. This bit is valid only when scan mode is s 0: Single-cycle scan 1: Continuous scan When changing the operating mode, first clear th bit to 0.
5	ACE	0	R/W	Automatic Clear Enable Enables or disables the automatic clearing of AD ADDR is read by the CPU or DMAC. When this to 1, ADDR is automatically cleared to H'0000 a CPU or DMAC reads ADDR. This function allow detection of any renewal failures of ADDR. 0: Automatic clearing of ADDR after being read disabled. 1: Automatic clearing of ADDR after being read

				In addition, ADIE activates the DMAC when an A generated. At this time, no interrupt to the CPU is generated. 0: Generation of A/D conversion end interrupt is d 1: Generation of A/D conversion end interrupt is e
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
1	TRGE	0	R/W	Trigger Enable Enables or disables A/D conversion start by the e trigger input ($\overline{\text{ADTRG}}$) or A/D conversion start trig from the MTU2 and MTU2S (TRGAN, TRG0N, TI and TRG4BN from the MTU2 and TRGAN, TRG4 TRG4BN from the MTU2S). For selection of the e trigger and A/D conversion start trigger from the M MTU2S, see the description of the EXTRG bit. 0: A/D conversion start by the external trigger or a conversion start trigger from the MTU or MTU2 disabled 1: A/D conversion start by the external trigger or a conversion start trigger from the MTU2 or MTU enabled

ADST bit by software is subsequently performed conversion start by the external trigger input is e only when the ADST bit is cleared to 0.

When the external trigger is used as an A/D con start trigger, the low-level pulse input to the ADT must be at least 1.5 Aϕ clock cycles in width.

0: A/D converter is started by the A/D conversio trigger from the MTU2 or MTU2S

1: A/D converter is started by the external pin ($\overline{A}$

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
0	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the completion of A/D conversion. [Setting condition] • When A/D conversion on all specified chann completed in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DMAC is activated by an ADI inter ADDR is read

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
6	STR6	0	R/W	Start Trigger 6 Enables or disables the A/D conversion start re input from the MTU2S. 0: Disables the A/D conversion start by TRGAN (MTU2S). 1: Enables the A/D conversion start by TRGAN (MTU2S).
5	STR5	0	R/W	Start Trigger 5 Enables or disables the A/D conversion start re input from the MTU2S. 0: Disables the A/D conversion start by TRG4A (MTU2S). 1: Enables the A/D conversion start by TRG4A (MTU2S).
4	STR4	0	R/W	Start Trigger 4 Enables or disables the A/D conversion start re input from the MTU2S. 0: Disables the A/D conversion start by TRG4B (MTU2S). 1: Enables the A/D conversion start by TRG4B (MTU2S).

				Enables or disables the A/D conversion start req input from the MTU2. 0: Disables the A/D conversion start by TRGAN (MTU2). 1: Enables the A/D conversion start by TRGAN (MTU2).
1	STR1	0	R/W	Start Trigger 1 Enables or disables the A/D conversion start req input from the MTU2. 0: Disables the A/D conversion start by TRG4AN (MTU2). 1: Enables the A/D conversion start by TRG4AN (MTU2).
0	STR0	0	R/W	Start Trigger 0 Enables or disables the A/D conversion start req input from the MTU2. 0: Disables the A/D conversion start by TRG4BN (MTU2). 1: Enables the A/D conversion start by TRG4BN (MTU2).

7	ANS7	0	R/W	Setting bits in the A/D analog input channel sel register to 1 selects a channel that correspond specified bit. For the correspondence between input pins and bits, see table 17.3. When changing the analog input channel, the A ADCR must be cleared to 0 to prevent incorrec operations.
6	ANS6	0	R/W	
5	ANS5	0	R/W	
4	ANS4	0	R/W	
3	ANS3	0	R/W	
2	ANS2	0	R/W	
1	ANS1	0	R/W	
0	ANS0	0	R/W	

Table 17.3 Channel Select List

Bit Name	Analog Input Channels
ANS0	AN0
ANS1	AN1
ANS2	AN2
ANS3	AN3
ANS4	AN4
ANS5	AN5
ANS6	AN6
ANS7	AN7

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	-	-	-	ADD[11:0]									
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved
11 to 0	ADD[11:0]	All 0	R	12-bit data

Table 17.4 Correspondence between Analog Channels and Registers (ADDR0 to A

Analog Input Channels	A/D Data Registers
AN0	ADDR0
AN1	ADDR1
AN2	ADDR2
AN3	ADDR3
AN4	ADDR4
AN5	ADDR5
AN6	ADDR6
AN7	ADDR7

AN0 to AN7.

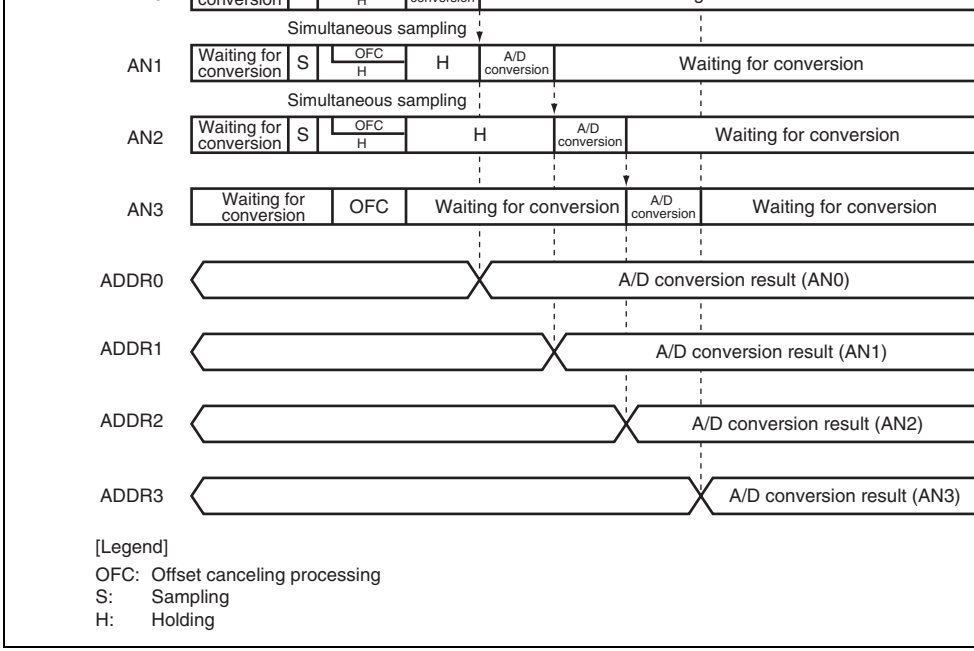
In single-cycle scan mode, when one cycle of A/D conversion on all specified channels completed, the ADF bit in ADSR is set to 1 and the ADST bit is automatically cleared t continuous scan mode, when conversion on all specified channels is completed, the AD ADSR is set to 1. To stop A/D conversion, write 0 to the ADST bit. When the ADF bit i if the ADIE bit in ADCR is set to 1, an A/D conversion end interrupt (ADI) is generated clearing the ADF bit to 0, read the ADF bit while set to 1 and then write 0. However, wh DMAC is activated by an ADI interrupt, the ADF bit is automatically cleared to 0.

17.4.1 Single-Cycle Scan Mode

The following example shows the operation when analog input channels 0 to 3 (AN0 to selected and the A/D conversion is performed in single-cycle scan mode using four char

1. Set the ADCS bit in the A/D control register (ADCR) to 0.
2. Set all bits ANS0 to ANS3 in the A/D analog input channel select register (ADANS
3. Set the ADST bit in the A/D control register (ADCR) to 1 to start A/D conversion.
4. After channels 0 to 2 (GrA) are sampled simultaneously, offset canceling processing performed. Then, A/D conversion is performed on channel 0. Upon completion of th conversion, the A/D conversion result is transferred to ADDR0. Following this, char converted. Upon completion of the conversion, the A/D conversion result is transfer ADDR1. In the same way, channel 2 is converted and the A/D conversion result is tr to ADDR2.

   A/D conversion of channel 3 is then started. Upon completion of the A/D conversion conversion result is transferred to ADDR3.



[Legend]
OFC: Offset canceling processing
S: Sampling
H: Holding

Figure 17.2 Example of A/D_0 Converter Operation (Single-Cycle Scan Mod

4. Channels 0 and 2 (GrA) are sampled simultaneously. As the ANS1 bit in ADANSR channel 1 is not sampled. After this, offset canceling processing (OFC) is performed A/D conversion on channel 0 is started. Upon completion of the A/D conversion, the conversion result is transferred to ADDR0. In the same way, channel 2 is converted A/D conversion result is transferred to ADDR2. The A/D conversion is not performe channel 1.
5. The A/D conversion of channel 3 starts. Upon completion of the A/D conversion, th conversion result is transferred to ADDR3.
6. When the A/D conversion ends on all the specified channels (AN0 to AN3), the ADI to 1. At this time, if the ADIE bit is set to 1, an ADI interrupt is generated after the A conversion.
7. Steps 4 to 6 are repeated as long as the ADST bit remains set to 1. When the ADST cleared to 0, the A/D conversion stops. After this, if the ADST bit is set to 1, the A/D conversion starts again and repeats steps 4 to 6.

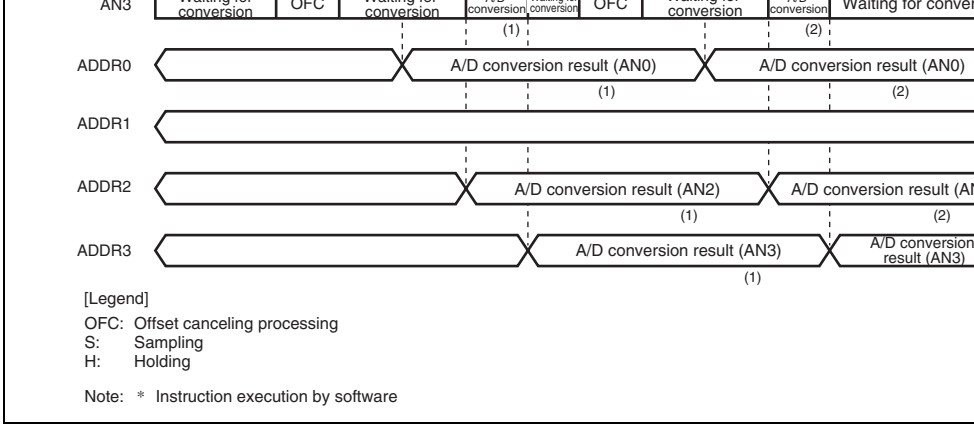


Figure 17.3 Example of A/D Converter Operation (Continuous Scan Mode

activation request from the MTU2, the MTU2S, or an external trigger signal occurs, the input is sampled by the dedicated sample-and-hold circuit for each channel after the A/D conversion start delay time (t_D) has passed and the offset canceling processing (OFC) is performed. After this, the sampling of the analog input using the sample-and-hold circuit to all the channels is performed and then the A/D conversion is started. Figure 17.4 shows conversion timing in this case. This A/D conversion time (t_{CONV}) includes the t_D, the offset canceling processing time (t_{OFC}), the analog input sampling time with a dedicated sample circuit for each channel (t_{SPLSH}), and the analog input sampling time with the sample-and-hold circuit common to all the channels (t_{SPL}). The t_{SPLSH} does not depend on the number of channels simultaneously sampled.

In continuous scan mode, the A/D conversion time (t_{CONV}) given in table 17.6 applies to the conversion time of the first cycle. The conversion time of the second and subsequent cycles is expressed as ($t_{CONV} - t_D + 6$).

Table 17.5 Correspondence between Analog Input Channels and Groups being A Simultaneous Sampling

Analog Input Channels	Group
AN0	GrA
AN1	
AN2	
AN3	—
AN4	—
AN5	—
AN6	—
AN7	—

A/D conversion complete processing	t_{end}	—	4	—
A/D conversion time	t_{CONV}	50n + 95*3	—	50n

Notes: 1. A/D converter activation by the MTU2 or MTU2S trigger signal.
2. A/D converter activation by an external trigger signal.
3. n: number of A/D conversion channels (n = 1 to 8)

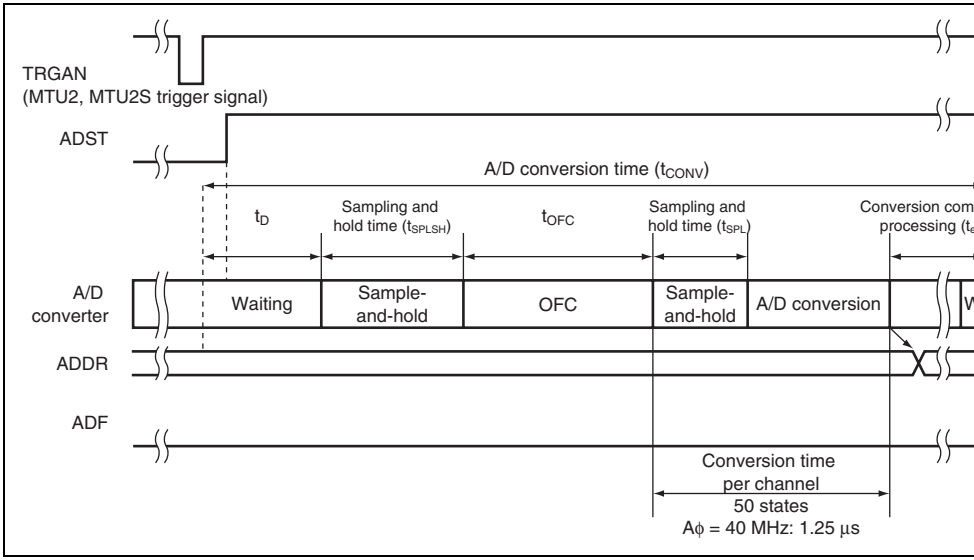


Figure 17.4 A/D Conversion Timing (Single-Cycle Scan Mode)

the ADCR, ADSTRGR, and ADANSR registers have been set.

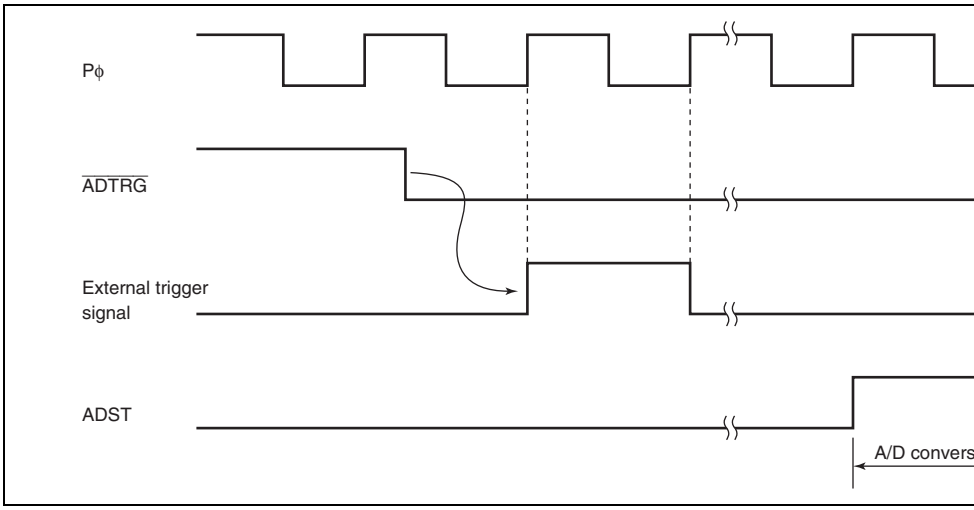


Figure 17.5 External Trigger Input Timing

17.4.6 Example of ADDR Auto-Clear Function

When the A/D data register (ADDR) is read by the CPU or DMAC, ADDR can be autom cleared to H'0000 by setting the ACE bit in ADCR to 1. This function allows the detectio updated ADDR states.

Figure 17.6 shows an example of when the auto-clear function of ADDR is disabled (nor and enabled.

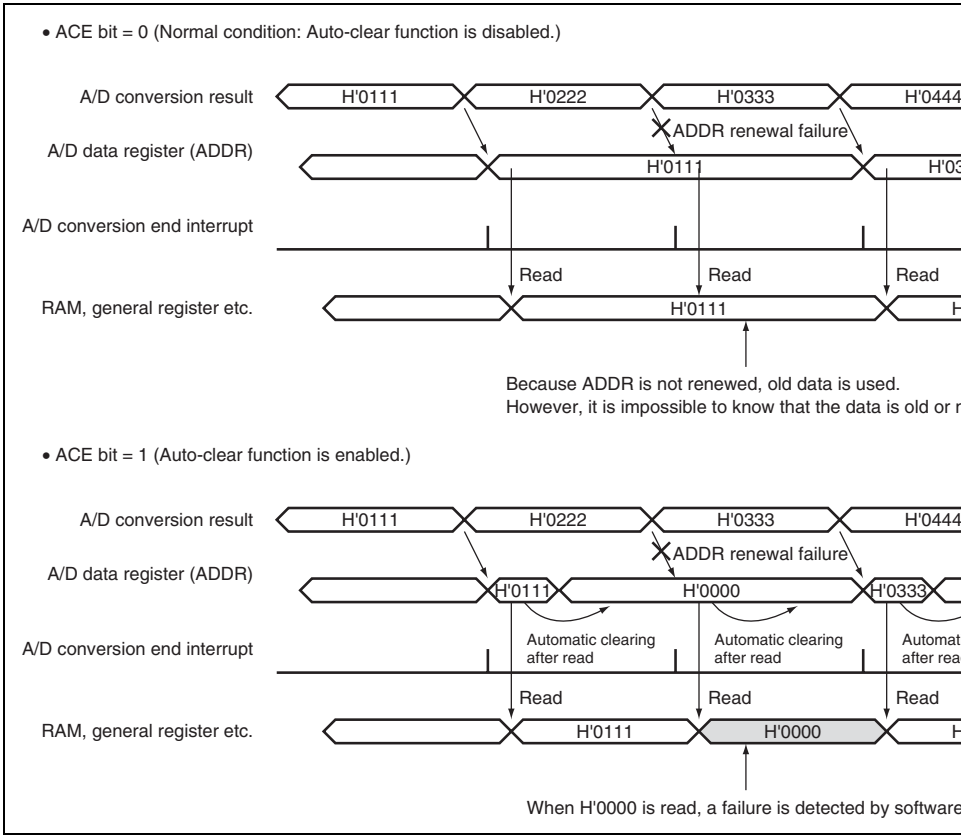


Figure 17.6 Example of When ADDR Auto-clear Function is Disabled (Normal Condition)/Enabled

figure 17.7).

- Full-scale error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conver characteristic when the digital output value changes from B'111111111110 to the ma voltage value (full-scale voltage) B'111111111111. Does not include a quantization figure 17.7).

- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.7).

- Nonlinearity error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conver characteristic between zero voltage and full-scale voltage. Does not include offset er scale error, or quantization error (see figure 17.7).

- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset e scale error, quantization error, and nonlinearity error.

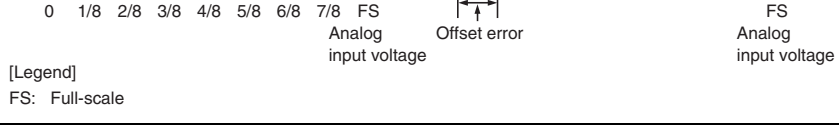


Figure 17.7 Definitions of A/D Conversion Accuracy

When using the A/D converter or D/A converter, set AVREF to a level between 4.5 V a
When the A/D converter and D/A converter are not used, make settings such that AVRE
AVcc, and do not leave the AVREF pin open.

The setting of the AVREFVss pin should always be such that AVREFVss = AVss, and d
leave AVREFVss open. If these conditions are not met, the reliability of the SH7211 ma
adversely affected.

17.7.3 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as p
and the layout in which the digital circuit signal lines and analog circuit signal lines cros
close proximity to each other should be avoided as much as possible. Failure to do so ma
the incorrect operation of the analog circuitry due to inductance, adversely affecting the
conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN7), ana
reference power supply (AVREF), the analog power supply (AVcc), the analog ground (
and the analog reference ground (AVREFVss). Also, AVss should be connected at one p
stable digital ground (Vss) on the board.

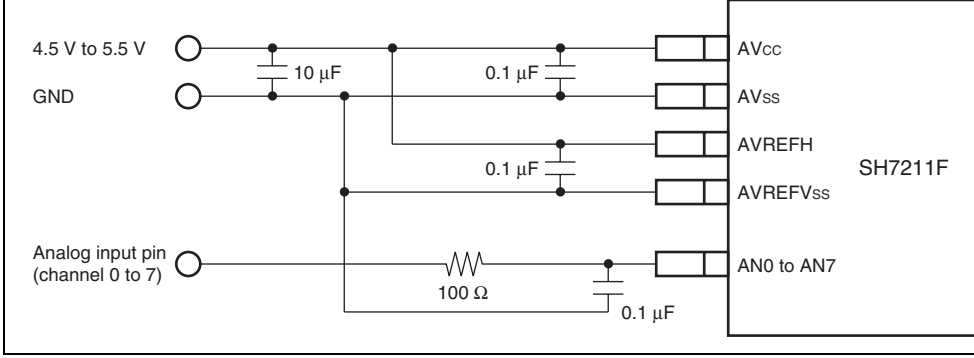


Figure 17.8 Example of Analog Input Pin Protection Circuit

17.7.5 Notes on Register Setting

- Set the ADST bit in the A/D control register (ADCR) after the A/D start trigger select (ADSTRGR) and the A/D analog input channel select register (ADANSR) have been not modify the settings of the ADCS, ACE, ADIE, TRGE, and EXTRG bits while the bit in the ADCR register is set to 1.
- Do not start the A/D conversion when the ANS bits (ANS[7:0]) in the A/D analog inp channel select register (ADANSR) are all 0.

- Module standby mode can be set

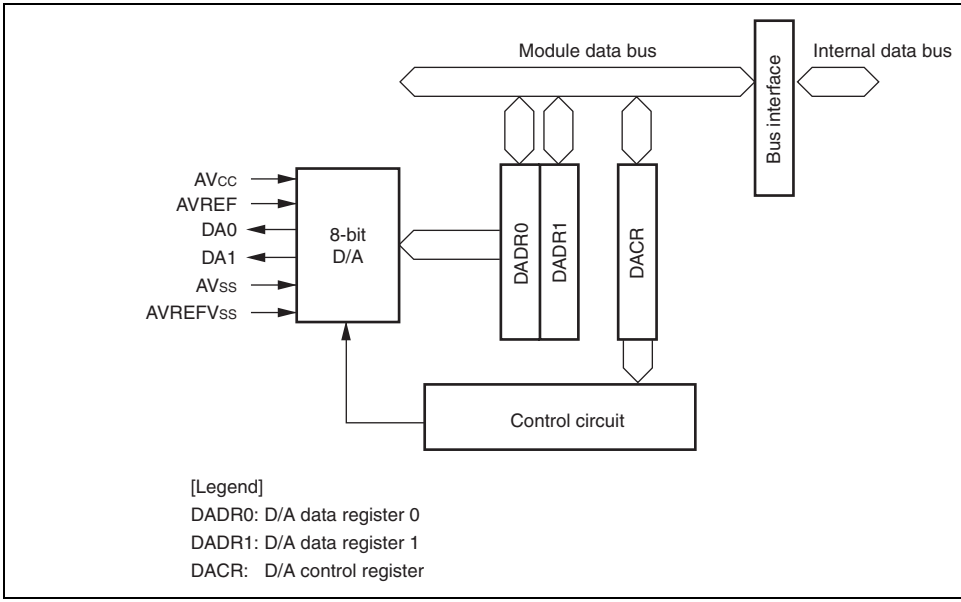


Figure 18.1 Block Diagram of D/A Converter

Reference ground pin	AVREFVss	Input	D/A conversion reference grou
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

18.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

DADR is an 8-bit readable/writable register that stores data to which D/A conversion is performed. Whenever analog output is enabled, the values in DADR are converted and the analog output pins.

DADR is initialized to H'00 by a power-on reset or in module standby mode.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output for c 0: Analog output of channel 1 (DA1) is disabled 1: D/A conversion of channel 1 is enabled. Analo of channel 1 (DA1) is enabled.
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output for c 0: Analog output of channel 0 (DA0) is disabled 1: D/A conversion of channel 0 is enabled. Analo 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. Output of conversion results is a controlled by the DAOE0 and DAOE1 bits. For d see table 18.3. 0: D/A conversion for channels 0 and 1 is contro independently 1: D/A conversion for channels 0 and 1 is contro together
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be r

1	0	0	D/A conversion is disabled.
		1	D/A conversion of channels 0 and 1 is enabled.
	1	0	
		1	

the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The conversio continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to output value is expressed by the following formula:

$$\frac{\text{Contents of DADR}}{256} \times \text{AVref}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion r output after the conversion time t_{DCONV} has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

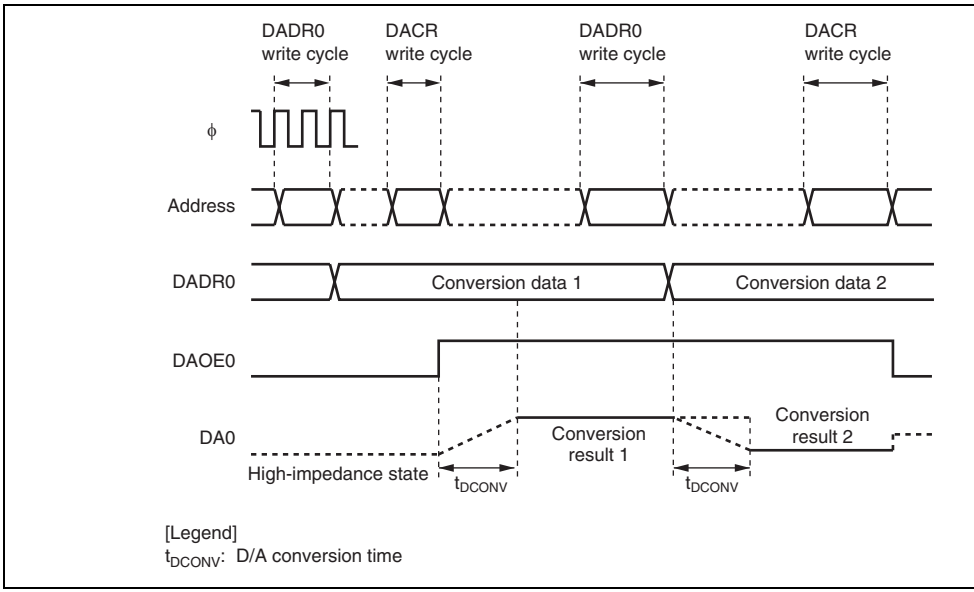


Figure 18.2 Example of D/A Converter Operation

When this LSI enters software standby mode with D/A conversion enabled, the D/A out
retained, and the analog power supply current is equal to as during D/A conversion. If th
power supply current needs to be reduced in software standby mode, clear the DAOE0,
and DAE bits to 0 to disable the D/A outputs.

18.5.3 Setting Analog Input Voltage

The reliability of this LSI may be adversely affected if the following voltage ranges are

1. AVcc and AVss input voltages

   Input voltages AVcc and AVss should be Vcc ≤ AVcc ≤ 5.0 V ± 0.5 V and AVss = V
   not leave the AVcc and AVss pins open when the A/D converter or D/A converter is
   and in software standby mode. When not in use, connect AVcc to the power supply
   and AVss to the ground (VssQ).
2. Setting range of AVREF input voltage

   Set the voltage range of the AVREF pin as AVREF = AVcc ± 0.3 V when the A/D c
   or D/A converter is used, or as AVREF = AVcc when no A/D converter or D/A conv
   used.

A	PA25 I/O (port)	A25 output (BSC)	—	IRQ7 input (INTC)	TIOC0D I/O (MTU2)	TXD1 output (SCIF1)	–
	PA24 I/O (port)	A24 output (BSC)	—	IRQ6 input (INTC)	TIOC0C I/O (MTU2)	RXD1 input (SCIF1)	–
	PA23 I/O (port)	A23 output (BSC)	—	IRQ5 input (INTC)	TIOC0B I/O (MTU2)	SCK1 I/O (SCIF1)	–
	PA22 I/O (port)	A22 output (BSC)	—	IRQ4 input (INTC)	TIOC0A I/O (MTU2)	—	–
	PA21 I/O (port)	A21 output (BSC)	—	IRQ3 input (INTC)	—	—	–
	PA20 I/O (port)	A20 output (BSC)	—	IRQ2 input (INTC)	—	—	–
	PA19 I/O (port)	A19 output (BSC)	—	IRQ1 input (INTC)	—	—	–
	PA18 I/O (port)	A18 output (BSC)	—	IRQ0 input (INTC)	—	—	–
	PA17 I/O (port)	A17 output (BSC)	—	—	—	TXD3 output (SCIF3)	–
	PA16 I/O (port)	A16 output (BSC)	—	—	—	RXD3 input (SCIF3)	–
	PA15 I/O (port)	A15 output (BSC)	—	—	—	SCK3 I/O (SCIF3)	–
	PA14 I/O (port)	A14 output (BSC)	—	—	—	—	–
	PA13 I/O (port)	A13 output (BSC)	—	—	—	—	–

PA8 I/O (port)	A8 output (BSC)	—	—	—	—	—
PA7 I/O (port)	A7 output (BSC)	—	—	—	—	—
PA6 I/O (port)	A6 output (BSC)	—	—	—	—	—
PA5 I/O (port)	A5 output (BSC)	—	—	—	—	—
PA4 I/O (port)	A4 output (BSC)	—	—	—	—	—
PA3 I/O (port)	A3 output (BSC)	—	—	—	—	—
PA2 I/O (port)	A2 output (BSC)	—	—	—	—	—
PA1 I/O (port)	A1 output (BSC)	—	—	—	—	—
PA0 I/O (port)	A0 output (BSC)	—	—	—	—	—

(port)		(DMAC)		(MTU2)	(SCIF3)	
PB27 I/O (port)	—	TEND0 output (DMAC)	—	TIOC2A I/O (MTU2)	TXD3 output (SCIF3)	A c (
PB26 I/O (port)	—	DREQ1 input (DMAC)	—	TIOC2B I/O (MTU2)	SCK3 I/O (SCIF3)	A c (
PB25 I/O (port)	—	DACK1 output (DMAC)	IRQ3 input (INTC)	TCLKA input (MTU2)	TXD3 output (SCIF3)	A c (
PB24 I/O (port)	—	TEND1 output (DMAC)	IRQ2 input (INTC)	TCLKB input (MTU2)	RXD3 input (SCIF3)	A c (
PB23 I/O (port)	—	DREQ2 input (DMAC)	—	TCLKC input (MTU2)	TXD2 output (SCIF2)	A (
PB22 I/O (port)	—	DACK2 output (DMAC)	—	TCLKD input (MTU2)	RXD2 input (SCIF2)	A c
PB21 I/O (port)	$\overline{CS2}$ output (BSC)	—	IRQ0 input (INTC)	TIOC3BS I/O (MTU2S)	RXD0 input (SCIF0)	-
PB20 I/O (port)	$\overline{BS}$ output (BSC)	—	—	TIOC3DS I/O (MTU2S)	—	-
PB19 I/O (port)	$\overline{CS6}$ output (BSC)	—	IRQ6 input (INTC)	TIOC3D I/O (MTU2)	—	-
PB18 I/O (port)	$\overline{CS4}$ output (BSC)	—	IRQ4 input (INTC)	TIOC3B I/O (MTU2)	—	-
PB17 I/O (port)	$\overline{CS3}$ output (BSC)	—	IRQ1 input (INTC)	TIOC3A I/O (MTU2)	—	-

PB12 I/O (port)	$\overline{\text{BREQ}}$ input (BSC)	—	—	TIOC4AS I/O (MTU2S)	TXD2 output (SCIF2)	—
PB11 I/O (port)	$\overline{\text{AH}}$ output (BSC)	DACK3 output (DMAC)	—	TIOC4DS I/O (MTU2S)	TXD2 output (SCIF2)	—
PB10 I/O (port)	$\overline{\text{WAIT}}$ input (BSC)	DREQ3 input (DMAC)	—	TIOC4CS I/O (MTU2S)	RXD2 input (SCIF2)	—
PB9 I/O (port)	$\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$ output (BSC)	—	—	TIOC3CS I/O (MTU2S)	TXD3 (SCIF3)	—
PB8 I/O (port)	$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$ output (BSC)	—	—	TIOC3AS I/O (MTU2S)	RXD3 (SCIF3)	—
PB7 I/O (port)	$\overline{\text{CS7}}$ output (BSC)	—	IRQ7 input (INTC)	TIOC4D I/O (MTU2)	—	—
PB6 I/O (port)	$\overline{\text{CASL}}$ output (BSC)	—	IRQ3 input (INTC)	TIOC4C I/O (MTU2)	—	—
PB5 I/O (port)	$\overline{\text{RASL}}$ output (BSC)	—	IRQ2 input (INTC)	TIOC4B I/O (MTU2)	—	—
PB4 I/O (port)	CKE output (BSC)	—	—	TIOC4A I/O (MTU2)	—	—
PB3 I/O (port)	CK output (CPG)	—	—	—	—	—
PB2 I/O (port)	$\overline{\text{CS0}}$ output (BSC)	—	—	$\overline{\text{POE4}}$ input (POE)	SCK0 I/O (SCIF0)	—
PB1 I/O (port)	RD/$\overline{\text{WR}}$ output (BSC)	—	—	$\overline{\text{POE8}}$ input (POE)	TXD0 output (SCIF0)	—
PB0 I/O (port)	$\overline{\text{RD}}$ output (BSC)	—	—	$\overline{\text{POE0}}$ input (POE)	RXD0 input (SCIF0)	—

PD12 I/O (port)	D12 I/O (BSC)	—	—	TIC5U input (MTU2)	—	–
PD11 I/O (port)	D11 I/O (BSC)	—	—	TIC5V input (MTU2)	—	–
PD10 I/O (port)	D10 I/O (BSC)	—	—	TIC5W input (MTU2)	—	–
PD9 I/O (port)	D9 I/O (BSC)	—	—	—	—	–
PD8 I/O (port)	D8 I/O (BSC)	—	—	—	—	–
PD7 I/O (port)	D7 I/O (BSC)	—	—	—	—	–
PD6 I/O (port)	D6 I/O (BSC)	—	—	—	—	–
PD5 I/O (port)	D5 I/O (BSC)	—	—	—	—	–
PD4 I/O (port)	D4 I/O (BSC)	—	—	—	—	–
PD3 I/O (port)	D3 I/O (BSC)	—	—	—	—	–
PD2 I/O (port)	D2 I/O (BSC)	—	—	—	—	–
PD1 I/O (port)	D1 I/O (BSC)	—	—	—	—	–
PD0 I/O (port)	D0 I/O (BSC)	—	—	—	—	–

Port A control register H2	PACRH2	R/W	H'0000/ H'1111	H'FFFE380C	8, 1
Port A control register H1	PACRH1	R/W	H'0000/ H'1111	H'FFFE380E	8, 1
Port A control register L4	PACRL4	R/W	H'0000/ H'1111	H'FFFE3810	8, 1
Port A control register L3	PACRL3	R/W	H'0000/ H'1111	H'FFFE3812	8, 1
Port A control register L2	PACRL2	R/W	H'0000/ H'1111	H'FFFE3814	8, 1
Port A control register L1	PACRL1	R/W	H'0000/ H'1111	H'FFFE3816	8, 1
Port B I/O register H	PBIORH	R/W	H'0000	H'FFFE3884	8, 1
Port B I/O register L	PBIORL	R/W	H'0000	H'FFFE3886	8, 1
Port B control register H4	PBCRH4	R/W	H'0000	H'FFFE3888	8, 1
Port B control register H3	PBCRH3	R/W	H'0000	H'FFFE388A	8, 1
Port B control register H2	PBCRH2	R/W	H'0000	H'FFFE388C	8, 1
Port B control register H1	PBCRH1	R/W	H'0000/ H'0001	H'FFFE388E	8, 1
Port B control register L4	PBCRL4	R/W	H'0000	H'FFFE3890	8, 1
Port B control register L3	PBCRL3	R/W	H'0000/ H'0011	H'FFFE3892	8, 1
Port B control register L2	PBCRL2	R/W	H'0000	H'FFFE3894	8, 1
Port B control register L1	PBCRL1	R/W	H'0000/ H'0101/ H'1101	H'FFFE3896	8, 1

Port F control register L1	PFCRL1	R/W	H'0000	H'FFFE3A96	8, 16
IRQOUT function control register	IFCR	R/W	H'0000	H'FFFE38A2	16
WAVE function control register 2	WAVECR2	R/W	H'0001	H'FFFE3A14	8, 16
WAVE function control register 1	WAVECR1	R/W	H'1111	H'FFFE3A16	8, 16

PAIORH and PAIORL are initialized to H'0000 by a power-on reset; but are not initializ
manual reset or in sleep mode or software standby mode.

(1) Port A I/O Register H (PAIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	-	-	-	-	-	PA25 IOR	PA24 IOR	PA23 IOR	PA22 IOR	PA21 IOR	PA20 IOR	PA19 IOR	PA18 IOR
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(2) Port A I/O Register L (PAIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	-	-	-	-	-	-	-	-	-	PA25MD[2:0]			-	PA24	
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/W	R/W	R/W	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
6 to 4	PA25MD[2:0]	000*	R/W	PA25 Mode Select the function of the PA25/A25/IRQ7/TIOC0D/TXD1 pin. • Area 0: 16-bit mode/8-bit mode 000: PA25 I/O (port) 001: A25 output (BSC) (initial value) 010: Setting prohibited 011: IRQ7 input (INTC) 100: TIOC0D I/O (MTU2) 101: TXD1 output (SCIF) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

		100: TIOC0C I/O (MTU2) 101: RXD1 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PA23MD[2:0]	000*	R/W	PA23 Mode Select the function of the PA23/A23/IRQ5/TIOC0B/SCK1 pin. • Area 0: 16-bit mode/8-bit mode 000: PA23 I/O (port) 001: A23 output (BSC) (initial value) 010: Setting prohibited 011: IRQ5 input (INTC) 100: TIOC0B I/O (MTU2) 101: SCK1 I/O (SCIF) 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

				100: TIOC0A I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
6 to 4	PA21MD[2:0]	000*	R/W	PA21 Mode Select the function of the PA21/A21/IRQ3 p • Area 0: 16-bit mode/8-bit mode 000: PA21 I/O (port) 001: A21 output (BSC) (initial value) 010: Setting prohibited 011: IRQ3 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

010: Setting prohibited
011: IRQ2 input (INTC)
100: Setting prohibited
101: Setting prohibited
110: Setting prohibited
111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PA19MD[2:0]	000*	R/W	PA19 Mode Select the function of the PA19/A19/IRQ1 p • Area 0: 16-bit mode/8-bit mode 000: PA19 I/O (port) 001: A19 output (BSC) (initial value) 010: Setting prohibited 011: IRQ1 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PA18MD[2:0]	000*	R/W	PA18 Mode Select the function of the PA18/A18/IRQ0 p • Area 0: 16-bit mode/8-bit mode 000: PA18 I/O (port) 001: A18 output (BSC) (initial value) 010: Setting prohibited 011: IRQ0 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TXD3 output (SCIF) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2 to 0	PA16MD[2:0]	000*	R/W	PA16 Mode Select the function of the PA16/A16/RXD3 pi • Area 0: 16-bit mode/8-bit mode 000: PA16 I/O (port) 001: A16 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: RXD3 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PA15MD[2:0]	000*	R/W	PA15 Mode Select the function of the PA15/A15/SCK3 p • Area 0: 16-bit mode/8-bit mode 000: PA15 I/O (port) 001: A15 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: SCK3 I/O (SCIF) 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PA14MD[2:0]	000*	R/W	PA14 Mode Select the function of the PA14/A14 pin. • Area 0: 16-bit mode/8-bit mode 000: PA14 I/O (port) 001: A14 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2 to 0	PA12MD[2:0]	000*	R/W	PA12 Mode Select the function of the PA12/A12 pin. • Area 0: 16-bit mode/8-bit mode 000: PA12 I/O (port) 001: A12 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PA11MD[2:0]	000*	R/W	PA11 Mode Select the function of the PA11/A11 pin. • Area 0: 16-bit mode/8-bit mode 000: PA11 I/O (port) 001: A11 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PA10MD[2:0]	000*	R/W	PA10 Mode Select the function of the PA10/A10 pin. • Area 0: 16-bit mode/8-bit mode 000: PA10 I/O (port) 001: A10 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2 to 0	PA8MD[2:0]	000*	R/W	PA8 Mode Select the function of the PA8/A8 pin. • Area 0: 16-bit mode/8-bit mode 000: PA8 I/O (port) 001: A8 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PA7MD[2:0]	000*	R/W	PA7 Mode Select the function of the PA7/A7 pin. • Area 0: 16-bit mode/8-bit mode 000: PA7 I/O (port) 001: A7 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PA6MD[2:0]	000*	R/W	PA6 Mode Select the function of the PA6/A6 pin. • Area 0: 16-bit mode/8-bit mode 000: PA6 I/O (port) 001: A6 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2 to 0	PA4MD[2:0]	000*	R/W	PA4 Mode Select the function of the PA4/A4 pin. • Area 0: 16-bit mode/8-bit mode 000: PA4 I/O (port) 001: A4 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PA3MD[2:0]	000*	R/W	PA3 Mode Select the function of the PA3/A3 pin. • Area 0: 16-bit mode/8-bit mode 000: PA3 I/O (port) 001: A3 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PA2MD[2:0]	000*	R/W	PA2 Mode Select the function of the PA2/A2 pin. • Area 0: 16-bit mode/8-bit mode 000: PA2 I/O (port) 001: A2 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2 to 0	PA0MD[2:0]	000*	R/W	PA0 Mode Select the function of the PA0/A0 pin. • Area 0: 16-bit mode/8-bit mode 000: PA0 I/O (port) 001: A0 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

PBIORH and PBIORL are initialized to H'0000 by a power-on reset; but are not initializ
manual reset or in sleep mode or software standby mode.

