PanaXSeries

The One to Watch for Constant Innovation-Making the Future Come Alive

MICROCOMPUTER MN103E

MN103E010H/040H

LSI User's Manual

Pub.No.23301-020E

Panasonic

Request for your special attention and precautions in using the technical information and semiconductors described in this book

(1) An export permit needs to be obtained from the competent authorities of the Japanese Government if any of the products or technologies described in this book and controlled under the "Foreign Exchange and Foreign Trade Law" is to be exported or taken out of Japan.

(2) The technical information described in this book is limited to showing representative characteristics and applied circuits examples of the products. It neither warrants non-infringement of intellectual property right or any other rights owned by our company or a third party, nor grants any license.

(3) We are not liable for the infringement of rights owned by a third party arising out of the use of the product or technologies as described in this book.

(4) The products described in this book are intended to be used for standard applications or general electronic equipment (such as office equipment, communications equipment, measuring instruments and household appliances).
Consult our sales staff in advance for information on the following applications:

- Special applications (such as for airplanes, aerospace, automobiles, traffic control equipment, combustion equipment, life support systems and safety devices) in which exceptional quality and reliability are required, or if the failure or malfunction of the products may directly jeopardize life or harm the human body.
- Any applications other than the standard applications intended.

(5) The products and product specifications described in this book are subject to change without notice for modification and/or improvement. At the final stage of your design, purchasing, or use of the products, therefore, ask for the most up-to-date Product Standards in advance to make sure that the latest specifications satisfy your requirements.

(6) When designing your equipment, comply with the guaranteed values, in particular those of maximum rating, the range of operating power supply voltage, and heat radiation characteristics. Otherwise, we will not be liable for any defect which may arise later in your equipment.
Even when the products are used within the guaranteed values, take into the consideration of incidence of break down and failure mode, possible to occur to semiconductor products. Measures on the systems such as redundant design, arresting the spread of fire or preventing glitch are recommended in order to prevent physical injury, fire, social damages, for example, by using the products.

(7) When using products for which damp-proof packing is required, observe the conditions (including shelf life and amount of time let standing of unsealed items) agreed upon when specification sheets are individually exchanged.

(8)

If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book.

Table of contents

Table of Contents

TABLE OF CONTENTS

FIGURES

TABLES

Chapter 1 Overview

1

CHAPTER 1

Overview

1.1. General

This LSI is a 32-bit microprocessor chip from the MN103E (AM33) Series, which is upward compatible with Matsushita's MN103S (AM30) Series of 32-bit microcontrollers.
The LSI is designed around a compact 32-bit CPU core with an instruction set that uses a basic instruction length of 1 byte. The microcontroller also includes an instruction cache, data cache, MMU, bus controller, system bus controller, memory bus controller, interrupt controller, timer, serial interface, DMA controller, A/D converter, analog front end interface, real-time clock, I2C controller, IrDA controller, and I/O ports, all are implemented in a 292-pin PBGA package in MN103E010HRA and in a 424-pin C-CSP in MN103E040HYB.
This microcontroller is suited for multimedia devices that must be able to process large volumes of data at fast speed (for audio, still images, video, etc.), and for real-time control devices that require fast and precise control.

1.2. Features

Operating Frequency

- 133MHz (internal: 1.8V; I/O: 3.3V)
- 3.3V, LVTTL level input interface (excluding some input signals)

AM33, CPU core

- Supports instructions that are upward compatible with the AM30/AM31/AM32 core.
- Supports AM33 extended instructions.

  LIW-type parallel instructions, extended operation instructions (sum-of-products instruction, etc.)
- Extended general-purpose registers

  Added four address registers (A0 to A3), four data registers (D0 to D3), and eight general-purpose registers (E0 to E7).
- Two operation modes (normal/debugging) and three privilege levels (user/supervisor/monitor)
- Load/store architecture with five-stage pipeline
- Proprietary high-speed branch processing techniques
- Supports a linear address space of up to 4GB.

FPU (Floating-point Operation Unit)

- Supports a data type that conforms with the IEEE754 standard.
- Supports rounding to the nearest value that conforms with the IEEE754 standard.
- 32 single-precision floating-point operation registers (FS0 to FS31)

  These registers can also be referenced as 16 double-precision floating-point operation registers (FD0 to FD30).
- Supports five floating-point operation exceptions that conform with the IEEE754 standard, and a floating-point unimplemented instruction exception.

MMU (Memory Management Unit)

- Memory storage protection function

  Permits setting separate access permission to a logical address space for supervisor level and user level.
- Address translation function

  Paging-based address translation (The page size can be to: 1KB/4KB/128KB/4MB.)
- TLB (instruction/data separation type)

  32 entries each for instruction and data; full associative.