(1) Port B I/O Register H (PBIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	PB30 IOR	PB29 IOR	PB28 IOR	PB27 IOR	PB26 IOR	PB25 IOR	PB24 IOR	PB23 IOR	PB22 IOR	PB21 IOR	PB20 IOR	PB19 IOR	PB18 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	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

(2) Port B I/O Register L (PBIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	PB15 IOR	PB14 IOR	PB13 IOR	PB12 IOR	PB11 IOR	PB10 IOR	PB9 IOR	PB8 IOR	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	-	-	-	-	-	PB30MD[2:0]			-	PB29MD[2:0]			-	PB28	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
10 to 8	PB30MD[2:0]	000	R/W	PB30 Mode Select the function of the PB30/IRQOUT/REFOUT/UBCTRG pin. 000: PB30 I/O (port) 001: IRQOUT/REFOUT output (INTC/BSC) 010: Setting prohibited 011: UBCTRG output (UBC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

				110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
2 to 0	PB28MD[2:0]	000	R/W	PB28 Mode Select the function of the PB28/DACK0/TIOC1A/RXD3 pin. 000: PB28 I/O (port) 001: Setting prohibited 010: Dack0 output (DMAC) 011: Setting prohibited 100: TIOC1A I/O (MTU2) 101: RXD3 input (SCIF) 110: Setting prohibited 111: Setting prohibited

				always be 0.
14 to 12	PB27MD[2:0]	000	R/W	PB27 Mode Select the function of the PB27/TEND0/TIOC2A/TXD3/AUDATA0 pin 000: PB27 I/O (port) 001: Setting prohibited 010: TEND0 output (DMAC) 011: Setting prohibited 100: TIOC2A I/O (MTU2) 101: TXD3 output (SCIF) 110: AUDATA0 output (AUD) 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PB26MD[2:0]	000	R/W	PB26 Mode Select the function of the PB26/DREQ0/TIOC2B/SCK3/AUDATA1 pi 000: PB26 I/O (port) 001: Setting prohibited 010: DREQ1 input (DMAC) 011: Setting prohibited 100: TIOC2B I/O (MTU2) 101: SCK3 I/O (SCIF) 110: AUDATA1 output (AUD) 111: Setting prohibited

				010: DACK1 output (DMAC) 011: IRQ3 input (INTC) 100: TCLKA input (MTU2) 101: TXD3 output (SCIF) 110: AUDATA2 output (AUD) 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
2 to 0	PB24MD[2:0]	000	R/W	PB24 Mode Select the function of the PB24/TEND1/IRQ2/TCLKB/RXD3/AUDAT 000: PB24 I/O (port) 001: Setting prohibited 010: TEND1 output (DMAC) 011: IRQ2 input (INTC) 100: TCLKB input (MTU2) 101: RXD3 input (SCIF) 110: AUDATA3 output (AUD) 111: Setting prohibited.

				always be 0.
14 to 12	PB23MD[2:0]	000	R/W	PB23 Mode Select the function of the PB23/DREQ2/TCLKC/TXD2/AUDCK pin. 000: PB23 I/O (port) 001: Setting prohibited 010: DREQ2 input (DMAC) 011: Setting prohibited 100: TCLKC input (MTU2) 101: TXD2 output (SCIF) 110: AUDCK output (AUD) 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PB22MD[2:0]	000	R/W	PB22 Mode Select the function of the PB22/DACK2/TCLKD/RXD2/AUDSYNC pi 000: PB22 I/O (port) 001: Setting prohibited 010: DACK2 output (DMAC) 011: Setting prohibited 100: TCLKD input (MTU2) 101: RXD2 input (SCIF) 110: AUDSYNC output (AUD) 111: Setting prohibited

				010: Setting prohibited 011: IRQ0 input (INTC) 100: TIOC3BS I/O (MTU2S) 101: RXD0 input (SCIF) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
2 to 0	PB20MD[2:0]	000	R/W	PB20 Mode Select the function of the PB20/$\overline{BS}$/TIOC3 000: PB20 I/O (port) 001: $\overline{BS}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC3DS I/O (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				always be 0.
14 to 12	PB19MD[2:0]	000	R/W	PB19 Mode Select the function of the PB19/$\overline{\text{CS6}}$/IRQ6/ pin. 000: PB19 I/O (port) 001: $\overline{\text{CS6}}$ output (BSC) 010: Setting prohibited 011: IRQ6 input (INTC) 100: TIOC3D I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PB18MD[2:0]	000	R/W	PB18 Mode Select the function of the PB18/$\overline{\text{CS4}}$/IRQ4/ pin. 000: PB18 I/O (port) 001: $\overline{\text{CS4}}$ output (BSC) 010: Setting prohibited 011: IRQ4 input (INTC) 100: TIOC3B I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: IRQ1 input (INTC) 100: TIOC3A I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
2 to 0	PB16MD[2:0]	000*	R/W	PB16 Mode Select the function of the PB16/$\overline{CS1}$/$\overline{POE}$ pin. • Area 0: 16-bit mode/8-bit mode 000: PB16 I/O (port) 001: $\overline{CS1}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: $\overline{POE1}$ input (POE2) 101: TXD0 output (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PB15MD[2:0]	000	R/W	PB15 Mode Select the function of the PB15/$\overline{CS5}$/IRQ5/ pin. 000: PB15 I/O (port) 001: $\overline{CS5}$ output (BSC) 010: Setting prohibited 011: IRQ5 input (INTC) 100: TIOC3C I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PB14MD[2:0]	000	R/W	PB14 Mode Select the function of the PB14/$\overline{ADTRG}$/RXD2/$\overline{MRES}$ pin. 000: PB14 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: $\overline{ADTRG}$ input (ADC) 100: Setting prohibited 101: RXD2 input (SCIF) 110: $\overline{MRES}$ input (system control) 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: TIOC4BS I/O (MTU2S) 101: SCK2 I/O (SCIF) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
2 to 0	PB12MD[2:0]	000	R/W	PB12 Mode Select the function of the PB12/$\overline{\text{BREQ}}$/TIOC4AS/TXD2 pin. 000: PB12 I/O (port) 001: $\overline{\text{BREQ}}$ input (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC4AS I/O (MTU2S) 101: TXD2 output (SCIF) 110: Setting prohibited 111: Setting prohibited

				always be 0.
14 to 12	PB11MD[2:0]	000	R/W	PB11 Mode Select the function of the PB11/$\overline{\text{AH}}$/DACK3/TIOC4DS/TXD2 pin. 000: PB11 I/O (port) 001: $\overline{\text{AH}}$ output (BSC) 010: DACK3 output (DMAC) 011: Setting prohibited 100: TIOC4DS I/O (MTU2S) 101: TXD2 output (SCIF) 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PB10MD[2:0]	000	R/W	PB10 Mode Select the function of the PB10/$\overline{\text{WAIT}}$/DREQ3/TIOC4CS/RXD2 pin. 000: PB10 I/O (port) 001: $\overline{\text{WAIT}}$ input (BSC) 010: DREQ3 input (DMAC) 011: Setting prohibited 100: TIOC4CS I/O (MTU2S) 101: RXD2 input (SCIF) 110: Setting prohibited 111: Setting prohibited

				001: WE1/DQMLU output (BSC) (initia 010: Setting prohibited 011: Setting prohibited 100: TIOC3CS I/O (MTU2S) 101: TXD3 output (SCIF) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
2 to 0	PB8MD[2:0]	000*	R/W	PB8 Mode Select the function of the PB8/$\overline{WE0}$/$\overline{DQMLL}$/TIOC3AS/RXD3 pin. • Area 0: 16-bit mode/8-bit mode 000: PB8 I/O (port) 001: $\overline{WE0}$/$\overline{DQMLL}$ output (BSC) (initia 010: Setting prohibited 011: Setting prohibited 100: TIOC3AS I/O (MTU2S) 101: RXD3 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PB7MD[2:0]	000	R/W	PB7 Mode Select the function of the PB7/$\overline{CS7}$/IRQ7/TIO 000: PB7 I/O (port) 001: $\overline{CS7}$ output (BSC) 010: Setting prohibited 011: IRQ7 input (INTC) 100: TIOC4D I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
10 to 8	PB6MD[2:0]	000	R/W	PB6 Mode Select the function of the PB6/$\overline{CASL}$/IRQ3/TI pin. 000: PB6 I/O (port) 001: $\overline{CASL}$ output (BSC) 010: Setting prohibited 011: IRQ3 input (INTC) 100: TIOC4C I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: IRQ2 input (INTC) 100: TIOC4B I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
2 to 0	PB4MD[2:0]	000	R/W	PB4 Mode Select the function of the PB4/CKE/TIOC4 000: PB4 I/O (port) 001: CKE output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC4A I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				always be 0.
14 to 12	PB3MD[2:0]	000*	R/W	PB3 Mode Select the function of the PB3/CK pin. • Area 0: 16-bit mode/8-bit mode 000: PB3 I/O (port) 001: CK output (CPG) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PB2MD[2:0]	000*	R/W	PB2 Mode Select the function of the PB2/$\overline{CS0}$/$\overline{POE4}$/S • Area 0: 16-bit mode/8-bit mode 000: PB2 I/O (port) 001: $\overline{CS0}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: $\overline{POE4}$ input (POE2) 101: SCK0 I/O (SCIF) 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: $\overline{\text{POE8}}$ input (POE2) 101: TXD0 output (SCIF) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
2 to 0	PB0MD[2:0]	000*	R/W	PB0 Mode Select the function of the PB0/$\overline{\text{RD}}$/$\overline{\text{POE0}}$/P • Area 0: 16-bit mode/8-bit mode 000: PB0 I/O (port) 001: $\overline{\text{RD}}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: $\overline{\text{POE0}}$ input (POE2) 101: RXD0 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	PD15 IOR	PD14 IOR	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	
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	

19.1.6 Port D Control Registers L1 to L4 (PDCRL1 to PDCRL4)

PDCRL1 to PDCRL4 are 16-bit readable/writable registers that are used to select the fun the multiplexed pins on port D.

PDCRL1 to PDCRL4 are initialized to the values shown in table 19.6 by a power-on rese not initialized by a manual reset or in sleep mode or software standby mode.

Table 19.6 Initial Values of Port D Control Registers

	Initial Value	
Register Name	**Area 0: 16-Bit Mode**	**Area 0: 8-Bit Mode**
PDCRL4	H'1111	H'0000
PDCRL3	H'1111	H'0000
PDCRL2	H'1111	H'1111
PDCRL1	H'1111	H'1111

				always be 0.
14 to 12	PD15MD[2:0]	000*	R/W	PD15 Mode Select the function of the PD15/D15/TIC5US • Area 0: 16-bit mode 000: PD15 I/O (port) 001: D15 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIC5US input (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD15 I/O (port) (initial value) 001: D15 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5US input (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

				101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD14 I/O (port) (initial value) 001: D14 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5VS input (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

				101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD13 I/O (port) (initial value) 001: D13 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5WS input (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

101: Setting prohibited
110: Setting prohibited
111: Setting prohibited

- Area 0: 8-bit mode
  000: PD12 I/O (port) (initial value)
  001: D12 I/O (data)
  010: Setting prohibited
  011: Setting prohibited
  100: TIC5U input (MTU2)
  101: Setting prohibited
  110: Setting prohibited
  111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PD11MD[2:0]	000*	R/W	PD11 Mode Select the function of the PD11/D11/TIC5V • Area 0: 16-bit mode 000: PD11 I/O (port) 001: D11 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIC5V input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD11 I/O (port) (initial value) 001: D11 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5V input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

				101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD10 I/O (port) (initial value) 001: D10 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5W input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

				101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD9 I/O (port) (initial value) 001: D9 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

101: Setting prohibited
110: Setting prohibited
111: Setting prohibited

- Area 0: 8-bit mode
  000: PD8 I/O (port) (initial value)
  001: D8 I/O (data)
  010: Setting prohibited
  011: Setting prohibited
  100: Setting prohibited
  101: Setting prohibited
  110: Setting prohibited
  111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PD7MD[2:0]	000*	R/W	PD7 Mode Select the function of the PD7/D7 pin. • Area 0: 8-bit mode 000: PD7 I/O (port) 001: D7 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PD6MD[2:0]	000*	R/W	PD6 Mode Select the function of the PD6/D6 pin. • Area 0: 8-bit mode 000: PD6 I/O (port) 001: D6 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2 to 0	PD4MD[2:0]	000*	R/W	PD4 Mode Select the function of the PD4/D4 pin. • Area 0: 8-bit mode 000: PD4 I/O (port) 001: D4 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

				always be 0.
14 to 12	PD3MD[2:0]	000*	R/W	PD3 Mode Select the function of the PD3/D3 pin. • Area 0: 8-bit mode 000: PD3 I/O (port) 001: D3 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	PD2MD[2:0]	000*	R/W	PD2 Mode Select the function of the PD2/D2 pin. • Area 0: 8-bit mode 000: PD2 I/O (port) 001: D2 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

				010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2 to 0	PD0MD[2:0]	000*	R/W	PD0 Mode Select the function of the PD0/D0 pin. • Area 0: 8-bit mode 000: PD0 I/O (port) 001: D0 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
6 to 4	PF1MD[2:0]	000	R/W	PF1 Mode Select the function of the PF1/IRQ1/$\overline{\text{POE3}}$ 000: PF1 input (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ1 input (INTC) 100: $\overline{\text{POE3}}$ input (POE2) 101: SDA I/O (IIC3) 110: Setting prohibited 111: Setting prohibited

011: IRQ0 input (INTC)

100: $\overline{\text{POE7}}$ input (POE2)

101: SCL I/O (IIC3)

110: Setting prohibited

111: Setting prohibited

	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
1, 0	IRQMD[1:0]	00	R/W	IRQOUT Mode Select the function of the $\overline{\text{IRQOUT}}$/REFOUT bits 10 to 8 (PB30MD[2:0]) in PBCRH4 are B'001. 00: Interrupt request accept signal output 01: Refresh signal output 10: Interrupt request accept signal output or signal output (depends on the operating 11: Always high-level output

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/W	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
2 to 0	WVRMD[2:0]	001	R/W	WRXD Mode Select the function of the WRXD pin. 000: Setting prohibited 001: Initial value 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: WRXD input 111: Setting prohibited

14 to 12	WVTMD[2:0]	001	R/W	WTXD Mode Select the function of the WTXD pin. 000: Setting prohibited 001: Initial value 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: WTXD output 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
10 to 8	WVSMD[2:0]	001	R/W	WSCK Mode Select the function of the WSCK pin. 000: Setting prohibited 001: Initial value 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: WSCK output 111: Setting prohibited

				should always be 0.
0	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.

20.1 Port A

Port A is an input/output port with the 26 pins shown in figure 20.1.

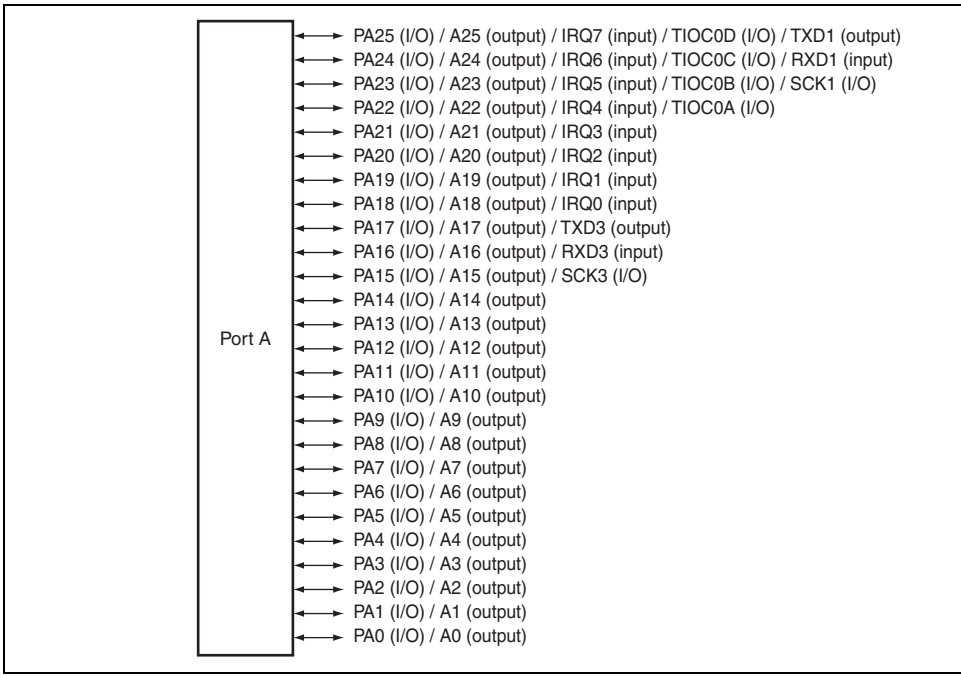


Figure 20.1 Port A

Port A port register L	PAPRL	R	H'xxxx	H'FFFE381E	8, 16

20.1.2 Port A Data Registers H, L (PADRH, PADRL)

PADRH and PADRL are 16-bit readable/writable registers that store port A data. Bits PA
PA0DR correspond to pins PA25 to PA0, respectively.

When a pin function is general output, if a value is written to PADRH or PADRL, the val
output directly from the pin, and if PADRH or PADRL is read, the register value is retur
directly regardless of the pin state.

When a pin function is general input, if PADRH or PADRL is read, the pin state, not the
value, is returned directly. If a value is written to PADRH or PADRL, although that valu
written into PADRH or PADRL, it does not affect the pin state. Table 20.2 summarizes P
and PADRL read/write operations.

PADRH and PADRL are initialized to the respective values shown in table 20.1 by a pow
reset. PADRH and PADRL are not initialized by a manual reset or in sleep mode or softw
standby mode.

				always be 0.
9	PA25DR	0	R/W	See table 20.2.
8	PA24DR	0	R/W	
7	PA23DR	0	R/W	
6	PA22DR	0	R/W	
5	PA21DR	0	R/W	
4	PA20DR	0	R/W	
3	PA19DR	0	R/W	
2	PA18DR	0	R/W	
1	PA17DR	0	R/W	
0	PA16DR	0	R/W	

13	PA13DR	0	R/W
12	PA12DR	0	R/W
11	PA11DR	0	R/W
10	PA10DR	0	R/W
9	PA9DR	0	R/W
8	PA8DR	0	R/W
7	PA7DR	0	R/W
6	PA6DR	0	R/W
5	PA5DR	0	R/W
4	PA4DR	0	R/W
3	PA3DR	0	R/W
2	PA2DR	0	R/W
1	PA1DR	0	R/W
0	PA0DR	0	R/W

	PADRL value	
Other than general output	PADRH or PADRL value	Can write to PADRH and PADRL, but it effect on pin state.

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
9	PA25PR	Pin state	R	The pin state is returned regardless of the PFC These bits cannot be modified.
8	PA24PR	Pin state	R	
7	PA23PR	Pin state	R	
6	PA22PR	Pin state	R	
5	PA21PR	Pin state	R	
4	PA20PR	Pin state	R	
3	PA19PR	Pin state	R	
2	PA18PR	Pin state	R	
1	PA17PR	Pin state	R	
0	PA16PR	Pin state	R	

13	PA13PR	Pin state	R
12	PA12PR	Pin state	R
11	PA11PR	Pin state	R
10	PA10PR	Pin state	R
9	PA9PR	Pin state	R
8	PA8PR	Pin state	R
7	PA7PR	Pin state	R
6	PA6PR	Pin state	R
5	PA5PR	Pin state	R
4	PA4PR	Pin state	R
3	PA3PR	Pin state	R
2	PA2PR	Pin state	R
1	PA1PR	Pin state	R
0	PA0PR	Pin state	R

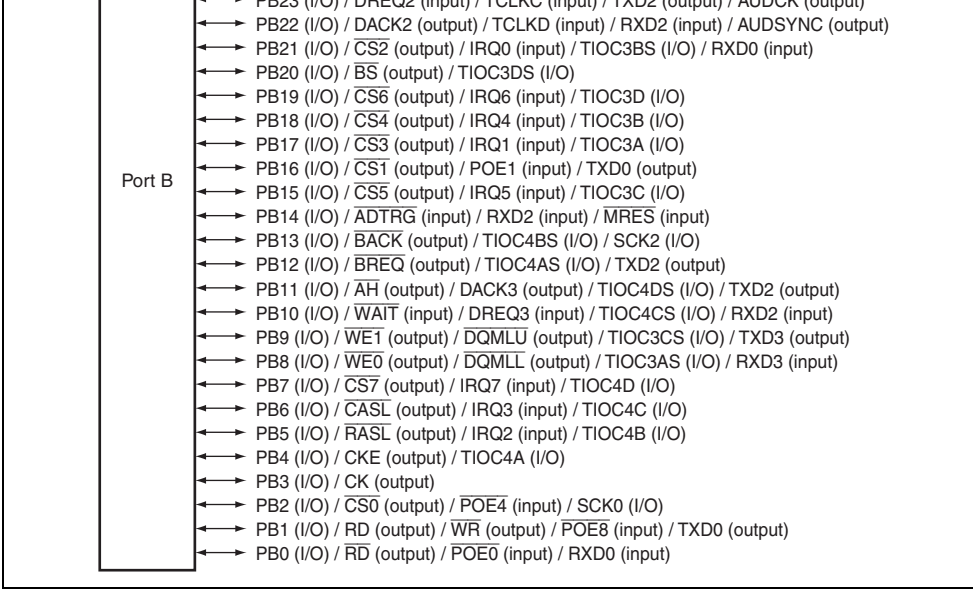


Figure 20.2 Port B

Port B port register L	PBPRL	R	H'xxxx	H'FFFE389E	8, 1

20.2.2 Port B Data Registers H, L (PBDRH, PBDRL)

PBDRH and PBDRL are 16-bit readable/writable registers that store port B data. Bits PB and PB0DR correspond to pins PB30 to PB0, respectively.

When a pin function is general output, if a value is written to PBDRH or PBDRL, the va output directly from the pin, and if PBDRH or PBDRL is read, the register value is retur directly regardless of the pin state.

When a pin function is general input, if PBDRH or PBDRL is read, the pin state, not the value, is returned directly. If a value is written to PBDRH or PBDRL, although that valu written into PBDRH or PBDRL, it does not affect the pin state. Table 20.4 summarizes and PBDRL read/write operations.

PBDRH and PBDRL are initialized to the value shown in table 20.3 by a power-on reset not initialized by a manual reset or in sleep mode or software standby mode.

				always be 0.
14	PB30DR	0	R/W	See table 20.4.
13	PB29DR	0	R/W	
12	PB28DR	0	R/W	
11	PB27DR	0	R/W	
10	PB26DR	0	R/W	
9	PB25DR	0	R/W	
8	PB24DR	0	R/W	
7	PB23DR	0	R/W	
6	PB22DR	0	R/W	
5	PB21DR	0	R/W	
4	PB20DR	0	R/W	
3	PB19DR	0	R/W	
2	PB18DR	0	R/W	
1	PB17DR	0	R/W	
0	PB16DR	0	R/W	

13	PB13DR	0	R/W
12	PB12DR	0	R/W
11	PB11DR	0	R/W
10	PB10DR	0	R/W
9	PB9DR	0	R/W
8	PB8DR	0	R/W
7	PB7DR	0	R/W
6	PB6DR	0	R/W
5	PB5DR	0	R/W
4	PB4DR	0	R/W
3	PB3DR	0	R/W
2	PB2DR	0	R/W
1	PB1DR	0	R/W
0	PB0DR	0	R/W

	value	
Other than general output	PBDRH/PBDRL value	Can write to PBDRH or PBDRL, but it ha effect on pin state.

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
14	PB30PR	Pin state	R	The pin state is returned regardless of the PF These bits cannot be modified.
13	PB29PR	Pin state	R	
12	PB28PR	Pin state	R	
11	PB27PR	Pin state	R	
10	PB26PR	Pin state	R	
9	PB25PR	Pin state	R	
8	PB24PR	Pin state	R	
7	PB23PR	Pin state	R	
6	PB22PR	Pin state	R	
5	PB21PR	Pin state	R	
4	PB20PR	Pin state	R	
3	PB19PR	Pin state	R	
2	PB18PR	Pin state	R	
1	PB17PR	Pin state	R	
0	PB16PR	Pin state	R	

13	PB13PR	Pin state	R
12	PB12PR	Pin state	R
11	PB11PR	Pin state	R
10	PB10PR	Pin state	R
9	PB9PR	Pin state	R
8	PB8PR	Pin state	R
7	PB7PR	Pin state	R
6	PB6PR	Pin state	R
5	PB5PR	Pin state	R
4	PB4PR	Pin state	R
3	PB3PR	Pin state	R
2	PB2PR	Pin state	R
1	PB1PR	Pin state	R
0	PB0PR	Pin state	R

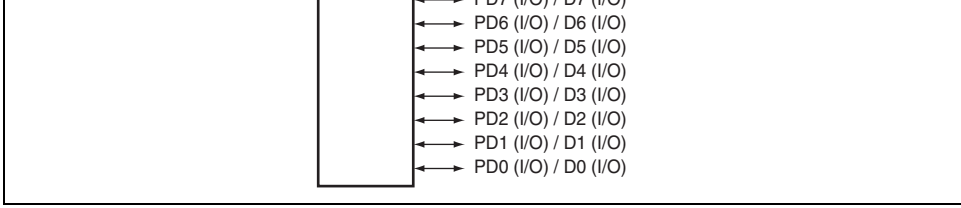


Figure 20.3 Port D

20.3.1 Register Descriptions

Table 20.5 lists the port D registers.

Table 20.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Acc
Port D data register L	PDDRL	R/W	H'0000	H'FFFE3982	8, 1
Port D port register L	PDPRL	R	H'xxxx	H'FFFE399E	8, 1

does not affect the pin state. Table 20.8 summarizes PDDRL read/write operations.

PDDRL is initialized to the respective values shown in table 20.5 by a power-on reset, bu
initialized by a manual reset or in sleep mode or software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	PD15 DR	PD14 DR	PD13 DR	PD12 DR	PD11 DR	PD10 DR	PD9 DR	PD8 DR	PD7 DR	PD6 DR	PD5 DR	PD4 DR	PD3 DR	PD2 DR	
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

8	PD8DR	0	R/W
7	PD7DR	0	R/W
6	PD6DR	0	R/W
5	PD5DR	0	R/W
4	PD4DR	0	R/W
3	PD3DR	0	R/W
2	PD2DR	0	R/W
1	PD1DR	0	R/W
0	PD0DR	0	R/W

Table 20.6 Port D Data Register L (PDDRL) Read/Write Operations

- PDDRL bits 15 to 0

PDIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PDDRL, but it has no effec state.
	Other than general input	Pin state	Can write to PDDRL, but it has no effec state.
1	General output	PDDRL value	The value written is output from the pin.
	Other than general output	PDDRL value	Can write to PDDRL, but it has no effec state.

Bit	Bit Name	Value	R/W	Description
15	PD15PR	Pin state	R	The pin state is returned regardless of the PFC These bits cannot be modified.
14	PD14PR	Pin state	R	
13	PD13PR	Pin state	R	
12	PD12PR	Pin state	R	
11	PD11PR	Pin state	R	
10	PD10PR	Pin state	R	
9	PD9PR	Pin state	R	
8	PD8PR	Pin state	R	
7	PD7PR	Pin state	R	
6	PD6PR	Pin state	R	
5	PD5PR	Pin state	R	
4	PD4PR	Pin state	R	
3	PD3PR	Pin state	R	
2	PD2PR	Pin state	R	
1	PD1PR	Pin state	R	
0	PD0PR	Pin state	R	

20.4.1 Register Descriptions

Table 20.7 lists the port F register.

Table 20.7 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Acc
Port F data register	PFDR	R	H'0000	H'FFFE3A82	8, 1

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

Note: * Depends on the external pin state.

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
1	PF1DR	Pin state	R	See table 20.8.
0	PF0DR	Pin state	R	

Table 20.8 Port F Data Register (PFDR) Read/Write Operations

- PFDR bits 1 and 0

Pin Function	Read	Write
General input	Pin state	Ignored (no effect on pin state)
Other than general input	Pin state	Ignored (no effect on pin state)

- $\overline{\text{WE1}}$/DQMLU and $\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$
- $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$
- CKE
- $\overline{\text{WAIT}}$
- $\overline{\text{BREQ}}$
- $\overline{\text{BACK}}$
- $\overline{\text{MRES}}$

LSI has started up.

— Size of the user MAT, from which booting-up proceeds after a power-on reset in mode: 384/512 kbytes*

— Size of the user boot MAT, from which booting-up proceeds after a power-on res boot mode: 12 kbytes

- Three on-board programming modes and one off-board programming mode

  On-board programming modes

  Boot Mode: The on-chip SCIF interface is used for programming in this mode. Eithe MAT or user-boot MAT can be programmed, and the bit rate for data transfer betwe and this LSI are automatically adjusted.

  User Program Mode: This mode allows programming of the user MAT via any des interface.

  User Boot Mode: This mode allows writing of a user boot program via any desired i and programming of the user MAT.

  Off-board programming mode

  Programmer Mode: This mode allows programming of the user MAT and user boo with the aid of a PROM programmer.

- Downloading of an on-chip program to provide an interface for programming/erasur

  This LSI has a dedicated programming/erasing program. After this program has beer downloaded to the on-chip RAM, programming or erasing can be performed by setti parameters as arguments. “User branching” is also supported.

Note: * See Appendix B. Product Lineup.

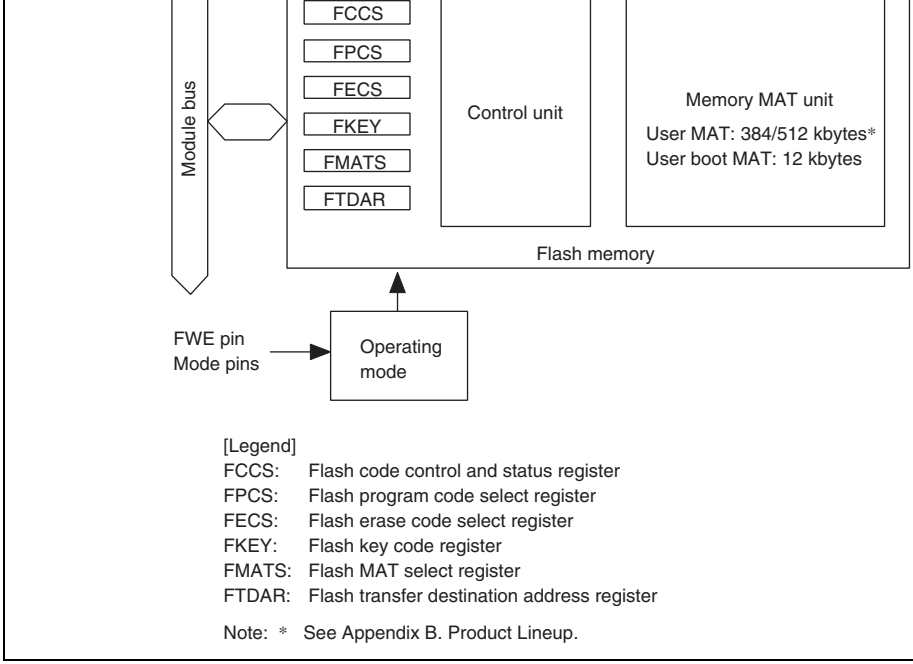
When an abnormal state is detected, such as runaway execution of programming/erasi
protection modes initiate the transition to the error protection state and suspend
programming/erasing processing.

- Programming/erasing time

  The time taken to program 256 bytes of flash memory in a single round is 2 ms (typ.),
  equivalent to 7.8 µs per byte. The erasing time is 80 ms (typ.) per 8-Kbyte block, 600
  per 64-Kbyte block, and 1200 ms (typ.) per 128-Kbyte block.
- Number of programming operations

  The flash memory can be programmed up to 100 times.
- Operating frequency for programming/erasing

  The operating frequency range for programming/erasing Iϕ = 32 to 40 MHz



[Legend]
FCCS: Flash code control and 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

Note: * See Appendix B. Product Lineup.

Figure 21.1 Block Diagram of Flash Memory

- Flash memory can be read, programmed, or erased on the board only in user program user boot mode, and boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programm programmer mode.

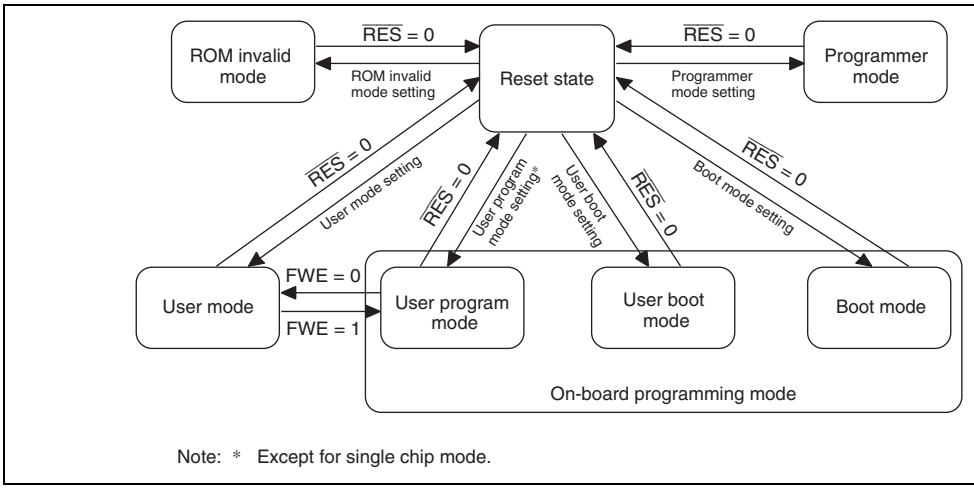


Figure 21.2 Mode Transition of Flash Memory

2. MD0 = 0: External bus can be used, MD0 = 1: Single-chip mode (external bus
be used)

Programming/ erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MA User boo
Programming/ erasing control	Command method	Programming/ erasing interface	Programming/ erasing interface	—
All erasure	Possible (Automatic)	Possible	Possible	Possible (Automa
Block division erasure	Possible*1	Possible	Possible	Not poss
Program data transfer	From host via SCIF	From optional device via RAM	From optional device via RAM	Via prog
User branch function	Not possible	Possible	Possible	Not poss
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*2	Embedd program MAT
Transition to user mode	Mode setting change and reset	FWE setting change	Mode setting change and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.
2. Initiation starts from the embedded program storage MAT. After checking the f memory related registers, initiation starts from the reset vector of the user MA

- The user boot MAT can be programmed or erased only in boot mode and programme
- The user MAT and user boot MAT are all erased in boot mode. Then, the user MAT a boot MAT can be programmed by means of the command method. However, the cont the MAT cannot be read until this state.
  Only user boot MAT is programmed and the user MAT is programmed in user boot m only user MAT is programmed because user boot mode is not used.
- In user boot mode, the boot operation of the optional interface can be performed by a setting different from user program mode.

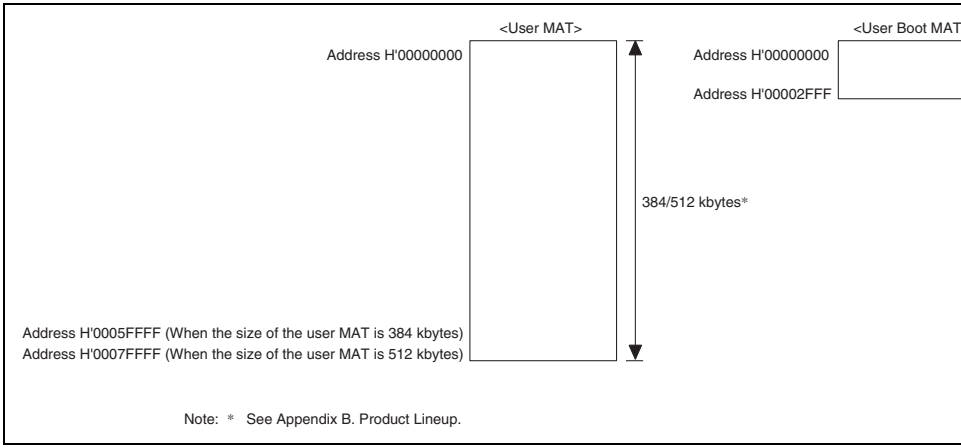


Figure 21.3 Flash Memory Configuration

The user MAT and user boot MAT have different memory sizes. Do not access a user bo
that is 12 kbytes or more. When a user boot MAT exceeding 12 kbytes is read from, an u
value is read.

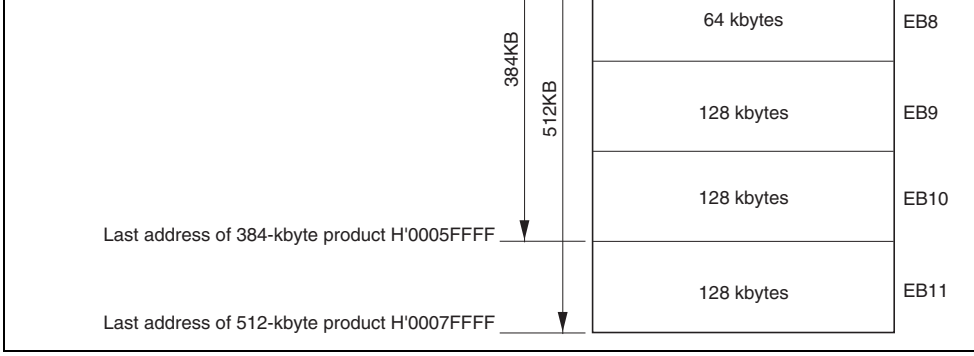


Figure 21.4 Block Division of User MAT

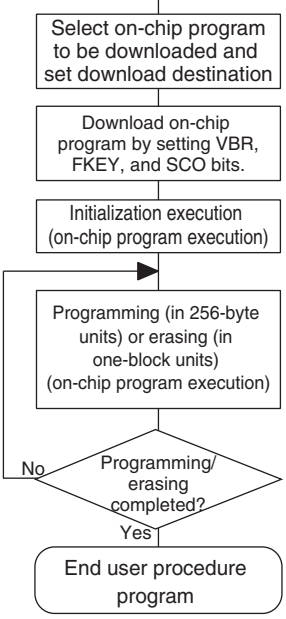


Figure 21.5 Overview of User Procedure Program

(1) Selection of On-Chip Program to be Downloaded and Setting of Download Destinat

This LSI has programming/erasing programs and they can be downloaded to the on-
RAM. The on-chip program to be downloaded is selected by setting the correspondi
the programming/erasing interface registers. The download destination can be specif
FTDAR.

Note that VBR can be changed after download is completed.

(3) Initialization of Programming/Erasing

The operating frequency and user branch are set before execution of programming/era
The user branch destination must be in an area other than the user MAT area which is
middle of programming and the area where the on-chip program is downloaded. Thes
are performed by using the programming/erasing interface parameters.

(4) Programming/Erasing Execution

To program or erase, the FWE pin must be brought high and user program mode mus
entered.

The program data/programming destination address is specified in 256-byte units whe
programming.

The block to be erased is specified in erase-block units when erasing.

These specifications are set by using the programming/erasing interface parameters a
chip program is initiated. The on-chip program is executed by using the JSR or BSR
instruction to perform the subroutine call of the specified address in the on-chip RAM
execution result is returned to the programming/erasing interface parameters.

The area to be programmed must be erased in advance when programming flash mem

Ensure that NMI, IRQ, and all other interrupts are not generated during programming
erasing.

(5) When Programming/Erasing is Executed Consecutively

When the processing is not ended by the 256-byte programming or one-block erasure
program address/data and erase-block number must be updated and consecutive
programming/erasing is required.

Since the downloaded on-chip program is left in the on-chip RAM after the processin
download and initialization are not required when the same processing is executed
consecutively.

Mode 0	MD0	Input	Sets operating mode of this L
Transmit data	TXD1 (PA25)	Output	Serial transmit data output (u boot mode)
Receive data	RXD1 (PA24)	Input	Serial receive data input (use mode)

21.4 Register Descriptions

21.4.1 Registers

The registers/parameters which control flash memory when the on-chip flash memory is shown in table 21.4.

There are several operating modes for accessing flash memory, for example, read mode/ mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/p are allocated for each operating mode and MAT selection. The correspondence of opera and registers/parameters for use is shown in table 21.5.

Flash transfer destination address register	FTDAR	R/W	H'00	H'8000C006

Notes: 1. The bits except the SCO bit are read-only bits. The SCO bit is a programming-
(The value which can be read is always 0.)
2. The initial value of the FWE bit is 0 when the FWE pin goes low.
The initial value of the FWE bit is 1 when the FWE pin goes high.
3. The initial value at initiation in user mode or user program mode is H'00.
The initial value at initiation in user boot mode is H'AA.

Table 21.4 (2) Parameter Configuration

Name	Abbreviation	R/W	Initial Value	Address	A Si
Download pass/fail result	DPFR	R/W	Undefined	On-chip RAM*	8,
Flash pass/fail result	FPFR	R/W	Undefined	R0 of CPU	8,
Flash multipurpose address area	FMPAR	R/W	Undefined	R5 of CPU	8,
Flash multipurpose data destination area	FMPDR	R/W	Undefined	R4 of CPU	8,
Flash erase block select	FEBS	R/W	Undefined	R4 of CPU	8,
Flash program and erase frequency control	FPEFEQ	R/W	Undefined	R4 of CPU	8,
Flash user branch address set parameter	FUBRA	R/W	Undefined	R5 of CPU	8,

Note: * One byte of the start address in the on-chip RAM area specified by FTDAR is

Programming/ erasing interface parameters	DPFR	√	—	—	—	—	—
	FPFR	—	√	√	√	—	—
	FPEFEQ	—	√	—	—	—	—
	FUBRA	—	√	—	—	—	—
	FMPAR	—	—	√	—	—	—
	FMPDR	—	—	√	—	—	—
	FEBS	—	—	—	√	—	—

Notes: 1. The setting is required when programming or erasing user MAT in user boot m
2. The setting may be required according to the combination of initiation mode a target MAT.

	FWE	MAT	-	FLER	-	-	-	SCO
Initial value:	1/0	1/0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	(R)/W

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1/0	R	Flash Programming Enable Monitors the level which is input to the FWE pin t performs hardware protection of the flash memor programming or erasing. The initial value is 0 or 1 according to the FWE pin state. 0: When the FWE pin goes low (in hardware prot state) 1: When the FWE pin goes high
6	MAT	1/0	R	MAT Bit Indicates whether the user MAT or user boot MA selected. 0: User MAT is selected 1: User boot MAT is selected
5	—	0	R	Reserved This bit is always read as 0. The write value shou be 0.

				0: Flash memory operates normally Programming/erasing protection for flash mer protection) is invalid. [Clearing condition] At a power-on reset 1: Indicates an error occurs during programming flash memory. Programming/erasing protection for flash mer protection) is valid. [Setting condition] See section 21.6.3, Error Protection.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

immediately after setting this bit to 1.

For interrupts during download, see section 21.7. Interrupts during Programming/Erasing. For the d time, see section 21.7.3, Other Notes.

Since this bit is cleared to 0 when download is co this bit cannot be read as 1.

Download by setting the SCO bit to 1 requires a s interrupt processing that performs bank switching on-chip program storage area. Therefore, before download request (SCO = 1), set VBR to H'80000 Otherwise, the CPU gets out of control. Once dow end is confirmed, VBR can be changed to any oth

The mode in which the FWE pin is high must be u when using the SCO function.

0: Download of the on-chip programming/erasing to the on-chip RAM is not executed.

[Clearing condition]

When download is completed

1: Request that the on-chip programming/erasing is downloaded to the on-chip RAM is generate

[Setting conditions]

When all of the following conditions are satisfied a written to this bit

- FKEY is written to H'A5
- During execution in the on-chip RAM

7 to 1		All 0	R	Reserved These bits are always read as 0. The write value always be 0.
0	PPVS	0	R/W	Program Pulse Single Selects the programming program. 0: On-chip programming program is not selected [Clearing condition] When transfer is completed 1: On-chip programming program is selected

(3) Flash Erase Code Select Register (FECS)

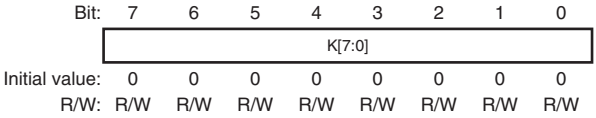
FECS selects download of the on-chip erasing program.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	EPVB
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

(2) Flash Key Code Register (FKEY)

FKEY is a register for software protection that enables download of the on-chip program programming/erasing of flash memory. Before setting the SCO bit to 1 in order to downlo on-chip program or executing the downloaded programming/erasing program, these proc cannot be executed if the key code is not written.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	K[7:0]	All 0	R/W	Key Code Only when H'A5 is written, writing to the SCO bit When a value other than H'A5 is written to FKEY, cannot be written to the SCO bit. Therefore down to the on-chip RAM cannot be executed. Only when H'5A is written, programming/erasing memory can be executed. Even if the on-chip programming/erasing program is executed, flash cannot be programmed or erased when a value o H'5A is written to FKEY. H'A5: Writing to the SCO bit is enabled (The SCO cannot be set by a value other than H'A5.) H'5A: Programming/erasing is enabled (A value o H'5A enables software protection state.) H'00: Initial value

7	MS7	0/1	R/W	MAT Select
6	MS6	0	R/W	These bits are in user-MAT selection state when other than H'AA is written and in user-boot-MAT state when H'AA is written. The MAT is switched by writing a value in FMAT on-chip RAM instruction. When the MAT is switched, follow section 21.7.1 Switching between User MAT and User Boot MAT user boot MAT cannot be programmed in user program mode if user boot MAT is selected by FMATS. The boot MAT must be programmed in boot mode or programmer mode.) H'AA: The user boot MAT is selected (in user-MAT selection state when the value of these bits other than H'AA) Initial value when these bits are initialized in boot mode. H'00: Initial value when these bits are initialized in except for user boot mode (in user-MAT state) [Programmable condition] These bits are in the execution state in the on-chip
5	MS5	0/1	R/W	
4	MS4	0	R/W	
3	MS3	0/1	R/W	
2	MS2	0	R/W	
1	MS1	0/1	R/W	
0	MS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	TDER	0	R/W	Transfer Destination Address Setting Error This bit is set to 1 when there is an error in the downloa address set by bits 6 to 0 (TDA6 to TDA0). Whether the setting is erroneous or not is tested by checking whethe setting of TDA6 to TDA0 is in the range of H'00 to H'05 setting the SCO bit in FCCS to 1 and performing downlo Before setting the SCO bit to 1 be sure to set the FTDA between H'00 to H'05 as well as clearing this bit to 0. 0: Setting of TDA6 to TDA0 is normal 1: Setting of TDER and TDA6 to TDA0 is H'06 to H'FF a download has been aborted
6 to 0	TDA[6:0]	All 0	R/W	Transfer Destination Address These bits specify the download start address. A value H'00 to H'05 can be set to specify the download start ad on-chip RAM in 2-kbyte units. A value from H'06 to H'7F cannot be set. If such a value the TDER bit (bit 7) in this register is set to 1 to prevent download from being executed. H'00: Download start address is set to H'FFF81000 H'01: Download start address is set to H'FFF81800 H'02: Download start address is set to H'FFF82000 H'03: Download start address is set to H'FFF82800 H'04: Download start address is set to H'FFF83000 H'05: Download start address is set to H'FFF83800 H'06 to H'7F: Setting prohibited. If this value is set, the T (bit 7) is set to 1 to abort the download pro

must be saved at the processing start. (The maximum size of a stack area to be used is 1

The programming/erasing interface parameters are used in the following four items.

1. Download control
2. Initialization before programming or erasing
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 21.6.

The processing results of initialization, programming, and erasing are returned, but the b
have different meanings according to the processing program. See the description of FP
each processing.

control								
Flash user branch address set	FUBRA	—	√	—	—	R/W	Undefined	R5
Flash multipurpose address area	FMPAR	—	—	√	—	R/W	Undefined	R5
Flash multipurpose data destination area	FMPDR	—	—	√	—	R/W	Undefined	R4
Flash erase block select	FEBS	—	—	—	√	R/W	Undefined	R4

Note: * One byte of start address of download destination specified by FTDAR

(1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-ch
area to be downloaded is the area as much as 3 kbytes starting from the start address spec
FTDAR. For the address map of the on-chip RAM, see figure 21.10.

The download control is set by using the programming/erasing interface registers. The re
value is given by the DPFR parameter.

(a) Download Pass/Fail Result Parameter (DPFR: One Byte of Start Address of O
RAM Specified by FTDAR)

This parameter indicates the return value of the download result. The value of this param
be used to determine if downloading is executed or not. Since the confirmation whether t
bit is set to 1 is difficult, the certain determination must be performed by setting one byte
start address of the on-chip RAM area specified by FTDAR to a value other than the retu
of download (for example, H'FF) before the download start (before setting the SCO bit to

				Return 0.
2	SS	Undefined	R/W	Source Select Error Detect The on-chip program which can be download specified as only one type. When more than t of the program are selected, the program is n selected, or the program is selected without n an error occurs. 0: Download program can be selected norma 1: Download error occurs (Multi-selection or p which is not mapped is selected)
1	FK	Undefined	R/W	Flash Key Register Error Detect Returns the check result whether the value of set to H'A5. 0: FKEY setting is normal (FKEY = H'A5) 1: FKEY setting is abnormal (FKEY = value o H'A5)
0	SF	Undefined	R/W	Success/Fail Returns the result whether download has end normally or not. 0: Downloading on-chip program has ended n (no error) 1: Downloading on-chip program has ended a (error occurs)

(2.1) Flash Programming/Erasing Frequency Parameter (FPEFEQ: General Registe CPU)

This parameter sets the operating frequency of the CPU.

The flash programming/erasing frequency Iϕ of this LSI is limited to 32 to 40 MHz.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

2. For example, when Iϕ = 33.333 MHz, set a

(1) Iϕ = 3333 × 10^4 Hz

(2) F[15:0] = 3333 (H'0D05)

(3) Set R4 (FPEFEQ) to H'00000D05.

(2.2) Flash User Branch Address Setting Parameter (FUBRA: General Register R5

This parameter sets the user branch destination address. The user program which has be
be executed in specified processing units when programming and erasing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24	UA23	UA22	UA21	UA20	UA19	UA18
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8	UA7	UA6	UA5	UA4	UA3	UA2
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

program download area and stack area must n
overwritten. If CPU runaway occurs or the dow
area or stack area is overwritten, the value of f
memory cannot be guaranteed.

The download of the on-chip program, initializa
initiation of the programming/erasing program
be executed in the processing of the user bran
destination. Programming or erasing cannot be
guaranteed when returning from the user branc
destination. The program data which has alrea
prepared must not be programmed.

Store general registers R8 to R15. General reg
to R7 are available without storing them.

Moreover, the programming/erasing interface r
must not be written to in the processing of the
branch destination.

After the processing of the user branch has en
programming/erasing program must be returne
using the RTS instruction.

For the execution intervals of the user branch
processing, see note 2 (User branch processin
intervals) in section 21.7.3, Other Notes.

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	Undefined	R/W	Unused Return 0.
2	BR	Undefined	R/W	User Branch Error Detect Returns the check result whether the specifie branch destination address is in the area othe storage area of the programming/erasing prog which has been downloaded. 0: User branch address setting is normal 1: User branch address setting is abnormal
1	FQ	Undefined	R/W	Frequency Error Detect Returns the check result whether the specifie operating frequency of the CPU is in the rang supported operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	Undefined	R/W	Success/Fail Indicates whether initialization is completed n 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error o

data must be in the consecutive space which can be accessed by using the MOV.B ins
of the CPU and is not the flash memory space.
When data to be programmed does not satisfy 256 bytes, the 256-byte program data n
prepared by embedding the dummy code (H'FF).
The start address of the area in which the prepared program data is stored must be set
general register R4. This parameter is called FMPDR (flash multipurpose data destina
parameter).

For details on the programming procedure, see section 21.5.2, User Program Mode.

(3.1) Flash Multipurpose Address Area Parameter (FMPAR: General Register R5 o

This parameter indicates the start address of the programming destination on the user MA

When an address in an area other than the flash memory space is set, an error occurs.

The start address of the programming destination must be at the 256-byte boundary. If thi
boundary condition is not satisfied, an error occurs. The error occurrence is indicated by t
bit (bit 1) in FPFR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	M
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	M
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

of CPU)

This parameter indicates the start address in the area which stores the data to be program the user MAT. When the storage destination of the program data is in flash memory, an occurs. The error occurrence is indicated by the WD bit (bit 2) in FPFR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	Undefined	R/W	MOD31 to MOD0 Store the start address of the area which stor program data for the user MAT. The consecu byte data is programmed to the user MAT sta the specified start address.

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	Undefined	R/W	Unused Return 0.
6	MD	Undefined	R/W	Programming Mode Related Setting Error Dete Returns the check result of whether the signal the FWE pin is high and whether the error prot state is not entered. When a low-level signal is input to the FWE pir error protection state is entered, 1 is written to The input level to the FWE pin and the error pr state can be confirmed with the FWE bit (bit 7) FLER bit (bit 4) in FCCS, respectively. For con enter the error protection state, see section 21 Error Protection. 0: FWE and FLER settings are normal (FWE = = 0) 1: FWE = 0 or FLER = 1, and programming ca performed

				If FMATS is set to H'AA and the user boot MA selected, an error occurs when programming performed. In this case, both the user MAT ar boot MAT are not rewritten. Programming of the user boot MAT must be e in boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally (progr result is not guaranteed)
4	FK	Undefined	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY the start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other
3	—	Undefined	R/W	Unused Return 0.
2	WD	Undefined	R/W	Write Data Address Error Detect When an address in the flash memory area is as the start address of the storage destination program data, an error occurs. 0: Setting of write data address is normal 1: Setting of write data address is abnormal

				1: Setting of programming destination address abnormal
0	SF	Undefined	R/W	Success/Fail Indicates whether the program processing has normally or not. 0: Programming has ended normally (no error) 1: Programming has ended abnormally (error o

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	-	-	-	-	-	-	-	-	EBS[7:0]					
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R/W	Unused Return 0.
7 to 0	EBS[7:0]	Undefined	R/W	Set the erase-block number in the range from corresponds to the EB0 block and 11 corresp the EB11 block. An error occurs when a numb than 0 to 11 (H'00 to H'0B) is set.

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	Undefined	R/W	Unused Return 0.
6	MD	Undefined	R/W	Erasure Mode Related Setting Error Detect Returns the check result of whether the signal the FWE pin is high and whether the error prot state is not entered. When a low-level signal is input to the FWE pir error protection state is entered, 1 is written to The input level to the FWE pin and the error pr state can be confirmed with the FWE bit (bit 7) FLER bit (bit 4) in FCCS, respectively. For con enter the error protection state, see section 21 Error Protection. 0: FWE and FLER settings are normal (FWE = = 0) 1: FWE = 0 or FLER = 1, and erasure cannot b performed

				selected, an error occurs when erasure is per this case, both the user MAT and user boot M not erased. Erasure of the user boot MAT must be execut mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure res guaranteed)
4	FK	Undefined	R/W	Flash Key Register Error Detect Returns the check result of FKEY value befor the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other
3	EB	Undefined	R/W	Erase Block Select Error Detect Returns the check result whether the specifie block number is in the block range of the user 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal
2, 1	—	Undefined	R/W	Unused Return 0.
0	SF	Undefined	R/W	Success/Fail Indicates whether the erasing processing has normally or not. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occu

Boot mode executes programming/erasing user MAT and user boot MAT by means of th
command and program data transmitted from the host using the on-chip SCI. The tool for
transmitting the control command and program data must be prepared in the host. The SC
communication mode is set to asynchronous mode. When reset start is executed after this
is set in boot mode, the boot program in the microcontroller is initiated. After the SCI bit
automatically adjusted, the communication with the host is executed by means of the con
command method.

The system configuration diagram in boot mode is shown in figure 21.6. For details on th
setting in boot mode, see table 21.1. Interrupts are ignored in boot mode, so do not genera
Note that the AUD cannot be used during boot mode operation.

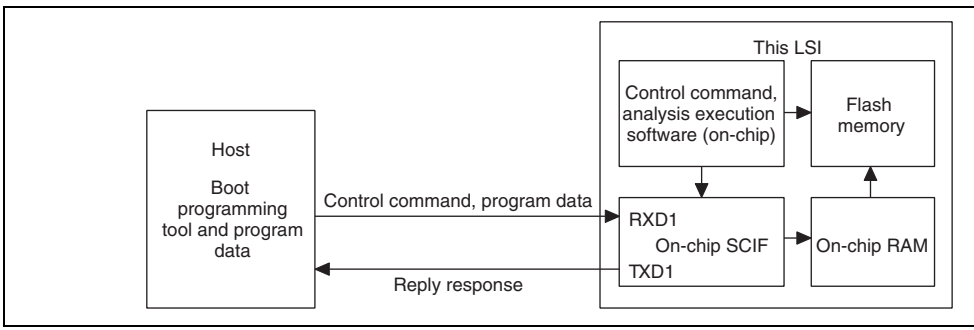


Figure 21.6 System Configuration in Boot Mode

the transfer bit rate of the host must be set to 9,600 bps or 19,200 bps.

The system clock frequency which can automatically adjust the transfer bit rate of the ho
bit rate of this LSI is shown in table 21.7. Boot mode must be initiated in the range of th
clock.

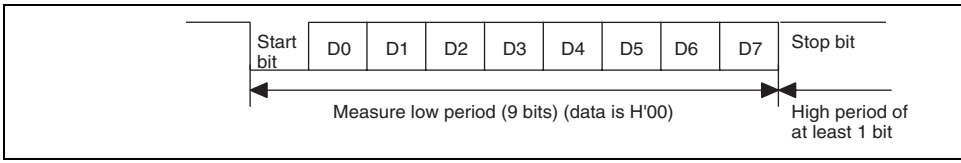


Figure 21.7 Automatic Adjustment Operation of SCIF Bit Rate

Table 21.7 Peripheral Clock (Pϕ) Frequency that Can Automatically Adjust Bit R This LSI

Host Bit Rate	Peripheral Clock (Pϕ) Frequency That Can Automatically Adjust Rate
9,600 bps	32 to 40 MHz
19,200 bps	32 to 40 MHz

and configuration of the user MAT, start addresses of the MATs, information on supp
devices, etc.

3. Automatic erasure of the entire user MAT and user boot MAT

   After all necessary inquiries and selections have been made and the command for tran
   the programming/erasure state is sent by the host, the entire user MAT and user boot
   automatically erased.

4. Waiting for programming/erasure command

   — On receiving the programming selection command, the chip waits for data to be
   programmed. To program data, the host transmits the programming command cod
   followed by the address where programming should start and the data to be progra
   This is repeated as required while the chip is in the programming-selected state. T
   terminate programming, H'FFFFFFFF should be transmitted as the first address of
   for programming. This makes the chip return to the programming/erasure comman
   waiting state from the programming data waiting state.

   — On receiving the erasure select command, the chip waits for the block number of a
   be erased. To erase a block, the host transmits the erasure command code followed
   number of the block to be erased. This is repeated as required while the chip is in
   erasure-selected state. To terminate erasure, H'FF should be transmitted as the blo
   number. This makes the chip return to the programming/erasure command waiting
   from the erasure block number waiting state. Erasure should only be executed whe
   specific block is to be reprogrammed without executing a reset-start of the chip af
   flash memory has been programmed in boot mode. If all desired programming is d
   single operation, such erasure processing is not necessary because all blocks are er
   before the chip enters the programming/erasure/other command waiting state.

   — In addition to the programming and erasure commands, commands for sum check
   blank checking (checking for erasure) of the user MAT and user boot MAT, readi
   from the user MAT/user boot MAT, and acquiring current state information are pr

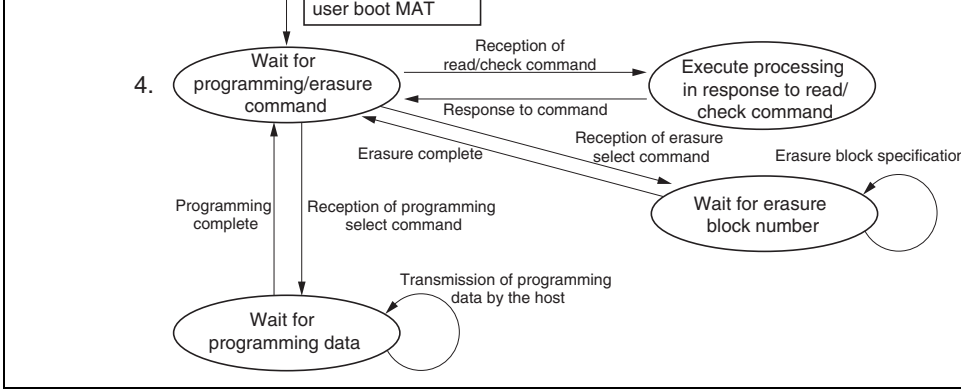


Figure 21.8 State Transitions in Boot Mode

memory. If reset is executed accidentally, the reset signal must be released after the reset
period, which is longer than the normal 100 µs.

For details on the programming procedure, see the description in section 21.5.2 (2), Progr
Procedure in User Program Mode. For details on the erasing procedure, see the descriptio
section 21.5.2 (3), Erasing Procedure in User Program Mode.

For the overview of a processing that repeats erasing and programming by downloading t
programming program and the erasing program in separate on-chip ROM areas using FTI
the description in section 21.5.2 (4), Erasing and Programming Procedure in User Progra

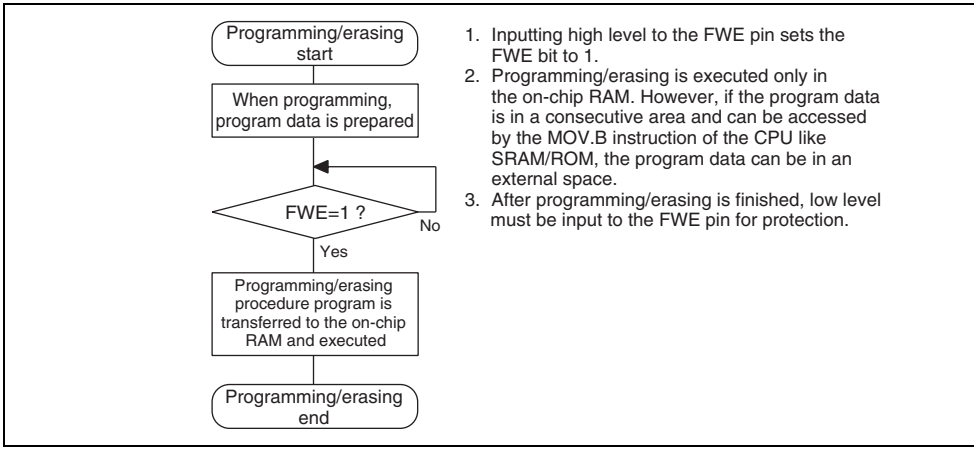


Figure 21.9 Programming/Erasing Overview Flow

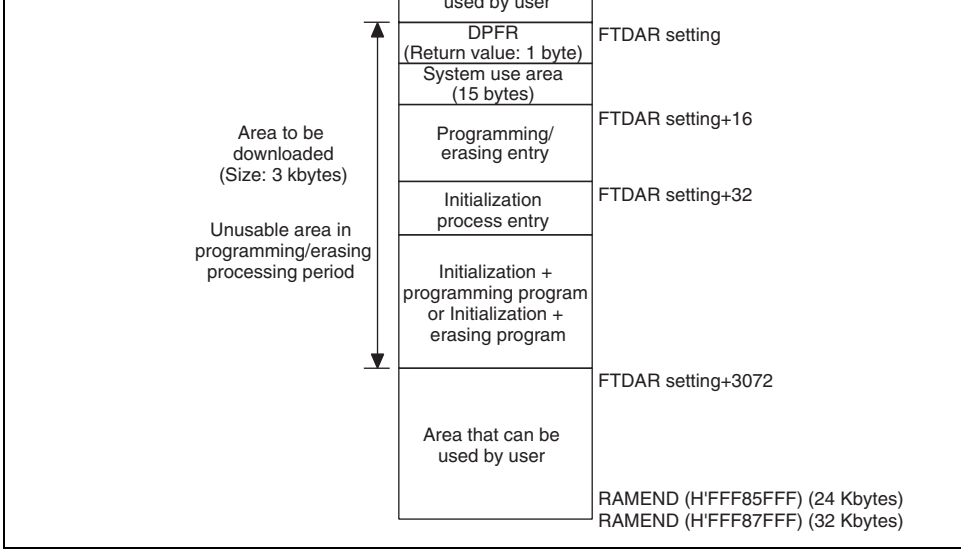


Figure 21.10 RAM Map after Download

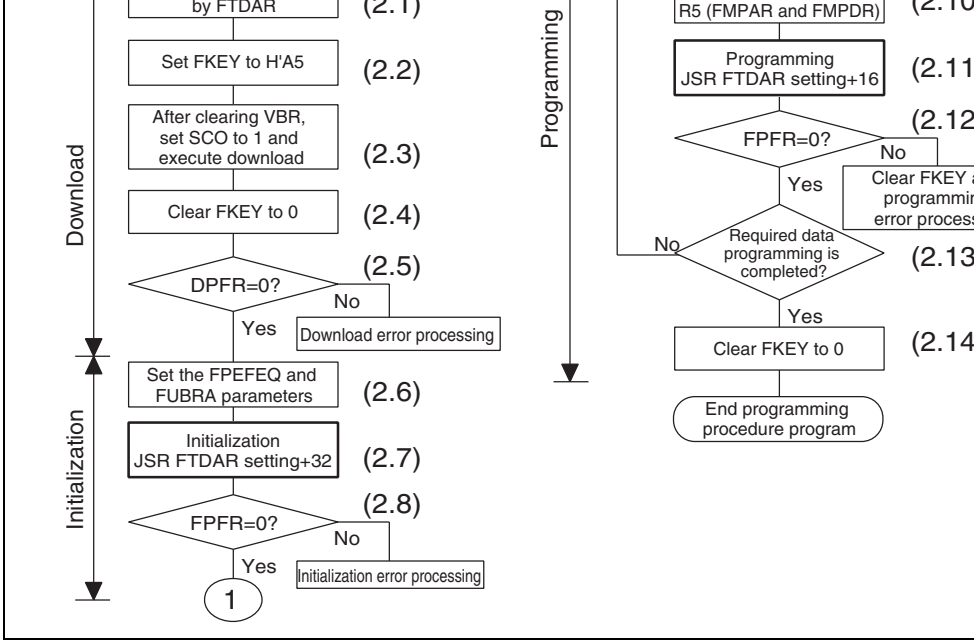


Figure 21.11 Programming Procedure

The details of the programming procedure are described below. The procedure program m executed in an area other than the flash memory to be programmed. Especially the part w SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM. Spec as the frequency division ratios of an internal clock (Iϕ), a bus clock (Bϕ), and a peripher (Pϕ) through the frequency control register (FRQCR).

The area that can be executed in the steps of the user procedure program (on-chip RAM, MAT, and external space) is shown in section 21.8.2, Areas for Storage of the Procedural and Data for Programming.

(2.1) Select the on-chip program to be downloaded

When the PPVS bit of FPCS is set to 1, the programming program is selected.
Several programming/erasing programs cannot be selected at one time. If several pro
set, download is not performed and a download error is returned to the source select
detect (SS) bit in the DPFR parameter.
Specify the start address of the download destination by FTDAR.

(2.2) Write H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be written to the SCO bit for
download request.

(2.3) VBR is set to 0 and 1 is written to the SCO bit of FCCS, and then download is exe

VBR must always be set to H'80000000 before setting the SCO bit to 1.
To write 1 to the SCO bit, the following conditions must be satisfied.

- H'A5 is written to FKEY.
- The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When execution ret
user procedure program, the SCO bit is cleared to 0. Therefore, the SCO bit cannot b
confirmed to be 1 in the user procedure program.
The download result can be confirmed only by the return value of the DPFR parame
the SCO bit is set to 1, incorrect decision must be prevented by setting the DPFR par
that is one byte of the start address of the on-chip RAM area specified by FTDAR, to
other than the return value (H'FF).
When download is executed, particular interrupt processing, which is accompanied b
switch as described below, is performed as an internal microcontroller processing, so
need to be set to H'80000000. Thirty-two NOP instructions are executed immediatel
instructions that set the SCO bit to 1.

be changed.

The notes on download are as follows.

In the download processing, the values of the general registers of the CPU are retaine

During the download processing, interrupts must not be generated. For details on the relationship between download and interrupts, see section 21.7.2, Interrupts during Programming/Erasing.

Since a stack area of maximum 256 bytes is used, an area of at least 128 bytes must b before setting the SCO bit to 1.

If flash memory is accessed by the DMAC during downloading, operation cannot be guaranteed. Therefore, access by the DMAC must not be executed.

(2.4) FKEY is cleared to H'00 for protection.

(2.5) The value of the DPFR parameter must be checked to confirm the download result.

A recommended procedure for confirming the download result is shown below.

1. Check the value of the DPFR parameter (one byte of start address of the downloac destination specified by FTDAR). If the value is H'00, download has been perforn normally. If the value is not H'00, the source that caused download to fail can be investigated by the description below.
2. If the value of the DPFR parameter is the same as before downloading (e.g. H'FF) address setting of the download destination in FTDAR may be abnormal. In this c confirm the setting of the TDER bit (bit 7) in FTDAR.
3. If the value of the DPFR parameter is different from before downloading, check th (bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the download p selection and FKEY register setting were normal, respectively.

When the user branch processing is not required, 0 must be set to FUBRA.

When the user branch is executed, the branch destination is executed in flash me than the one that is to be programmed. The area of the on-chip program that is do cannot be set.

The program processing must be returned from the user branch processing by the instruction.

See the description in section 21.4.3 (2.2), Flash User Branch Address Setting Pa (FUBRA: General Register R5 of CPU).

(2.7) Initialization

When a programming program is downloaded, the initialization program is also dow on-chip RAM. There is an entry point of the initialization program in the area from (start address set by FTDAR) + 32 bytes. The subroutine is called and initialization is by using the following steps.

```
MOV.L  #DLTOP+32,R1          ; Set entry address to R1
JSR    @R1                   ; Call initialization routine
NOP
```

1. The general registers other than R0 are saved in the initialization program.
2. R0 is a return value of the FPFR parameter.
3. Since the stack area is used in the initialization program, a stack area of 256 byte must be reserved in RAM.
4. Interrupts can be accepted during the execution of the initialization program. Ho program storage area and stack area in on-chip RAM and register values must no destroyed.

programming is not executed and an error is returned to the return value paramete
Since the unit is 256 bytes, the lower eight bits (MOA7 to MOA0) must be in the
boundary of H'00.

2. FMPDR setting

   If the storage destination of the program data is flash memory, even when the prog
   execution routine is executed, programming is not executed and an error is returne
   FPFR parameter. In this case, the program data must be transferred to on-chip RA
   then programming must be executed.

(2.11) Programming

There is an entry point of the programming program in the area from (download start
set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and programmi
executed by using the following steps.

```
MOV.L  #DLTOP+16,R1          ; Set entry address to R1
JSR    @R1                   ; Call programming routine
NOP
```

1. The general registers other than R0 are saved in the programming program.
2. R0 is a return value of the FPFR parameter.
3. Since the stack area is used in the programming program, a stack area of maximu
   bytes must be reserved in RAM.

finished, secure a reset period (period of RES = 0) that is at least as long as the norm

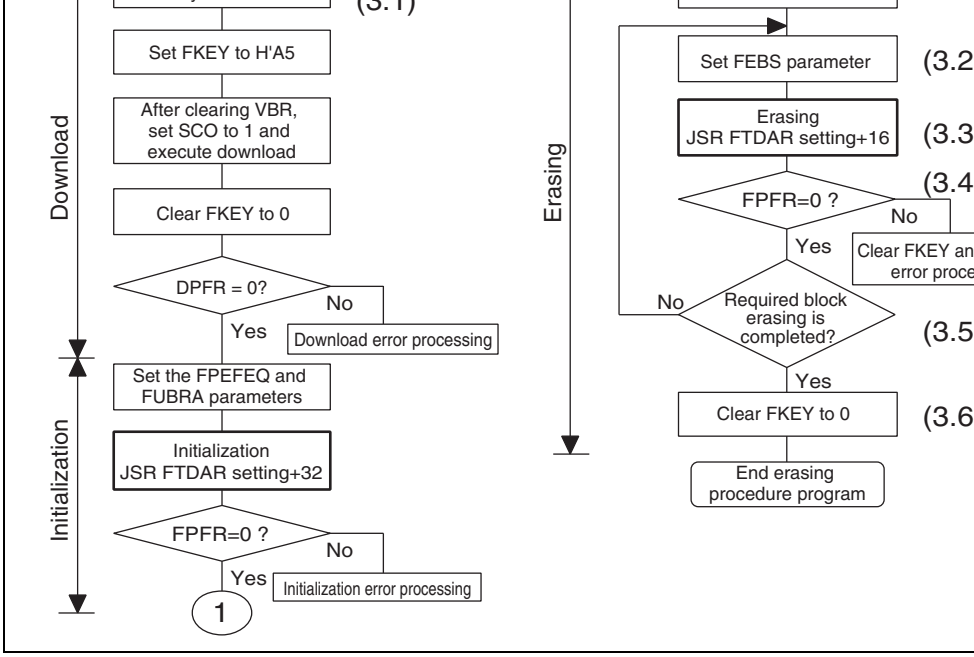


Figure 21.12 Erasing Procedure

The details of the erasing procedure are described below. The procedure program must be executed in an area other than the user MAT to be erased.

Especially the part where the SCO bit in FCCS is set to 1 for downloading must be execu chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, MAT, and external space) is shown in section 21.8.2, Areas for Storage of the Procedural and Data for Programming.

Set the LPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several pro
set, download is not performed and a download error is returned to the source select
detect (SS) bit in the DPFR parameter.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization
same as those in the programming procedure. For details, see the description in secti
(2), Programming Procedure in User Program Mode.

(3.2) Set the FEBS parameter necessary for erasure

Set the erase block number of the user MAT in the flash erase block select paramete
general register R4). If a value other than an erase block number of the user MAT is
block is erased even though the erasing program is executed, and an error is returned
return value parameter FPFR.

(3.3) Erasure

Similar to as in programming, there is an entry point of the erasing program in the ar
(download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine
and erasing is executed by using the following steps.

```
MOV.L  #DLTOP+16,R1          ; Set entry address to R1
JSR    @R1                   ; Call erasing routine
NOP
```

1. The general registers other than R0 are saved in the erasing program.
2. R0 is a return value of the FPFR parameter.
3. Since the stack area is used in the erasing program, a stack area of maximum 128
must be reserved in RAM.

By changing the on-chip RAM address of the download destination in FTDAR, the erasi
program and programming program can be downloaded to separate on-chip RAM areas.

Figure 21.13 shows an example of repetitively executing RAM emulation, erasing, and programming.

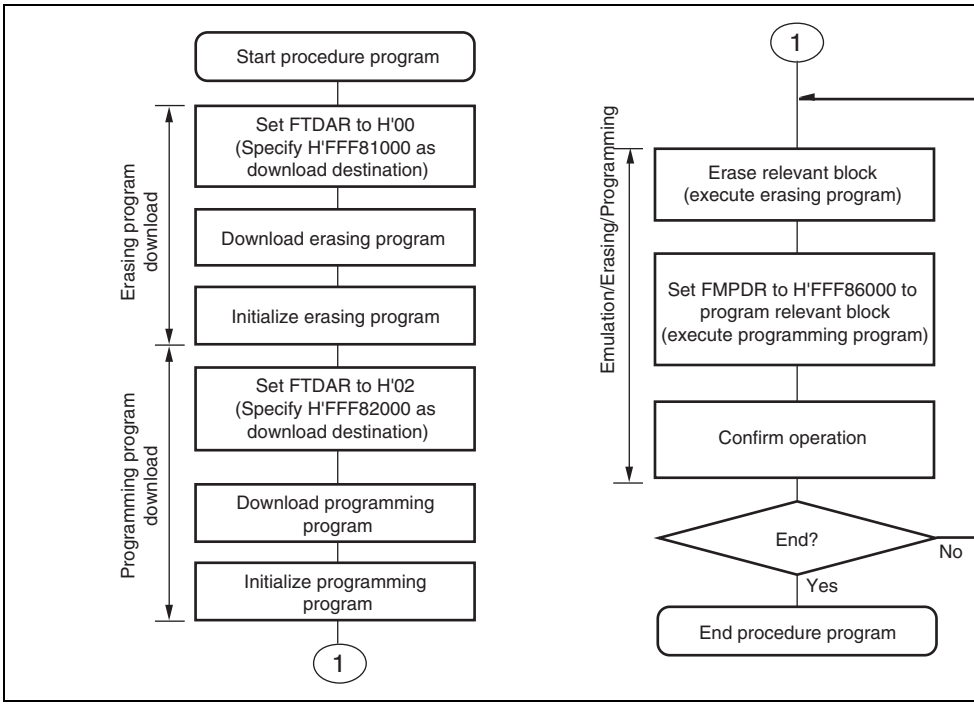


Figure 21.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Progr (Overview)

(H'FFF81020 in this example) and (download start address for programming pro
bytes (H'FFF82020 in this example).

21.5.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than tho
program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot
uses the on-chip SCIF.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing
user boot MAT is only enabled in boot mode or programmer mode.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 21.1.

When the reset start is executed in user boot mode, the check routine for flash-memory r
registers runs on the on-chip RAM. NMI and all other interrupts cannot be accepted. Ne
the AUD be used in this period. This period is 100 µs while operating at an internal freq
40 MHz.

Next, processing starts from the execution start address of the reset vector in the user bo
At this point, H'AA is set to the flash MAT select register (FMATS) because the executi
is the user boot MAT.

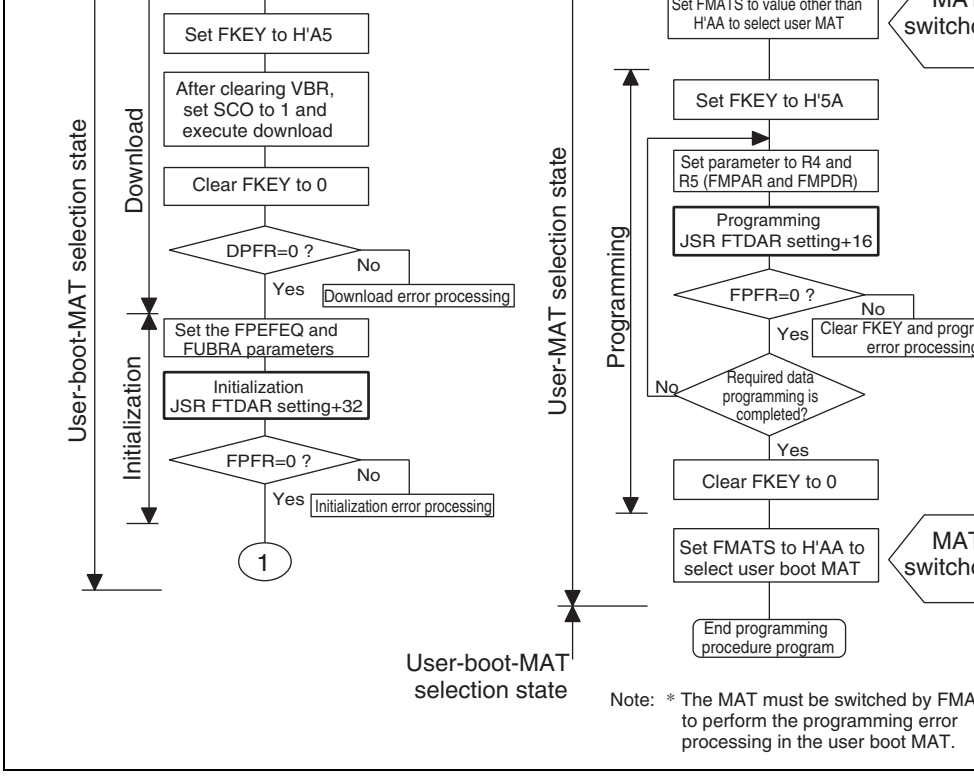


Figure 21.14 Procedure for Programming User MAT in User Boot Mode

MAT the interrupt vector is read from is undetermined. Perform MAT switching in
with the description in section 21.7.1, Switching between User MAT and User Boot

Except for MAT switching, the programming procedure is the same as that in user p
mode.

The area that can be executed in the steps of the user procedure program (on-chip R
MAT, and external space) is shown in section 21.8.2, Areas for Storage of the Proce
Program and Data for Programming.

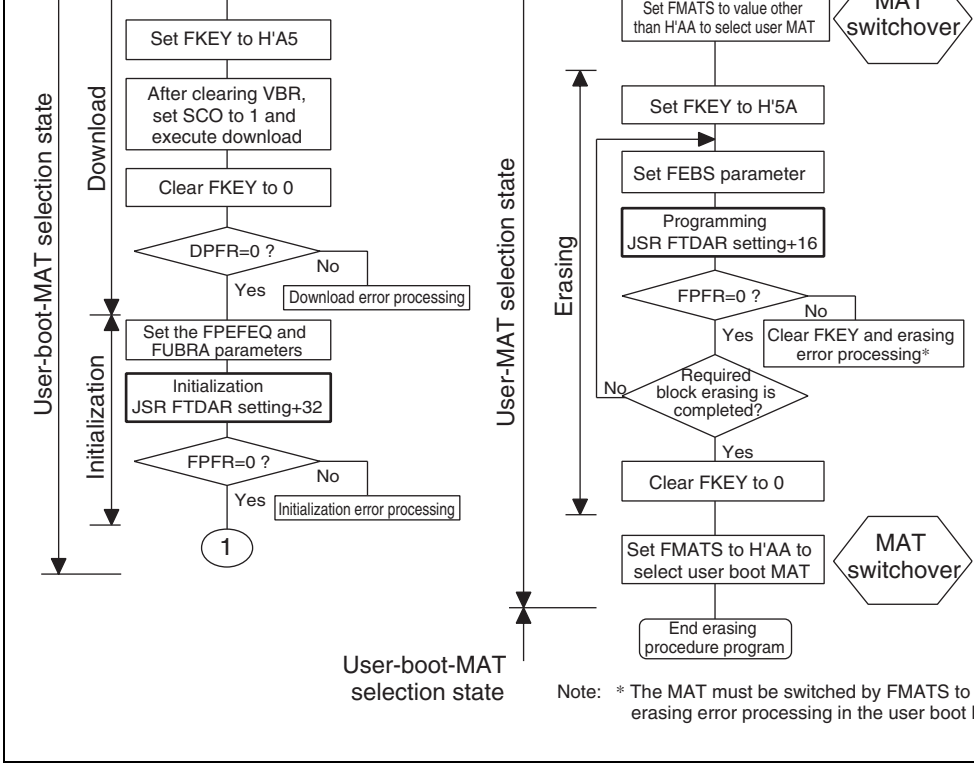


Figure 21.15 Procedure for Erasing User MAT in User Boot Mode

The area that can be executed in the steps of the user procedure program (on-chip RAM, MAT, and external space) is shown in section 21.8.2, Areas for Storage of the Procedure and Data for Programming.

or erase locations in a user MAT, and the error in programming/erasing is reported in the parameter.

Table 21.8 Hardware Protection

		Function to be Pro	
Item	**Description**	**Download**	**Progr Erasu**
FWE-pin protection	The input of a low-level signal on the FWE pin clears the FWE bit of FCCS and the LSI enters a programming/erasing-protected state.	—	√
Reset/standby protection	• A power-on reset (including a power-on reset by the WDT) and entry to standby mode initializes the programming/erasing interface registers and the LSI enters a programming/erasing-protected state. • Resetting by means of the $\overline{\text{RES}}$ pin after power is initially supplied will not make the LSI enter the reset state unless the $\overline{\text{RES}}$ pin is held low until oscillation has stabilized. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width that is specified in the section on AC characteristics. If the LSI is reset during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	√	√

SCO bit	downloading of the programming/erasing program, thus making the LSI enter a programming/erasing-protected state.		
Protection by FKEY	Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and for programming/erasing.	√	√

21.6.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occ
form of the microcontroller getting out of control during programming/erasing of the fla
memory or operations that are not in accordance with the established procedures for
programming/erasing. Aborting programming or erasure in such cases prevents damage
flash memory due to excessive programming or erasing.

If the microcontroller malfunctions during programming/erasing of the flash memory, th
bit in FCCS is set to 1 and the LSI enters the error protection state, thus aborting progra
erasure.

memory, some voltage may still remain even after the error protection state has been elite
this reason, it is necessary to reduce the risk of damage to the flash memory by extending
period so that the charge is released.

The state-transition diagram in figure 21.16 shows transitions to and from the error protec
state.

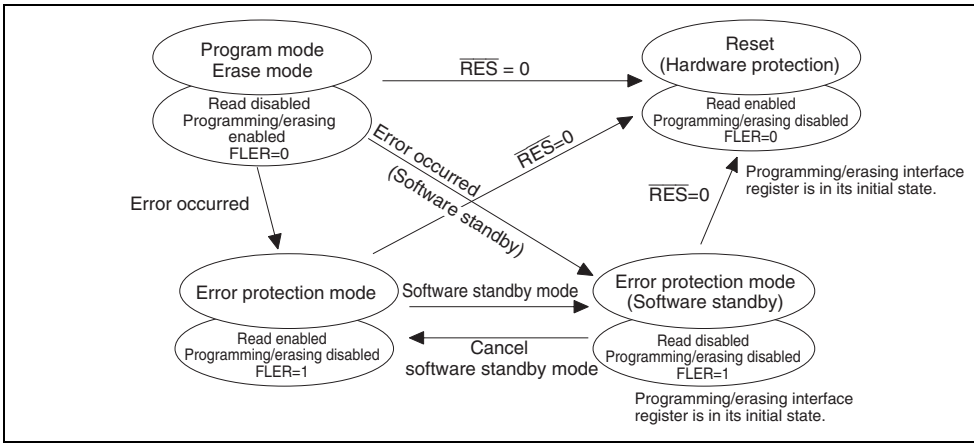


Figure 21.16 Transitions to and from Error Protection State

1. MAT switching by FMATS should always be executed from the on-chip RAM. The microcontroller prefetches execution instructions. Therefore, a switchover during pro execution in the user MAT causes an instruction code in the user MAT to be prefetch instruction in the newly selected user boot MAT to be prefetched, thus resulting in u operation.
2. To ensure that the MAT that has been switched to is accessible, execute thirty-two N instructions in on-chip RAM immediately after writing to FMATS of on-chip RAM prevents access to the flash memory during MAT switching).
3. If an interrupt occurs during switching, there is no guarantee of which memory MAT accessed.
   Always mask the maskable interrupts before switching MATs. In addition, configuri system so that NMI interrupts do not occur during MAT switching is recommended.
4. After the MATs have been switched, take care because the interrupt vector table wil been switched.
   If the same interrupt processings are to be executed before and after MAT switching interrupt requests cannot be disabled, transfer the interrupt processing routine to on-c and use the VBR setting to place the interrupt vector table in on chip RAM. In this c sure the VBR setting change does not conflict with the interrupt occurrence.
5. Memory sizes of the user MAT and user boot MAT are different. When accessing th boot MAT, do not access addresses exceeding the 12-kbyte memory space. If access beyond the 12-kbyte space, the values read are undefined.

(2) Write a value other than H'AA to FMATS.
(3) Execute thirty-two NOP instructions before accessing the user MAT.

Figure 21.17 Switching between User MAT and User Boot MAT

21.7.2 Interrupts during Programming/Erasing

(1) Download of On-Chip Program

(a) VBR Setting Change

Before downloading the on-chip program, VBR must be set to H'80000000. If VBR is set
value other than H'80000000, the interrupt vector table is placed in the user MAT (FMAT
H'AA) or the user boot MAT (FMATS is H'AA) on setting H'80000000 to VBR.

When VBR setting change conflicts with interrupt occurrence, whether the vector table b
after VBR is changed is referenced may cause an error.

Therefore, for cases where VBR setting change may conflict with interrupt occurrence, p
vector table to be referenced when VBR is H'00000000 (initial value) at the start of the u
or user boot MAT.

(b) SCO Download Request and Interrupt Request

Download of the on-chip programming/erasing program that is initiated by setting the SC
FCCS to 1 generates a particular interrupt processing accompanied by MAT switchover.
Operation when the SCO download request and interrupt request conflicts is described be

1. Contention between SCO download request and interrupt request
   Figure 21.18 shows the timing of contention between execution of the instruction that
   SCO bit in FCCS to 1 and interrupt acceptance.

2. Generation of interrupt requests during downloading
   Ensure that interrupts are not generated during downloading that is initiated by the S

frequency is 40 MHz, the download for each program takes approximately 10 ms at maxi

(2) User Branch Processing Intervals

The intervals for executing the user branch processing differs in programming and erasin processing phase also differs. Table 21.10 lists the maximum intervals for initiating the u branch processing when the CPU clock frequency is 40 MHz.

Table 21.10 Initiation Intervals of User Branch Processing

Processing Name	Maximum Interval
Programming	Approximately 2 ms*
Erasing	Approximately 15 ms*

Note: * Reference value

However, when operation is done with CPU clock of 40 MHz, maximum values of the ti first user branch processing are as shown in table 21.11.

Table 21.11 Initial User Branch Processing Time

Processing Name	Maximum
Programming	Approximately 2 ms*
Erasing	Approximately 15 ms*

Note: * Reference value

- Boot mode
- Programmer mode

(5) Compatibility with Programming/Erasing Program of Conventional F-ZTAT Microcontroller

A programming/erasing program for flash memory used in the conventional F-ZTAT SH microcontroller which does not support download of the on-chip program by a SCO tran request cannot run in this LSI.
Be sure to download the on-chip program to execute programming/erasing of flash mem LSI.

(6) Monitoring Runaway by WDT

Unlike the conventional F-ZTAT SH microcontroller, no countermeasures are available runaway by WDT during programming/erasing by the downloaded on-chip program.
Prepare countermeasures (e.g. use of the user branch routine and periodic timer interrup WDT while taking the programming/erasing time into consideration as required.

H'1111*

H'1333*

Note: * The CKOEN bit (bit 12) can be specified as either 0 or1.

(8) Programming the User MAT in User Program Mode

This LSI does not allow transitions from single chip mode to user program mode. Therefo
order to program the user MAT in user program mode, be sure to activate the LSI in MCU
extension mode 2 rather than in single chip mode.

The boot program has three states.

1. Bit-rate matching state

   In this state, the boot program adjusts the bit rate to match that of the host. When the starts up in boot mode, the boot program is activated and enters the bit-rate matching which it receives commands from the host and adjusts the bit rate accordingly. After matching is complete, the boot program proceeds to the inquiry-and-selection state.

2. Inquiry-and-selection state

   In this state, the boot program responds to inquiry commands from the host. The dev mode, and bit rate are selected in this state. After making these selections, the boot p enters the programming/erasure state in response to the transition-to-programming/e state command. The boot program transfers the erasure program to RAM and execut of the user MAT and user boot MAT before it enters the programming/erasure state.

3. Programming/erasure state

   In this state, programming/erasure are executed. The boot program transfers the prog programming/erasure to RAM in line with the command received from the host and programming/erasure. It also performs sum checking and blank checking as directed respective commands.

Figure 21.19 shows the flow of processing by the boot program.

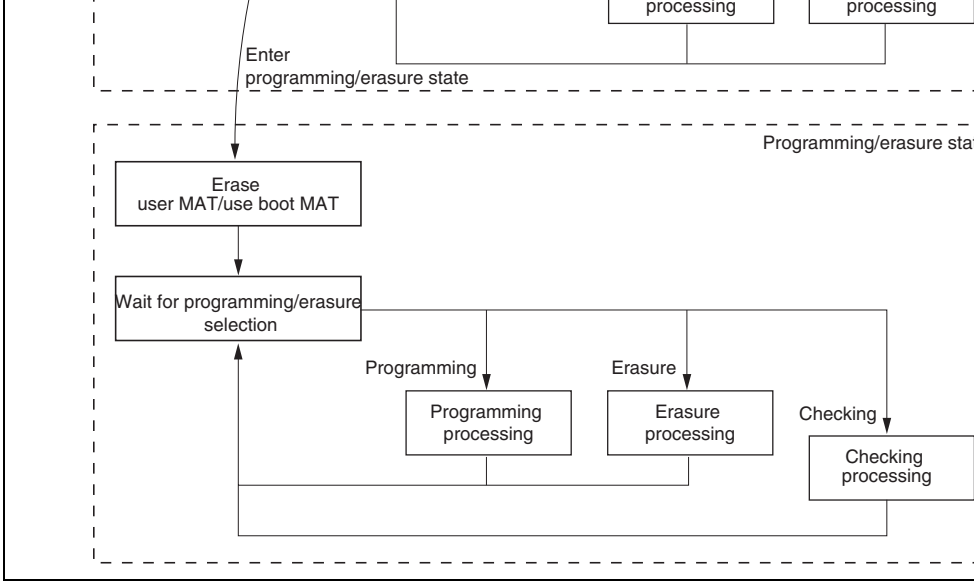


Figure 21.19 Flow of Processing by the Boot Program

(2) Bit-Rate Matching State

In bit-rate matching, the boot program measures the low-level intervals in a signal carryi
data that is transmitted by the host, and calculates the bit rate from this. The bit rate can b
changed by the new-bit-rate selection command. On completion of bit-rate matching, the
program goes to the inquiry and selection state. The sequence of processing in bit-rate ma
shown in figure 21.20.

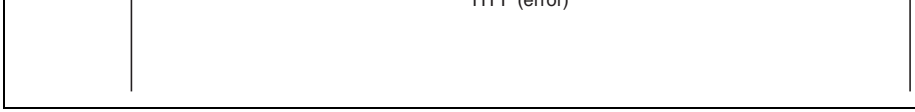


Figure 21.20 Sequence of Bit-Rate Matching

(3) Communications Protocol

Formats in the communications protocol between the host and boot program after compl
the bit-rate matching are as follows.

1. One-character command or one-character response

   A command or response consisting of a single character used for an inquiry or the A
   indicating normal completion.
2. n-character command or n-character response

   A command or response that requires n bytes of data, which is used as a selection co
   response to an inquiry. The length of programming data is treated separately below.
3. Error response

   Response to a command in case of an error: two bytes, consisting of the error respon
   error code.
4. 256-byte programming command

   The command itself does not include data-size information. The data length is know
   response to the command for inquiring about the programming size.
5. Response to a memory reading command

   This response includes four bytes of size information.

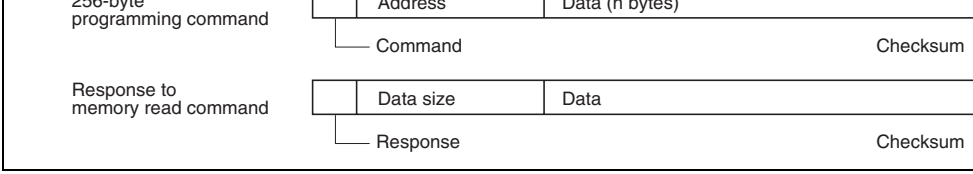


Figure 21.21 Formats in the Communications Protocol

- Command (1 byte): Inquiry, selection, programming, erasure, checking, etc.
- Response (1 byte): Response to an inquiry
- Size (one or two bytes): The length of data for transfer, excluding the command/respo and checksum.
- Data (n bytes): Particular data for the command or response
- Checksum (1 byte): Set so that the total sum of byte values from the command code t checksum is H'00.
- Error response (1 byte): Error response to a command
- Error code (1 byte): Indicates the type of error.
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed. "n" is known from the response to the comma to inquire about the programming size.
- Data size (4 bytes): Four-byte field included in the response to a memory reading com

	devices	program names.
H'10	Device selection	Selects a device code.
H'21	Inquiry on clock modes	Requests the number of available clock modes a respective values.
H'11	Clock-mode selection	Selects a clock mode.
H'22	Inquiry on frequency multipliers	Requests the number of clock signals for which fr multipliers and divisors are selectable, the numbe multiplier and divisor settings for the respective cl the values of the multipliers and divisors.
H'23	Inquiry on operating frequency	Requests the minimum and maximum values for frequency of the main clock and peripheral clock.
H'24	Inquiry on user boot MATs	Requests the number of user boot MAT areas alo their start and end addresses.
H'25	Inquiry on user MATs	Requests the number of user MAT areas along w start and end addresses.
H'26	Inquiry on erasure blocks	Requests the number of erasure blocks along wit start and end addresses.
H'27	Inquiry on programming size	Requests the unit of data for programming.
H'3F	New bit rate selection	Selects a new bit rate.
H'40	Transition to programming/erasure state	On receiving this command, the boot program era user MAT and user boot MAT and enters the programming/erasure state.
H'4F	Inquiry on boot program state	Requests information on the current state of boot processing.

(a) Inquiry on Supported Devices

In response to the inquiry on supported devices, the boot program returns the device code devices it supports and the product names of their respective boot programs.

Command	H'20

- Command H'20 (1 byte): Inquiry on supported devices

Response	H'30	Size	No. of devices	
	Number of characters	Device code		Product name
	…			
	SUM			

- Response H'30 (1 byte): Response to the inquiry on supported devices
- Size (1 byte): The length of data for transfer excluding the command code, this field (the checksum. Here, it is the total number of bytes taken up by the number of devices, of characters, device code, and product name fields.
- Number of devices (1 byte): The number of device models supported by the boot prog embedded in the microcontroller.
- Number of characters (1 byte): The number of characters in the device code and produ fields.
- Device code (4 bytes): Device code of a supported device (ASCII encoded)
- Product name (n bytes): Product code of the boot program (ASCII encoded)
- SUM (1 byte): Checksum
  This is set so that the total sum of all bytes from the command code to the checksum i

devices (ASCII encoded)

- SUM (1 byte): Checksum

Response	H'06	

- Response H'06 (1 byte): Response to device selection
  This is the ACK code and is returned when the specified device code matches one of
  supported devices.