Cache Memory

- Instruction cache
  Size: 16K Bytes, 4K Bytes x 4-way set associative
  Number of entries: 256; Line size: 16 bytes
  Pseudo LRU replacement algorithm (Each way)
  Each way has an entry lock function. (Conversion to RAM is possible for individual ways.)
- Data Cache
  Size: 16K Bytes, 4K Bytes x 4-way set associative
  Number of entries: 256; line size: 16 bytes
  Pseudo LRU replacement algorithm (each way)
  The writing policy can be switched between write-through and write-back (write allocate/write nonallocate)
  Each way has an entry lock function. (Conversion to RAM is possible for individual ways.)

On-chip RAM

- SRAM
  - 16K Bytes
  - Can be used as instruction RAM and data RAM.
  - Can be mapped to the instruction cache or data cache as a cacheable space.
  - Can be used in DMA transfer as either the transfer source or the transfer destination memory.

Clock Control

- Self-excited oscillation
  - Clock is supplied by connecting an oscillator.
- Low power consumption mode
  - Three modes are implemented: HALT, STOP and SLEEP

Bus Controller (BCU)

- Concurrent bus access
  - Permits concurrent access of four slave devices from three master devices.

System Bus Interface

- The external memory space can be allocated to eight banks.
  - A chip select signal is output for each bank.
  - The bus width can be set to 16 or 32 bits for each bank.
  - Each bank can be set for either fixed wait insertion and handshaking.
  - Each bank can be set for either synchronous mode (external bus clock synchronous)/asynchronous mode (external bus clock asynchronous)
- On-chip RAM, ROM, and SDRAM interface can be used by an external master.

Memory Bus Interface

- Internal SDRAM Direct Link Interface
  - Address multiplexing function
  - Programmable CAS latency setting
  - Refresh control

Internal Peripheral Functions

- Interrupts
  - 41 sources (42groups: 3 of them are system-reserved.)

- External interrupts: 9 sources (external interrupt pins (XIRQ) x 8, NMI pin (XNMI) interrupt x 1)
- Internal interrupts: 32 sources (asynchronous bus error: 1; WDT: 1; double timer: 14; DMAC: 4; SIO: 6; I2C: 2; IrDA: 1; AFE: 1; A/D: 1; real-time clock: 1)

❍ Timer

- 8-bit timers x 4 (all are down counters)

  Cascaded connection is permitted (can be used as a 16-, 24-, or 32-bit timer).
  Timer output support (with a duty ratio of 1:1).
  Either an internal clock source or an external clock source can be selected.
  Can be selected as the clock for the serial interface.
- 16-bit timer x 7 (down counter)

  Cascaded connection is permitted (can be used as a 32-bit timer).
  Timer output support (with a duty ratio of 1:1).
  Either an internal clock source or an external clock source can be selected.
  Some can be selected as the clock for the serial interface.
- 16-bit timer x 1 (up counter)

  Either an internal clock source or an external clock source can be selected.
  Provides an input capture function. (The rising edge, falling edge, or both edges can be selected.)
  Provides a PWM generation function (with two compare/capture registers built in).
- Watchdog timer x 1

❍ Serial interface

- UART/synchronous/I2C(dual-purpose) x 2 channels
- UART (with CTS control) x 1 channel

❍ DMAC

- Number of channels: 4 channels
- Unit of transfer: 1/2/4/16 bytes
- Maximum number of bytes that can be transferred: 1MB
- Initiation sources: External requests, interrupts, software
- Transfer format: 2 bus cycle transfer
- Transfer mode: Batch transfer, intermittent transfer
- Addressing mode

  "Fixed," "increment" and "decrement" can be specified for both the source and the destination address.
  Incrementing and decrementing are executed automatically according to the unit of transfer.

❍ Analog front end interface

- Interface for external AFE (analog front end) device
- Parallel-serial conversion for output data, and serial-parallel conversion for input data
- Internal send/receive FIFO (16 bits wide, 16 stages)
- NCU control through parallel I/O ports
- Eye pattern output