Error response	H'90	ERROR

- Error response H'90 (1 byte): Error response to device selection
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'21: Non-matching device code

(c) Inquiry on Clock Modes

In response to the inquiry on clock modes, the boot program returns the number of avail
modes.

Command	H'21

- Command H'21 (1 byte): Inquiry on clock modes

Response	H'31	Size	Mode	…	SUM

Command	H'11	Size	Mode	SUM

- Command H'11 (1 byte): Clock mode selection
- Size (1 byte): Number of characters in the clock-mode field (fixed at 1)
- Mode (1 byte): A clock mode returned in response to the inquiry on clock modes
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): Response to clock mode selection
  This is the ACK code and is returned when the specified clock-mode matches one of t
  available clock modes.

Error response	H'91	ERROR

- Error response H'91 (1 byte): Error response to clock mode selection
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'21: Non-matching clock mode

No. of multipliers	Multiplier	…				
…						
SUM						

- Response H'32 (1 byte): Response to the inquiry on frequency multipliers
- Size (1 byte): The total length of the number of operating clocks, number of multipli multiplier fields.
- Number of operating clocks (1 byte): The number of operating clocks for which mul can be selected
  (for example, if frequency multiplier settings can be made for the frequencies of the peripheral operating clocks, the value should be H'02).
- Number of multipliers (1 byte): The number of multipliers selectable for the operatir frequency of the main or peripheral modules
- Multiplier (1 byte):
  Multiplier: Numerical value in the case of frequency multiplication (e.g. H'04 for ×4
  Divisor: Two's complement negative numerical value in the case of frequency divisi H'FE [-2] for ×1/2)
  As many multiplier fields are included as there are multipliers or divisors, and comb the number of multipliers and multiplier fields are repeated as many times as there a operating clocks.
- SUM (1 byte): Checksum

…	
SUM	

- Response H'33 (1 byte): Response to the inquiry on operating frequency
- Size (1 byte): The total length of the number of operating clocks, and maximum and m values of operating frequency fields.
- Number of operating clocks (1 byte): The number of operating clock frequencies requ within the device.
  For example, the value two indicates main and peripheral operating clock frequencies
- Minimum value of operating frequency (2 bytes): The minimum frequency of a frequ multiplied or -divided clock signal.
  The value in this field and in the maximum value field is the frequency in MHz to two places, multiplied by 100 (for example, if the frequency is 20.00 MHz, the value mult 100 is 2000, so H'07D0 is returned here).
- Maximum value of operating frequency (2 bytes): The maximum frequency of a frequ multiplied or -divided clock signal.
  As many pairs of minimum/maximum values are included as there are operating clock
- SUM (1 byte): Checksum

…				
SUM				

- Response H'34 (1 byte): Response to the inquiry on user boot MATs
- Size (1 byte): The total length of the number of areas and first and last address fields
- Number of areas (1 byte): The number of user boot MAT areas.
  H'01 is returned if the entire user boot MAT area is continuous.
- First address of the area (4 bytes)
- Last address of the area (4 bytes)
  As many pairs of first and last address field are included as there are areas.
- SUM (1 byte): Checksum

(h) Inquiry on User MATs

In response to the inquiry on user MATs, the boot program returns the number of user M
and their addresses.

Command	H'25	

- Command H'25 (1 byte): Inquiry on user MAT information

Response	H'35	Size	No. of areas	
	First address of the area			Last address of the area
	…			
	SUM			

In response to the inquiry on erasure blocks, the boot program returns the number of eras
blocks in the user MAT and the addresses where each block starts and ends.

Command	H'26

- Command H'26 (1 byte): Inquiry on erasure blocks

Response	H'36	Size	No. of blocks	
	First address of the block			Last address of the block
	…			
	SUM			

- Response H'36 (1 byte): Response to the inquiry on erasure blocks
- Size (2 bytes): The total length of the number of blocks and first and last address field
- Number of blocks (1 byte): The number of erasure blocks in flash memory
- First address of the block (4 bytes)
- Last address of the block (4 bytes)
  As many pairs of first and last address data are included as there are blocks.
- SUM (1 byte): Checksum

- Response H'37 (1 byte): Response to the inquiry on programming size
- Size (1 byte): The number of characters in the programming size field (fixed at 2)
- Programming size (2 bytes): The size of the unit for programming
  This is the unit for the reception of data to be programmed.
- SUM (1 byte): Checksum

(k) New Bit Rate Selection

In response to the new-bit-rate selection command, the boot program changes the bit rat
the new bit rate and, if the setting was successful, responds to the ACK sent by the host
returning another ACK at the new bit rate.

The new-bit-rate selection command should be sent after clock-mode selection.

Command

H'3F	Size	Bit rate	Input frequency
No. of multipliers	Multiplier 1	Multiplier 2	
SUM			

- Command H'3F (1 byte): New bit rate selection
- Size (1 byte): The total length of the bit rate, input frequency, number of multipliers,
  multiplier fields
- Bit rate (2 bytes): New bit rate
  The bit rate value divided by 100 should be set here (for example, to select 19200 bp
  H'00C0, which is 192 in decimal notation).
- Input frequency (2 bytes): The frequency of the clock signal fed to the boot program
  This should be the frequency in MHz to the second decimal place, multiplied by 100
  example, if the frequency is 8.882 MHz, the value is truncated to the second decimal
  multiplied by 100, making 888; so H'0378 should be set in this field).

H'FE [-2] for ×1/2)

- SUM (1 byte): Checksum

Response	H'06	

- Response H'06 (1 byte): Response to the new-bit-rate selection command
  This is the ACK code and is returned if the specified bit rate is selected.

Error response	H'BF	ERROR

- Error response H'BF (1 byte): Error response to the new-bit-rate selection command
- ERROR (1 byte): Error code

  H'11: Sum-check error

  H'24: Bit rate selection error (the specified bit rate is not selectable).

  H'25: Input frequency error (the specified input frequency is not within the range from minimum to the maximum value).

  H'26: Frequency multiplier error (the specified multiplier does not match an available

  H'27: Operating frequency error (the specified operating frequency is not within the ra from the minimum to the maximum value).

3. Operating frequency

The operating frequency is calculated from the received input frequency and the freq multiplier or divisor. The input frequency is the frequency of the clock signal supplie LSI, while the operating frequency is the frequency at which the LSI is actually drive following formulae are used for this calculation.

Operating frequency = input frequency × multiplier, or

Operating frequency = input frequency / divisor

The calculated operating frequency is checked to see if it is within the range of the n and maximum values of the operating frequency for the selected clock mode of the s device. A value outside the range generates an operating frequency error.

4. Bit rate

From the peripheral operating frequency (Pϕ) and the bit rate (B), the value (= n) of select bits (CKS) in the serial mode register (SCSMR) and the value (= N) of the bit register (SCBRR) are calculated, after which the error in the bit rate is calculated. Th checked to see if it is smaller than 4%. A result greater than or equal to 4% generates selection error. The following formula is use to calculate the error.

$$\text{Error (\%)} = \left\{ \left[\frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} \right] - 1 \right\} \times 100$$

- Response H'06 (1 byte): The ACK code transferred in response to acknowledgement o
  new bit rate

The sequence of new bit rate selection is shown in figure 21.22.

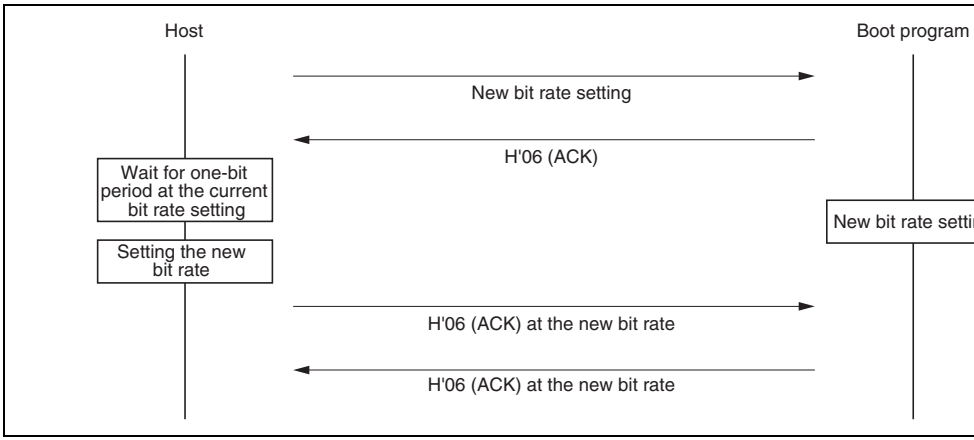


Figure 21.22 Sequence of New Bit Rate Selection

Command	H'40

- Command H'40 (1 byte): Transition to programming/erasure state

Response	H'06

- Response H'06 (1 byte): Response to the transition-to-programming/erasure state co
  This is returned as ACK when erasure of the user boot MAT and user MAT has succ
  after transfer of the erasure program.

Error response	H'C0	H'51

- Error response H'C0 (1 byte): Error response to the transition-to-programming/erasu
  command
- ERROR (1 byte): Error code
  H'51: Erasure error (Erasure did not succeed because of an error.)

- Command H'xx (1 byte): Received command

(6) Order of Commands

In the inquiry-and-selection state, commands should be sent in the following order.

1. Send the inquiry on supported devices command (H'20) to get the list of supported de
2. Select a device from the returned device information, and send the device selection co (H'10) to select that device.
3. Send the inquiry on clock mode command (H'21) to get the available clock modes.
4. Select a clock mode from among the returned clock modes, and send the clock-mode command (H'11).
5. After selection of the device and clock mode, send the commands to inquire about fre multipliers (H'22) and operating frequencies (H'23) to get the information required to new bit rate.
6. Taking into account the returned information on the frequency multipliers and operati frequencies, send a new-bit-rate selection command (H'3F).
7. After the device and clock mode have been selected, get the information required for programming and erasure of the user boot MAT and user MAT by sending the comm inquire about the user boot MAT (H'24), user MAT (H'25), erasure block (H'26), and programming size (H'27).
8. After making all necessary inquiries and the new bit rate selection, send the transition programming/erasure state command (H'40) to place the boot program in the programming/erasure state.

H'42	Selection of user boot MAT programming	Selects transfer of the program for user boot MAT programming.
H'43	Selection of user MAT programming	Selects transfer of the program for user MAT prog
H'50	256-byte programming	Executes 256-byte programming.
H'48	Erasure selection	Selects transfer of the erasure program.
H'58	Block erasure	Executes erasure of the specified block.
H'52	Memory read	Reads from memory.
H'4A	Sum checking of user boot MAT	Executes sum checking of the user boot MAT.
H'4B	Sum checking of user MAT	Executes sum checking of the user MAT.
H'4C	Blank checking of user boot MAT	Executes blank checking of the user boot MAT.
H'4D	Blank checking of user MAT	Executes blank checking of the user MAT.
H'4F	Inquiry on boot program state	Requests information on the state of boot process

Next, the host issues a 256-byte programming command. 256 bytes of data for programm
the method selected by the preceding programming selection command are expected to fo
command. To program more than 256 bytes, repeatedly issue 256-byte programming com
To terminate programming, the host should send another 256-byte programming commar
the address H'FFFFFFFF. On completion of programming, the boot program waits for the
programming/erasure selection command.

To then program the other MAT, start by sending the programming select command.

The sequence of programming by programming-selection and 256-byte programming cor
is shown in figure 21.23.

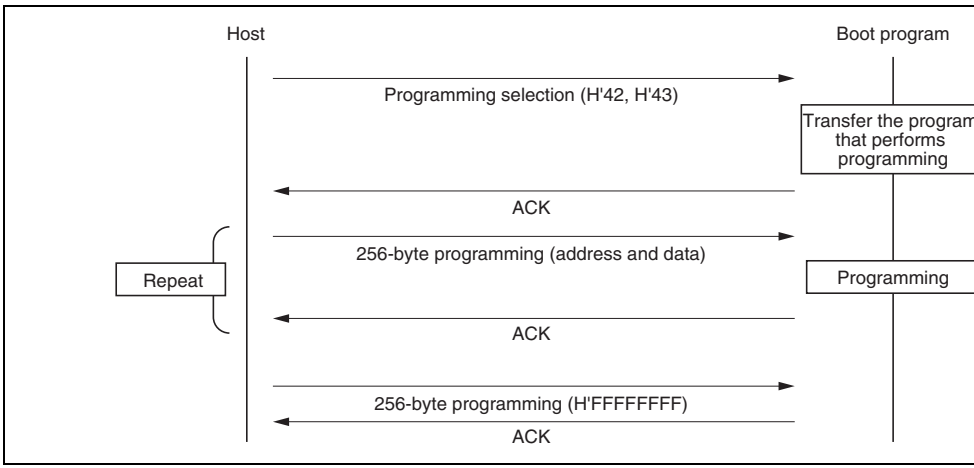


Figure 21.23 Sequence of Programming

- Response H'06 (1 byte): Response to selection of user boot MAT programming
  This ACK code is returned after transfer of the program that performs writing to the
  MAT.

Error response	H'C2	ERROR

- Error response H'C2 (1 byte): Error response to selection of user boot MAT program
- ERROR (1 byte): Error code
  H'54: Error in selection processing (processing was not completed because of a trans

(b) Selection of User MAT Programming

In response to the command for selecting programming of the user MAT, the boot progr
transfers the corresponding flash-writing program, i.e. the program for writing to the use

Command	H'43

- Command H'43 (1 byte): Selects programming of the user MAT.

Response	H'06

(c) 256-Byte Programming

In response to the 256-byte programming command, the boot program executes the flash-
program transferred in response to the command to select programming of the user boot M
user MAT.

Command	H'50	Address for programming					
	Data	…					
	…						
	SUM						

- Command H'50 (1 byte): 256-byte programming
- Address for programming (4 bytes): Address where programming starts
  Specify the address of a 256-byte boundary.
  [Example] H'00, H01, H'00, H'00: H'00010000
- Programming data (n bytes): Data for programming
  The length of the programming data is the size returned in response to the programmi
  inquiry command.
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): Response to 256-byte programming
  The ACK code is returned on completion of the requested programming.

Error response	H'D0	ERROR

address field H'FFFFFFFF. This informs the boot program that all data for the selected I
been sent; the boot program then waits for the next programming/erasure selection comm

Command	H'50	Address for programming	SUM

- Command H'50 (1 byte): 256-byte programming
- Address for programming (4 bytes): Terminating code (H'FF, H'FF, H'FF, H'FF)
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): Response to 256-byte programming
  This ACK code is returned on completion of the requested programming.

Error response	H'D0	ERROR

- Error response H'D0 (1 byte): Error response to 256-byte programming
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'53: Programming error

The sequence of erasure by the erasure selection command and block erasure command is in figure 21.24.

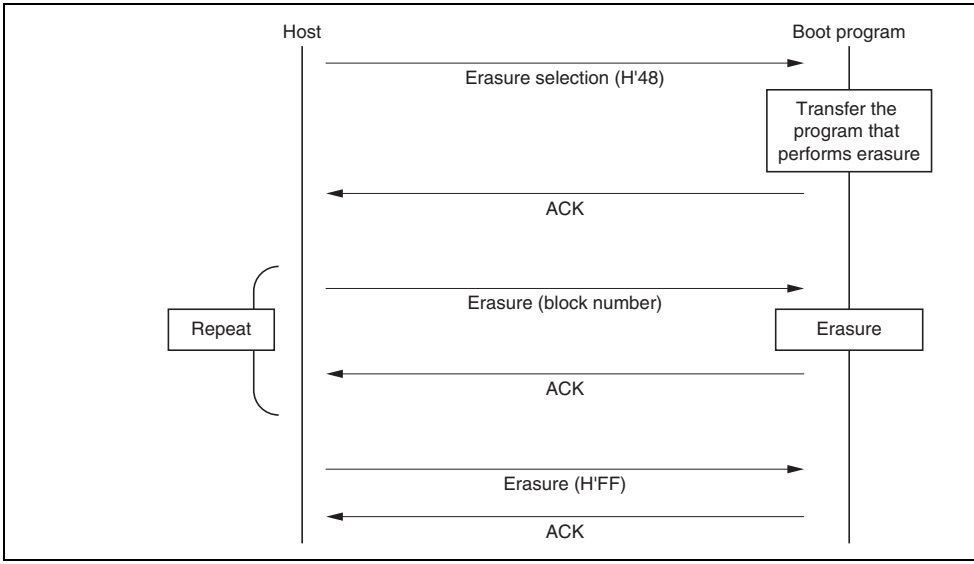


Figure 21.24 Sequence of Erasure

- Response H'06 (1 byte): Response to selection of erasure
  This ACK code is returned after transfer of the program that performs erasure.

Error response	H'C8	ERROR

- Error response H'C8 (1 byte): Error response to selection of erasure
- ERROR (1 byte): Error code
  H'54: Error in selection processing (processing was not completed because of a trans

(b) Block Erasure

In response to the block erasure command, the boot program erases the data in a specifie
the user MAT.

Command	H'58	Size	Block number	SUM

- Command H'58 (1 byte): Erasure of a block
- Size (1 byte): The number of characters in the block number field (fixed at 1)
- Block number (1 byte): Block number of the block to be erased
- SUM (1 byte): Checksum

Response	H'06

H'51: Erasure error (an error occurred during erasure.)

On receiving the command with H'FF as the block number, the boot program stops erasu processing and waits for the next programming/erasure selection command.

Command	H'58	Size	Block number	SUM

- Command H'58 (1 byte): Erasure of a block
- Size (1 byte): The number of characters in the block number field (fixed at 1)
- Block number (1 byte): H'FF (erasure terminating code)
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): ACK code to indicate response to the request for termination erasure

To perform erasure again after having issued the command with the block number specifi H'FF, execute the process from the selection of erasure.

(10) Memory Read

In response to the memory read command, the boot program returns the data from the spe address.

Command	H'52	Size	Area	First address for reading	
	Amount to read				SUM

Response	H'52	Amount to read					
	Data	…					
	SUM						

- Response H'52 (1 byte): Response to the memory read command
- Amount to read (4 bytes): The amount to read as specified in the memory read comm
- Data (n bytes): The specified amount of data read out from the specified address
- SUM (1 byte): Checksum

Error response	H'D2	ERROR

- Error response H'D2 (1 byte): Error response to memory read command
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'2A: Address error (the address specified for reading is beyond the range of the MA
  H'2B: Size error (the specified amount is greater than the size of the MAT,
  the last address for reading as calculated from the specified address for the start of re
  the amount to read is beyond the MAT area, or "0" was specified as the amount to re

- Response H'5A (1 byte): Response to sum checking of the user boot MAT
- Size (1 byte): The number of characters in the checksum for the MAT (fixed at 4)
- Checksum for the MAT (4 bytes): Result of checksum calculation for the user boot M the total of all data in the MAT, in byte units.
- SUM (1 byte): Checksum (for the transmitted data)

(12) Sum Checking of the User MAT

In response to the command for sum checking of the user MAT, the boot program adds al data in the user MAT and returns the result.

Command	H'4B	

- Command H'4B (1 byte): Sum checking of the user MAT

Response	H'5B	Size	Checksum for the MAT	SUM

- Response H'5B (1 byte): Response to sum checking of the user MAT
- Size (1 byte): The number of characters in the checksum for the MAT (fixed at 4)
- Checksum for the MAT (4 bytes): Result of checksum calculation for the user MAT: the total of all data in the MAT, in byte units.
- SUM (1 byte): Checksum (for the transmitted data)

- Response H'06 (1 byte): Response to blank checking of the user boot MAT
  This ACK code is returned when the whole area is blank (all bytes are H'FF).

Error response	H'CC	H'52

- Error response H'CC (1 byte): Error response to blank checking of the user boot MA
- Error code H'52 (1 byte): Non-erased error

(14) Blank Checking of the User MAT

In response to the command for blank checking of the user MAT, the boot program chec
if the whole of the user MAT is blank; the value returned indicates the result.

Command	H'4D

- Command H'4D (1 byte): Blank checking of the user boot MAT

Response	H'06

- Response H'06 (1 byte): Response to blank checking of the user MAT
  The ACK code is returned when the whole area is blank (all bytes are H'FF).

Error response	H'CD	H'52

- Error response H'CD (1 byte): Error response to blank checking of the user MAT
- Error code H'52 (1 byte): Non-erased error

- Response H'5F (1 byte): Response to the inquiry regarding boot-program state
- Size (1 byte): The number of characters in STATUS and ERROR (fixed at 2)
- STATUS (1 byte): State of the standard boot program
  See table 21.14, Status Codes.
- ERROR (1 byte): Error state (indicates whether the program is in normal operation or has occurred)
  ERROR = 0: Normal
  ERROR ≠ 0: Error
  See table 21.15, Error Codes.
- SUM (1 byte): Checksum

Table 21.14 Status Codes

Code	Description
H'11	Waiting for device selection
H'12	Waiting for clock-mode selection
H'13	Waiting for bit-rate selection
H'1F	Waiting for transition to programming/erasure status (bit-rate selection co
H'31	Erasing the user MAT or user boot MAT
H'3F	Waiting for programming/erasure selection (erasure complete)
H'4F	Waiting to receive data for programming (programming complete)
H'5F	Waiting for erasure block specification (erasure complete)

H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data length error (size error)
H'51	Erasure error
H'52	Non-erased error
H'53	Programming error
H'54	Selection processing error
H'80	Command error
H'FF	Bit-rate matching acknowledge error

21.8.2 Areas for Storage of the Procedural Program and Data for Programming

In the descriptions in the previous section, storable areas for the programming/erasing pr
programs and program data are assumed to be in on-chip RAM. However, the procedure
and data can be stored in and executed from other areas (e.g. external address space) as l
following conditions are satisfied.

1. The on-chip programming/erasing program is downloaded from the address set by F
   on-chip RAM, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use 128 bytes or more as a stack. M
   this area is reserved.
3. Since download by setting the SCO bit to 1 will cause the MATs to be switched, it sh
   executed in on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, unt
   result of downloading has been decided.

before the reset signal is released.

7. Switching of the MATs by FMATS is needed for programming/erasing of the user M user boot mode. The program which switches the MATs should be executed from the RAM. For details, see section 21.7.1, Switching between User MAT and User Boot M Please make sure you know which MAT is selected when switching the MATs.
8. When the program data storage area indicated by the FMPDR parameter in the progra processing is within the flash memory area, an error will occur. Therefore, temporaril the program data to on-chip RAM to change the address set in FMPDR to an address than flash memory.

Based on these conditions, tables 21.16 and 21.17 show the areas in which the program d be stored and executed according to the operation type and mode.

Table 21.16 Executable MAT

	Initiated Mode	
Operation	**User Program Mode**	**User Boot Mode***
Programming	Table 21.17 (1)	Table 21.17 (3)
Erasing	Table 21.17 (2)	Table 21.17 (4)

Note: * Programming/Erasing is possible to user MATs.

	Writing 1 to SCO in FCCS (download)	√	X	X		√
	Key register clearing	√	√	√	√	
	Judging download result	√	√	√	√	
	Download error processing	√	√	√	√	
	Setting initialization parameters	√	√	√	√	
Pro-gram-ming proce-dure	Initialization	√	X	X	√	
	Judging initialization result	√	√	√	√	
	Initialization error processing	√	√	√	√	
	Writing H'5A to key register	√	√	√	√	
	Setting programming parameters	√	X	√	√	
	Programming	√	X	X	√	
	Judging programming result	√	X	√	√	
	Programming error processing	√	X	√	√	
	Key register clearing	√	X	√	√	

Note: * If the data has been transferred to on-chip RAM in advance, this area can be

	(download)				
	Key register clearing	√	√	√	√
	Judging download result	√	√	√	√
	Download error processing	√	√	√	√
	Setting initialization parameters	√	√	√	√
	Initialization	√	X	X	√
Erasing procedure	Judging initialization result	√	√	√	√
	Initialization error processing	√	√	√	√
	Writing H'5A to key register	√	√	√	√
	Setting erasure parameters	√	X	√	√
	Erasure	√	X	X	√
	Judging erasure result	√	X	√	√
	Erasing error processing	√	X	√	√
	Key register clearing	√	X	√	√

	Writing H'A5 to key register	√	√	√		√	
	Writing 1 to SCO in FCCS (download)	√	X	X			√
	Key register clearing	√	√	√		√	
Programming procedure	Judging download result	√	√	√		√	
	Download error processing	√	√	√		√	
	Setting initialization parameters	√	√	√		√	
	Initialization	√	X	X		√	
	Judging initialization result	√	√	√		√	
	Initialization error processing	√	√	√		√	
	Switching MATs by FMATS	√	X	X	√		
	Writing H'5A to Key Register	√	X	√	√		

gram-ming proce-dure	result					
	Programming error processing	√	X*2	√	√	
	Key register clearing	√	X	√	√	
	Switching MATs by FMATS	√	X	X		√

Notes: 1. If the data has been transferred to on-chip RAM in advance, this area can be u
2. If the MATs have been switched by FMATS in on-chip RAM, this MAT can be

	Writing 1 to SCO in FCCS (download)	√	X	X			√
	Key register clearing	√	√	√		√	
	Judging download result	√	√	√		√	
	Download error processing	√	√	√		√	
Erasing procedure	Setting initialization parameters	√	√	√		√	
	Initialization	√	X	X		√	
	Judging initialization result	√	√	√		√	
	Initialization error processing	√	√	√		√	
	Switching MATs by FMATS	√	X	X		√	
	Writing H'5A to key register	√	X	√	√		
	Setting erasure parameters	√	X	√	√		

dure						
	Key register clearing	√	X	√	√	
	Switching MATs by FMATS	√	X	X		√

Note: * If the MATs have been switched by FMATS in on-chip RAM, this MAT can be u

21.9 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing socket adapter, just as for a discrete flash memory. Use a PROM programmer that suppor Renesas 512-kbyte flash memory on-chip MCU device type (F-ZTAT512DV3_15A).

The 32 Kbyte on-chip RAM is divided into four pages (pages 0 to 3).

The 24 Kbyte on-chip RAM is divided into three pages (pages 0 to 2).

- Memory map

  The on-chip RAM is located in the address spaces shown in table 22.1, 22.2.

Table 22.1 32 Kbyte On-Chip RAM Address Spaces

Page	Address
Page 0	H'FFF80000 to H'FFF81FFF
Page 1	H'FFF82000 to H'FFF83FFF
Page 2	H'FFF84000 to H'FFF85FFF
Page 3	H'FFF86000 to H'FFF87FFF

Table 22.2 24 Kbyte On-Chip RAM Address Spaces

Page	Address
Page 0	H'FFF80000 to H'FFF81FFF
Page 1	H'FFF82000 to H'FFF83FFF
Page 2	H'FFF84000 to H'FFF85FFF

- Ports

  Each page has two independent read and write ports and is connected to the internal
  bus), CPU instruction fetch bus (F bus), and CPU memory access bus (M bus). (Note
  bus is connected only to the read ports.)

  The F bus and M bus are used for access by the CPU, and the I bus is used for access
  DMAC.

- Priority

far as possible. For example, no conflict will arise if different memory or pages are acces
each bus.

22.2.2 RAME and RAMWE Bits

Before disabling memory operation or write access through the RAME or RAMWE bit, b
read from any address and then write to the same address in each page; otherwise, the las
data in each page may not be actually written to the RAM. For setting the RAME and RA
bits, see section 23.3.5, System Control Register 1 (SYSCR1), and section 23.3.6, System
Register 2 (SYSCR2).

```
// For page 2
  MOV.L  #H'FFF84000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For page 3
  MOV.L  #H'FFF86000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0
```

Figure 22.1 Examples of Read/Write before Disabling RAM

1. Sleep mode
2. Software standby mode
3. Module standby function

Table 23.1 shows the transition conditions for entering the modes from the program exe state, as well as the CPU and peripheral module states in each mode and the procedures canceling each mode.

Software standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts	Held	Halts (contents are held)	Halts	Self-refreshing	• NM • IR • Ma • Po
Module standby function	Set the MSTP bits in STBCR2, STBCR3, and STBCR4 to 1	Runs	Runs	Held	Specified module halts (contents are held)	Specified module halts	Auto-refreshing	• Cle to • Po (or UB DM

Note: * The pin state is retained or set to high impedance. For details, see appendix A States.

2. The watchdog timer (WDT) starts counting with the WT/$\overline{IT}$ bit in WTCSR set to 1 the RSTS bit in WRCSR set to 0 while the RSRE bit in WRCSR is 1, and the coun overflows.
3. The H-UDI reset is generated (for details on the H-UDI reset, see section 24, User Debugging Interface (H-UDI)).

(2) Manual Reset

1. A low level is input to the $\overline{MRES}$ pin.
2. The WDT starts counting with the WT/$\overline{IT}$ bit in WTCSR set to 1 and with the RST WRCSR set to 1 while the RSRE bit in WRCSR is 1, and the counter overflows.

23.3 Register Descriptions

The following registers are used in power-down modes.

Table 23.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address
Standby control register	STBCR	R/W	H'00	H'FFFE0014
Standby control register 2	STBCR2	R/W	H'00	H'FFFE0018
Standby control register 3	STBCR3	R/W	H'7E	H'FFFE0408
Standby control register 4	STBCR4	R/W	H'F4	H'FFFE040C
System control register 1	SYSCR1	R/W	H'FF	H'FFFE0402
System control register 2	SYSCR2	R/W	H'FF	H'FFFE0404

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby Specifies transition to software standby m 0: Executing SLEEP instruction puts chip mode. 1: Executing SLEEP instruction puts chip software standby mode.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP10	0	R/W	Module Stop 10 When the MSTP10 bit is set to 1, the supply clock to the H-UDI is halted. 0: H-UDI runs. 1: Clock supply to H-UDI halted.
6	MSTP9	0	R/W	Module Stop 9 When the MSTP9 bit is set to 1, the supply o clock to the UBC is halted. 0: UBC runs. 1: Clock supply to UBC halted.
5	MSTP8	0	R/W	Module Stop 8 When the MSTP8 bit is set to 1, the supply o clock to the DMAC is halted. 0: DMAC runs. 1: Clock supply to DMAC halted.
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	HIZ	0	R/W	Port High Impedance Selects whether the state of a specified pin retained or the pin is placed in the high-imp state in software standby mode. See appen States to determine the pin to which this co applied. Do not set this bit when the TME bit of WTS WDT is 1. When setting the output pin to th impedance state, set the HIZ bit with the TM being 0. 0: The pin state is held in software standby 1: The pin state is set to the high-impedanc software standby mode.
6	MSTP36	1	R/W	Module Stop 36 When the MSTP36 bit is set to 1, the supply clock to the MTU2S is halted. 0: MTU2S runs. 1: Clock supply to MTU2S halted.
5	MSTP35	1	R/W	Module Stop 35 When the MSTP35 bit is set to 1, the supply clock to the MTU2 is halted. 0: MTU2 runs. 1: Clock supply to MTU2 halted.

				0: IIC3 runs. 1: Clock supply to IIC3 halted.
2	MSTP32	1	R/W	Module Stop 32 When the MSTP32 bit is set to 1, the supply clock to the ADC is halted. 0: ADC runs. 1: Clock supply to ADC halted.
1	MSTP31	1	R/W	Module Stop 31 When the MSTP31 bit is set to 1, the supply clock to the DAC is halted. 0: DAC runs. 1: Clock supply to DAC halted.
0	MSTP30	0	R/W	Module Stop 30 When the MSTP30 bit is set to 1, the supply clock to the flash memory is halted. 0: Flash memory runs. 1: Clock supply to flash memory halted.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP47	1	R/W	Module Stop 47 When the MSTP47 bit is set to 1, the supply clock to the SCIF0 is halted. 0: SCIF0 runs. 1: Clock supply to SCIF0 halted.
6	MSTP46	1	R/W	Module Stop 46 When the MSTP46 bit is set to 1, the supply clock to the SCIF1 is halted. 0: SCIF1 runs. 1: Clock supply to SCIF1 halted.
5	MSTP45	1	R/W	Module Stop 45 When the MSTP45 bit is set to 1, the supply clock to the SCIF2 is halted. 0: SCIF2 runs. 1: Clock supply to SCIF2 halted.
4	MSTP44	1	R/W	Module Stop 44 When the MSTP44 bit is set to 1, the supply clock to the SCIF3 is halted. 0: SCIF3 runs. 1: Clock supply to SCIF3 halted.
3	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

				0: WAVEIF runs. 1: Clock supply to WAVEIF halted.
0	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

Note that when clearing the RAME bit to 0 to disable the on-chip RAM, be sure to exec
instruction to read from or write to the same arbitrary address in each page before settin
RAME bit. If such an instruction is not executed, the data last written to each page may
written to the on-chip RAM. Furthermore, an instruction to access the on-chip RAM sho
located immediately after the instruction to write to SYSCR1. If an on-chip RAM acces
instruction is set, normal access is not guaranteed.

To enable the on-chip RAM by setting the RAME bit to 1, place an instruction to read d
SYSCR1 immediately after an instruction to write to SYSCR1. If an instruction to acces
chip RAM is placed immediately after the instruction to write to SYSCR1, normal acces
guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAME3	RAME2	RAME1	RAME0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write v should always be 1.
3	RAME3	1	R/W	RAM Enable 3 (corresponding RAM addres H'FFF86000 to H'FFF87FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled Note: This is a reserved bit on versions with RAM. Its value is always 1 when read write 1 to this bit.

0	RAME0	1	R/W	RAM Enable 0 (corresponding RAM address H'FFF80000 to H'FFF81FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled

instruction to read from or write to the same arbitrary address in each page before setting RAMWE bit. If such an instruction is not executed, the data last written to each page ma written to the on-chip RAM. Furthermore, an instruction to access the on-chip RAM sho located immediately after the instruction to write to SYSCR2. If an on-chip RAM access instruction is set, normal access is not guaranteed.

To enable the on-chip RAM by setting the RAMWE bit to 1, locate an instruction to rea from SYSCR2 immediately after an instruction to write to SYSCR2. If an instruction to on-chip RAM is located immediately after the instruction to write to SYSCR2, normal a not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAM WE3	RAM WE2	RAM WE1	RAM WE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write v should always be 1.
3	RAMWE3	1	R/W	RAM Write Enable 3 (corresponding RAM a H'FFF86000 to H'FFF87FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled Note: This is a reserved bit on versions with RAM. Its value is always 1 when read write 1 to this bit.

0	RAMWE0	1	R/W	RAM Write Enable 0 (corresponding RAM ac H'FFF80000 to H'FFF81FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled

(2) Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, and on-chip peripheral module), DM error, or reset (manual reset or power-on reset).

- Canceling with an interrupt

  When an NMI, IRQ, or on-chip peripheral module interrupt occurs, sleep mode is ca interrupt exception handling is executed. When the priority level of the generated int equal to or lower than the interrupt mask level that is set in the status register (SR) o or the interrupt by the on-chip peripheral module is disabled on the module side, the request is not accepted and sleep mode is not canceled.
- Canceling with a DMA address error

  When a DMA address error occurs, sleep mode is canceled and DMA address error handling is executed.
- Canceling with a reset

  Sleep mode is canceled by a power-on reset or a manual reset.

23.4.2 Software Standby Mode

(1) Transition to Software Standby Mode

The LSI switches from a program execution state to software standby mode by executin SLEEP instruction when the STBY bit in STBCR is 1. In software standby mode, not or CPU but also the clock and on-chip peripheral modules halt. The clock output from the also halts.

The contents of the CPU remain unchanged. Some registers of on-chip peripheral modu however, initialized. Table 23.4 shows the states of peripheral module registers in softw standby mode.

User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
A/D converter (ADC)	All registers	—
I/O port	—	All registers
User debugging interface (H-UDI)	—	All registers
Serial communication interface with FIFO (SCIF)	—	All registers
Direct memory access controller (DMAC)	—	All registers
Multi-function timer pulse unit 2 (MTU2)	—	All registers
Multi-function timer pulse unit 2S (MTU2S)	—	All registers
Port output enable 2 (POE2)	—	All registers
Compare match timer (CMT)	All registers	—
I^2C bus interface 3 (IIC3)	BC2 and BC0 bits in ICMR register	Other than BC[2: ICMR
D/A converter (DAC)	—	All registers

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the WD
2. Set the WDT's timer counter (WTCNT) to 0 and the CKS[2:0] bits in WTCSR to appr values to secure the specified oscillation settling time.
3. After setting the STBY bit in STBCR to 1, read STBCR. Then, execute a SLEEP inst

(WDT) used to count the oscillation settling time.

After the elapse of the time set in the clock select bits (CKS[2:0]) in the watchdog timer control/status register (WTCSR) of the WDT before the transition to software standby m WDT overflow occurs. Since this overflow indicates that the clock has been stabilized, t pulse will be supplied to the entire chip after this overflow. Software standby mode is cl NMI interrupt exception handling (IRQ interrupt exception handling in the case of IRQ

When canceling software standby mode by the NMI interrupt or IRQ interrupt, set the C bits so that the WDT overflow period will be equal to or longer than the oscillation settl

The clock output phase of the CK pin may be unstable immediately after detecting an in and until software standby mode is canceled. When software standby mode is canceled b falling edge of the NMI pin, the NMI pin should be high when the CPU enters software mode (when the clock pulse stops) and should be low when the CPU returns from softw standby mode (when the clock is initiated after the oscillation settling). When software s mode is canceled by the rising edge of the NMI pin, the NMI pin should be low when th enters software standby mode (when the clock pulse stops) and should be high when the returns from software standby mode (when the clock is initiated after the oscillation sett is the same with the IRQ pin.)

(b) Exit from Software Standby by a Reset

When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin is driven low, this LSI enters the power-on reset or manual state, and software standby mode is exited.

Keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until the clock oscillation settles.

Internal clock pulses are output continuously on the CK pin.

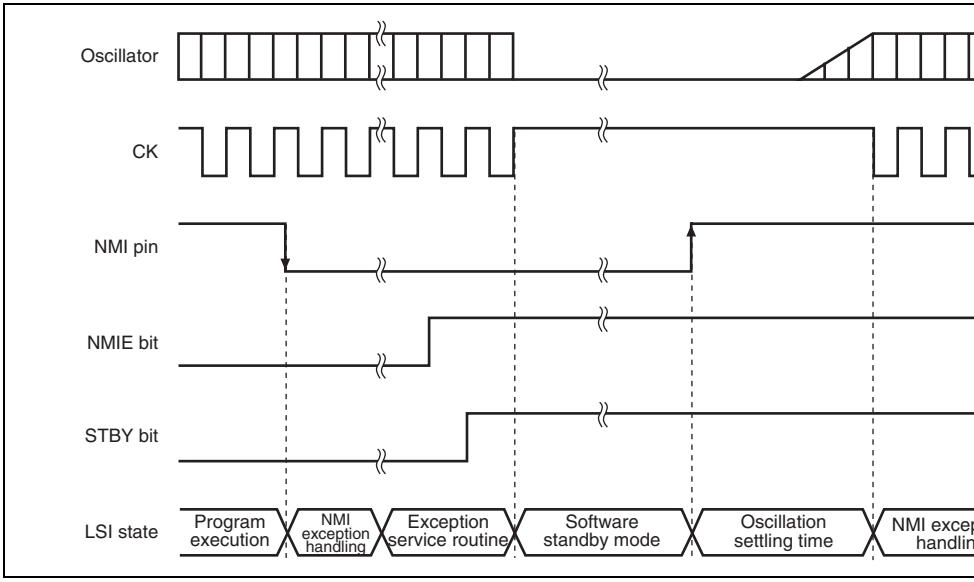


Figure 23.1 NMI Timing in Software Standby Mode (Application Example

However, the states of the CMT and DAC registers are exceptional. In the CMT, all regi
initialized in software standby mode, but retain their previous values in module standby
the DAC, all registers retain their previous values in software standby mode, but are init
module standby mode.

(2) Canceling Module Standby Function

The module standby function can be canceled by clearing the MSTP bits to 0, or by a po
reset (only possible for H-UDI, UBC, and DMAC). When taking a module out of the mo
standby state by clearing the corresponding MSTP bit to 0, read the MSTP bit to confirm
has been cleared to 0.

Figure 24.1 shows a block diagram of the H-UDI.

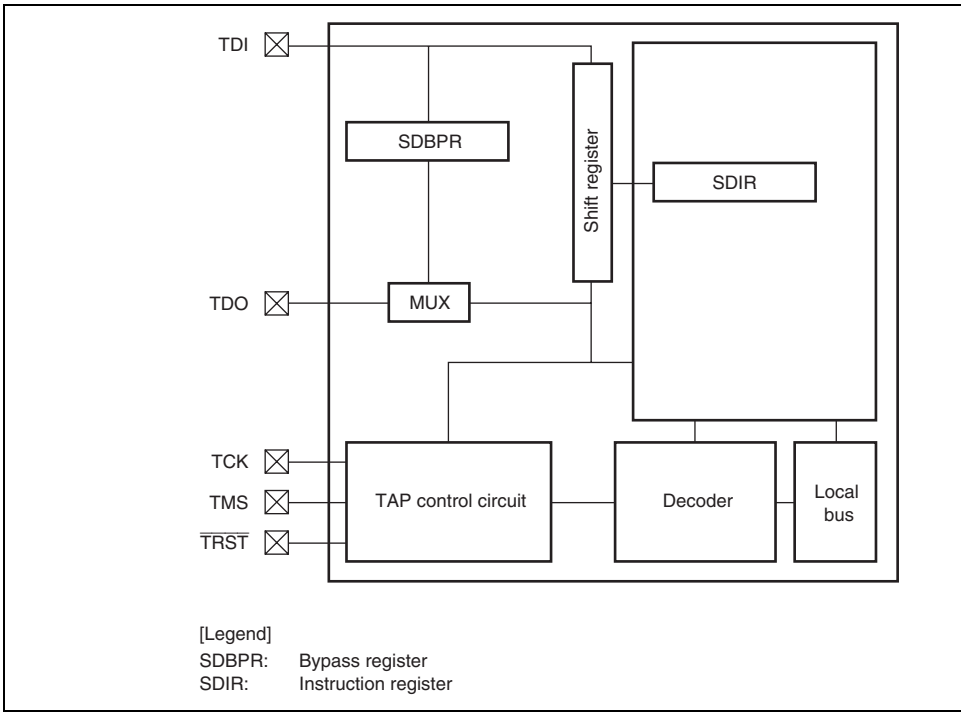


Figure 24.1 Block Diagram of H-UDI

			synchronization with TCK. For the p see figure 24.2.
H-UDI reset input pin	$\overline{\text{TRST}}$	Input	Input is accepted asynchronously w respect to TCK, and when low, the reset. $\overline{\text{TRST}}$ must be low for a cons period when power is turned on reg of using the H-UDI function. See se 24.4.2, Reset Configuration, for mo information.
H-UDI serial data input pin	TDI	Input	Data transfer to the H-UDI is execu changing this signal in synchroniza TCK.
H-UDI serial data output pin	TDO	Output	Data read from the H-UDI is execut reading this pin in synchronization TCK. The initial value of the data o timing is the TCK falling edge. This changed to the TCK rising edge by inputting the TDO change timing sw command to SDIR. See section 24 TDO Output Timing, for more inform
ASE mode select pin	$\overline{\text{ASEMD}}$*	Input	If a low level is input at the $\overline{\text{ASEMD}}$ while the $\overline{\text{RES}}$ pin is asserted, ASE entered; if a high level is input, norm mode is entered. In ASE mode, dec emulator function can be used. The level at the $\overline{\text{ASEMD}}$ pin should be h at least one cycle after $\overline{\text{RES}}$ negati

Note: * When the emulator is not in use, fix this pin to the high level.

24.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to BYP mode, SDBPR is connected between H-UDI pins TDI and TDO. The initial value is und

24.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. It is initialized by $\overline{\text{TRST}}$ assertion or in the TAP test reset state, and can be written to by the H-UDI irrespective of CPU mode. Operation is n guaranteed if a reserved command is set in this register. The initial value is H'EFFD.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	TI[7:0]								-	-	-	-	-	-	-
Initial value:	1*	1*	1*	0*	1*	1*	1*	1*	1	1	1	1	1	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The initial value of the TI[7:0] bits is a reserved value. When setting a command, the TI[7:0] bits must be set to

This bit is always read as 1.

Note: * The initial value of the TI[7:0] bits is a reserved value. When setting a comman
TI[7:0] bits must be set to another value.

Table 24.3 H-UDI Commands

Bits 15 to 8								
TI7	**TI6**	**TI5**	**TI4**	**TI3**	**TI2**	**TI1**	**TI0**	**Description**
0	1	1	0	—	—	—	—	H-UDI reset negate
0	1	1	1	—	—	—	—	H-UDI reset assert
1	0	0	1	1	1	0	0	TDO change timing
1	0	1	1	—	—	—	—	H-UDI interrupt
1	1	1	1	—	—	—	—	BYPASS mode
Other than above								Reserved

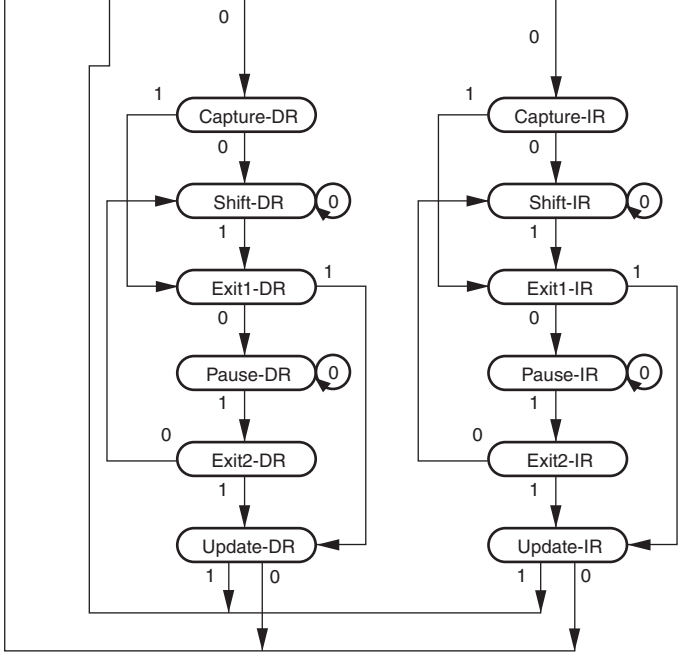


Figure 24.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI va sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. F on change timing of the TDO value, see section 24.4.3, TDO Output Timing. Th at high impedance, except with shift-DR and shift-IR states. During the change t 0, there is a transition to test-logic-reset asynchronously with TCK.