- Real-time clock
  - Clock/calendar function
  - Interrupts: periodic, alarm, update ended
  - BCD/binary support
  - Automatic compensation for leap years
  - 24-hour/12-hour system selectable
  - Daylight savings time support
- A/D converter
  - 10-bit load redistribution system (error +/- 5LSB)
  - Number of channels: 8-channel time division
- IrDA controller
  - IrDA 1.0 SIR (-115.2Kb/s, half-duplex)
  - IrDA 1.1 MIR (0.579, 1.152Mb/s, half-duplex)
  - IrDA 1.1 FIR (4.0Mbp/s, half-duplex)
  - UART (- 1.5Mbp/s, full-duplex)
  - 48MHz clock input (internal baud rate generation function)
- I2C interface
  - 2 ports
  - Master/slave interface (multi-master support)
  - 3.3V interface (open drain output)
- Parallel I/O
  - 6 ports

1.3. Block Diagram

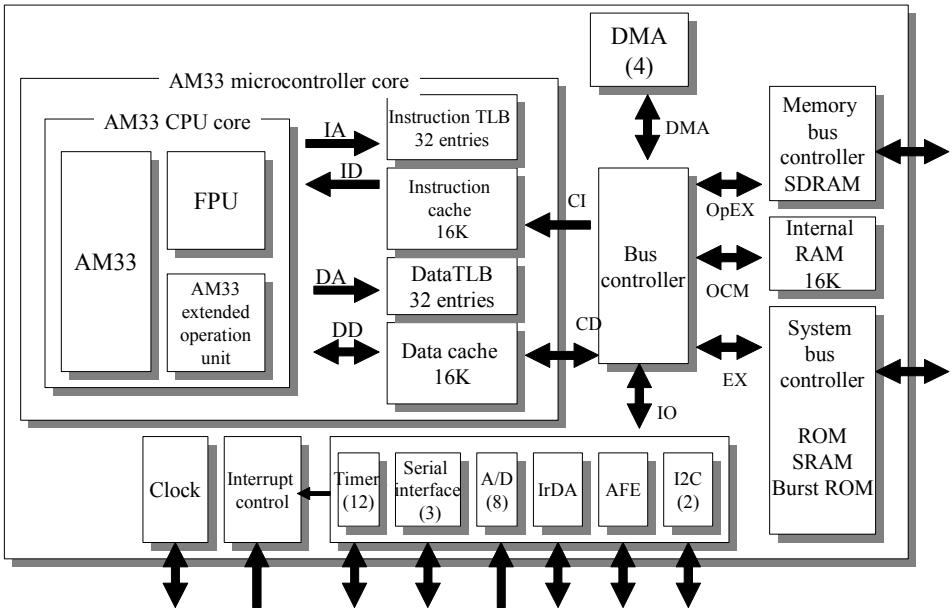


Figure 1 Block Diagram

1.4. Pin Descriptions Pin Assignments

1.4.1. Pin assingments

1.4.1.1. MN103E040HYB Pin assignments (Top view)

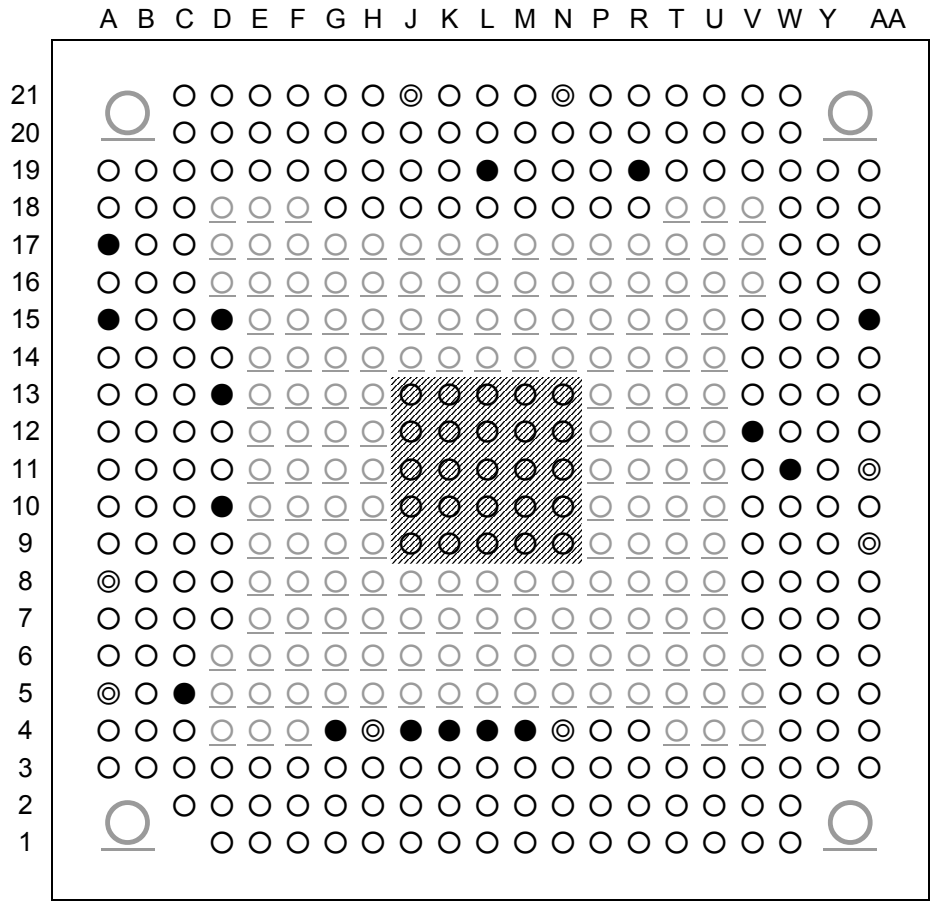
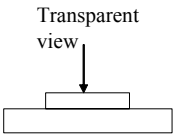