	H	Power-on reset
H	L	H-UDI reset only
	H	Normal operation

Notes: 1. Performs normal mode and ASE mode settings
$\overline{\text{ASEMD}}$ = H, normal mode
$\overline{\text{ASEMD}}$ = L, ASE mode
2. In ASE mode, reset hold is entered if the $\overline{\text{TRST}}$ pin is driven low while the $\overline{\text{RES}}$ negated. In this state, the CPU does not start up. When $\overline{\text{TRST}}$ is driven high, H operation is enabled, but the CPU does not start up. The reset hold state is ca by a power-on reset.

24.4.3 TDO Output Timing

The initial value of the TDO change timing is to perform data output from the TDO pin o TCK falling edge. However, setting a TDO change timing switch command in SDIR via t UDI pin and passing the Update-IR state synchronizes the TDO change timing to the TCI edge. Thereafter the TDO change timing cannot be changed unless a power-on reset that the $\overline{\text{TRST}}$ pin simultaneously is performed.

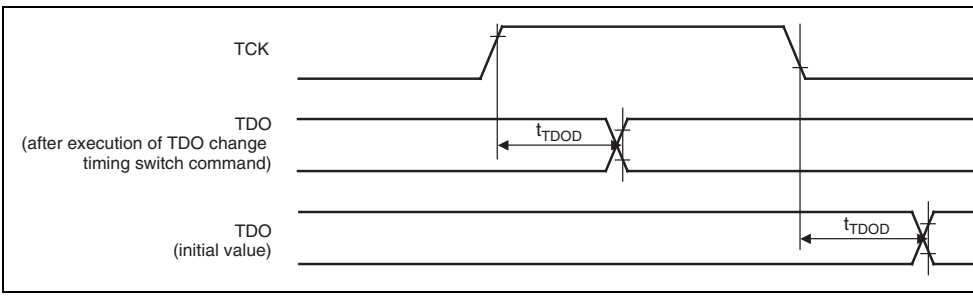


Figure 24.3 H-UDI Data Transfer Timing

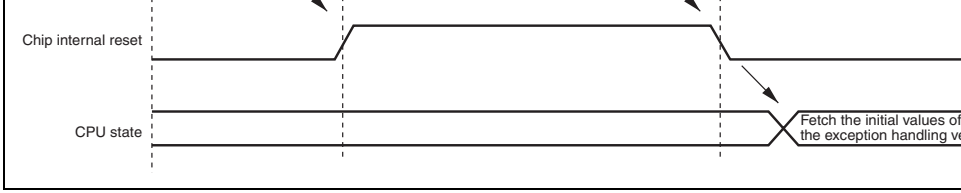


Figure 24.4 H-UDI Reset

24.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-U
SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in fetchi
exception service routine start address from the exception handling vector table, jumpin
address, and starting program execution from that address. This interrupt request has a f
priority level of 15.

H-UDI interrupts are accepted in sleep mode, but not in software standby mode.

Table 25.1 Pin Configuration

Pin Name	Symbol	I/O	Function
WAVE clock pin	WSCK	Output	WAVE interface clock output
WAVE receive data	WRXD	Input	WAVE interface receive data input
WAVE transmit data	WTXD	Output	WAVE interface transmit data output

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Ad
(by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a co
for holding data.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functio
module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the correspo
section.
- The register states described are for the basic operating modes. If there is a specific
on-chip peripheral module, refer to the section on that on-chip peripheral module.

4. Notes when Writing to the On-Chip Peripheral Modules

- To access an on-chip module register, two or more peripheral module clock (Pf) cyc
required. Care must be taken in system design. When the CPU writes data to the inte
peripheral registers, the CPU performs the succeeding instructions without waiting f
completion of writing to registers. For example, a case is described here in which the
transferring to the software standby mode for power savings. To make this transition
SLEEP instruction must be performed after setting the STBY bit in the STBCR regis
However a dummy read of the STBCR register is required before executing the SLE
instruction. If a dummy read is omitted, the CPU executes the SLEEP instruction be
STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A
read of the STBCR register is indispensable to complete writing to the STBY bit. To
change by internal peripheral registers while performing the succeeding instructions,
dummy read of registers to which write instruction is given and then perform the suc
instructions.

	IRQ interrupt request register	IRQRR	16	H'FFFE0806
	Bank control register	IBCR	16	H'FFFE080C
	Bank number register	IBNR	16	H'FFFE080E
	Interrupt priority register 01	IPR01	16	H'FFFE0818
	Interrupt priority register 02	IPR02	16	H'FFFE081A
	Interrupt priority register 05	IPR05	16	H'FFFE0820
	Interrupt priority register 06	IPR06	16	H'FFFE0C00
	Interrupt priority register 07	IPR07	16	H'FFFE0C02
	Interrupt priority register 08	IPR08	16	H'FFFE0C04
	Interrupt priority register 09	IPR09	16	H'FFFE0C06
	Interrupt priority register 10	IPR10	16	H'FFFE0C08
	Interrupt priority register 11	IPR11	16	H'FFFE0C0A
	Interrupt priority register 12	IPR12	16	H'FFFE0C0C
	Interrupt priority register 13	IPR13	16	H'FFFE0C0E
	Interrupt priority register 14	IPR14	16	H'FFFE0C10
	Interrupt priority register 15	IPR15	16	H'FFFE0C12
UBC	Break address register_0	BAR_0	32	H'FFFC0400
	Break address mask register_0	BAMR_0	32	H'FFFC0404
	Break bus cycle register_0	BBR_0	16	H'FFFC04A0
	Break address register_1	BAR_1	32	H'FFFC0410
	Break address mask register_1	BAMR_1	32	H'FFFC0414
	Break bus cycle register_1	BBR_1	16	H'FFFC04B0
	Break address register_2	BAR_2	32	H'FFFC0420

	CS0 space bus control register	CS0BCR	32	H'FFFC0004
	CS1 space bus control register	CS1BCR	32	H'FFFC0008
	CS2 space bus control register	CS2BCR	32	H'FFFC000C
	CS3 space bus control register	CS3BCR	32	H'FFFC0010
	CS4 space bus control register	CS4BCR	32	H'FFFC0014
	CS5 space bus control register	CS5BCR	32	H'FFFC0018
	CS6 space bus control register	CS6BCR	32	H'FFFC001C
	CS7 space bus control register	CS7BCR	32	H'FFFC0020
	CS0 space wait control register	CS0WCR	32	H'FFFC0028
	CS1 space wait control register	CS1WCR	32	H'FFFC002C
	CS2 space wait control register	CS2WCR	32	H'FFFC0030
	CS3 space wait control register	CS3WCR	32	H'FFFC0034
	CS4 space wait control register	CS4WCR	32	H'FFFC0038
	CS5 space wait control register	CS5WCR	32	H'FFFC003C
	CS6 space wait control register	CS6WCR	32	H'FFFC0040
	CS7 space wait control register	CS7WCR	32	H'FFFC0044
	SDRAM control register	SDCR	32	H'FFFC004C
	Refresh timer control/status register	RTCSR	16	H'FFFC0050
	Refresh timer counter	RTCNT	16	H'FFFC0054
	Refresh time constant register	RTCOR	16	H'FFFC0058
DMAC	DMA source address register_0	SAR_0	32	H'FFFE1000
	DMA destination address register_0	DAR_0	32	H'FFFE1004
	DMA transfer count register_0	DMATCR_0	32	H'FFFE1008
	DMA channel control register_0	CHCR_0	32	H'FFFE100C

DMA reload source address register_1	RSAR_1	32	H'FFFE1110
DMA reload destination address register_1	RDAR_1	32	H'FFFE1114
DMA reload transfer count register_1	RDMATCR_1	32	H'FFFE1118
DMA source address register_2	SAR_2	32	H'FFFE1020
DMA destination address register_2	DAR_2	32	H'FFFE1024
DMA transfer count register_2	DMATCR_2	32	H'FFFE1028
DMA channel control register_2	CHCR_2	32	H'FFFE102C
DMA reload source address register_2	RSAR_2	32	H'FFFE1120
DMA reload destination address register_2	RDAR_2	32	H'FFFE1124
DMA reload transfer count register_2	RDMATCR_2	32	H'FFFE1128
DMA source address register_3	SAR_3	32	H'FFFE1030
DMA destination address register_3	DAR_3	32	H'FFFE1034
DMA transfer count register_3	DMATCR_3	32	H'FFFE1038
DMA channel control register_3	CHCR_3	32	H'FFFE103C
DMA reload source address register_3	RSAR_3	32	H'FFFE1130
DMA reload destination address register_3	RDAR_3	32	H'FFFE1134
DMA reload transfer count register_3	RDMATCR_3	32	H'FFFE1138
DMA source address register_4	SAR_4	32	H'FFFE1040
DMA destination address register_4	DAR_4	32	H'FFFE1044
DMA transfer count register_4	DMATCR_4	32	H'FFFE1048
DMA channel control register_4	CHCR_4	32	H'FFFE104C

DMA reload source address register_5	RSAR_5	32	H'FFFE1150
DMA reload destination address register_5	RDAR_5	32	H'FFFE1154
DMA reload transfer count register_5	RDMATCR_5	32	H'FFFE1158
DMA source address register_6	SAR_6	32	H'FFFE1060
DMA destination address register_6	DAR_6	32	H'FFFE1064
DMA transfer count register_6	DMATCR_6	32	H'FFFE1068
DMA channel control register_6	CHCR_6	32	H'FFFE106C
DMA reload source address register_6	RSAR_6	32	H'FFFE1160
DMA reload destination address register_6	RDAR_6	32	H'FFFE1164
DMA reload transfer count register_6	RDMATCR_6	32	H'FFFE1168
DMA source address register_7	SAR_7	32	H'FFFE1070
DMA destination address register_7	DAR_7	32	H'FFFE1074
DMA transfer count register_7	DMATCR_7	32	H'FFFE1078
DMA channel control register_7	CHCR_7	32	H'FFFE107C
DMA reload source address register_7	RSAR_7	32	H'FFFE1170
DMA reload destination address register_7	RDAR_7	32	H'FFFE1174
DMA reload transfer count register_7	RDMATCR_7	32	H'FFFE1178
DMA operation register	DMAOR	16	H'FFFE1200
DMA extension resource selector 0	DMARS0	16	H'FFFE1300
DMA extension resource selector 1	DMARS1	16	H'FFFE1304
DMA extension resource selector 2	DMARS2	16	H'FFFE1308
DMA extension resource selector 3	DMARS3	16	H'FFFE130C

Timer general register A_0	TGRA_0	16	H'FFFE4308
Timer general register B_0	TGRB_0	16	H'FFFE430A
Timer general register C_0	TGRC_0	16	H'FFFE430C
Timer general register D_0	TGRD_0	16	H'FFFE430E
Timer general register E_0	TGRE_0	16	H'FFFE4320
Timer general register F_0	TGRF_0	16	H'FFFE4322
Timer interrupt enable register2_0	TIER2_0	8	H'FFFE4324
Timer status register2_0	TSR2_0	8	H'FFFE4325
Timer buffer operation transfer mode register_0	TBTM_0	8	H'FFFE4326
Timer control register_1	TCR_1	8	H'FFFE4380
Timer mode register_1	TMDR_1	8	H'FFFE4381
Timer I/O control register_1	TIOR_1	8	H'FFFE4382
Timer interrupt enable register_1	TIER_1	8	H'FFFE4384
Timer status register_1	TSR_1	8	H'FFFE4385
Timer counter_1	TCNT_1	16	H'FFFE4386
Timer general register A_1	TGRA_1	16	H'FFFE4388
Timer general register B_1	TGRB_1	16	H'FFFE438A
Timer input capture control register	TICCR	8	H'FFFE4390
Timer control register_2	TCR_2	8	H'FFFE4000
Timer mode register_2	TMDR_2	8	H'FFFE4001
Timer I/O control register_2	TIOR_2	8	H'FFFE4002
Timer interrupt enable register_2	TIER_2	8	H'FFFE4004
Timer status register_2	TSR_2	8	H'FFFE4005

Timer interrupt enable register_3	TIER_3	8	H'FFFE4208
Timer status register_3	TSR_3	8	H'FFFE422C
Timer counter_3	TCNT_3	16	H'FFFE4210
Timer general register A_3	TGRA_3	16	H'FFFE4218
Timer general register B_3	TGRB_3	16	H'FFFE421A
Timer general register C_3	TGRC_3	16	H'FFFE4224
Timer general register D_3	TGRD_3	16	H'FFFE4226
Timer buffer operation transfer mode register_3	TBTM_3	8	H'FFFE4238
Timer control register_4	TCR_4	8	H'FFFE4201
Timer mode register_4	TMDR_4	8	H'FFFE4203
Timer I/O control register H_4	TIORH_4	8	H'FFFE4206
Timer I/O control register L_4	TIORL_4	8	H'FFFE4207
Timer interrupt enable register_4	TIER_4	8	H'FFFE4209
Timer status register_4	TSR_4	8	H'FFFE422D
Timer counter_4	TCNT_4	16	H'FFFE4212
Timer general register A_4	TGRA_4	16	H'FFFE421C
Timer general register B_4	TGRB_4	16	H'FFFE421E
Timer general register C_4	TGRC_4	16	H'FFFE4228
Timer general register D_4	TGRD_4	16	H'FFFE422A
Timer buffer operation transfer mode register_4	TBTM_4	8	H'FFFE4239
Timer A/D converter start request control register	TADCR	16	H'FFFE4240

Timer control register V_5	TCRV_5	8	H'FFFE4094
Timer control register W_5	TCRW_5	8	H'FFFE40A4
Timer I/O control register U_5	TIORU_5	8	H'FFFE4086
Timer I/O control register V_5	TIORV_5	8	H'FFFE4096
Timer I/O control register W_5	TIORW_5	8	H'FFFE40A6
Timer interrupt enable register_5	TIER_5	8	H'FFFE40B2
Timer status register_5	TSR_5	8	H'FFFE40B0
Timer start register_5	TSTR_5	8	H'FFFE40B4
Timer counter U_5	TCNTU_5	16	H'FFFE4080
Timer counter V_5	TCNTV_5	16	H'FFFE4090
Timer counter W_5	TCNTW_5	16	H'FFFE40A0
Timer general register U_5	TGRU_5	16	H'FFFE4082
Timer general register V_5	TGRV_5	16	H'FFFE4092
Timer general register W_5	TGRW_5	16	H'FFFE40A2
Timer compare match clear register	TCNTCMPCLR	8	H'FFFE40B6
Timer start register	TSTR	8	H'FFFE4280
Timer synchronous register	TSYR	8	H'FFFE4281
Timer counter synchronous start register	TCSYSTR	8	H'FFFE4282
Timer read/write enable register	TRWER	8	H'FFFE4284
Timer output master enable register	TOER	8	H'FFFE420A
Timer output control register 1	TOCR1	8	H'FFFE420E
Timer output control register 2	TOCR2	8	H'FFFE420F
Timer gate control register	TGCR	8	H'FFFE420D

	Timer dead time enable register	TDER	8	H'FFFE4234
	Timer synchronous clear register	TSYCR	8	H'FFFE4250
	Timer waveform control register	TWCR	8	H'FFFE4260
	Timer output level buffer register	TOLBR	8	H'FFFE4236
MTU2S	Timer control register_3S	TCR_3S	8	H'FFFE4A00
	Timer mode register_3S	TMDR_3S	8	H'FFFE4A02
	Timer I/O control register H_3S	TIORH_3S	8	H'FFFE4A04
	Timer I/O control register L_3S	TIORL_3S	8	H'FFFE4A05
	Timer interrupt enable register_3S	TIER_3S	8	H'FFFE4A08
	Timer status register_3S	TSR_3S	8	H'FFFE4A2C
	Timer counter_3S	TCNT_3S	16	H'FFFE4A10
	Timer general register A_3S	TGRA_3S	16	H'FFFE4A18
	Timer general register B_3S	TGRB_3S	16	H'FFFE4A1A
	Timer general register C_3S	TGRC_3S	16	H'FFFE4A24
	Timer general register D_3S	TGRD_3S	16	H'FFFE4A26
	Timer buffer operation transfer mode register_3S	TBTM_3S	8	H'FFFE4A38
	Timer control register_4S	TCR_4S	8	H'FFFE4A01
	Timer mode register_4S	TMDR_4S	8	H'FFFE4A03
	Timer I/O control register H_4S	TIORH_4S	8	H'FFFE4A06
	Timer I/O control register L_4S	TIORL_4S	8	H'FFFE4A07
	Timer interrupt enable register_4S	TIER_4S	8	H'FFFE4A09
	Timer status register_4S	TSR_4S	8	H'FFFE4A2D
	Timer counter_4S	TCNT_4S	16	H'FFFE4A12

Timer A/D converter start request cycle set register A_4S	TADCORA_4S	16	H'FFFE4A44
Timer A/D converter start request cycle set register B_4S	TADCORB_4S	16	H'FFFE4A46
Timer A/D converter start request cycle set buffer register A_4S	TADCOBRA_4S	16	H'FFFE4A48
Timer A/D converter start request cycle set buffer register B_4S	TADCOBRB_4S	16	H'FFFE4A4A
Timer control register U_5S	TCRU_5S	8	H'FFFE4884
Timer control register V_5S	TCRV_5S	8	H'FFFE4894
Timer control register W_5S	TCRW_5S	8	H'FFFE48A4
Timer I/O control register U_5S	TIORU_5S	8	H'FFFE4886
Timer I/O control register V_5S	TIORV_5S	8	H'FFFE4896
Timer I/O control register W_5S	TIORW_5S	8	H'FFFE48A6
Timer interrupt enable register_5S	TIER_5S	8	H'FFFE48B2
Timer status register_5S	TSR_5S	8	H'FFFE48B0
Timer start register_5S	TSTR_5S	8	H'FFFE48B4
Timer counter U_5S	TCNTU_5S	16	H'FFFE4880
Timer counter V_5S	TCNTV_5S	16	H'FFFE4890
Timer counter W_5S	TCNTW_5S	16	H'FFFE48A0
Timer general register U_5S	TGRU_5S	16	H'FFFE4882
Timer general register V_5S	TGRV_5S	16	H'FFFE4892
Timer general register W_5S	TGRW_5S	16	H'FFFE48A2
Timer compare match clear register S	TCNTCMPCLRS	8	H'FFFE48B6
Timer start register S	TSTRS	8	H'FFFE4A80

	Timer dead time data register S	TDDRS	16	H'FFFE4A16
	Timer subcounter S	TCNTSS	16	H'FFFE4A20
	Timer cycle buffer register S	TCBRS	16	H'FFFE4A22
	Timer interrupt skipping set register S	TITCRS	8	H'FFFE4A30
	Timer interrupt skipping counter S	TITCNTS	8	H'FFFE4A31
	Timer buffer transfer set register S	TBTERS	8	H'FFFE4A32
	Timer dead time enable register S	TDERS	8	H'FFFE4A34
	Timer synchronous clear register S	TSYCRS	8	H'FFFE4A50
	Timer waveform control register S	TWCRS	8	H'FFFE4A60
	Timer output level buffer register S	TOLBRS	8	H'FFFE4A36
POE2	Input level control/status register 1	ICSR1	16	H'FFFE5000
	Output level control/status register 1	OCSR1	16	H'FFFE5002
	Input level control/status register 2	ICSR2	16	H'FFFE5004
	Output level control/status register 2	OCSR2	16	H'FFFE5006
	Input level control/status register 3	ICSR3	16	H'FFFE5008
	Software port output enable register	SPOER	8	H'FFFE500A
	Port output enable control register 1	POECR1	8	H'FFFE500B
	Port output enable control register 2	POECR2	16	H'FFFE500C
CMT	Compare match timer start register	CMSTR	16	H'FFFEC000
	Compare match timer control/status register_0	CMCSR_0	16	H'FFFEC002
	Compare match counter_0	CMCNT_0	16	H'FFFEC004
	Compare match constant register_0	CMCOR_0	16	H'FFFEC006

Bit rate register_0	SCBRR_0	8	H'FFFE8004
Serial control register_0	SCSCR_0	16	H'FFFE8008
Transmit FIFO data register_0	SCFTDR_0	8	H'FFFE800C
Serial status register_0	SCFSR_0	16	H'FFFE8010
Receive FIFO data register_0	SCFRDR_0	8	H'FFFE8014
FIFO control register_0	SCFCR_0	16	H'FFFE8018
FIFO data count register_0	SCFDR_0	16	H'FFFE801C
Serial port register_0	SCSPTR_0	16	H'FFFE8020
Line status register_0	SCLSR_0	16	H'FFFE8024
Serial mode register_1	SCSMR_1	16	H'FFFE8800
Bit rate register_1	SCBRR_1	8	H'FFFE8804
Serial control register_1	SCSCR_1	16	H'FFFE8808
Transmit FIFO data register_1	SCFTDR_1	8	H'FFFE880C
Serial status register_1	SCFSR_1	16	H'FFFE8810
Receive FIFO data register_1	SCFRDR_1	8	H'FFFE8814
FIFO control register_1	SCFCR_1	16	H'FFFE8818
FIFO data count register_1	SCFDR_1	16	H'FFFE881C
Serial port register_1	SCSPTR_1	16	H'FFFE8820
Line status register_1	SCLSR_1	16	H'FFFE8824
Serial extended mode register_1	SCSEMR_1	8	H'FFFE8900
Serial mode register_2	SCSMR_2	16	H'FFFE9000
Bit rate register_2	SCBRR_2	8	H'FFFE9004
Serial control register_2	SCSCR_2	16	H'FFFE9008

	Serial extended mode register_2	SCSEMR_2	8	H'FFFE9F00
	Serial mode register_3	SCSMR_3	16	H'FFFE9800
	Bit rate register_3	SCBRR_3	8	H'FFFE9804
	Serial control register_3	SCSCR_3	16	H'FFFE9808
	Transmit FIFO data register_3	SCFTDR_3	8	H'FFFE980C
	Serial status register_3	SCFSR_3	16	H'FFFE9810
	Receive FIFO data register_3	SCFRDR_3	8	H'FFFE9814
	FIFO control register_3	SCFCR_3	16	H'FFFE9818
	FIFO data count register_3	SCFDR_3	16	H'FFFE981C
	Serial port register_3	SCSPTR_3	16	H'FFFE9820
	Line status register_3	SCLSR_3	16	H'FFFE9824
IIC3	I^2C bus control register 1	ICCR1	8	H'FFFEE000
	I^2C bus control register 2	ICCR2	8	H'FFFEE001
	I^2C bus mode register	ICMR	8	H'FFFEE002
	I^2C bus interrupt enable register	ICIER	8	H'FFFEE003
	I^2C bus status register	ICSR	8	H'FFFEE004
	Slave address register	SAR	8	H'FFFEE005
	I^2C bus transmit data register	ICDRT	8	H'FFFEE006
	I^2C bus receive data register	ICDRR	8	H'FFFEE007
	NF2CYC register	NF2CYC	8	H'FFFEE008
ADC	A/D control register	ADCR	16	H'FFFFE800
	A/D status register	ADSR	16	H'FFFFE802
	A/D start trigger select register	ADSTRGR	8	H'FFFFE81C
	A/D analog input channel select register	ADANSR	8	H'FFFFE820

	A/D data register 7	ADDR7	16	H'FFFE384E
DAC	D/A data register 0	DADR0	8	H'FFFE6800
	D/A data register 1	DADR1	8	H'FFFE6801
	D/A control register	DACR	8	H'FFFE6802
PFC	Port A I/O register H	PAIORH	16	H'FFFE3804
	Port A I/O register L	PAIORL	16	H'FFFE3806
	Port A control register H3	PACRH3	16	H'FFFE380A
	Port A control register H2	PACRH2	16	H'FFFE380C
	Port A control register H1	PACRH1	16	H'FFFE380E
	Port A control register L4	PACRL4	16	H'FFFE3810
	Port A control register L3	PACRL3	16	H'FFFE3812
	Port A control register L2	PACRL2	16	H'FFFE3814
	Port A control register L1	PACRL1	16	H'FFFE3816
	Port B I/O register H	PBIORH	16	H'FFFE3884
	Port B I/O register L	PBIORL	16	H'FFFE3886
	Port B control register H4	PBCRH4	16	H'FFFE3888
	Port B control register H3	PBCRH3	16	H'FFFE388A
	Port B control register H2	PBCRH2	16	H'FFFE388C
	Port B control register H1	PBCRH1	16	H'FFFE388E
	Port B control register L4	PBCRL4	16	H'FFFE3890
	Port B control register L3	PBCRL3	16	H'FFFE3892
	Port B control register L2	PBCRL2	16	H'FFFE3894
	Port B control register L1	PBCRL1	16	H'FFFE3896

	WAVE function control register 2	WAVECR2	16	H'FFFE3A14
	WAVE function control register 1	WAVECR1	16	H'FFFE3A16
I/O port	Port A data register H	PADRH	16	H'FFFE3800
	Port A data register L	PADRL	16	H'FFFE3802
	Port A port register H	PAPRH	16	H'FFFE381C
	Port A port register L	PAPRL	16	H'FFFE381E
	Port B data register H	PBDRH	16	H'FFFE3880
	Port B data register L	PBDRL	16	H'FFFE3882
	Port B port register H	PBPRH	16	H'FFFE389C
	Port B port register L	PBPRL	16	H'FFFE389E
	Port D data register L	PDDRL	16	H'FFFE3982
	Port D port register L	PDPRL	16	H'FFFE399E
	Port F data register	PFDR	16	H'FFFE3A82
FLASH	Flash code control and status register	FCCS	8	H'8000C000
	Flash program code select register	FPCS	8	H'8000C001
	Flash erase code select register	FECS	8	H'8000C002
	Flash key code register	FKEY	8	H'8000C004
	Flash MAT select register	FMATS	8	H'8000C005
	Flash transfer destination address register	FTDAR	8	H'8000C006

Note: The access sizes of the WDT registers are different between the read and writ
prevent incorrect writing.

		—	—	—	—	—	—	—
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S
	IRQRR	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F
	IBCR	E15	E14	E13	E12	E11	E10	E9
		E7	E6	E5	E4	E3	E2	E1
	IBNR	BE[1:0]		BOVE	—	—	—	—
		—	—	—	—	BN[3:0]		
	IPR01	IRQ0[3:0]				IRQ1[3:0]		
		IRQ2[3:0]				IRQ3[3:0]		
	IPR02	IRQ4[3:0]				IRQ5[3:0]		
		IRQ6[3:0]				IRQ7[3:0]		
	IPR05	—	—	—	—	—	—	—
		ADI[3:0]				—	—	—
	IPR06	DMAC0[3:0]				DMAC1[3:0]		
		DMAC2[3:0]				DMAC3[3:0]		
	IPR07	DMAC4[3:0]				DMAC5[3:0]		
		DMAC6[3:0]				DMAC7[3:0]		
	IPR08	CMT0[3:0]				CMT1[3:0]		
		BSC[3:0]				WDT[3:0]		
	IPR09	MTU0(TGI0A to TGI0D)[3:0]				MTU0(TCI0V, TGI0E, TGI0		
		MTU1(TGI1A, TGI1B)[3:0]				MTU1(TCI1V, TCI1U)[3		

		IIC3[3:0]				—	—	—
	IPR14	SCIF0[3:0]				SCIF1[3:0]		
		SCIF2[3:0]				SCIF3[3:0]		
	IPR15	WAVEIF[3:0]				—	—	—
		—	—	—	—	—	—	—
UBC	BAR_0	BA0_31	BA0_30	BA0_29	BA0_28	BA0_27	BA0_26	BA0_25
		BA0_23	BA0_22	BA0_21	BA0_20	BA0_19	BA0_18	BA0_17
		BA0_15	BA0_14	BA0_13	BA0_12	BA0_11	BA0_10	BA0_9
		BA0_7	BA0_6	BA0_5	BA0_4	BA0_3	BA0_2	BA0_1
	BAMR_0	BAM0_31	BAM0_30	BAM0_29	BAM0_28	BAM0_27	BAM0_26	BAM0_25
		BAM0_23	BAM0_22	BAM0_21	BAM0_20	BAM0_19	BAM0_18	BAM0_17
		BAM0_15	BAM0_14	BAM0_13	BAM0_12	BAM0_11	BAM0_10	BAM0_9
		BAM0_7	BAM0_6	BAM0_5	BAM0_4	BAM0_3	BAM0_2	BAM0_1
	BBR_0	—	—	UBID0	—	—	—	CP
		CD0[1:0]		ID0[1:0]		RW0[1:0]		SZ
	BAR_1	BA1_31	BA1_30	BA1_29	BA1_28	BA1_27	BA1_26	BA1_25
		BA1_23	BA1_22	BA1_21	BA1_20	BA1_19	BA1_18	BA1_17
		BA1_15	BA1_14	BA1_13	BA1_12	BA1_11	BA1_10	BA1_9
		BA1_7	BA1_6	BA1_5	BA1_4	BA1_3	BA1_2	BA1_1
	BAMR_1	BAM1_31	BAM1_30	BAM1_29	BAM1_28	BAM1_27	BAM1_26	BAM1_25
		BAM1_23	BAM1_22	BAM1_21	BAM1_20	BAM1_19	BAM1_18	BAM1_17
		BAM1_15	BAM1_14	BAM1_13	BAM1_12	BAM1_11	BAM1_10	BAM1_9
		BAM1_7	BAM1_6	BAM1_5	BAM1_4	BAM1_3	BAM1_2	BAM1_1

		BAM2_23	BAM2_22	BAM2_21	BAM2_20	BAM2_19	BAM2_18	BAM2_1
		BAM2_15	BAM2_14	BAM2_13	BAM2_12	BAM2_11	BAM2_10	BAM2_9
		BAM2_7	BAM2_6	BAM2_5	BAM2_4	BAM2_3	BAM2_2	BAM2_
	BBR_2	—	—	UBID2	—	—	—	C
		CD2[1:0]		ID2[1:0]		RW2[1:0]		S
	BAR_3	BA3_31	BA3_30	BA3_29	BA3_28	BA3_27	BA3_26	BA3_25
		BA3_23	BA3_22	BA3_21	BA3_20	BA3_19	BA3_18	BA3_17
		BA3_15	BA3_14	BA3_13	BA3_12	BA3_11	BA3_10	BA3_9
		BA3_7	BA3_6	BA3_5	BA3_4	BA3_3	BA3_2	BA3_1
	BAMR_3	BAM3_31	BAM3_30	BAM3_29	BAM3_28	BAM3_27	BAM3_26	BAM3_2
		BAM3_23	BAM3_22	BAM3_21	BAM3_20	BAM3_19	BAM3_18	BAM3_1
		BAM3_15	BAM3_14	BAM3_13	BAM3_12	BAM3_11	BAM3_10	BAM3_9
		BAM3_7	BAM3_6	BAM3_5	BAM3_4	BAM3_3	BAM3_2	BAM3_
	BBR_3	—	—	UBID3	—	—	—	C
		CD3[1:0]		ID3[1:0]		RW3[1:0]		S
	BRCR	—	—	—	—	—	—	—
		—	—	—	—	—	—	C
		SCMFC0	SCMFC1	SCMFC2	SCMFC3	SCMFD0	SCMFD1	SCMFD
		PCB3	PCB2	PCB1	PCB0	—	—	—

		—	—	—	—	—	—	—
	CS1BCR	—	IWW[2:0]			IWRWD[2:0]		
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0	
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]	
		—	—	—	—	—	—	—
	CS2BCR	—	IWW[2:0]			IWRWD[2:0]		
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0	
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]	
		—	—	—	—	—	—	—
	CS3BCR	—	IWW[2:0]			IWRWD[2:0]		
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0	
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]	
		—	—	—	—	—	—	—
	CS4BCR	—	IWW[2:0]			IWRWD[2:0]		
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0	
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]	
		—	—	—	—	—	—	—
	CS5BCR	—	IWW[2:0]			IWRWD[2:0]		
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0	
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]	
		—	—	—	—	—	—	—

		—	—	—	—	—	—	—
	CS0WCR*1	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—
		—	—	—	SW[1:0]			WR[3:1]
		WR[0]	WM	—	—	—	—	H
	CS0WCR*2	—	—	—	—	—	—	—
		—	—	BST[1:0]		—	—	B
		—	—	—	—	—	W[3:1]	
		W[0]	WM	—	—	—	—	—
	CS0WCR*4	—	—	—	—	—	—	—
		—	—	—	—	—	—	B
		—	—	—	—	—	W[3:1]	
		W[0]	WM	—	—	—	—	—
	CS1WCR*1	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0	
		—	—	—	SW[1:0]			WR[3:1]
		WR[0]	WM	—	—	—	—	H
	CS2WCR*1	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—
		—	—	—	—	—	WR[3:1]	
		WR[0]	WM	—	—	—	—	—

		WR[0]	WM	—	—	—	—	—
	CS3WCR*3	—	—	—	—	—	—	—
		—	—	—	—	—	—	—
		—	WTRP[1:0]		—	WTRCD[1:0]		—
		A3CL[0]	—	—	TRWL[1:0]		—	WTF
	CS4WCR*1	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]	
		—	—	—	SW[1:0]		WR[3:1]	
		WR[0]	WM	—	—	—	—	HW
	CS4WCR*2	—	—	—	—	—	—	—
		—	—	BST[1:0]		—	—	BW
		—	—	—	SW[1:0]		W[3:1]	
		W[0]	WM	—	—	—	—	HW
	CS5WCR*1	—	—	—	—	—	—	—
		—	—	SZSEL	MPXW/BAS	—	WW[2:0]	
		—	—	—	SW[1:0]		WR[3:1]	
		WR[0]	WM	—	—	—	—	HW
	CS6WCR*1	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—
		—	—	—	SW[1:0]		WR[3:1]	
		WR[0]	WM	—	—	—	—	HW

		—	—	—	A3ROW[1:0]		—	A3
	RTCSR	—	—	—	—	—	—	—
		—	—	—	—	—	—	—
		—	—	—	—	—	—	—
		CMF	CMIE	CKS[2:0]			RRC[2:0	
	RTCNT	—	—	—	—	—	—	—
		—	—	—	—	—	—	—
		—	—	—	—	—	—	—
	RTCOR	—	—	—	—	—	—	—
		—	—	—	—	—	—	—
		—	—	—	—	—	—	—
DMAC	SAR_0							
	DAR_0							

		DL	DS	TB	TS[1:0]		IE	TE
	RSAR_0							
	RDAR_0							
	RDMATCR_0							
	SAR_1							
	DAR1							

		DL	DS	TB	TS[1:0]		IE	TE
	RSAR_1							
	RDAR_1							
	RDMATCR_1	—	—	—	—	—	—	—
	SAR_2							
	DAR_2							

		DL	DS	TB	TS[1:0]		IE	TE
	RSAR_2							
	RDAR_2							
	RDMATCR_2	—	—	—	—	—	—	—
	SAR_3							
	DAR_3							

		DL	DS	TB	TS[1:0]		IE	TE
	RSAR_3							
	RDAR_3							
	RDMATCR_3	—	—	—	—	—	—	—
	SAR_4							
	DAR_4							

		—	—	TB	TS[1:0]		IE	TE
	RSAR_4							
	RDAR_4							
	RDMATCR_4	—	—	—	—	—	—	—
	SAR_5							
	DAR_5							

		—	—	TB	TS[1:0]		IE	TE
	RSAR_5							
	RDAR_5							
	RDMATCR_5	—	—	—	—	—	—	—
	SAR_6							
	DAR_6							

		—	—	TB	TS[1:0]		IE	TE
	RSAR_6							
	RDAR_6							
	RDMATCR_6	—	—	—	—	—	—	—
	SAR_7							
	DAR_7							

		—	—	TB	TS[1:0]		IE	TE
	RSAR_7							
	RDAR_7							
	RDMATCR_7	—	—	—	—	—	—	—
	DMAOR	—	—	CMS[1:0]		—	—	P
		—	—	—	—	—	AE	NMIF
	DMARS0	CH1 MID[5:0]						CH1
		CH0 MID[5:0]						CH0
	DMARS1	CH3 MID[5:0]						CH3
		CH2 MID[5:0]						CH2
	DMARS2	CH5 MID[5:0]						CH5
		CH4 MID[5:0]						CH4
	DMARS3	CH7 MID[5:0]						CH7
		CH6 MID[5:0]						CH6

	TGRA_0						
	TGRB_0						
	TGRC_0						
	TGRD_0						
	TGRE_0						
	TGRF_0						
	TIER2_0	TTGE2	—	—	—	—	—	TGIEF
	TSR2_0	—	—	—	—	—	—	TGFF
	TBTM_0	—	—	—	—	—	TTSE	TTSB
	TCR_1	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]	
	TMDR_1	—	—	—	—	MD[3:0]		
	TIOR_1	IOB[3:0]				IOA[3:0]		
	TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB
	TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB
	TCNT_1							
	TGRA_1							
	TGRB_1							
	TICCR	—	—	—	—	I2BE	I2AE	I1BE
	TCR_2	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]	
	TMDR_2	—	—	—	—	MD[3:0]		
	TIOR_2	IOB[3:0]				IOA[3:0]		
	TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB
	TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB
	TCNT_2							

	TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB
	TCNT_3							
	TGRA_3							
	TGRB_3							
	TGRC_3							
	TGRD_3							
	TBTM_3	—	—	—	—	—	—	TTSB
	TCR_4	CCLR[2:0]			CKEG[1:0]		TPSC[2:	
	TMDR_4	—	—	BFB	BFA	MD[3:0]		
	TIORH_4	IOB[3:0]				IOA[3:0]		
	TIORL_4	IOD[3:0]				IOC[3:0]		
	TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB
	TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB
	TCNT_4							
	TGRA_4							
	TGRB_4							
	TGRC_4							
	TGRD_4							
	TBTM_4	—	—	—	—	—	—	TTSB
	TADCR	BF[1:0]		—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE
	TADCORA_4							
	TADCORB_4							
	TADCOBRA_4							

	TIER_5	—	—	—	—	—	TGIE5U	TGIE5V
	TSR_5	—	—	—	—	—	CMFU5	CMFV5
	TSTR_5	—	—	—	—	—	CSTU5	CSTV5
	TCNTU_5							
	TCNTV_5							
	TCNTW_5							
	TGRU_5							
	TGRV_5							
	TGRW_5							
	TCNTCMPCLR	—	—	—	—	—	CMPCLR 5U	CMPCLR 5V
	TSTR	CST4	CST3	—	—	—	CST2	CST1
	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1
	TCSYSTR	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH3S
	TRWER	—	—	—	—	—	—	—
	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A
	TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN
	TOCR2	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N
	TGCR	—	BDC	N	P	FB	WF	VF
	TCDR							
	TDDR							

	TDER	—	—	—	—	—	—	—
	TSYCR	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A
	TWCR	CCE	—	—	—	—	—	—
	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N
MTU2S	TCR_3S	CCLR[2:0]			CKEG[1:0]		TPSC[2:	
	TMDR_3S	—	—	BFB	BFA	MD[3:0]		
	TIORH_3S	IOB[3:0]				IOA[3:0]		
	TIORL_3S	IOD[3:0]				IOC[3:0]		
	TIER_3S	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB
	TSR_3S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB
	TCNT_3S							
	TGRA_3S							
	TGRB_3S							
	TGRC_3S							
	TGRD_3S							
	TBTM_3S	—	—	—	—	—	—	TTSB
	TCR_4S	CCLR[2:0]			CKEG[1:0]		TPSC[2:	
	TMDR_4S	—	—	BFB	BFA	MD[3:0]		
	TIORH_4S	IOB[3:0]				IOA[3:0]		
	TIORL_4S	IOD[3:0]				IOC[3:0]		
	TIER_4S	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB
	TSR_4S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB
	TCNT_4S							
	TGRA_4S							

	TADCORB_4S						
	TADCOBRA_4S						
	TADCOBRB_4S						
	TCRU_5S	—	—	—	—	—	—	TPS
	TCRV_5S	—	—	—	—	—	—	TPS
	TCRW_5S	—	—	—	—	—	—	TPS
	TIORU_5S	—	—	—	IOC[4:0]			
	TIORV_5S	—	—	—	IOC[4:0]			
	TIORW_5S	—	—	—	IOC[4:0]			
	TIER_5S	—	—	—	—	—	TGIE5U	TGIE5V
	TSR_5S	—	—	—	—	—	CMFU5	CMFV5
	TSTR_5S	—	—	—	—	—	CSTU5	CSTV5
	TCNTU_5S							
	TCNTV_5S							
	TCNTW_5S							
	TGRU_5S							
	TGRV_5S							
	TGRW_5S							
	TCNTCMPCLRS	—	—	—	—	—	CMPCLR5U	CMPCLR5V
	TSTRS	CST4	CST3	—	—	—	CST2	CST1
	TSYRS	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1
	TRWERS	—	—	—	—	—	—	—
	TOERS	—	—	OE4D	OE4C	OE3D	OE4B	OE4A

	TCNTSS							
	TCBRS							
	TITCRS	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2	
	TITCNTS	—	3ACNT[2:0]			—	4VCNT[2	
	TBTERS	—	—	—	—	—	—	B
	TDERS	—	—	—	—	—	—	—
	TSYCRS	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A
	TWCRS	CCE	—	—	—	—	—	SCC
	TOLBRS	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N
POE2	ICSR1	POE3F	—	POE1F	POE0F	—	—	—
		POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		PO
	OCSR1	OSF1	—	—	—	—	—	OCE1
		—	—	—	—	—	—	—
	ICSR2	POE7F	—	—	POE4F	—	—	—
		POE7M[1:0]		—	—	—	—	PO
	OCSR2	OSF2	—	—	—	—	—	OCE2
		—	—	—	—	—	—	—
	ICSR3	—	—	—	POE8F	—	—	POE8E
		—	—	—	—	—	—	PO
	SPOER	—	—	—	—	—	MTU2S HIZ	MTU2 CH0HIZ

		CMF	CMIE	—	—	—	—	CK
	CMCNT_0							
	CMCOR_0							
	CMCSR_1	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CK
	CMCNT_1							
	CMCOR_1							
WDT	WTCSR	IOVF	WT/$\overline{IT}$	TME	—	—	CKS[2:0]	
	WTCNT							
	WRCSR	WOVF	RSTE	RSTS	—	—	—	—
SCIF	SCSMR_0	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CK
	SCBRR_0							
	SCSCR_0	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CK
	SCFTDR_0	—	—	—	—	—	—	—
	SCFSR_0	PER[3:0]				FER[3:0]		
		ER	TEND	TDFE	BRK	FER	PER	RDF
	SCFRDR_0							

		—	—	—	—	—	—	—
	SCSMR_1	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	C
	SCBRR_1							
	SCSCR_1	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	C
	SCFTDR_1	—	—	—	—	—	—	—
	SCFSR_1	PER[3:0]				FER[3:0]		
		ER	TEND	TDFE	BRK	FER	PER	RDF
	SCFRDR_1							
	SCFCR_1	—	—	—	—	—	—	—
		RTRG[1:0]		TTRG[1:0]		—	TFRST	RFRST
	SCFDR_1	—	—	—	T[4:0]			
		—	—	—	R[4:0]			
	SCSPTR_1	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO
	SCLSR_1	—	—	—	—	—	—	—
		—	—	—	—	—	—	—
	SCSEMR_1	ABCS	—	—	—	—	—	—
	SCSMR_2	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	C
	SCBRR_2							
	SCSCR_2	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	C

		—	—	—	R[4:0]			
	SCSPTR_2	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO
	SCLSR_2	—	—	—	—	—	—	—
		—	—	—	—	—	—	—
	SCSEMR_2	ABCS	—	—	—	—	—	—
	SCSMR_3	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CK
	SCBRR_3							
	SCSCR_3	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CK
	SCFTDR_3							
	SCFSR_3	PER[3:0]				FER[3:0]		
		ER	TEND	TDFE	BRK	FER	PER	RDF
	SCFRDR_3	—	—	—	—	—	—	—
	SCFCR_3	—	—	—	—	—	—	—
		RTRG[1:0]		TTRG[1:0]		—	TFRST	RFRST
	SCFDR_3	—	—	—	T[4:0]			
		—	—	—	R[4:0]			
	SCSPTR_3	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO
	SCLSR_3	—	—	—	—	—	—	—
		—	—	—	—	—	—	—

	ICDRR							
	NF2CYC	—	—	—	—	—	—	—
ADC	ADCR	ADST	ADCS	ACE	ADIE	—	—	TRGE
	ADSR	—	—	—	—	—	—	—
	ADSTRGR	—	STR6	STR5	STR4	STR3	STR2	STR1
	ADANSR	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1
	ADDR0	—	—	—	—	ADD0[11:8]		
		ADD0[7:0]						
	ADDR1	—	—	—	—	ADD1[11:8]		
		ADD1[7:0]						
	ADDR2	—	—	—	—	ADD2[11:8]		
		ADD2[7:0]						
	ADDR3	—	—	—	—	ADD3[11:8]		
		ADD3[7:0]						
	ADDR4	—	—	—	—	ADD4[11:8]		
		ADD4[7:0]						
	ADDR5	—	—	—	—	ADD5[11:8]		
		ADD5[7:0]						
	ADDR6	—	—	—	—	ADD6[11:8]		
		ADD6[7:0]						
	ADDR7	—	—	—	—	ADD7[11:8]		
		ADD7[7:0]						

	PACRH3	—	—	—	—	—	—	—
		—	PA25MD[2:0]			—	PA24MD[2:0]	
	PACRH2	—	PA23MD[2:0]			—	PA22MD[2:0]	
		—	PA21MD[2:0]			—	PA20MD[2:0]	
	PACRH1	—	PA19MD[2:0]			—	PA18MD[2:0]	
		—	PA17MD[2:0]			—	PA16MD[2:0]	
	PACRL4	—	PA15MD[2:0]			—	PA14MD[2:0]	
		—	PA13MD[2:0]			—	PA12MD[2:0]	
	PACRL3	—	PA11MD[2:0]			—	PA10MD[2:0]	
		—	PA9MD[2:0]			—	PA8MD[2:0]	
	PACRL2	—	PA7MD[2:0]			—	PA6MD[2:0]	
		—	PA5MD[2:0]			—	PA4MD[2:0]	
	PACRL1	—	PA3MD[2:0]			—	PA2MD[2:0]	
		—	PA1MD[2:0]			—	PA0MD[2:0]	
	PBIORH	—	PB30IOR	PB29IOR	PB28IOR	PB27IOR	PB26IOR	PB25IOR
		PB23IOR	PB22IOR	PB21IOR	PB20IOR	PB19IOR	PB18IOR	PB17IOR
	PBIORL	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR
		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR
	PBCRH4	—	—	—	—	—	PB30MD[2:0]	
		—	PB29MD[2:0]			—	PB28MD[2:0]	
	PBCRH3	—	PB27MD[2:0]			—	PB26MD[2:0]	
		—	PB25MD[2:0]			—	PB24MD[2:0]	

		—	PB9MD[2:0]			—	PB8MD[2	
	PBCRL2	—	PB7MD[2:0]			—	PB6MD[2	
		—	PB5MD[2:0]			—	PB4MD[2	
	PBCRL1	—	PB3MD[2:0]			—	PB2MD[2	
		—	PB1MD[2:0]			—	PB0MD[2	
	PDIOR	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR
	PDCRL4	—	PD15MD[2:0]			—	PD14MD[2	
		—	PD13MD[2:0]			—	PD12MD[2	
	PDCRL3	—	PD11MD[2:0]			—	PD10MD[2	
		—	PD9MD[2:0]			—	PD8MD[2	
	PDCRL2	—	PD7MD[2:0]			—	PD6MD[2	
		—	PD5MD[2:0]			—	PD4MD[2	
	PDCRL1	—	PD3MD[2:0]			—	PD2MD[2	
		—	PD1MD[2:0]			—	PD0MD[2	
	PFCRL1	—	—	—	—	—	—	—
		—	PF1MD[2:0]			—	PF0MD[2	
	IFCR	—	—	—	—	—	—	—
		—	—	—	—	—	—	IRQ
	WAVECR2	—	—	—	—	—	—	—
		—	—	—	—	—	WVRMD[2	
	WAVECR1	—	WVTMD[2:0]			—	WVSMD[2	
		—	—	—	—	—	—	—

		PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR
	PBDRH	—	PB30DR	PB29DR	PB28DR	PB27DR	PB26DR	PB25DR
		PB23DR	PB22DR	PB21DR	PB20DR	PB19DR	PB18DR	PB17DR
	PBDRL	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR
	PBPRH	—	PB30PR	PB29PR	PB28PR	PB27PR	PB26PR	PB25PR
		PB23PR	PB22PR	PB21PR	PB20PR	PB19PR	PB18PR	PB17PR
	PBPRL	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR
	PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR
	PDPRL	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR
		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR
	PFDR	—	—	—	—	—	—	—
		—	—	—	—	—	—	PF1DR
FLASH	FCCS	FWE	MAT	—	FLER	—	—	—
	FPCS	—	—	—	—	—	—	—
	FECS	—	—	—	—	—	—	—
	FKEY	K[7:0]						
	FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1
	FTDAR	TDER	TDA[6:0]					

		—	—	—	—	—	—	—

Notes: 1. When normal memory, SRAM with byte selection, or MPX-I/O is the memory
2. When burst ROM (clocked asynchronous) is the memory type
3. When SDRAM is the memory type
4. When burst ROM (clocked synchronous) is the memory type

	IRQRR	Initialized	Retained	Retained	—	Ret
	IBCR	Initialized	Retained	Retained	—	Ret
	IBNR	Initialized	Retained*[2]	Retained	—	Ret
	IPR01	Initialized	Retained	Retained	—	Ret
	IPR02	Initialized	Retained	Retained	—	Ret
	IPR05	Initialized	Retained	Retained	—	Ret
	IPR06	Initialized	Retained	Retained	—	Ret
	IPR07	Initialized	Retained	Retained	—	Ret
	IPR08	Initialized	Retained	Retained	—	Ret
	IPR09	Initialized	Retained	Retained	—	Ret
	IPR10	Initialized	Retained	Retained	—	Ret
	IPR11	Initialized	Retained	Retained	—	Ret
	IPR12	Initialized	Retained	Retained	—	Ret
	IPR13	Initialized	Retained	Retained	—	Ret
	IPR14	Initialized	Retained	Retained	—	Ret
	IPR15	Initialized	Retained	Retained	—	Ret
UBC	BAR_0	Initialized	Retained	Retained	Retained	Ret
	BAMR_0	Initialized	Retained	Retained	Retained	Ret
	BBR_0	Initialized	Retained	Retained	Retained	Ret
	BAR_1	Initialized	Retained	Retained	Retained	Ret
	BAMR_1	Initialized	Retained	Retained	Retained	Ret
	BBR_1	Initialized	Retained	Retained	Retained	Ret
	BAR_2	Initialized	Retained	Retained	Retained	Ret
	BAMR_2	Initialized	Retained	Retained	Retained	Ret

CS1BCR	Initialized	Retained	Retained	—	Re
CS2BCR	Initialized	Retained	Retained	—	Re
CS3BCR	Initialized	Retained	Retained	—	Re
CS4BCR	Initialized	Retained	Retained	—	Re
CS5BCR	Initialized	Retained	Retained	—	Re
CS6BCR	Initialized	Retained	Retained	—	Re
CS7BCR	Initialized	Retained	Retained	—	Re
CS0WCR	Initialized	Retained	Retained	—	Re
CS1WCR	Initialized	Retained	Retained	—	Re
CS2WCR	Initialized	Retained	Retained	—	Re
CS3WCR	Initialized	Retained	Retained	—	Re
CS4WCR	Initialized	Retained	Retained	—	Re
CS5WCR	Initialized	Retained	Retained	—	Re
CS6WCR	Initialized	Retained	Retained	—	Re
CS7WCR	Initialized	Retained	Retained	—	Re
SDCR	Initialized	Retained	Retained	—	Re
RTCSR	Initialized	Retained (Flag processing continued)	Retained	—	Re (F pr co
RTCNT	Initialized	Retained (Count-up continued)	Retained	—	Re (C co
RTCOR	Initialized	Retained	Retained	—	Re

DAR_1	Initialized	Retained	Retained	Retained	Ret
DMATCR_1	Initialized	Retained	Retained	Retained	Ret
CHCR_1	Initialized	Retained	Retained	Retained	Ret
RSAR_1	Initialized	Retained	Retained	Retained	Ret
RDAR_1	Initialized	Retained	Retained	Retained	Ret
RDMATCR_1	Initialized	Retained	Retained	Retained	Ret
SAR_2	Initialized	Retained	Retained	Retained	Ret
DAR_2	Initialized	Retained	Retained	Retained	Ret
DMATCR_2	Initialized	Retained	Retained	Retained	Ret
CHCR_2	Initialized	Retained	Retained	Retained	Ret
RSAR_2	Initialized	Retained	Retained	Retained	Ret
RDAR_2	Initialized	Retained	Retained	Retained	Ret
RDMATCR_2	Initialized	Retained	Retained	Retained	Ret
SAR_3	Initialized	Retained	Retained	Retained	Ret
DAR_3	Initialized	Retained	Retained	Retained	Ret
DMATCR_3	Initialized	Retained	Retained	Retained	Ret
CHCR_3	Initialized	Retained	Retained	Retained	Ret
RSAR_3	Initialized	Retained	Retained	Retained	Ret
RDAR_3	Initialized	Retained	Retained	Retained	Ret
RDMATCR_3	Initialized	Retained	Retained	Retained	Ret
SAR_4	Initialized	Retained	Retained	Retained	Ret
DAR_4	Initialized	Retained	Retained	Retained	Ret
DMATCR_4	Initialized	Retained	Retained	Retained	Ret

CHCR_5	Initialized	Retained	Retained	Retained	Re
RSAR_5	Initialized	Retained	Retained	Retained	Re
RDAR_5	Initialized	Retained	Retained	Retained	Re
RDMATCR_5	Initialized	Retained	Retained	Retained	Re
SAR_6	Initialized	Retained	Retained	Retained	Re
DAR_6	Initialized	Retained	Retained	Retained	Re
DMATCR_6	Initialized	Retained	Retained	Retained	Re
CHCR_6	Initialized	Retained	Retained	Retained	Re
RSAR_6	Initialized	Retained	Retained	Retained	Re
RDAR_6	Initialized	Retained	Retained	Retained	Re
RDMATCR_6	Initialized	Retained	Retained	Retained	Re
SAR_7	Initialized	Retained	Retained	Retained	Re
DAR_7	Initialized	Retained	Retained	Retained	Re
DMATCR_7	Initialized	Retained	Retained	Retained	Re
CHCR_7	Initialized	Retained	Retained	Retained	Re
RSAR_7	Initialized	Retained	Retained	Retained	Re
RDAR_7	Initialized	Retained	Retained	Retained	Re
RDMATCR_7	Initialized	Retained	Retained	Retained	Re
DMAOR	Initialized	Retained	Retained	Retained	Re
DMARS0	Initialized	Retained	Retained	Retained	Re
DMARS1	Initialized	Retained	Retained	Retained	Re
DMARS2	Initialized	Retained	Retained	Retained	Re
DMARS3	Initialized	Retained	Retained	Retained	Re

TGRA_0	Initialized	Retained	Retained	Initialized	Ret
TGRB_0	Initialized	Retained	Retained	Initialized	Ret
TGRC_0	Initialized	Retained	Retained	Initialized	Ret
TGRD_0	Initialized	Retained	Retained	Initialized	Ret
TGRE_0	Initialized	Retained	Retained	Initialized	Ret
TGRF_0	Initialized	Retained	Retained	Initialized	Ret
TIER2_0	Initialized	Retained	Retained	Initialized	Ret
TSR2_0	Initialized	Retained	Retained	Initialized	Ret
TBTM_0	Initialized	Retained	Retained	Initialized	Ret
TCR_1	Initialized	Retained	Retained	Initialized	Ret
TMDR_1	Initialized	Retained	Retained	Initialized	Ret
TIOR_1	Initialized	Retained	Retained	Initialized	Ret
TIER_1	Initialized	Retained	Retained	Initialized	Ret
TSR_1	Initialized	Retained	Retained	Initialized	Ret
TCNT_1	Initialized	Retained	Retained	Initialized	Ret
TGRA_1	Initialized	Retained	Retained	Initialized	Ret
TGRB_1	Initialized	Retained	Retained	Initialized	Ret
TICCR	Initialized	Retained	Retained	Initialized	Ret
TCR_2	Initialized	Retained	Retained	Initialized	Ret
TMDR_2	Initialized	Retained	Retained	Initialized	Ret
TIOR_2	Initialized	Retained	Retained	Initialized	Ret
TIER_2	Initialized	Retained	Retained	Initialized	Ret
TSR_2	Initialized	Retained	Retained	Initialized	Ret
TCNT_2	Initialized	Retained	Retained	Initialized	Ret

TCNT_3	Initialized	Retained	Retained	Initialized	Re
TGRA_3	Initialized	Retained	Retained	Initialized	Re
TGRB_3	Initialized	Retained	Retained	Initialized	Re
TGRC_3	Initialized	Retained	Retained	Initialized	Re
TGRD_3	Initialized	Retained	Retained	Initialized	Re
TBTM_3	Initialized	Retained	Retained	Initialized	Re
TCR_4	Initialized	Retained	Retained	Initialized	Re
TMDR_4	Initialized	Retained	Retained	Initialized	Re
TIORH_4	Initialized	Retained	Retained	Initialized	Re
TIORL_4	Initialized	Retained	Retained	Initialized	Re
TIER_4	Initialized	Retained	Retained	Initialized	Re
TSR_4	Initialized	Retained	Retained	Initialized	Re
TCNT_4	Initialized	Retained	Retained	Initialized	Re
TGRA_4	Initialized	Retained	Retained	Initialized	Re
TGRB_4	Initialized	Retained	Retained	Initialized	Re
TGRC_4	Initialized	Retained	Retained	Initialized	Re
TGRD_4	Initialized	Retained	Retained	Initialized	Re
TBTM_4	Initialized	Retained	Retained	Initialized	Re
TADCR	Initialized	Retained	Retained	Initialized	Re
TADCORA_4	Initialized	Retained	Retained	Initialized	Re
TADCORB_4	Initialized	Retained	Retained	Initialized	Re
TADCOBRA_4	Initialized	Retained	Retained	Initialized	Re
TADCOBRB_4	Initialized	Retained	Retained	Initialized	Re

TSTR_5	Initialized	Retained	Retained	Initialized	Ret
TCNTU_5	Initialized	Retained	Retained	Initialized	Ret
TCNTV_5	Initialized	Retained	Retained	Initialized	Ret
TCNTW_5	Initialized	Retained	Retained	Initialized	Ret
TGRU_5	Initialized	Retained	Retained	Initialized	Ret
TGRV_5	Initialized	Retained	Retained	Initialized	Ret
TGRW_5	Initialized	Retained	Retained	Initialized	Ret
TCNTCMPCLR	Initialized	Retained	Retained	Initialized	Ret
TSTR	Initialized	Retained	Retained	Initialized	Ret
TSYR	Initialized	Retained	Retained	Initialized	Ret
TCSYSTR	Initialized	Retained	Retained	Initialized	Ret
TRWER	Initialized	Retained	Retained	Initialized	Ret
TOER	Initialized	Retained	Retained	Initialized	Ret
TOCR1	Initialized	Retained	Retained	Initialized	Ret
TOCR2	Initialized	Retained	Retained	Initialized	Ret
TGCR	Initialized	Retained	Retained	Initialized	Ret
TCDR	Initialized	Retained	Retained	Initialized	Ret
TDDR	Initialized	Retained	Retained	Initialized	Ret
TCNTS	Initialized	Retained	Retained	Initialized	Ret
TCBR	Initialized	Retained	Retained	Initialized	Ret
TITCR	Initialized	Retained	Retained	Initialized	Ret
TITCNT	Initialized	Retained	Retained	Initialized	Ret
TBTER	Initialized	Retained	Retained	Initialized	Ret

TIER_3S	Initialized	Retained	Retained	Initialized	Re
TSR_3S	Initialized	Retained	Retained	Initialized	Re
TCNT_3S	Initialized	Retained	Retained	Initialized	Re
TGRA_3S	Initialized	Retained	Retained	Initialized	Re
TGRB_3S	Initialized	Retained	Retained	Initialized	Re
TGRC_3S	Initialized	Retained	Retained	Initialized	Re
TGRD_3S	Initialized	Retained	Retained	Initialized	Re
TBTM_3S	Initialized	Retained	Retained	Initialized	Re
TCR_4S	Initialized	Retained	Retained	Initialized	Re
TMDR_4S	Initialized	Retained	Retained	Initialized	Re
TIORH_4S	Initialized	Retained	Retained	Initialized	Re
TIORL_4S	Initialized	Retained	Retained	Initialized	Re
TIER_4S	Initialized	Retained	Retained	Initialized	Re
TSR_4S	Initialized	Retained	Retained	Initialized	Re
TCNT_4S	Initialized	Retained	Retained	Initialized	Re
TGRA_4S	Initialized	Retained	Retained	Initialized	Re
TGRB_4S	Initialized	Retained	Retained	Initialized	Re
TGRC_4S	Initialized	Retained	Retained	Initialized	Re
TGRD_4S	Initialized	Retained	Retained	Initialized	Re
TBTM_4S	Initialized	Retained	Retained	Initialized	Re
TADCRS	Initialized	Retained	Retained	Initialized	Re
TADCORA_4S	Initialized	Retained	Retained	Initialized	Re
TADCORB_4S	Initialized	Retained	Retained	Initialized	Re

TIER_5S	Initialized	Retained	Retained	Initialized	Ret
TSR_5S	Initialized	Retained	Retained	Initialized	Ret
TSTR_5S	Initialized	Retained	Retained	Initialized	Ret
TCNTU_5S	Initialized	Retained	Retained	Initialized	Ret
TCNTV_5S	Initialized	Retained	Retained	Initialized	Ret
TCNTW_5S	Initialized	Retained	Retained	Initialized	Ret
TGRU_5S	Initialized	Retained	Retained	Initialized	Ret
TGRV_5S	Initialized	Retained	Retained	Initialized	Ret
TGRW_5S	Initialized	Retained	Retained	Initialized	Ret
TCNTCMPCLRS	Initialized	Retained	Retained	Initialized	Ret
TSTRS	Initialized	Retained	Retained	Initialized	Ret
TSYRS	Initialized	Retained	Retained	Initialized	Ret
TRWERS	Initialized	Retained	Retained	Initialized	Ret
TOERS	Initialized	Retained	Retained	Initialized	Ret
TOCR1S	Initialized	Retained	Retained	Initialized	Ret
TOCR2S	Initialized	Retained	Retained	Initialized	Ret
TGCRS	Initialized	Retained	Retained	Initialized	Ret
TCDRS	Initialized	Retained	Retained	Initialized	Ret
TDDRS	Initialized	Retained	Retained	Initialized	Ret
TCNTSS	Initialized	Retained	Retained	Initialized	Ret
TCBRS	Initialized	Retained	Retained	Initialized	Ret
TITCRS	Initialized	Retained	Retained	Initialized	Ret

	ICSR2	Initialized	Retained	Retained	Retained	Re
	OCSR2	Initialized	Retained	Retained	Retained	Re
	ICSR3	Initialized	Retained	Retained	Retained	Re
	SPOER	Initialized	Retained	Retained	Retained	Re
	POECR1	Initialized	Retained	Retained	Retained	Re
	POECR2	Initialized	Retained	Retained	Retained	Re
CMT	CMSTR	Initialized	Retained	Initialized	Retained	Ini
	CMCSR_0	Initialized	Retained	Initialized	Retained	Ini
	CMCNT_0	Initialized	Retained	Initialized	Retained	Ini
	CMCOR_0	Initialized	Retained	Initialized	Retained	Ini
	CMCSR_1	Initialized	Retained	Initialized	Retained	Ini
	CMCNT_1	Initialized	Retained	Initialized	Retained	Ini
	CMCOR_1	Initialized	Retained	Initialized	Retained	Ini
WDT	WTCSR	Initialized	Retained	Retained	—	Re
	WTCNT	Initialized	Retained	Retained	—	Re
	WRCSR	Initialized*[1]	Retained	Retained	—	Re
SCIF	SCSMR_0	Initialized	Retained	Retained	Retained	Re
	SCBRR_0	Initialized	Retained	Retained	Retained	Re
	SCSCR_0	Initialized	Retained	Retained	Retained	Re
	SCFTDR_0	Undefined	Retained	Retained	Retained	Re
	SCFSR_0	Initialized	Retained	Retained	Retained	Re
	SCFRDR_0	Undefined	Retained	Retained	Retained	Re
	SCFCR_0	Initialized	Retained	Retained	Retained	Re

SCFRDR_1	Undefined	Retained	Retained	Retained	Ret
SCFCR_1	Initialized	Retained	Retained	Retained	Ret
SCFDR_1	Initialized	Retained	Retained	Retained	Ret
SCSPTR_1	Initialized	Retained	Retained	Retained	Ret
SCLSR_1	Initialized	Retained	Retained	Retained	Ret
SCSEMR_1	Initialized	Retained	Retained	Retained	Ret
SCSMR_2	Initialized	Retained	Retained	Retained	Ret
SCBRR_2	Initialized	Retained	Retained	Retained	Ret
SCSCR_2	Initialized	Retained	Retained	Retained	Ret
SCFTDR_2	Undefined	Retained	Retained	Retained	Ret
SCFSR_2	Initialized	Retained	Retained	Retained	Ret
SCFRDR_2	Undefined	Retained	Retained	Retained	Ret
SCFCR_2	Initialized	Retained	Retained	Retained	Ret
SCFDR_2	Initialized	Retained	Retained	Retained	Ret
SCSPTR_2	Initialized	Retained	Retained	Retained	Ret
SCLSR_2	Initialized	Retained	Retained	Retained	Ret
SCSEMR_2	Initialized	Retained	Retained	Retained	Ret
SCSMR_3	Initialized	Retained	Retained	Retained	Ret
SCBRR_3	Initialized	Retained	Retained	Retained	Ret
SCSCR_3	Initialized	Retained	Retained	Retained	Ret
SCFTDR_3	Undefined	Retained	Retained	Retained	Ret
SCFSR_3	Initialized	Retained	Retained	Retained	Ret
SCFRDR_3	Undefined	Retained	Retained	Retained	Ret

				(bc3-0)	(bc3-0)	
	ICIER	Initialized	Retained	Retained	Retained	Re
	ICSR	Initialized	Retained	Retained	Retained	Re
	SAR	Initialized	Retained	Retained	Retained	Re
	ICDRT	Initialized	Retained	Retained	Retained	Re
	ICDRR	Initialized	Retained	Retained	Retained	Re
	NF2CYC	Initialized	Retained	Retained	Retained	Re
ADC	ADCR	Initialized	Retained	Initialized	Retained	Re
	ADSR	Initialized	Retained	Initialized	Retained	Re
	ADSTRGR	Initialized	Retained	Initialized	Retained	Re
	ADANSR	Initialized	Retained	Initialized	Retained	Re
	ADDR0	Initialized	Retained	Initialized	Retained	Re
	ADDR1	Initialized	Retained	Initialized	Retained	Re
	ADDR2	Initialized	Retained	Initialized	Retained	Re
	ADDR3	Initialized	Retained	Initialized	Retained	Re
	ADDR4	Initialized	Retained	Initialized	Retained	Re
	ADDR5	Initialized	Retained	Initialized	Retained	Re
	ADDR6	Initialized	Retained	Initialized	Retained	Re
	ADDR7	Initialized	Retained	Initialized	Retained	Re
DAC	DADR0	Initialized	Retained	Retained	Initialized	Re
	DADR1	Initialized	Retained	Retained	Initialized	Re
	DACR	Initialized	Retained	Retained	Initialized	Re

PACRL2	Initialized	Retained	Retained	—	Ret
PACRL1	Initialized	Retained	Retained	—	Ret
PBIORH	Initialized	Retained	Retained	—	Ret
PBIORL	Initialized	Retained	Retained	—	Ret
PBCRH4	Initialized	Retained	Retained	—	Ret
PBCRH3	Initialized	Retained	Retained	—	Ret
PBCRH2	Initialized	Retained	Retained	—	Ret
PBCRH1	Initialized	Retained	Retained	—	Ret
PBCRL4	Initialized	Retained	Retained	—	Ret
PBCRL3	Initialized	Retained	Retained	—	Ret
PBCRL2	Initialized	Retained	Retained	—	Ret
PBCRL1	Initialized	Retained	Retained	—	Ret
PDIOR	Initialized	Retained	Retained	—	Ret
PDCRL4	Initialized	Retained	Retained	—	Ret
PDCRL3	Initialized	Retained	Retained	—	Ret
PDCRL2	Initialized	Retained	Retained	—	Ret
PDCRL1	Initialized	Retained	Retained	—	Ret
PFCRL1	Initialized	Retained	Retained	—	Ret
IFCR	Initialized	Retained	Retained	—	Ret
WAVECR2	Initialized	Retained	Retained	—	Ret
WAVECR1	Initialized	Retained	Retained	—	Ret

	PDDRL	Initialized	Retained	Retained	—	Re
	PDPRL	Undefined	Retained	Retained	—	Re
	PFDR	Initialized	Retained	Retained	—	Re
FLASH	FCCS	Initialized	Retained	Initialized	Initialized	Re
	FPCS	Initialized	Retained	Initialized	Initialized	Re
	FECS	Initialized	Retained	Initialized	Initialized	Re
	FKEY	Initialized	Retained	Initialized	Initialized	Re
	FMATS	Initialized	Retained	Initialized	Initialized	Re
	FTDAR	Initialized	Retained	Initialized	Initialized	Re
Power-down mode	STBCR	Initialized	Retained	Retained	—	Re
	STBCR2	Initialized	Retained	Retained	—	Re
	SYSCR1	Initialized	Retained	Retained	—	Re
	SYSCR2	Initialized	Retained	Retained	—	Re
	STBCR3	Initialized	Retained	Retained	—	Re
	STBCR4	Initialized	Retained	Retained	—	Re
H-UDI*3	SDIR	Retained	Retained	Retained	Retained	Re

Notes: 1. Retains the previous value after an internal power-on reset by means of the V
2. Bits BN[3:0] are initialized.
3. Initialized by $\overline{\text{TRST}}$ assertion or in the Test-Logic-Reset state of the TAP cont

Item	Symbol	Value
Power supply voltage (I/O)	$V_{CC}Q$	–0.3 to 4.6
Power supply voltage (Internal)	V_{CC}	–0.3 to 2.3
	$PLLV_{CC}$	
Input voltage (except analog input pins)	Vin	–0.3 to $V_{CC}Q$ +0.3
Analog power supply voltage	AV_{CC}	–0.3 to 7.0
Analog reference voltage	AVREF	–0.3 to AV_{CC} +0.3
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	–40 to +85
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exc

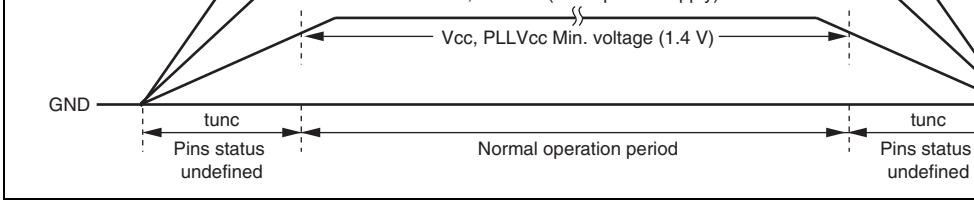


Figure 27.1 Power-on/Power-off Sequence

Table 27.2 Recommended Time for Power-on/Power-off Sequence

Item	Symbol	Maximum Allowance Value	Unit
Undefined time	tunc	100	ms

Note: VccQ ≥ Vcc = PLLVcc is recommended. Either VccQ, Vcc, or PLLVcc power supp
turned on or off first, though, an undefined period appears until Vcc rises to the Mi
or after Vcc passes the Min. voltage. During these periods, pin or internal states b
undefined. Design the system so that such undefined states do not cause a syster
malfunction. To avoid an increase in the current consumption during the undefined
at power-on, it is recommended that VccQ, Vcc, and PLLVcc be turned on simulta
This undefined period can be eliminated by turning on the power supplies in the or
shown in figure 27.2.

Figure 27.2 Power-On Sequence

Notes: To prevent the pin and internal states from being undefined, VccQ and AVcc
kept GND voltage level (0 V) and they should not be placed in floating state
reaches the Min. voltage. In addition, the $\overline{\text{RES}}$ pin should be input low to pla
on reset state. In this case, care must be taken for the power consumption inc
caused by sink current because each pin is placed in low-impedance state unt
reaches the Min. voltage.

		$PLLV_{CC}$					
Analog power supply voltage		AV_{CC}	4.5	5.0	5.5	V	
Current consumption*1	Normal operation	I_{CC}	–	200	310	mA	V_{CC} = 1.5 V Iϕ = 160 M Bϕ = 40 M Pϕ = 40 M
	Software standby mode	I_{stby}	–	5	60	mA	Ta = 25°C
		PI_{stby}	–	0.1	1	mA	V_{CC} = 3.3 V V_{CC} = 1.5 V
	Sleep mode	I_{sleep}	–	60	100	mA	Bϕ = 40 M Pϕ = 40 M
Input leakage current	All input pins (except PF0, PF1)	$\|I_{in}\|$	–	–	1	µA	V_{in} = 0.5 to V_{CC}
	PF0, PF1		–	–	1	µA	
Three-state leakage current	Input/output pins, all output pins (off state)	$\|I_{STI}\|$	–	–	1	µA	Vin = 0.5 to V_{CC}
Input capacitance	All pins	C_{in}	–	–	10	pF	
Analog power supply current	During A/D or D/A conversion	AI_{CC}	–	5	4	mA	Including A
	Waiting for A/D or D/A conversion		–	1	3	mA	
	Standby mode			–	4	µA	

Caution: When neither the A/D converter nor the D/A converter is in use, do not leave th AVss, AVREF, and AVREFVss pins open.

Notes: 1. Current consumption values are when all output pins are unloaded.

2. I_{CC}, I_{sleep}, and I_{stby} represent the total currents consumed in the Vcc and PLLVcc

3. PI_{stby} is the total current consumed in the VccQ power supply.

	$\overline{\text{ASEBRK}}$, $\overline{\text{FWE}}$					
	Input pins other than above (excluding Schmitt pins)		2.0	—	$V_{CC}Q + 0.3$	V
Input low voltage	$\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, MD1, MD0, MD_CLK2, MD_CLK0, $\overline{\text{ASEMD}}$, $\overline{\text{TRST}}$, EXTAL, $\overline{\text{ASEBRK}}$, $\overline{\text{FWE}}$	V_{IL}	–0.3	—	0.5	V
	Input pins other than above (excluding Schmitt pins)		–0.3	—	0.8	V

	TIOC4AS, TIOC4BS, TIOC4CS, TIOC4DS, TIC5US, TIC5VS, TIC5WS, $\overline{\text{POE8}}$, $\overline{\text{POE7}}$, $\overline{\text{POE4}}$, $\overline{\text{POE3}}$, $\overline{\text{POE1}}$, $\overline{\text{POE0}}$, SCK3 to SCK0, RXD3 to RXD0, IRQ7 to IRQ0						
Output high voltage	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	V_{OH}	$V_{CC}Q - 0.8$	–	–	V	$I_{OH} = -5$
	All output pins except for above pins		$V_{CC}Q - 0.5$	–	–	V	$I_{OH} = -2$

	All output pins except for above pins	–	–	0.4	V	I_{OL} = 2

Table 27.3 DC Characteristics (3) [I²C-Related Pins*]

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test C
Input high voltage	V_{IH}	$V_{CC}Q \times 0.7$	–	$V_{CC}Q + 0.3$	V	
Input low voltage	V_{IL}	–0.3	–	$PV_{CC} \times 0.3$	V	
Schmitt trigger input characteristics	$V_{IH} - V_{IL}$	0.4	–	–	V	
Output low voltage	V_{OL}	–	–	0.4	V	I_{OL} = 3.

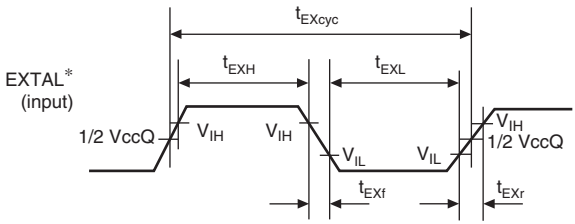
Note: * The PF0/IRQ0/$\overline{\text{POE7}}$/SCL and PF1/IRQ1/$\overline{\text{POE3}}$/SDA pins (open-drain pins)

	TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)				
	SCL, SDA				10
	Output pins other than above				2
Permissible output low current (total)		ΣI_{OL}	–	–	80
Permissible output high current (per pin)	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	$-I_{OH}$	–	–	5
	Output pins other than above		–	–	2
Permissible output high current (total)		$\Sigma-I_{OH}$	—	—	25

Caution: To protect the LSI's reliability, do not exceed the output current values in table

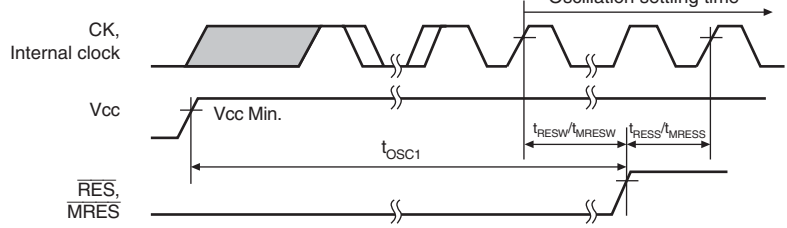
Operating frequency	CPU (Iϕ)	f	32	–	160	MHz
	Internal bus, external bus (Bϕ)		32	–	40	
	Peripheral module (Pϕ)		4	–	40	

EXTAL clock input pulse high width	t_{EXH}	6	–	ns	
EXTAL clock input rise time	t_{EXr}	–	3	ns	
EXTAL clock input fall time	t_{EXf}	–	3	ns	
CK clock output frequency	f_{OP}	16	40	MHz	Figure
CK clock output cycle time	t_{cyc}	25.0	62.5	ns	
CK clock output pulse low width	t_{CKOL}	6	–	ns	
CK clock output pulse high width	t_{CKOH}	6	–	ns	
CK clock output rise time	t_{CKOr}	–	3	ns	
CK clock output fall time	t_{CKOf}	–	3	ns	
Power-on oscillation setting time	t_{OSC1}	10	–	ms	Figure
Oscillation settling time on return from standby 1	t_{OSC2}	10	–	ms	Figure
Oscillation settling time on return from standby 2	t_{OSC3}	10	–	ms	Figure



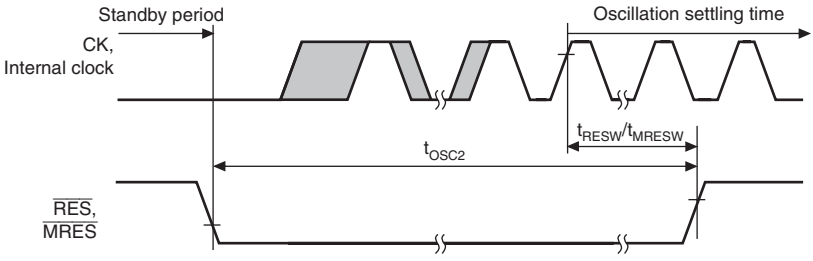
Note: * When the clock is input on the EXTAL pin.

Figure 27.3 EXTAL Clock Input Timing



Note: Oscillation settling time when the internal oscillator is used.

Figure 27.5 Power-On Oscillation Settling Time



Note: Oscillation settling time when the internal oscillator is used.

Figure 27.6 Oscillation Settling Time on Return from Standby (Return by R

$\overline{\text{RES}}$ hold time	t_{RESH}	15	–	ns	Figur
$\overline{\text{MRES}}$ pulse width	t_{MRESW}	20*[3]	—	t_{cyc}	
$\overline{\text{MRES}}$ setup time	t_{MRESS}	25	–	ns	
$\overline{\text{MRES}}$ hold time	t_{MRESH}	15	–	ns	
MD1, MD0 setup time	t_{MDS}	20	–	t_{cyc}	Figur
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	$1/2t_{cyc}$ + 10	–	ns	Figur
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	$1/2t_{cyc}$ + 4	–	ns	
NMI setup time *[1]	t_{NMIS}	15	–	ns	Figur
NMI hold time	t_{NMIH}	7	–	ns	
IRQ7 to IRQ0 setup time *[1]	t_{IRQS}	15	–	ns	
IRQ7 to IRQ0 hold time	t_{IRQH}	7	–	ns	
$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ output delay time	t_{IRQOD}	—	100	ns	Figur
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	$1/2t_{cyc}$ + 20	ns	Figur
Bus tri-state delay time 1	t_{BOFF1}	0	100	ns	
Bus tri-state delay time 2	t_{BOFF2}	0	100	ns	
Bus buffer on time 1	t_{BON1}	0	30	ns	
Bus buffer on time 2	t_{BON2}	0	30	ns	

Notes: 1. $\overline{\text{RES}}$, NMI, and IRQ7 to IRQ0 are asynchronous signals. When these setup ti observed, a change of these signals is detected at the clock rising edge. If the times are not observed, detection of a signal change may be delayed until the rising edge of the clock.

2. In standby mode or when the clock multiplication ratio is changed, $t_{RESW} = t_{OSC2}$ ms).
3. In standby mode, $t_{RESW} = t_{OSC2}$ (Min. 10 ms).

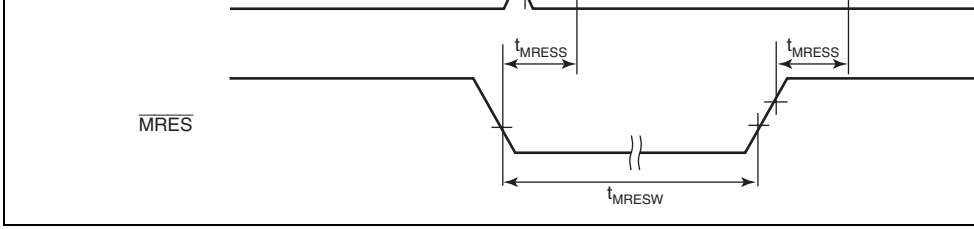


Figure 27.8 Reset Input Timing

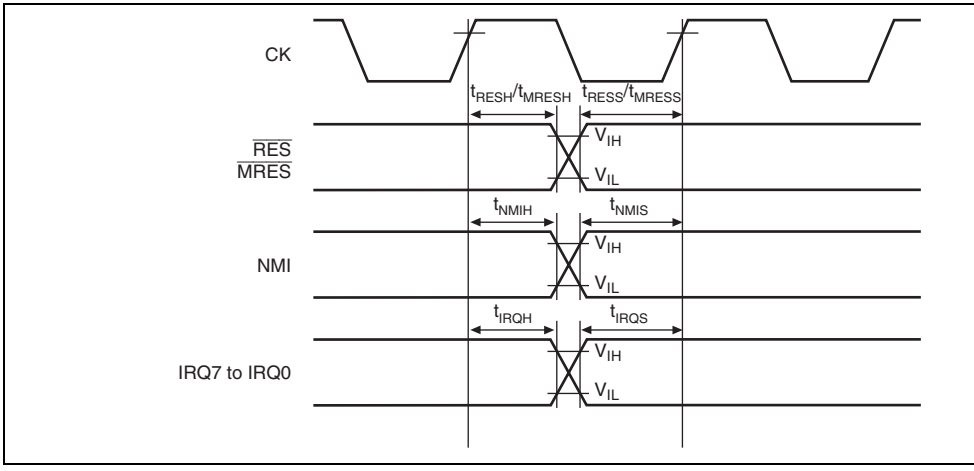


Figure 27.9 Interrupt Signal Input Timing

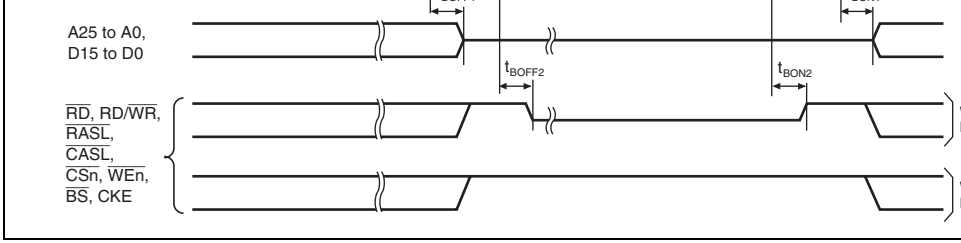


Figure 27.10 Interrupt Signal Output Timing

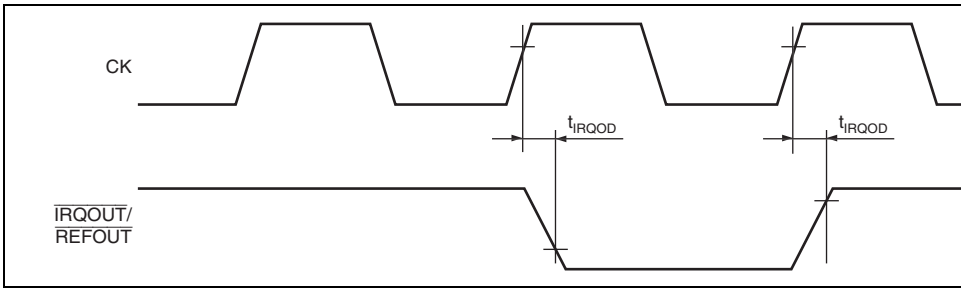


Figure 27.11 Bus Release Timing

Address delay time 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figure 27.1
Address delay time 3	t_{AD3}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.
Address setup time	t_{AS}	0	–	ns	Figures 27. 27.15, 27.1
Address hold time	t_{AH}	0	–	ns	Figures 27. 27.15
$\overline{BS}$ delay time	t_{BSD}	–	20	ns	Figures 27. 27.33, 27.3
$\overline{CS}$ delay time 1	t_{CSD1}	1	20	ns	Figures 27. 27.36
$\overline{CS}$ delay time 2	t_{CSD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.
$\overline{CS}$ setup time	t_{CSS}	0	–	ns	Figures 27. 27.15
$\overline{CS}$ hold time	t_{CSH}	0	–	ns	Figures 27. 27.15
Read write delay time 1	t_{RWD1}	1	20	ns	Figures 27. 27.36
Read write delay time 2	t_{RWD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27. 27.19
Read data setup time 1	t_{RDS1}	$1/2t_{cyc}$ + 13	–	ns	Figures 27. 27.18
Read data setup time 2	t_{RDS2}	10	–	ns	Figures 27. 27.23, 27.2 27.30
Read data setup time 3	t_{RDS3}	$1/2t_{cyc}$ + 20	–	ns	Figure 27.1
Read data setup time 4	t_{RDS4}	$1/2t_{cyc}$ + 20	–	ns	Figure 27.3

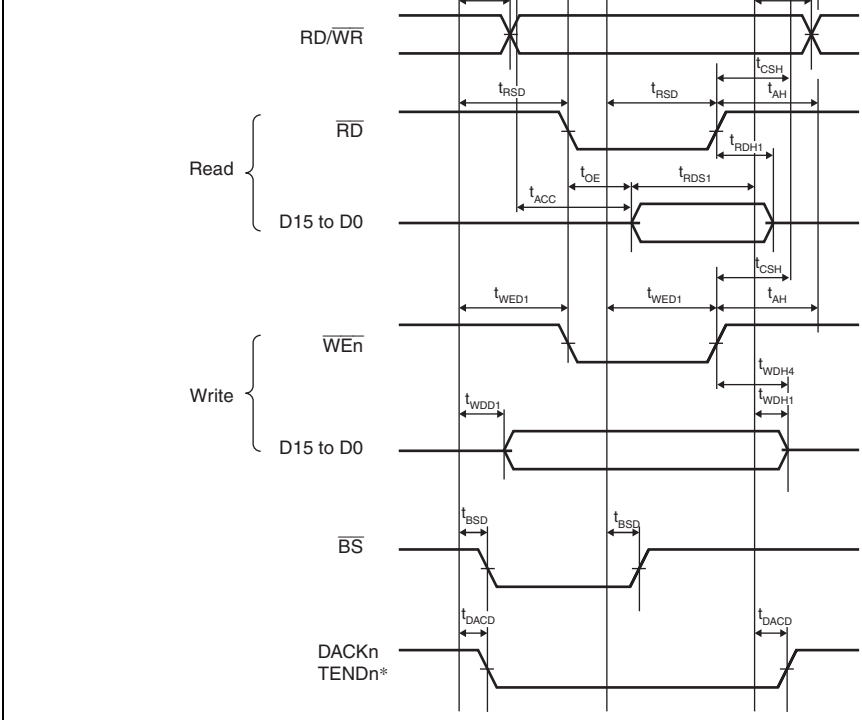
		1.5) – 31			27.15, 27. 27.18
Access time after read strobe	t_{OE}*3	t_{cyc}× (n + 1) – 31*2	—	ns	Figures 27 27.15, 27. 27.18
Write enable delay time 1	t_{WED1}	1/2t_{cyc}	1/2t_{cyc} + 20	ns	Figures 27 27.17
Write enable delay time 2	t_{WED2}	—	20	ns	Figure 27.
Write data delay time 1	t_{WDD1}	—	20	ns	Figures 27 27.18
Write data delay time 2	t_{WDD2}	—	20	ns	Figures 27 27.27, 27. 27.33
Write data delay time 3	t_{WDD3}	—	1/2t_{cyc} + 20	ns	Figure 27.
Write data hold time 1	t_{WDH1}	1	15	ns	Figures 27 27.18
Write data hold time 2	t_{WDH2}	1	—	ns	Figures 27 27.27, 27. 27.33
Write data hold time 3	t_{WDH3}	1/2t_{cyc}	—	ns	Figure 27.
Write data hold time 4	t_{WDH4}	0	—	ns	Figures 27
$\overline{\text{WAIT}}$ setup time	t_{WTS}	1/2t_{cyc} + 10	—	ns	Figures 27 27.19
$\overline{\text{WAIT}}$ hold time	t_{WTH}	1/2t_{cyc} + 5	—	ns	Figures 27 27.19
$\overline{\text{RAS}}$ delay time 1	t_{RASD1}	1	20	ns	Figures 27 27.31, 27. 27.36
$\overline{\text{RAS}}$ delay time 2	t_{RASD2}	1/2t_{cyc}	1/2t_{cyc} + 20	ns	Figures 27

CKE delay time 2	t_{CKED2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figure 27.3
$\overline{AH}$ delay time	t_{AHD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figure 27.1
Multiplexed address delay time	t_{MAD}	—	20	ns	Figure 27.1
Multiplexed address hold time	t_{MAH}	1	—	ns	Figure 27.1
DACK, TEND delay time	t_{DACD}	—	Refer to peripheral modules	ns	Figures 27. 27.33, 27.3

Note: *1 The maximum value (fmax) of Bϕ (external bus clock) depends on the number cycles and the system configuration of your board.

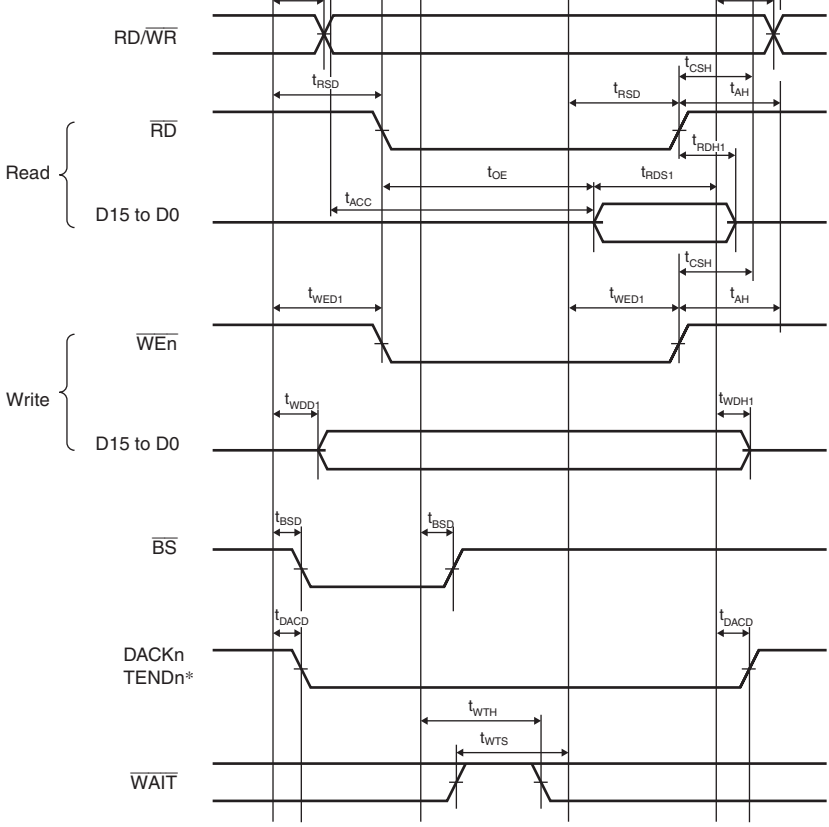
*2 n is the number of wait cycles.

*3 It is not necessary to accommodate t_{RDS1} if the access time is accommodated.