Figure 2 MN103E040HYB pin assignments



◎ 1.8 V-VDD

● 3.3 V-VDD(including AVDD,PVDD)

⊘ VSS (including AVSS, PVSS)

○ ND (Not Defined) has pins, but not guarantees NC (Not Connected).
Care should be taken not to cause a short with other wirings on the user substrate.
ND at the four corners are lands for reinforcing, and connecting to the printed

substrate is recommended.

- NP (No pin: This has no pins.)

Table 1 MN103E040HYB

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
A1	-	B1	-	C1	-	D1	MA2
A2	-	B2	-	C2	SA30	D2	MA3
A3	SA29	B3	SA31	C3	SA25	D3	SA20
A4	SA27	B4	SA26	C4	SA24	D4	-
A5	VDD18	B5	SA23	C5	VDD33	D5	-
A6	SA16	B6	SA19	C6	SA21	D6	-
A7	SA9	B7	SA13	C7	SA15	D7	SA22
A8	VDD18	B8	SA6	C8	SA7	D8	SA8
A9	XSBR	B9	SA1	C9	SA3	D9	SA14
A10	PIO2[1]	B10	SRXW	C10	SSZ1	D10	VDD33
A11	SYSCLK	B11	SA18	C11	SA12	D11	XSRE
A12	OSCO	B12	SA17	C12	SA11	D12	SA0
A13	OSCI	B13	SA10	C13	SA5	D13	VDD33
A14	PVSS	B14	SA4	C14	SA2	D14	XSCS5
A15	PVDD	B15	SSZ0	C15	XSBG	D15	VDD33
A16	TCPOUT	B16	PIO2[3]	C16	XSDK	D16	-
A17	RVDD	B17	PIO2[4]	C17	PIO2[2]	D17	-
A18	RCLKO	B18	PIO2[0]	C18	PWROK	D18	-
A19	RCLKI	B19	XSCS3	C19	XSWE0	D19	XSWE2
A20	-	B20	-	C20	XSWE3	D20	SD31
A21	-	B21	-	C21	XSWE1	D21	XSCS6

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
E1	MA6	F1	MA10	G1	MA9	H1	XMCS0
E2	MA5	F2	MA0	G2	MA8	H2	MA14
E3	SA28	F3	MA4	G3	MA1	H3	MA7
E4	-	F4	-	G4	VDD33	H4	VDD18
E5	-	F5	-	G5	-	H5	-
E6	-	F6	-	G6	-	H6	-
E7	-	F7	-	G7	-	H7	-
E8	-	F8	-	G8	-	H8	-
E9	-	F9	-	G9	-	H9	-
E10	-	F10	-	G10	-	H10	-
E11	-	F11	-	G11	-	H11	-
E12	-	F12	-	G12	-	H12	-
E13	-	F13	-	G13	-	H13	-
E14	-	F14	-	G14	-	H14	-
E15	-	F15	-	G15	-	H15	-
E16	-	F16	-	G16	-	H16	-
E17	-	F17	-	G17	-	H17	-
E18	-	F18	-	G18	SD26	H18	SD20
E19	XSCS0	F19	XSCS4	G19	SD28	H19	SD25
E20	XSCS7	F20	SD30	G20	XSAS	H20	SD23
E21	XSCS2	F21	XSCS1	G21	SD29	H21	SD27