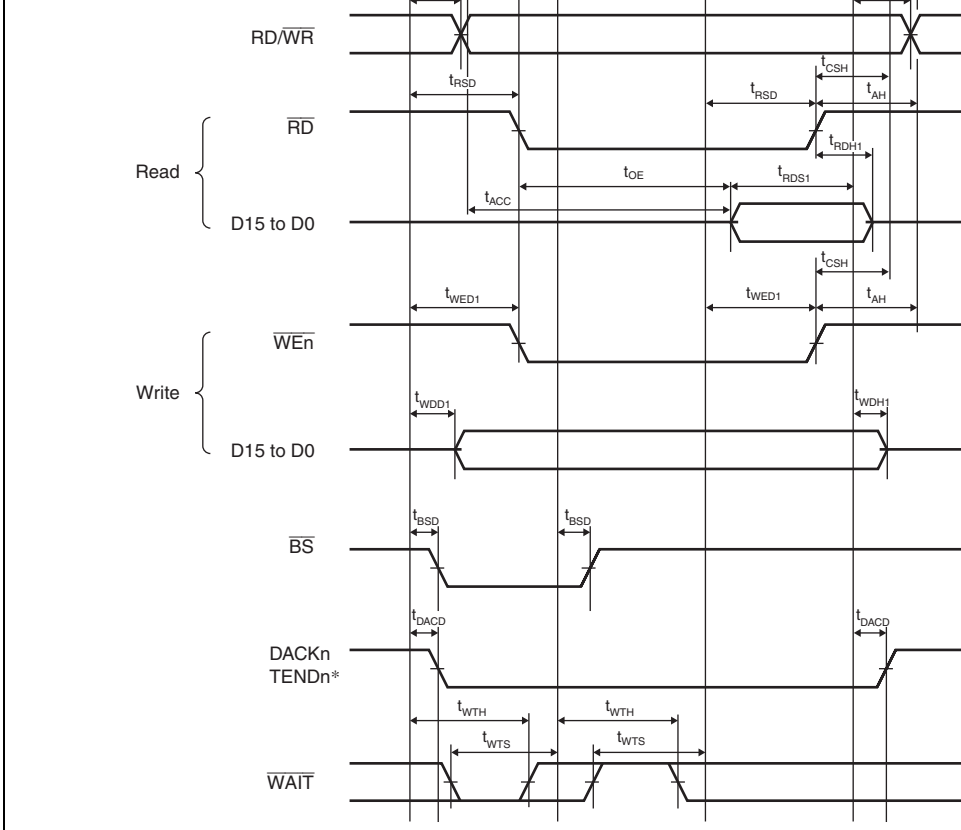
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.12 Basic Bus Timing for Normal Space (No Wait)



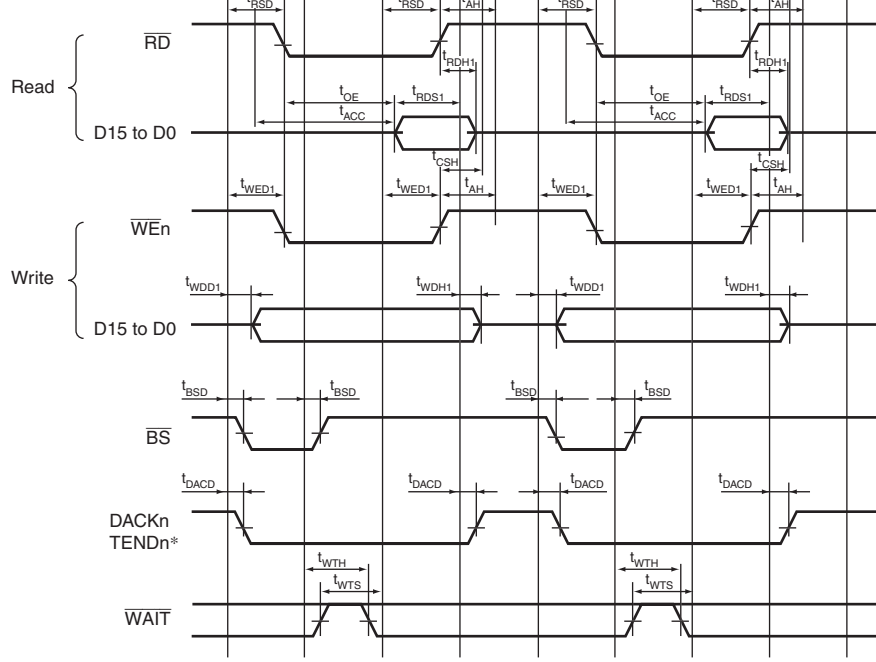
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.13 Basic Bus Timing for Normal Space (One Software Wait Cycl



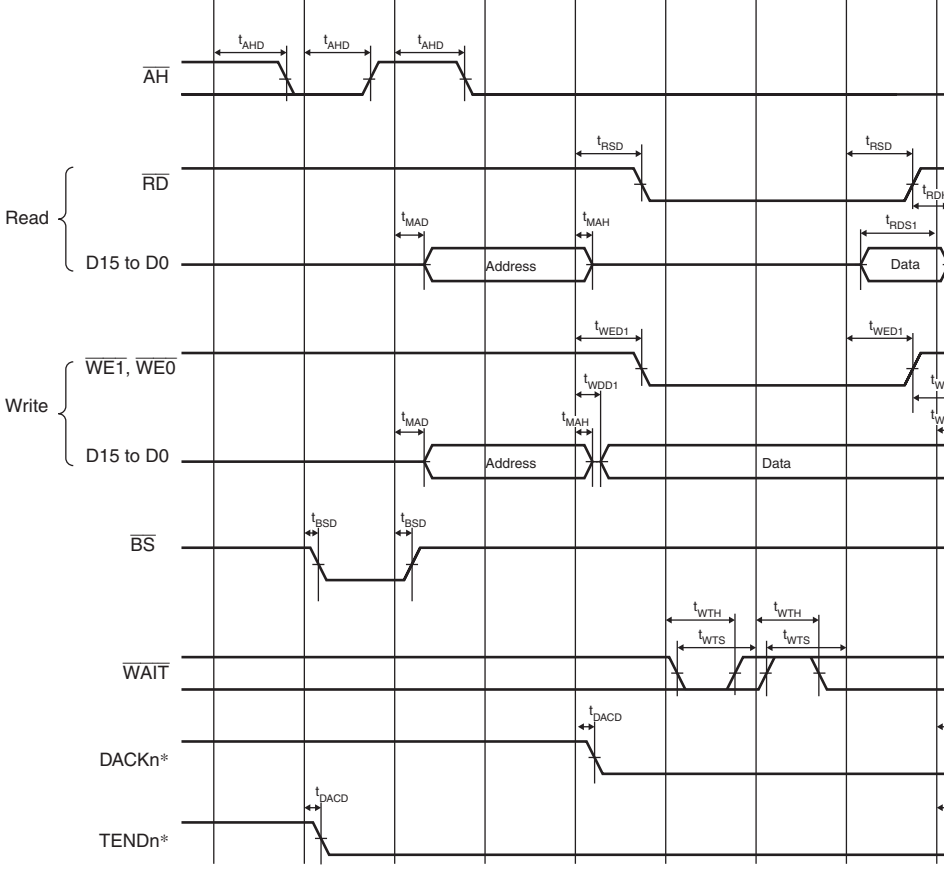
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.14 Basic Bus Timing for Normal Space (One External Wait Cyc



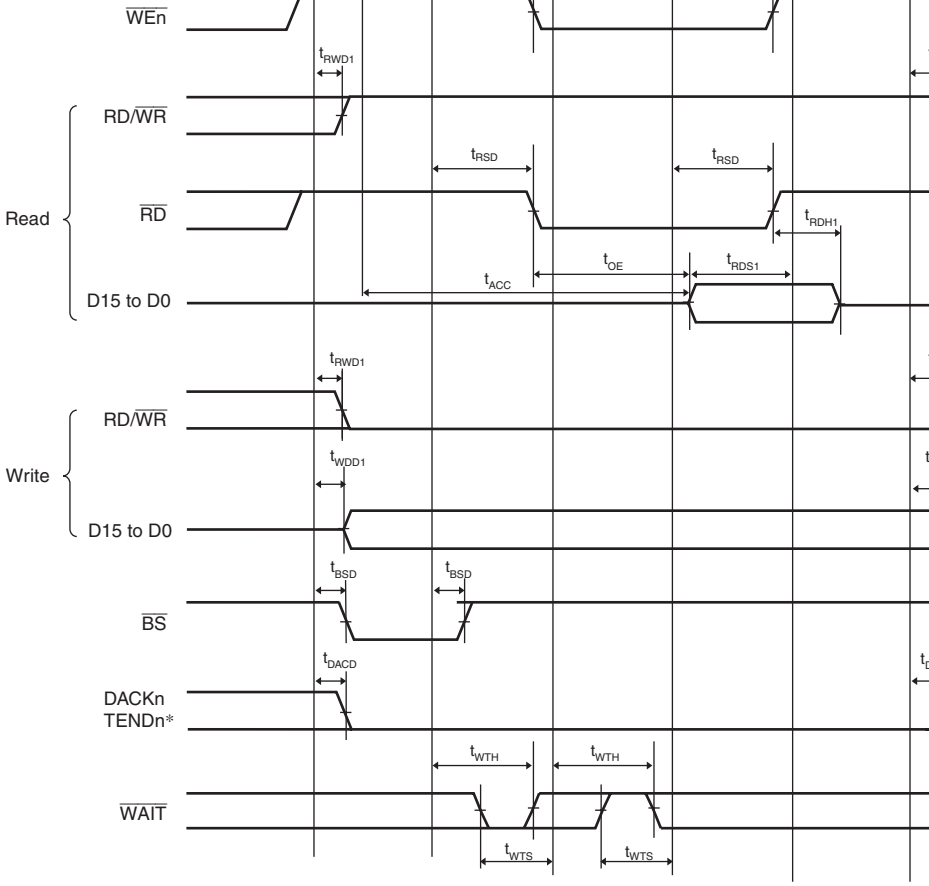
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.15 Basic Bus Timing for Normal Space
(One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle Cy



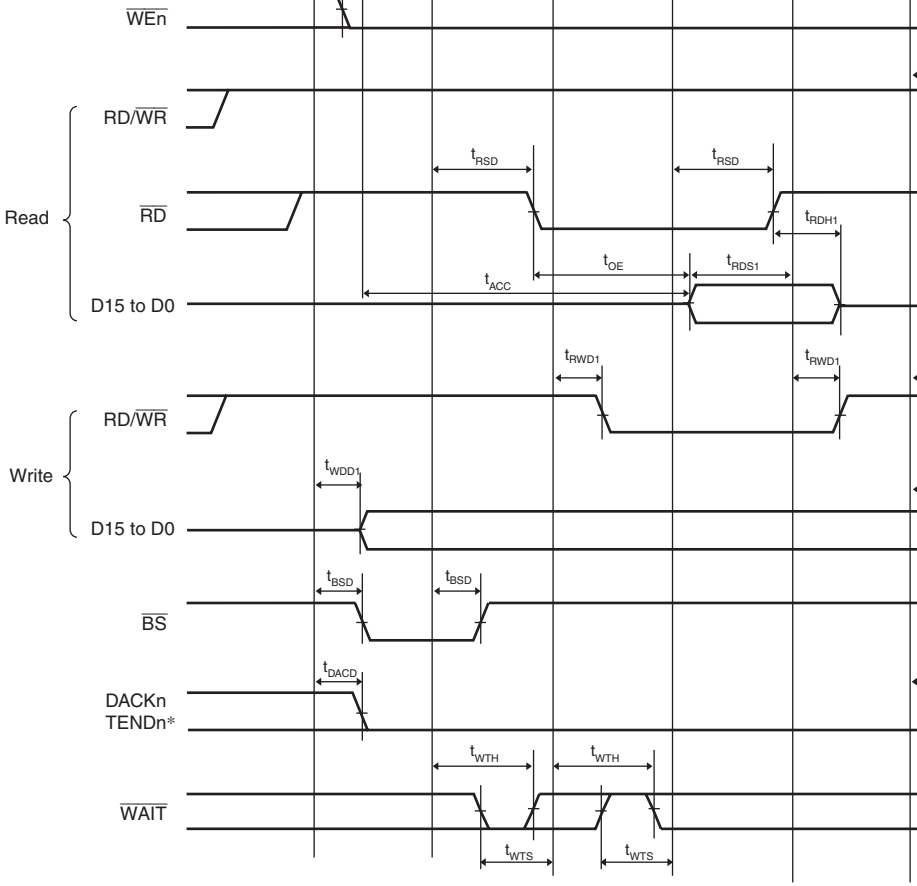
Note: * Waveforms for DACKn and TENDn are when active low is specified.

Figure 27.16 MPX-I/O Interface Bus Cycle (Three Address Cycles, One Software Wait Cycle, One External Wait Cycl



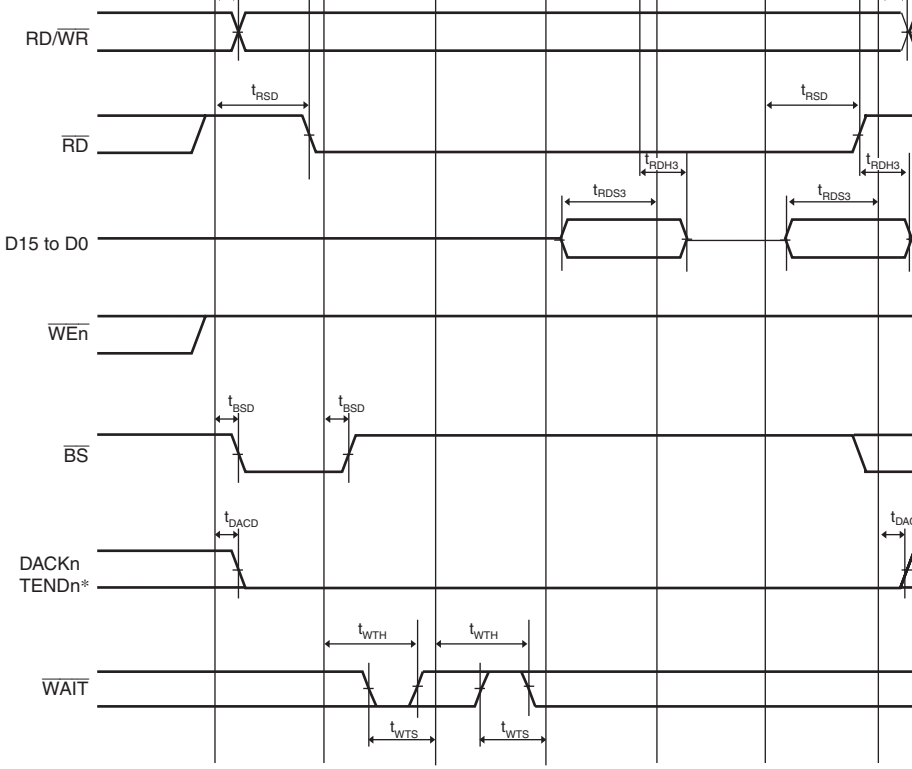
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.17 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 C One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Contro



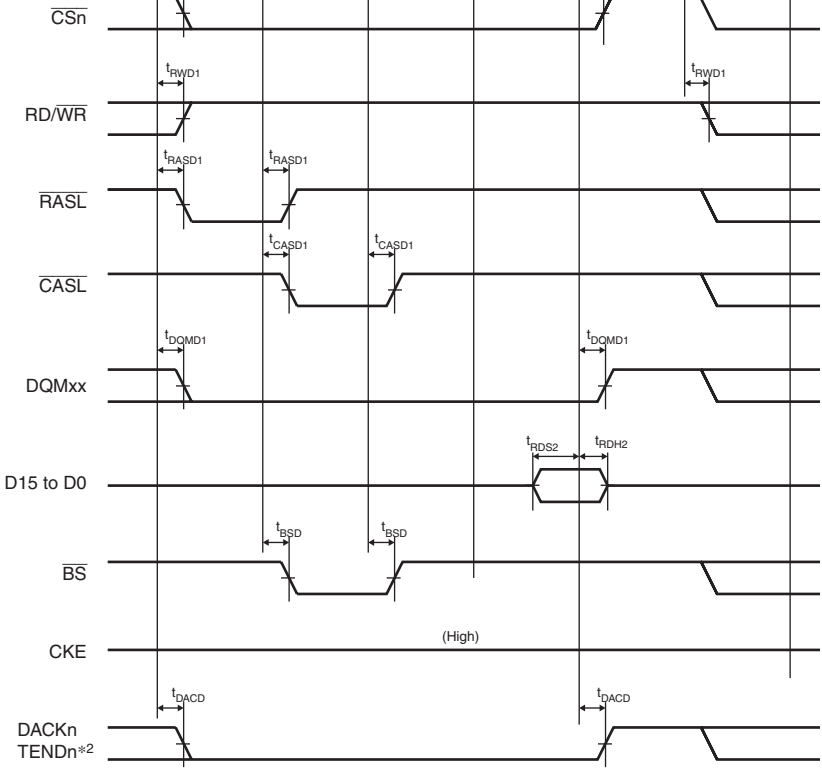
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.18 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Contro



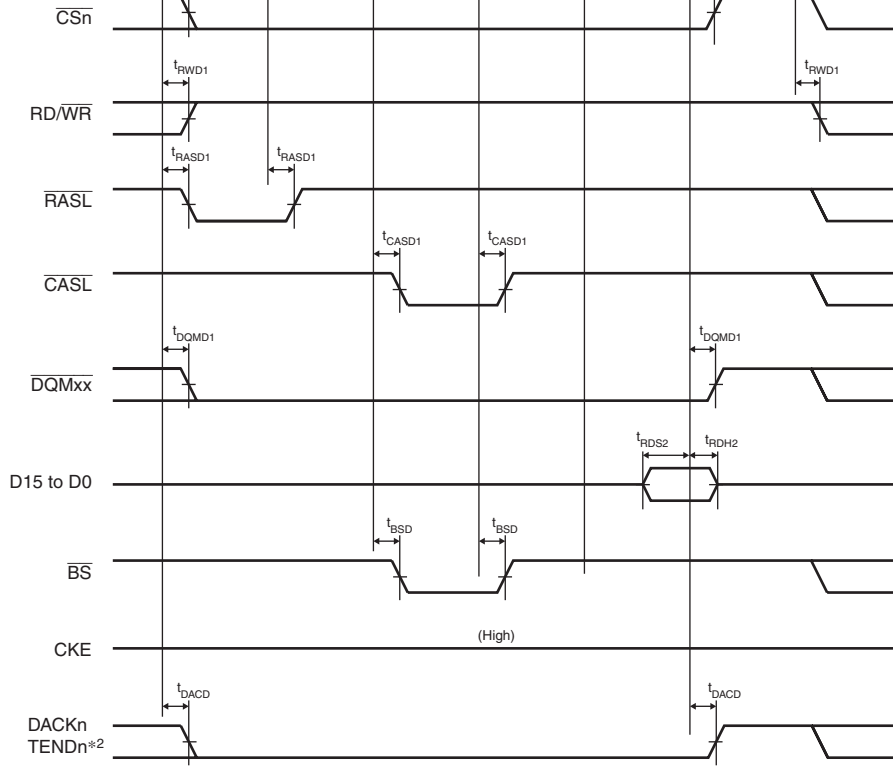
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.19 Burst ROM Read Cycle
(One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two-Cyc



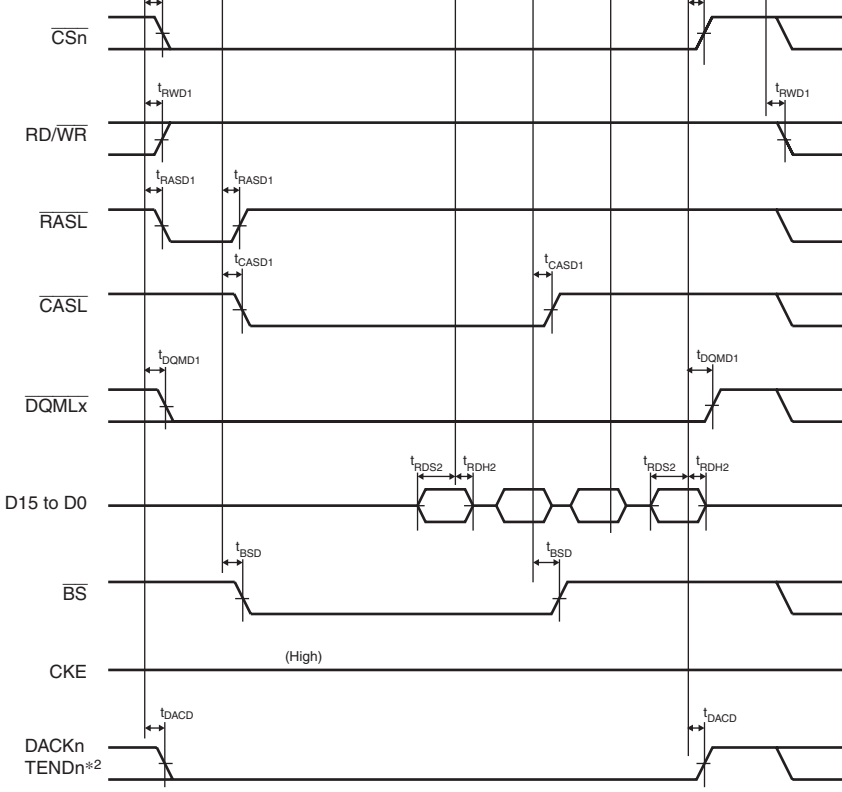
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.20 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)



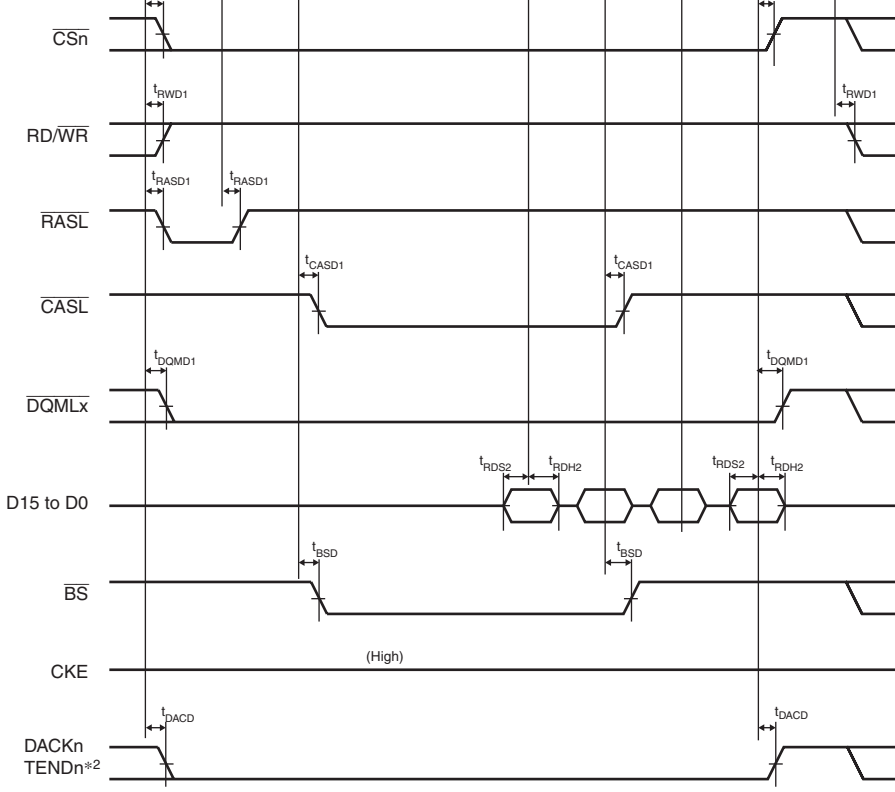
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.21 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



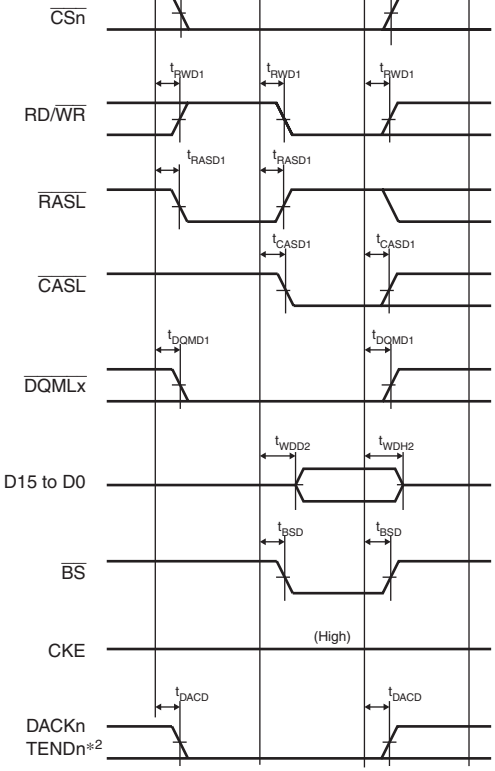
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.22 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycl
(Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)



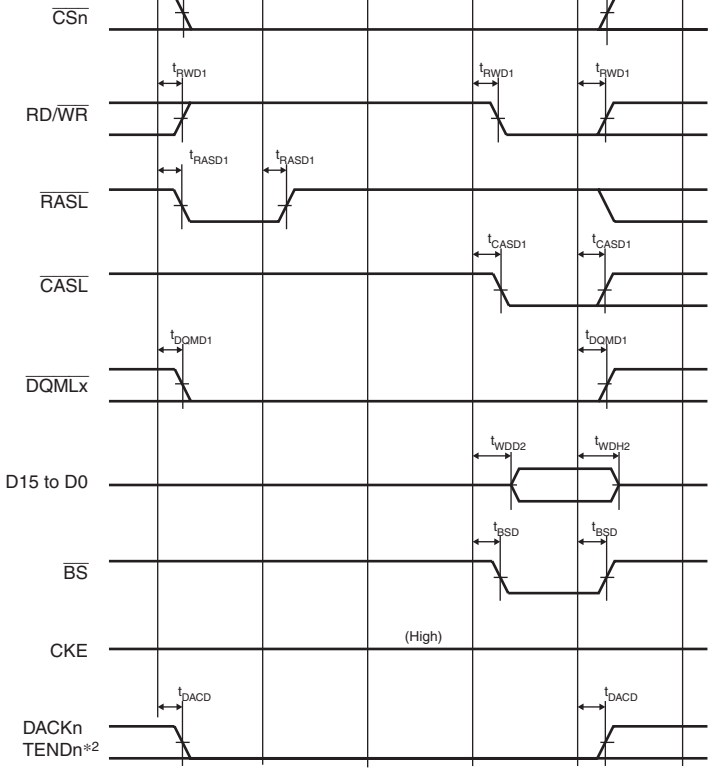
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.23 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



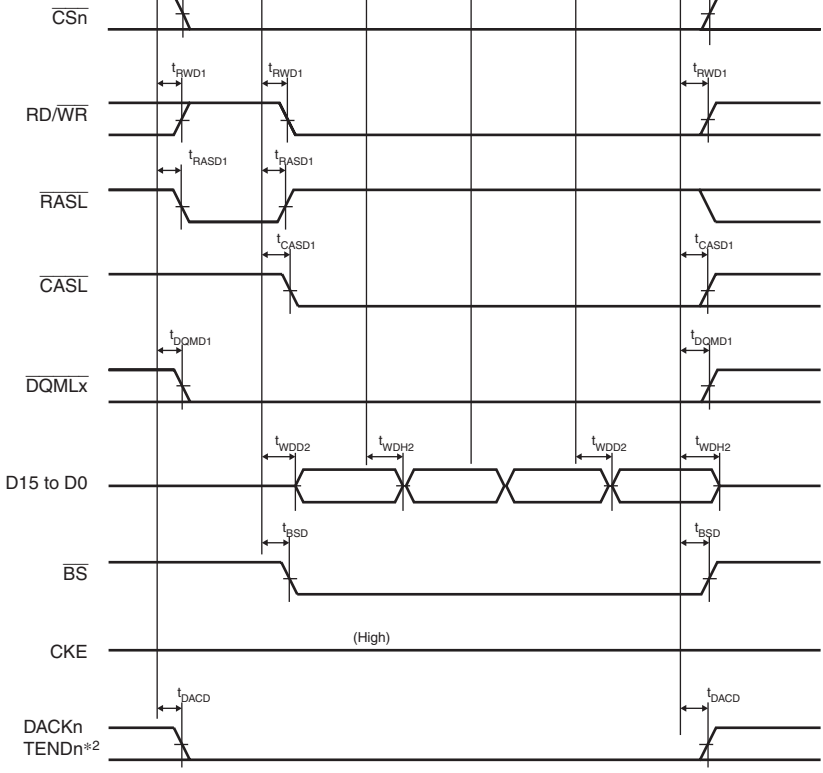
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.24 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 1 Cycle)



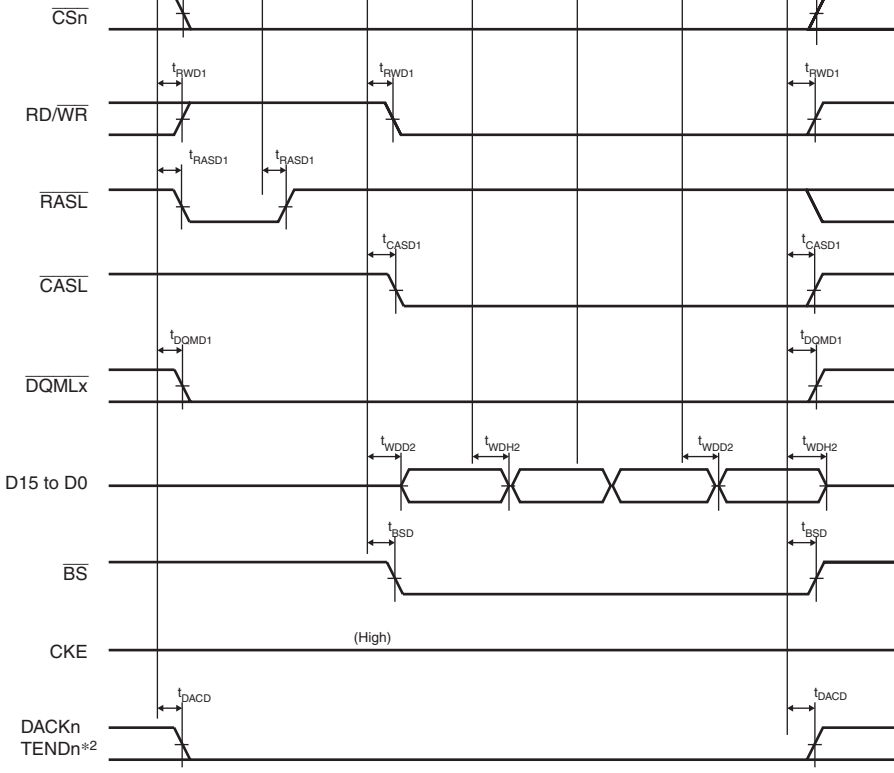
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.25 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



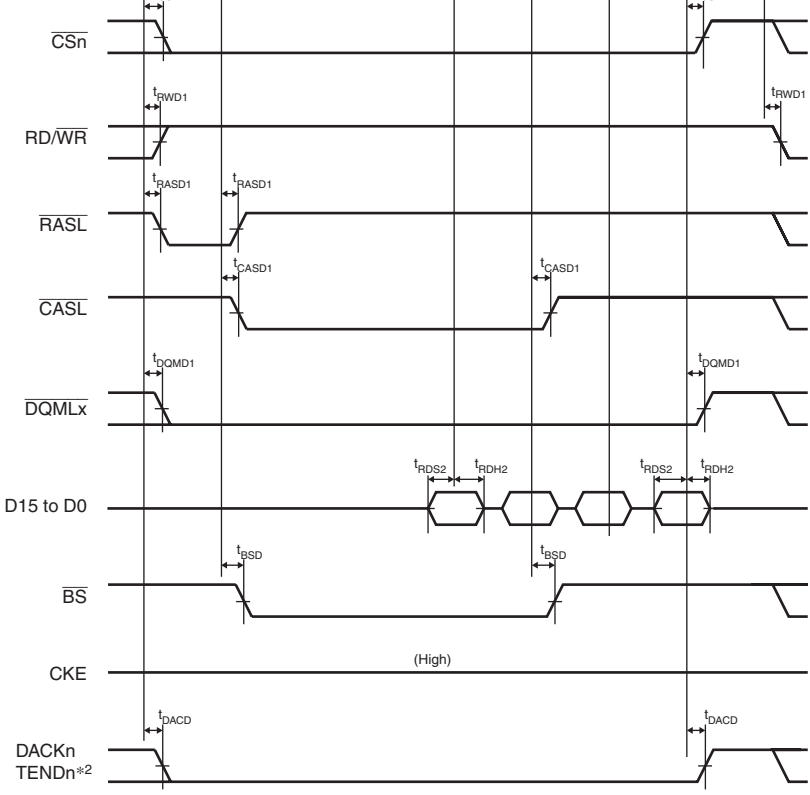
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.26 Synchronous DRAM Burst Write Bus Cycle (Four Write Cyc (Auto Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



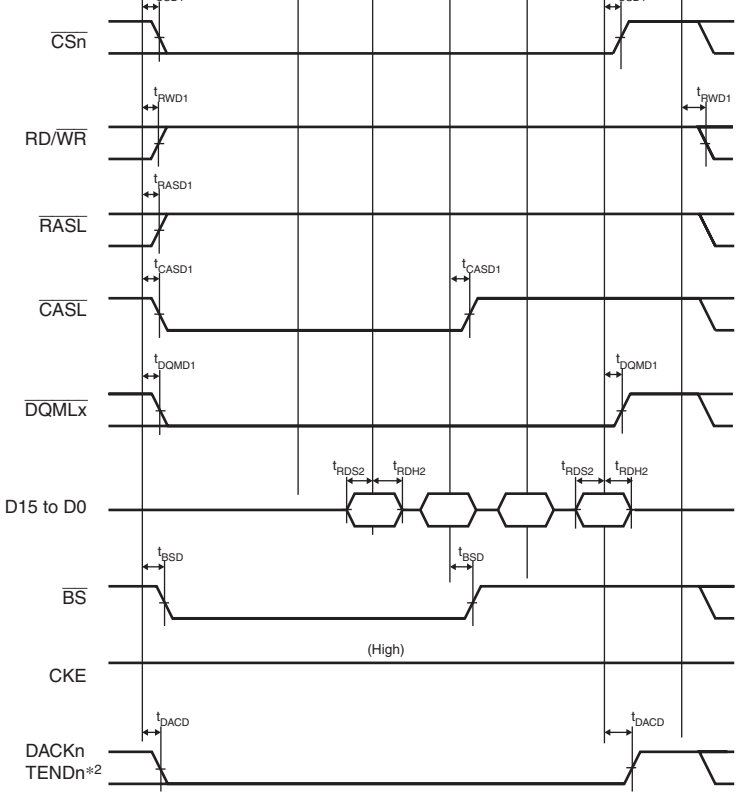
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.27 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycle (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)



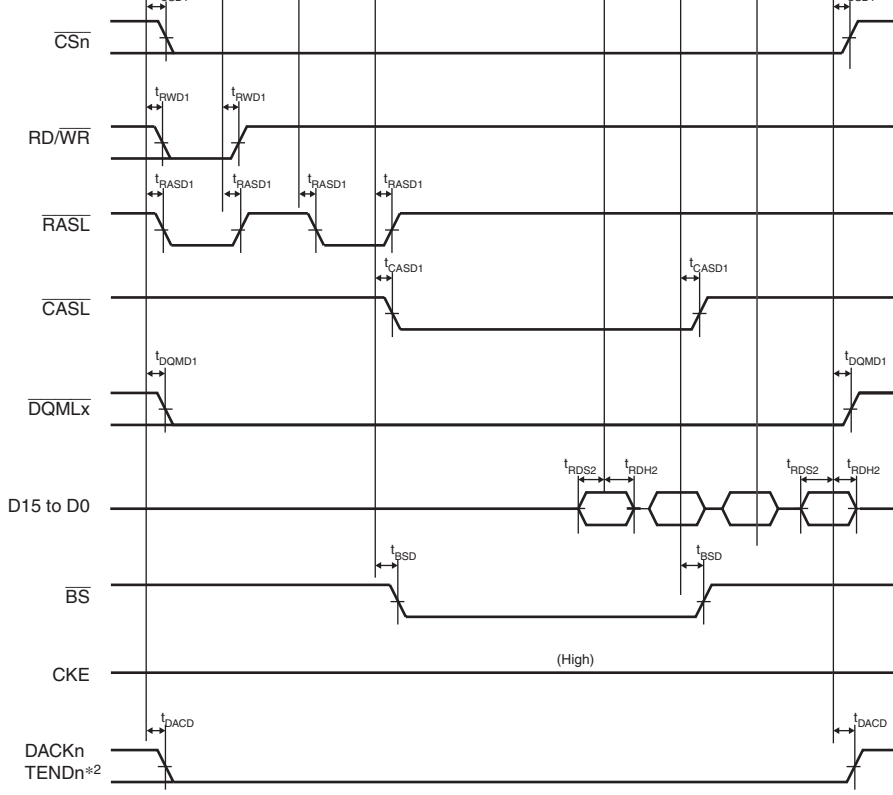
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.28 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycl
(Bank Active Mode: ACT + READ Commands, CAS Latency 2, WTRCD = 0 C



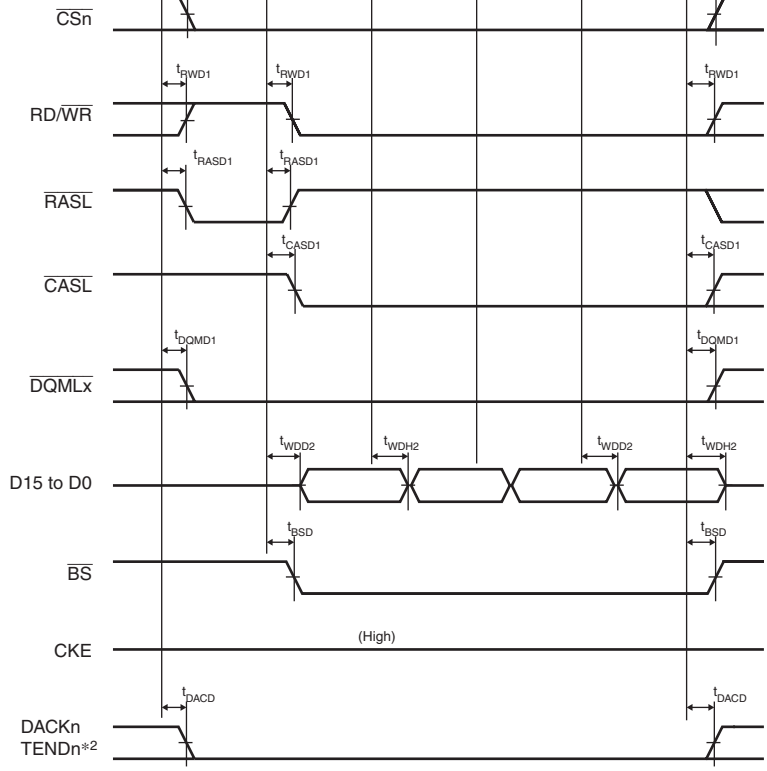
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.29 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycle
(Bank Active Mode: READ Command, Same Row Address, CAS Latency 2, WTRCD =



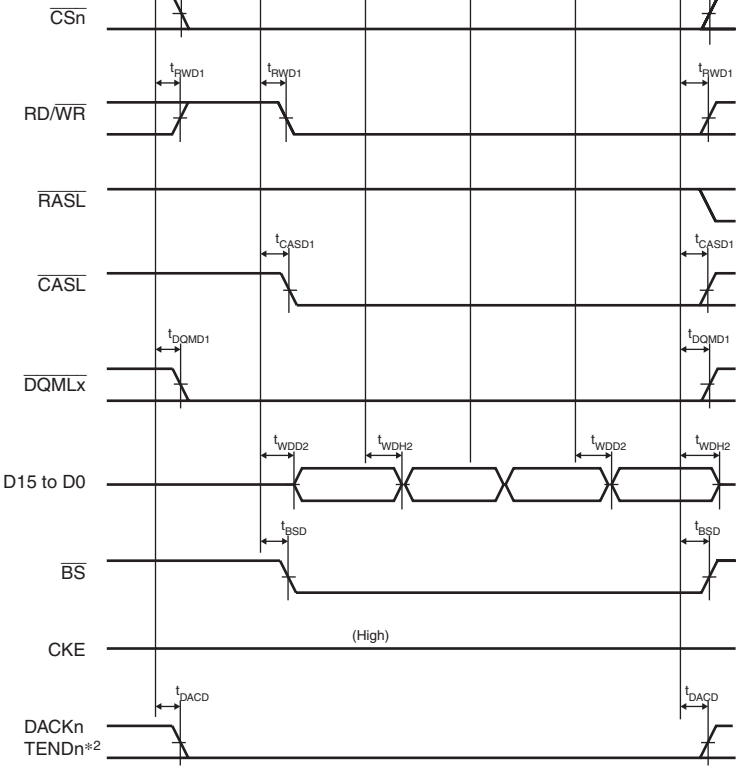
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.30 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycl (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addre CAS Latency 2, WTRCD = 0 Cycle)



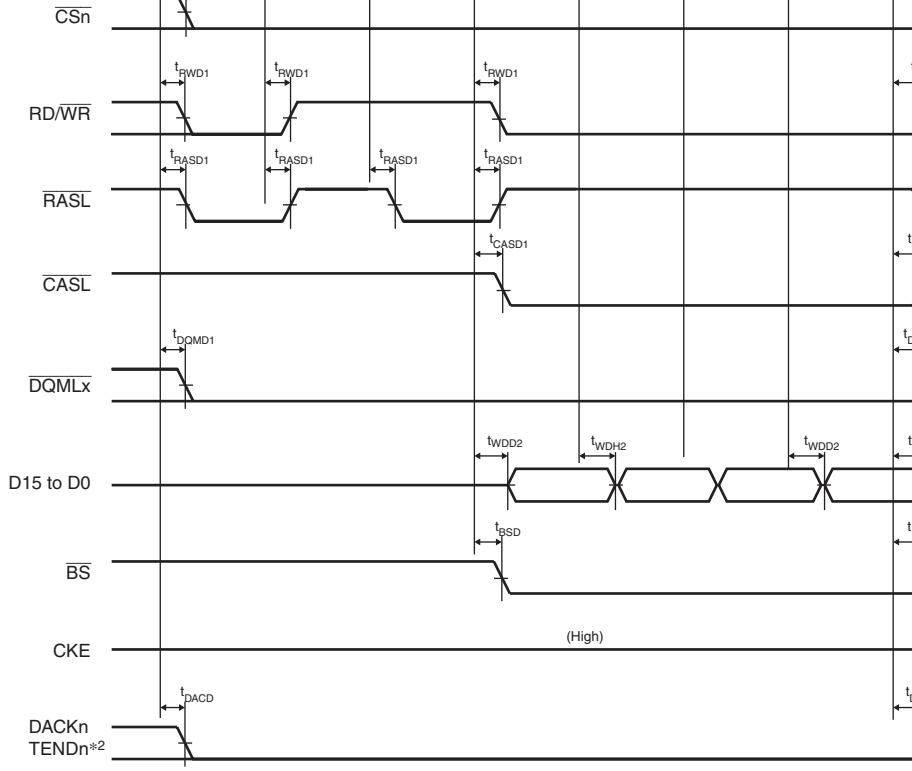
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.31 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycle
(Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0



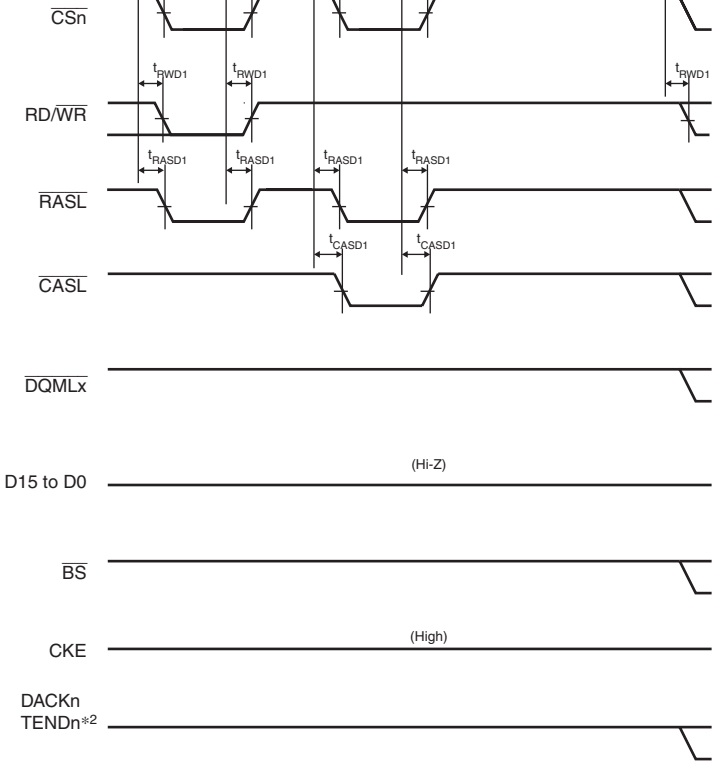
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.32 Synchronous DRAM Burst Write Bus Cycle (Four Write Cyc
(Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 C
TRWL = 0 Cycle)



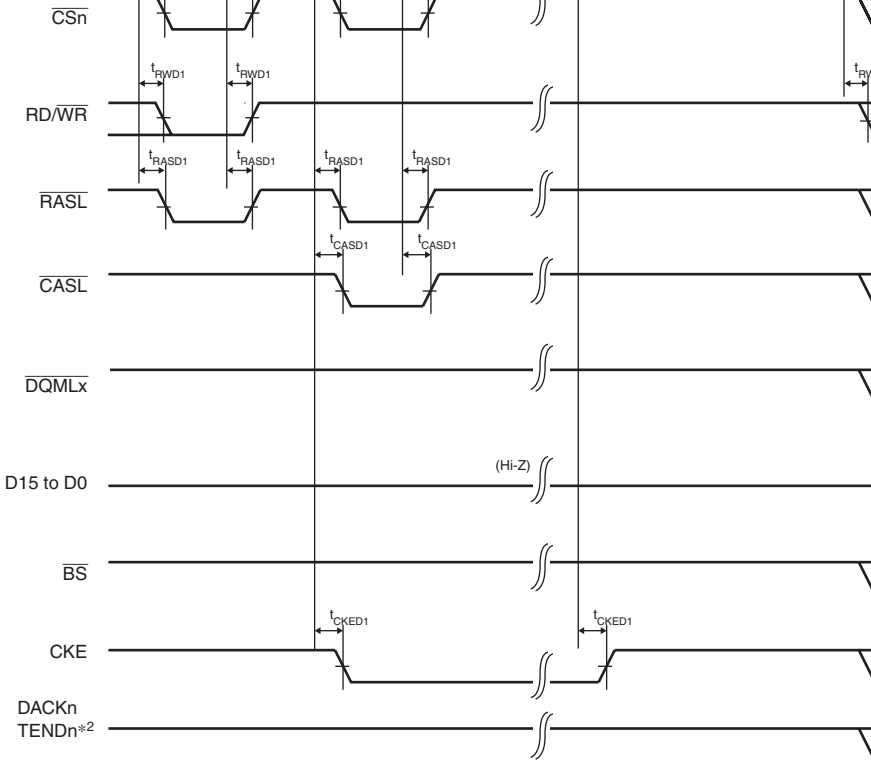
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.33 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycle (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addres WTRCD = 0 Cycle, TRWL = 0 Cycle)



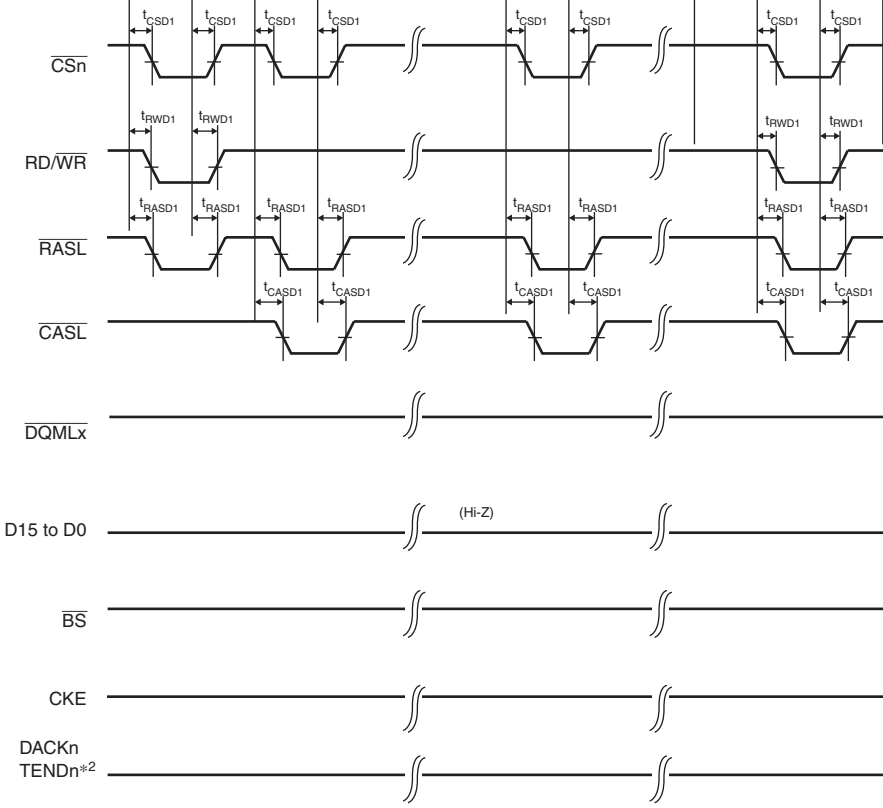
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.34 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



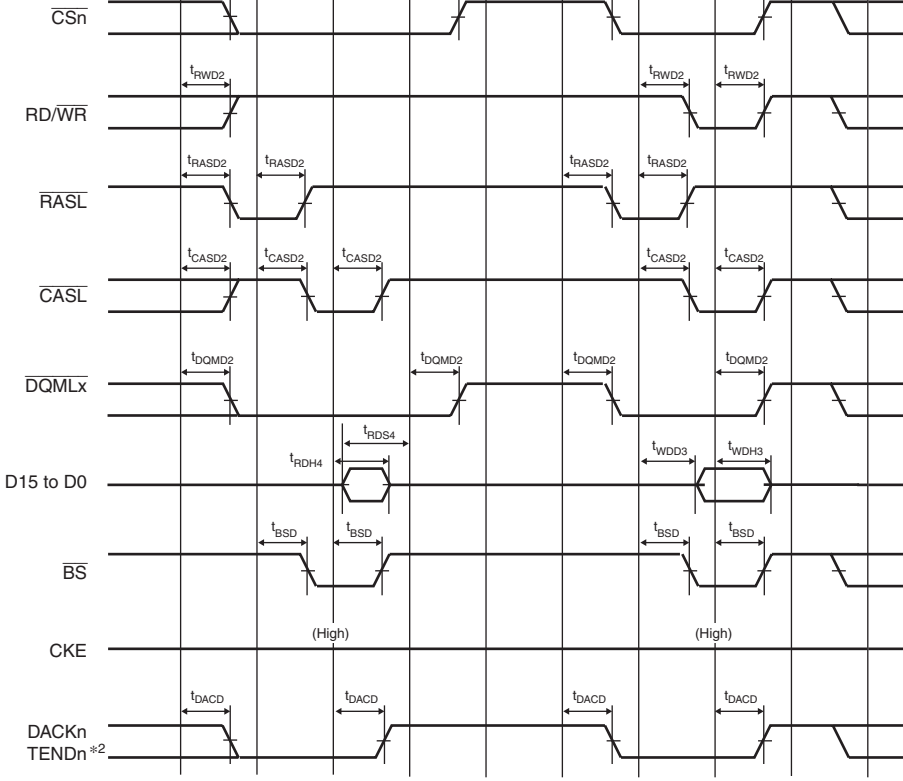
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.35 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)



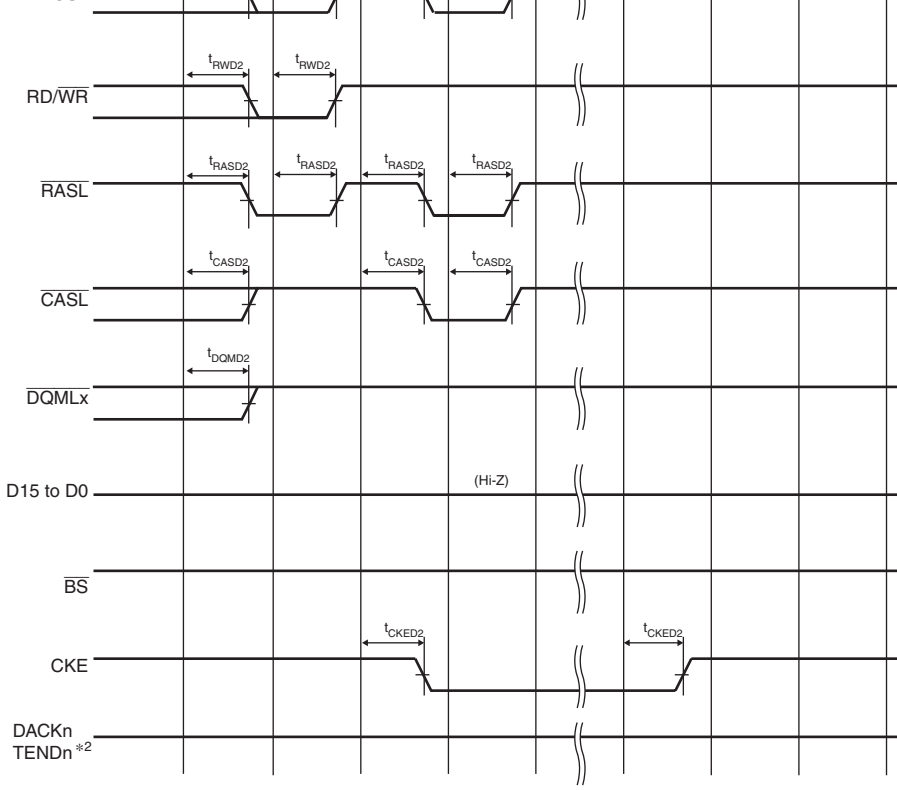
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.36 Synchronous DRAM Mode Register Write Timing (WTRP = 1 C



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.37 Synchronous DRAM Access Timing in Low-Frequency Mode (Auto-Precharge, TRWL = 2 Cycles)



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.38 Synchronous DRAM Self-Refreshing Timing in Low-Frequency (WTRP = 2 Cycles)

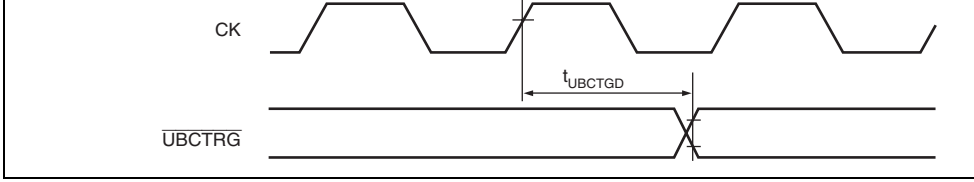


Figure 27.39 UBC Trigger Timing

27.4.5 DMAC Module Timing

Table 27.10 DMAC Module Timing

Conditions: Vcc = PLLVcc = 1.4 to 1.6 V, VccQ = 3.0 to 3.6 V, Vss = PLLVss = VssQ
Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	20	—	ns	Figure 27
DREQ hold time	t_{DRQH}	20	—		
DACK, TEND delay time	t_{DACD}	—	20		Figure 27

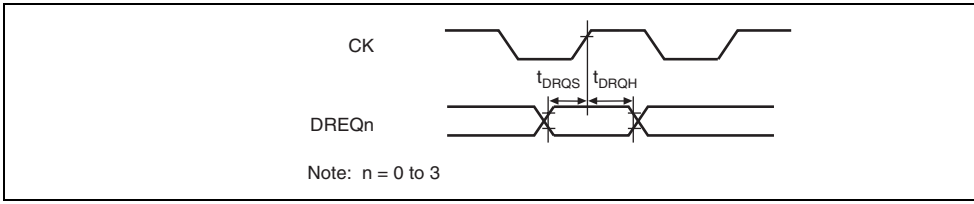


Figure 27.40 DREQ Input Timing

Timer input setup time	t_{TCKS}	t_{cyc} + 20	—	ns
Timer clock pulse width (single edge)	$t_{TCKWH/L}$	1.5	—	t_{pcyc}
Timer clock pulse width (both edges)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}
Timer clock pulse width (phase counting mode)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

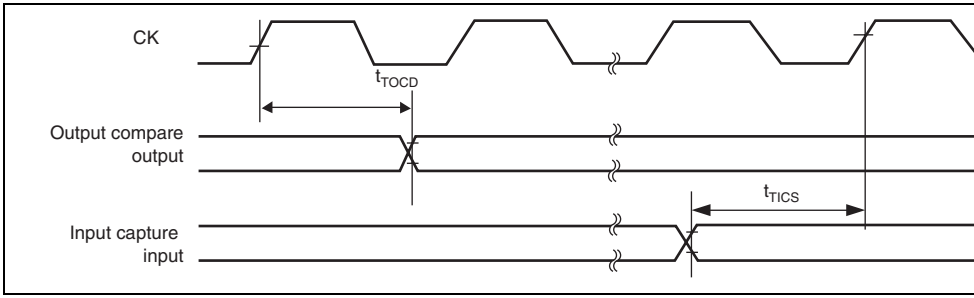


Figure 27.42 MTU2, MTU2S Input/Output Timing

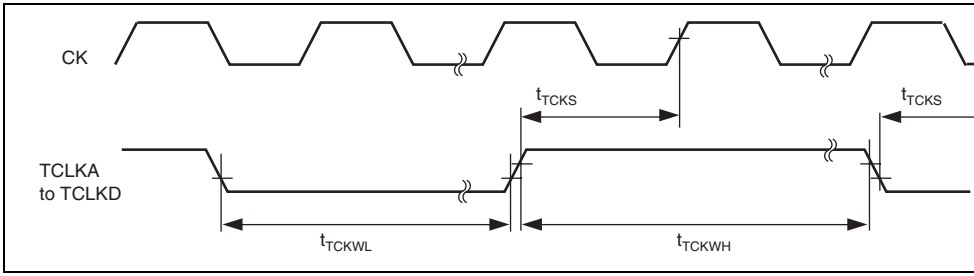


Figure 27.43 MTU2, MTU2S Clock Input Timing

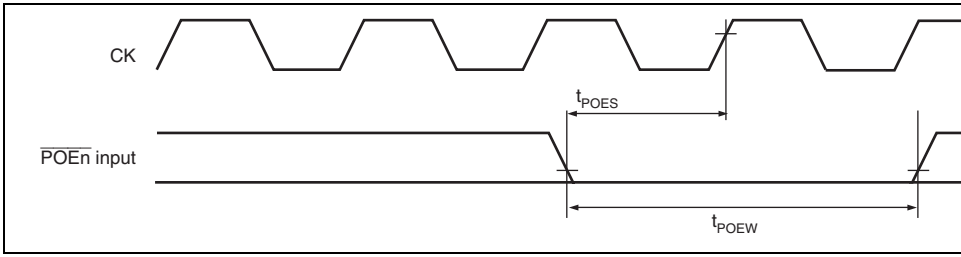


Figure 27.44 POE2 Input/Output Timing

27.4.8 Watchdog Timer Timing

Table 27.13 Watchdog Timer Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{WDTOVF}$ delay time	t_{WOVD}	—	50	ns	Figure 2

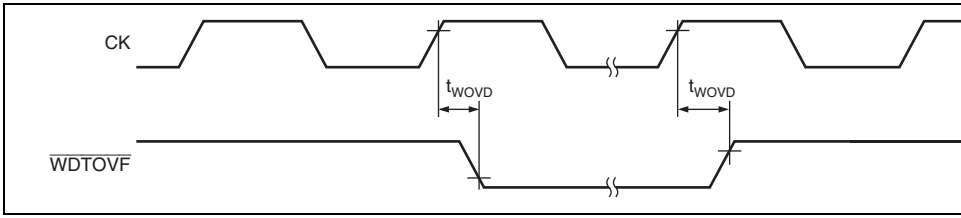


Figure 27.45 Watchdog Timer Timing

Input clock fall time	t_{SCKf}	—	1.5	t_{pcyc}	Figu
Input clock width	t_{SCKW}	0.4	0.6	t_{Scyc}	Figu
Transmit data delay time (clocked synchronous)	t_{TXD}	—	100	t_{pcyc}	Figu
Receive data setup time (clocked synchronous)	t_{RXS}	100	—	ns	Figu
Receive data hold time (clocked synchronous)	t_{RXH}	100	—	ns	Figu

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

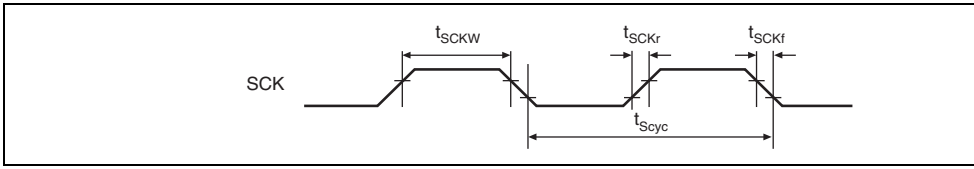


Figure 27.46 SCK Input Clock Timing

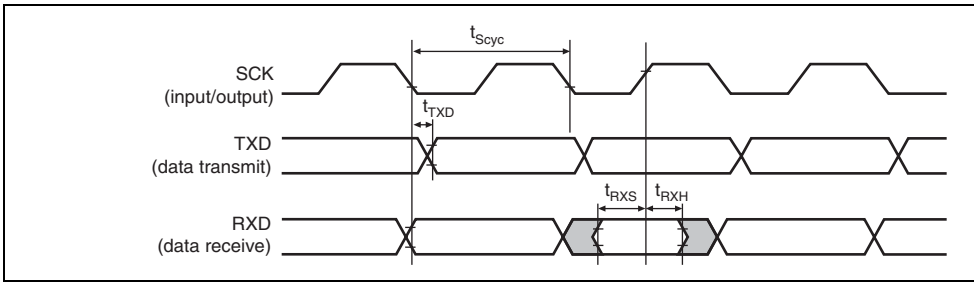


Figure 27.47 SCIF Input/Output Timing in Clocked Synchronous Mode

SCL input low pulse width	t_{SCLL}		$5\ t_{pcyc} + 300$	—	—	ns
SCL, SDA input rise time	t_{Sr}		—	—	300	ns
SCL, SDA input fall time	t_{Sf}		—	—	300	ns
SCL, SDA input spike pulse removal time*2	t_{SP}		—	—	$1\ t_{pcyc}$	ns
SDA input bus free time	t_{BUF}		5	—	—	t_{pcyc} *1
Start condition input hold time	t_{STAH}		3	—	—	t_{pcyc} *1
Retransmit start condition input setup time	t_{STAS}		3	—	—	t_{pcyc} *1
Stop condition input setup time	t_{STOS}		3	—	—	t_{pcyc} *1
Data input setup time	t_{SDAS}		$1\ t_{pcyc} + 20$	—	—	ns
Data input hold time	t_{SDAH}		0	—	—	ns
SCL, SDA capacitive load	Cb		0	—	400	pF
SCL, SDA output fall time*3	t_{Sf}	PV_{CC} = 3.0 to 3.6 V	—	—	300	ns

Notes: 1. t_{pcyc} indicates peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.

[Legend]
S: Start condition
P: Stop condition
Sr: Start condition for retransmission

Figure 27.48 I²C Bus Interface 3 Input/Output Timing

27.4.11 A/D Trigger Input Timing

Table 27.16 A/D Trigger Input Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Module	Item		Symbol	Min.	Max.	Unit	Figu
A/D converter	Trigger input setup time	B:P clock ratio = 1:1	t_{TRGS}	20	—	ns	Figur
		B:P clock ratio = 2:1		t_{cyc} + 20	—		
		B:P clock ratio = 4:1		3 × t_{cyc} + 20	—		

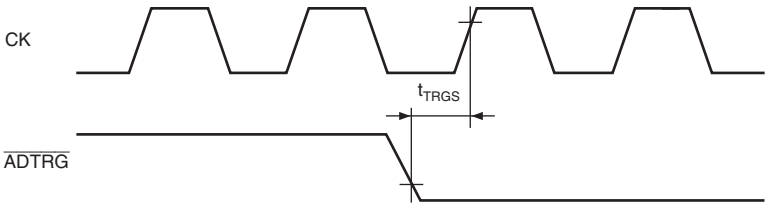


Figure 27.49 A/D Converter External Trigger Input Timing

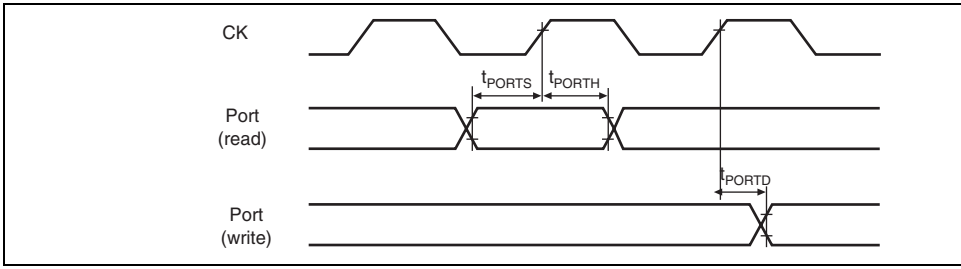


Figure 27.50 I/O Port Timing

TDI setup time	t_{TDIS}	15	—	ns	Figure 2
TDI hold time	t_{TDIH}	15	—	ns	
TMS setup time	t_{TMSS}	15	—	ns	
TMS hold time	t_{TMSH}	15	—	ns	
TDO delay time	t_{TDOD}	—	40	ns	

Note: * Should be greater than the peripheral clock (Pϕ) cycle time.

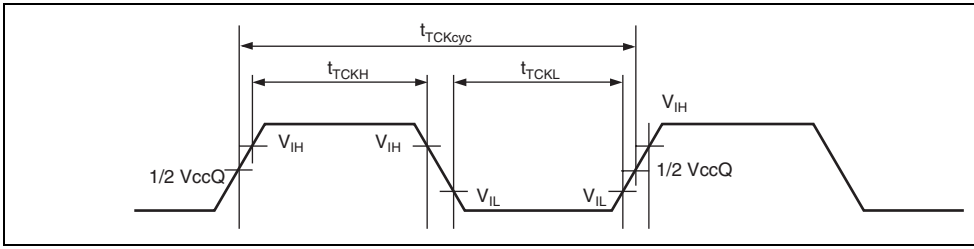


Figure 27.51 TCK Input Timing

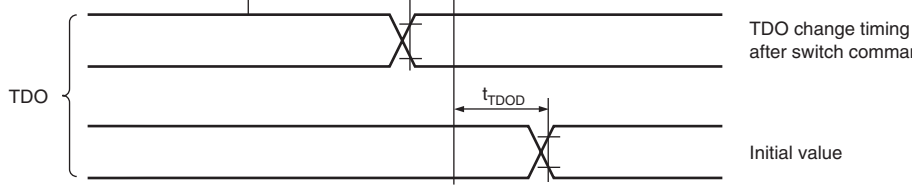
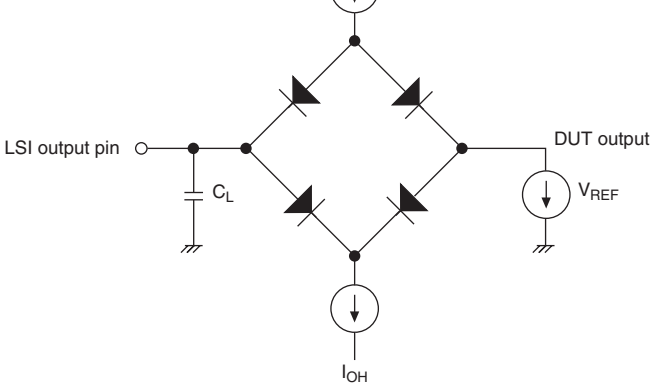


Figure 27.52 H-UDI Data Transfer Timing



Notes: 1. C_L is the total value that includes the capacitance of measurement tools. Each pin is set as follows:
75pF: CK
30pF: All pins
2. I_{OL} and I_{OH} are shown in table 27.4.

Figure 27.53 Output Load Circuit

Item	Min.	Typ.	Max.	U
Resolution	—	12	—	b
Conversion time per channel *2	1.25	—	—	µ
Analog input capacitance	—	—	5	p
Permissible signal-source impedance	—	—	3	k
Nonlinearity error	—	—	(±4.0)*1	L
Offset error	—	—	(±7.5)*1	L
Full-scale error	—	—	(±7.5)*1	L
Quantization error	—	—	(±0.5)*1	L
Absolute accuracy *3	—	—	±8.0	L

Notes: 1. The values in parentheses are reference values.

2. Conversion time per channel during continuous conversion. For the time from continuous conversion start to end, refer to section 17, A/D Converter (ADC).

3. The conversion error between 0 to 0.25 V of the AN0 to AN2 inputs and AVcc AVcc does not meet the above value.

Resolution	8	8	—	bits	
Conversion time	10	—	—	μs	Load capacitance
Absolute accuracy	—	±2.0	±3.0	LSB	Load resistance
	—	—	±2.5	LSB	Load resistance

		—	1200	3000	ms/128-K
Write time (total) *1*2*4	Σt_P	—	4.5	12	s/512 Kby
Erase time (total) *1*2*4	Σt_E	—	4.5	12	s/512 Kby
Write and erase time (total) *1*2*4	Σt_{PE}	—	9	24	s/512 Kby
Number of rewrite times	N_{WEC}	100*3	—	—	times

Notes: 1. Write time and erase time depend on data.
2. Data transfer time is not included in the write and erase time.
3. Minimum value that guarantees all characteristics after rewriting (guarantees range from 1 to Min. value).
4. Characteristics when the number of rewrite times falls within the range includ Min. value.

Type	Pin Name	Extended without ROM		Extended with ROM	Single chip	Manual	Software Standby	Sleep
		8 Bits	16 Bits					
Clock	CK (clock mode 6)	O			Z	O	O/Z*[4]	O
	XTAL (clock mode 6)	O				O	L	O
	EXTAL (clock mode 6)	I				I	I	I
System control	$\overline{\text{RES}}$	I				I	I	I
	$\overline{\text{MRES}}$	Z				I	I	I
	$\overline{\text{WDTOVF}}$	H				O	H	O
	$\overline{\text{BREQ}}$	Z				I	Z	I
	$\overline{\text{BACK}}$	Z				O	Z	O
Operating mode control	MD1, MD0	I				I	I	I
	MD_CLK2, MD_CLK0	I				I	I	I
Interrupt	NMI	I				I	I	I
	IRQ7 to IRQ0	Z				I	I	I
	$\overline{\text{IRQOUT}}$	Z				O	H/Z*[1]	O
Address bus	A25 to A0	O		Z		O	O/Z*[3]	O
Data bus	D15 to D0	Z				I/O	Z	I/O

	$\overline{BS}$	Z		O	H/Z*[4]	O
	$\overline{RD}$	H	Z	O	H/Z*[3]	O
	RD/$\overline{WR}$	Z		O	H/Z*[3]	O
	$\overline{WE0}$/$\overline{DQMLL}$	H	Z	O	H/Z*[3]	O
	$\overline{AH}$, $\overline{WE1}$/$\overline{DQMLU}$	Z		O	H/Z*[3]	O
	$\overline{RASL}$, $\overline{CASL}$	Z		O	O/Z*[2]	O
	CKE	Z		O	O/Z*[2]	O
	$\overline{REFOUT}$	Z		O	H/Z*[1]	O
DMAC	DREQ3 to DREQ0	Z		I	Z	I
	DACK3 to DACK0	Z		O	O/Z*[1]	O
	TEND1, TEND0	Z		O	O/Z*[1]	O
MTU2	TCLKA, TCLKB, TCLKC, TCLKD	Z		I	Z	I
	TIOC0A*[6], TIOC0B*[6], TIOC0C*[6], TIOC0D*[6]	Z		I/O	K/Z*[1]	I/O
	TIOC1A, TIOC1B	Z		I/O	K/Z*[1]	I/O
	TIOC2A, TIOC2B	Z		I/O	K/Z*[1]	I/O
	TIOC3A, TIOC3B*[6], TIOC3C, TIOC3D*[6]	Z		I/O	K/Z*[1]	I/O

	TIC5W				
MTU2S	TIOC3AS, TIOC3BS*6, TIOC3CS, TIOC3DS*6	Z	I/O	K/Z*1	I/O
	TIOC4AS*6, TIOC4BS*6, TIOC4CS*6, TIOC4DS*6	Z	I/O	K/Z*1	I/O
	TIC5US, TIC5VS, TIC5WS	Z	I	Z	I
POE2	$\overline{\text{POE8}}$, $\overline{\text{POE7}}$, $\overline{\text{POE4}}$, $\overline{\text{POE3}}$, $\overline{\text{POE1}}$, $\overline{\text{POE0}}$	Z	I	Z	I
SCIF	SCK3 to SCK0	Z	I/O	K/Z*1	I/O
	RXD3 to RXD0	Z	I	Z	I
	TXD3 to TXD0	Z	O	O/Z*1	O
WAVE	WSCK	Z	O	O/Z*1	O
	WRXD	Z	I	Z	I
	WTXD	Z	O	O/Z*1	O
A/D converter	AN7 to AN0	Z	I	Z	I
	$\overline{\text{ADTRG}}$	Z	I	Z	I
D/A converter	DA1, DA0	Z	O	O	O
IIC3	SCL	Z	I/O	Z	I/O
	SDA	Z	I/O	Z	I/O

	$\overline{\text{ASEMD}}$	I	I	I	I
	$\overline{\text{ASEBRK}}$/ $\overline{\text{ASEBRKAK}}$	O	O	I	O
	$\overline{\text{TRST}}$	I	I	I	I
	TCK	I	I	I	I
	TDI	I	I	I	I
	TDO	O/Z*5	O/Z*5	O/Z*5	O/Z*5
	TMS	I	I	I	I
UBC	$\overline{\text{UBCTRG}}$	Z	O	O/Z*1	O
I/O port	PA25 to PA0	Z	I/O	K/Z*1	I/O
	PB30 to PB22, PB21 to PB18*6, PB19 to PB12 PB13 to PB10*6, PB9, PB8, PB7 to PB4*6, PB3 to PB0	Z	I/O	K/Z*1	I/O
	PD15 to PD0	Z	I/O	K/Z*1	I/O
	PF1, PF0	Z	I	Z	I

[Legend]

I: Input

O: Output

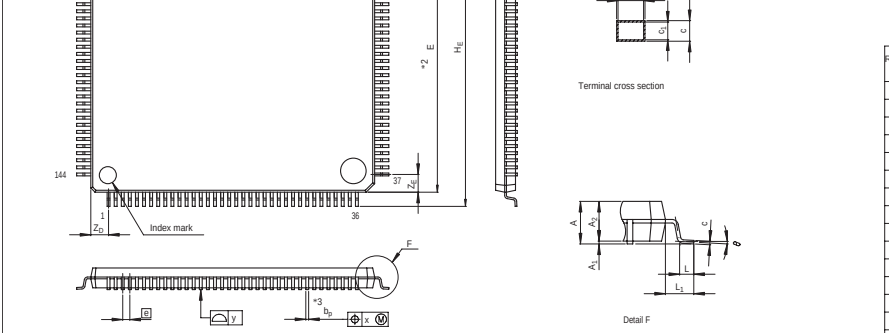
H: High-level output

L: Low-level output

Z: High-impedance

K: Input pins become high-impedance, and output pins retain their state.

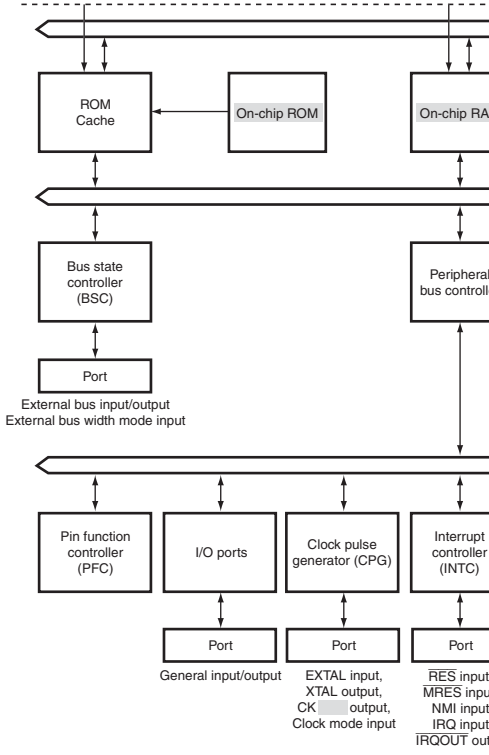
states at normal operation (see section 19, Pin Function Controller (PFC)).



Reference Symbol	D
D	1
E	1
A_2	-
H_D	2
H_E	2
A	-
A_1	0
b_p	0
b_1	-
c	0
c_1	-
θ	
e	-
x	-
y	-
Z_D	-
Z_E	-
L	0
L_1	-

Figure C.1 FP-144LV

1.2 Block Diagram Figure 1.1 Block Diagram	7	Figure amended



3.4 Address Map Figure 3.1 Address Map for Each Operating Mode (384-Kbyte On-Chip ROM Version)	58	Newly added

Bit	Bit Name	Initial Value	R/W	Description
2	AE	0	R/(W)*	Address Error Flag Indicates whether an address error has oc the DMAC. When this bit is set, even if the CHCR and the DME bit in DMAOR are set transfer is not enabled. This bit can only b writing 0 after reading 1. 0: No DMAC address error 1: DMAC address error occurred [Clearing condition] • Only write 0 to the AE bit after it has be 1. If the bit's value is 0 when read, writ

323 Table amended

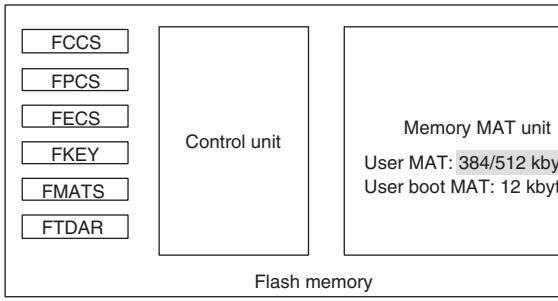
Bit	Bit Name	Initial Value	R/W	Description
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. W is set, even if the DE bit in CHCR and the DMAOR are set to 1, DMA transfer is not e bit can only be cleared by writing 0 after re When the NMI is input, the DMA transfer ir can be done in one transfer unit. Even if th interrupt is input while the DMAC is not in the NMIF bit is set to 1. 0: No NMI interrupt 1: NMI interrupt occurred [Clearing condition] • Only write 0 to the NMIF bit after it has as 1. If the bit's value is 0 when read, w

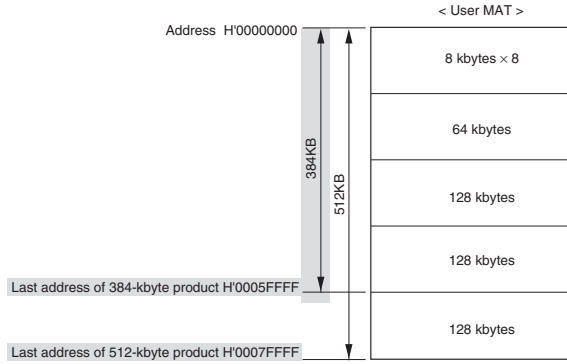
Note amended

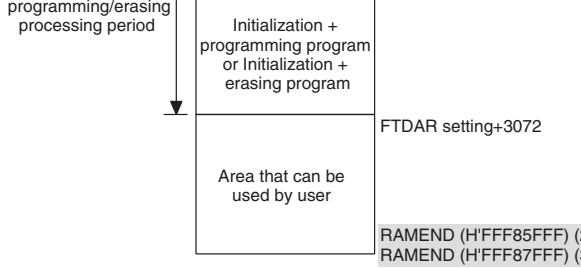
Note: * To clear flags, read the register and then write to the bits that were read as 1. Write 1 to the b were read as 0.

Voltage Range		
17.7.1 Relationship of AVcc and AVss to VccQ and VssQ	793	Description amended When using the A/D converter or D/A converter, mak such that AVcc = 5.0 V ±0.5 V and AVss = Vss. Whe converter and D/A converter are not used, make setti that AVcc = VccQ and AVss = VssQ, and do not leav AVcc and AVss pins open.
17.7.2 AVREF Pin Setting Range	793	Description amended When using the A/D converter or D/A converter, set A a level between 4.5 V and AVcc. When the A/D conv D/A converter are not used, make settings such that AVcc, and do not leave the AVREF pin open. The setting of the AVREFVss pin should always be s AVREFVss = AVss, and do not leave AVREFVss ope these conditions are not met, the reliability of the SH7 be adversely affected.
17.7.6 Treatment of AVcc and AVss When the A/D Converter is Not Used	—	Deleted

3		0	R	Reserved This bit is always read as 0. The write va always be 0.
2 to 0	PB28MD[2:0]	000	R/W	PB28 Mode Select the function of the PB28/DACK0/TIOC1A/RXD3 pin. 000: PB28 I/O (port) 001: Setting prohibited 010: Dack0 output (DMAC) 011: Setting prohibited 100: TIOC1A I/O (MTU2) 101: RXD3 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Section21 Flash Memory	885	Description amended This LSI has 384/512*kbyte on-chip flash memory. Th memory has the following features.
21.1 Features	885	Description amended Size of the user MAT, from which booting-up proceeds power-on reset in user mode: 384/512 kbytes* Note added Note: * See Appendix B. Product Lineup.
21.1.1 Block Diagram Figure 21.1 Block Diagram of Flash Memory	887	Figure amended 

		Address H'0005FFFF (When the size of the user MAT is 384 kbytes) Address H'0007FFFF (When the size of the user MAT is 512 kbytes)
		Note added Note: * See Appendix B. Product Lineup.
21.2.5 Block Division Figure 21.4 Block Division of User MAT	892	Figure amended 
21.2.6 Programming/Erasing Interface (4) Programming/Erasing Execution	894	Description amended The area to be programmed must be erased in advar programming flash memory. Ensure that NMI, IRQ, and all other interrupts are not generated during programming or erasing.



21.7.2 Interrupts during Programming/Erasing (2) Interrupts during Programming/Erasing	946	Description amended Ensure that NMI, IRQ, and all other interrupts are not generated during programming or erasing of on-chip p code.
21.8.2 Areas for Storage of the Procedural Program and Data for Programming	980	Description amended 5. The flash memory is not accessible during progra and erasing, so programs must be loaded into the RAM to perform these operations. Space in on-chi other than flash memory, or external bus space, m available for each procedure program for initiating programming or erasing, and for user programs at branch destinations during programming or erasin
Table 21.17 (1) Usable Area for Programming in User Program Mode	981	Table amended

	Storable/Executable Area			Selecte	
Item	On-Chip RAM	User MAT	External Space	User MAT	E P S M
Initialization error processing	√	√	√	√	
Writing H'5A to key register	√	√	√	√	

Item	On-Chip RAM	User Boot MAT	External Space	User MAT	User Boot MAT
Initialization error processing	√	√	√		√
Switching MATs by FMATS	√	X	X	√	

Table 21.17 (4) Usable Area for Erasure in User Boot Mode	985	Table amended

	Storable/Executable Area			Selected M	
Item	On-Chip RAM	User Boot MAT	External Space	User MAT	User Boot MAT
Initialization error processing	√	√	√		√
Switching MATs by FMATS	√	X	X		√

22.1 Features	987	Description amended • Pages The 32 Kbyte on-chip RAM is divided into four pa (pages 0 to 3). The 24 Kbyte on-chip RAM is divided into three p (pages 0 to 2). • Memory map The on-chip RAM is located in the address space in table 22.1, 22.2.
Table 22.2 24 Kbyte On-Chip RAM Address Spaces	987	Table added
23.3.5 System Control Register 1 (SYSCR1)	1001	Note added Note: This is a reserved bit on versions with 24 KB of value is always 1 when read. Always write 1 to

for the power consumption increase caused by sink cu
because each pin is placed in low-impedance state un
reaches the Min. voltage.

27.3 DC Characteristics 1084

Note amended

Table 27.3 DC Characteristics (1) [Common Items]

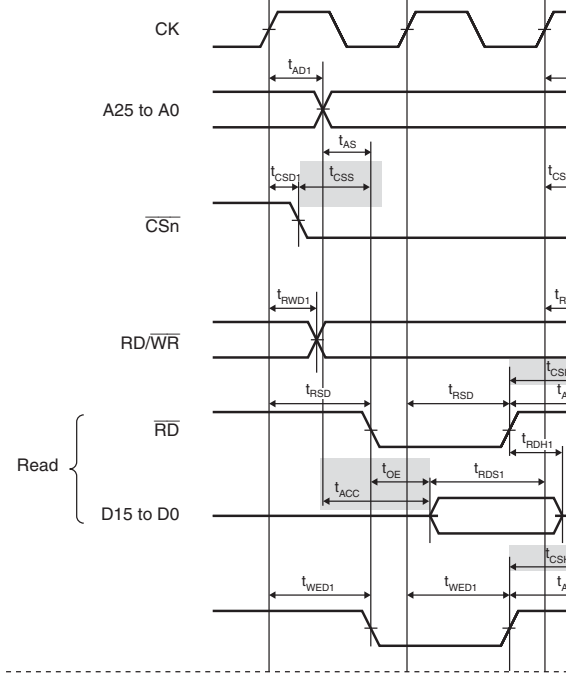
Caution: When neither the A/D converter nor the D/A converter is in use, do not leave the AVcc, AVREF, and AVREFVss pins open.

27.4.3 Bus Timing 1096, 1097

Table 27.8 Bus Timing

Table amended

		Bϕ = 40 MHz※1			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{CS}$ delay time 2	t_{CSD2}	$1/2t_{cyc}$	$1/2t_{cyc} + 20$	ns	Figures
$\overline{CS}$ setup time	t_{CSS}	0	—	ns	Figures 27.15
$\overline{CS}$ hold time	t_{CSH}	0	—	ns	Figures 27.15
Read write delay time 1	t_{RWD1}	1	20	ns	Figures 27.36
Read write delay time 2	t_{RWD2}	$1/2t_{cyc}$	$1/2t_{cyc} + 20$	ns	Figures
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc} + 20$	ns	Figures 27.19
Read data setup time 1	t_{RDS1}	$1/2t_{cyc} + 13$	—	ns	Figures 27.18
Read data hold time 4	t_{RDH4}	$1/2t_{cyc} + 5$	—	ns	Figure 2
Read data access time	t_{ACC} ※3	$t_{cyc} \times (n + 1.5)$ 31※2	—	ns	Figures 27.15, 2 27.18
Access time after read strobe	t_{OE} ※3	$t_{cyc} \times (n + 1)$ 31※2	—	ns	Figures 27.15, 2 27.18
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc} + 20$	ns	Figures 27.17
Write data hold time 1	t_{WDH1}	1	15	ns	Figures 27.18
Write data hold time 2	t_{WDH2}	1	—	ns	Figures 27.27, 2 27.33
Write data hold time 3	t_{WDH3}	$1/2t_{cyc}$	—	ns	Figure 2


CK
A25 to A0
CSn
RD/WR
Read
RD
D15 to D0
tAD1
tAS
tCSD1
tCSS
tRWD1
tRSD
tRSD
tRDH1
tOE
tRDS1
tACC
tWED1
tWED1

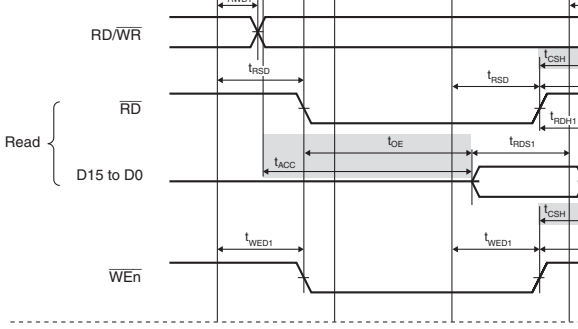
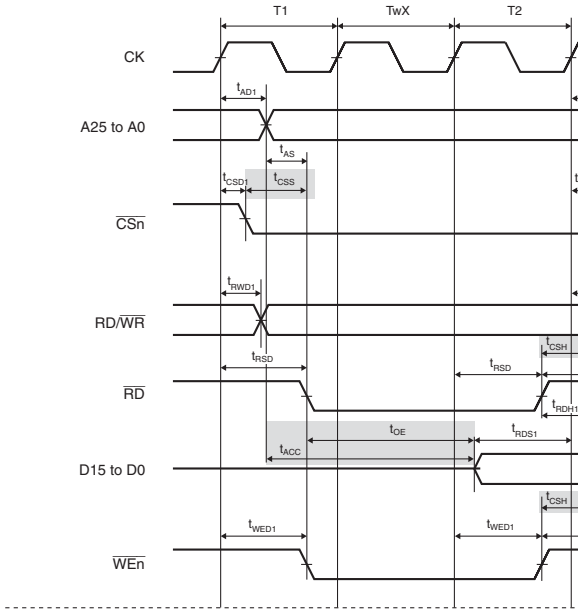


Figure 27.14 Basic Bus Timing for Normal Space (One External Wait Cycle)	1101	Figure amended



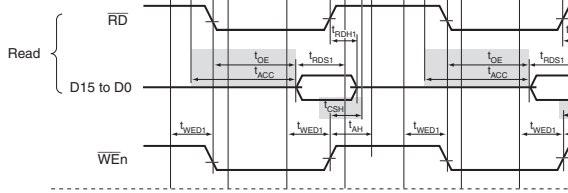
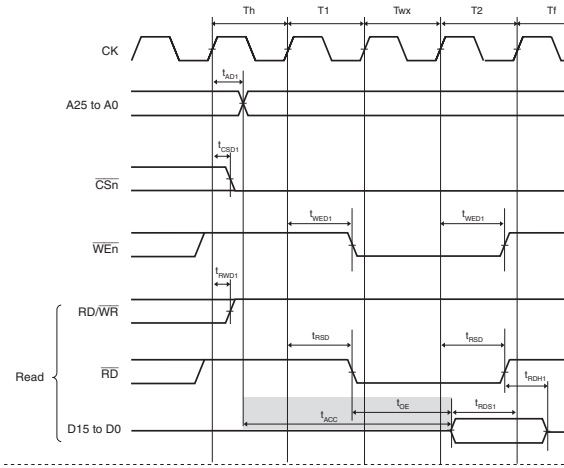
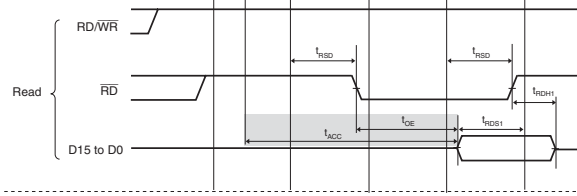


Figure 27.17 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Control))

1104

Figure amended





27.4.10 IIC3 Module Timing Table 27.15 I^2C Bus Interface 3 Timing	1131	Table amended

Item	Symbol	Test Conditions	Specifications Min.	Typ.	Max.	Unit
SCL input low pulse width	t_{SCLL}		5 t_{pcyc} + 300	—	—	ns
SCL, SDA input rise time	t_{sr}		—	—	300	ns
SCL, SDA input fall time	t_{sf}		—	—	300	ns
SCL, SDA input spike pulse removal time*2	t_{SP}		—	—	1 t_{pcyc}	ns

27.5 A/D Converter Characteristics	1137	Condition amended Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V, AVcc = 4.5 V to 5.5 V, AVREF = 4.5 V to AVcc, Vss = PLLVss = VssQ = AVss = AVREFVss = 0 V, Ta = –40°C to +85°C, V_{AN0-2} = 0.25 to AVcc –0.25 V, V_{AN3-7} = AVcc
27.6 D/A Converter Characteristics	1138	Condition amended Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V, 4.5 V to 5.5 V, AVREF = 4.5 V to AVcc, Vss = PLLVss = VssQ = AVss = AVREFVss = 0 V, Ta = –40°C to +85°C
Appendix B. Product Lineup Table B.1 Product Lineup	1146	Table amended

Product Type						
Product Name	Classification	ROM Capacity	RAM Capacity	Operating Temperature	Product Part No.	Pa (P
Sh7211	F-ZTAT version	384 Kbytes	24 Kbytes	40 to 85 C	R5F72114D160FPV	LC
		512 Kbytes	32 Kbytes	40 to 85 C	R5F72115D160FPV	(F

B

C

E

G

H

I

Q

R

T

V

W

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7211 Group

Publication Date:	Rev.1.00, October 30, 2006
	Rev.3.00, March 4, 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

SH7211 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0344-0300

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