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
J1	XMRAS	K1	SDCLK	L1	XMBE1	M1	MDK
J2	XMCS1	K2	SDCKE	L2	XMWE	M2	XMCAS
J3	MA13	K3	MA11	L3	MA12	M3	XMBE0
J4	VDD33	K4	VDD33	L4	VDD33	M4	VDD33
J5	-	K5	-	L5	-	M5	-
J6	-	K6	-	L6	-	M6	-
J7	-	K7	-	L7	-	M7	-
J8	-	K8	-	L8	-	M8	-
J9	VSS	K9	VSS	L9	VSS	M9	VSS
J10	VSS	K10	VSS	L10	VSS	M10	VSS
J11	VSS	K11	VSS	L11	VSS	M11	VSS
J12	VSS	K12	VSS	L12	VSS	M12	VSS
J13	VSS	K13	VSS	L13	VSS	M13	VSS
J14	-	K14	-	L14	-	M14	-
J15	-	K15	-	L15	-	M15	-
J16	-	K16	-	L16	-	M16	-
J17	-	K17	-	L17	-	M17	-
J18	SD13	K18	SD12	L18	SD7	M18	SD2
J19	SD24	K19	SD18	L19	VDD33	M19	SD15
J20	SD22	K20	SD17	L20	SA16	M20	SD9
J21	VDD18	K21	SD21	L21	SD19	M21	SD14

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
N1	MD7	P1	MD8	R1	MD5	T1	SDCKI
N2	MD6	P2	MD10	R2	MD11	T2	MD2
N3	MD9	P3	MD4	R3	MD12	T3	MD1
N4	VDD18	P4	TRCD2	R4	TRCD7	T4	-
N5	-	P5	-	R5	-	T5	-
N6	-	P6	-	R6	-	T6	-
N7	-	P7	-	R7	-	T7	-
N8	-	P8	-	R8	-	T8	-
N9	VSS	P9	-	R9	-	T9	-
N10	VSS	P10	-	R10	-	T10	-
N11	VSS	P11	-	R11	-	T11	-
N12	VSS	P12	-	R12	-	T12	-
N13	VSS	P13	-	R13	-	T13	-
N14	ND	P14	-	R14	-	T14	-
N15	ND	P15	-	R15	-	T15	-
N16	ND	P16	-	R16	-	T16	-
N17	ND	P17	-	R17	-	T17	-
N18	SD0	P18	SD5	R18	PIO0[1]	T18	-
N19	SD11	P19	SD10	R19	VDD33	T19	PIO0[7]
N20	SD8	P20	SD3	R20	SD4	T20	PIO5[1]
N21	VDD18	P21	SD6	R21	SD1	T21	CLK48

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
U1	MD3	V1	MD13	W1	MD0	Y1	-
U2	MD14	V2	TRCD3	W2	TRCD6	Y2	-
U3	MD15	V3	TRCD4	W3	TRCD0	Y3	EXTRG

U4	-	V4	-	W4	TRCD5	Y4	TMS
U5	-	V5	-	W5	TRCD1	Y5	PIO4[3]
U6	-	V6	-	W6	PIO4[1]	Y6	PIO3[4]
U7	-	V7	TRSTMOD	W7	PIO4[0]	Y7	PIO4[2]
U8	-	V8	PIO3[0]	W8	PIO3[1]	Y8	SBT0
U9	-	V9	PIO3[2]	W9	PIO3[3]	Y9	XRSTOUT
U10	-	V10	SBI1	W10	SBI2	Y10	SBO2
U11	-	V11	TRCST	W11	VDD33	Y11	XIRQ3
U12	-	V12	VDD33	W12	XRESET	Y12	PIO0[0]
U13	-	V13	SBO1	W13	XIRQ0	Y13	PIO0[6]
U14	-	V14	SBO0	W14	XIRQ5	Y14	XNMI
U15	-	V15	XIRQ4	W15	SBT2	Y15	PIO0[4]
U16	-	V16	-	W16	PIO0[3]	Y16	XIRQ7
U17	-	V17	-	W17	AN6	Y17	AN0
U18	-	V18	-	W18	AN5	Y18	PIO0[5]
U19	PIO5[0]	V19	AN1	W19	AN3	Y19	AN7
U20	PIO5[2]	V20	PIO1[4]	W20	PIO1[2]	Y20	-
U21	PIO1[3]	V21	PIO1[1]	W21	PIO1[0]	Y21	-

No.	Pin name
AA1	-
AA2	-
AA3	TRCCLK
AA4	TDO
AA5	TDI
AA6	TCK
AA7	SBI0
AA8	SBT1
AA9	VDD18
AA10	XIRQ1
AA11	VDD18
AA12	XIRQ6
AA13	PIO0[2]
AA14	XIRQ2
AA15	AVDD
AA16	AN4
AA17	AN2
AA18	AVSS
AA19	VREFH
AA20	-
AA21	-

1.4.1.2. MN103E010HRA Pin assignments (Top view)

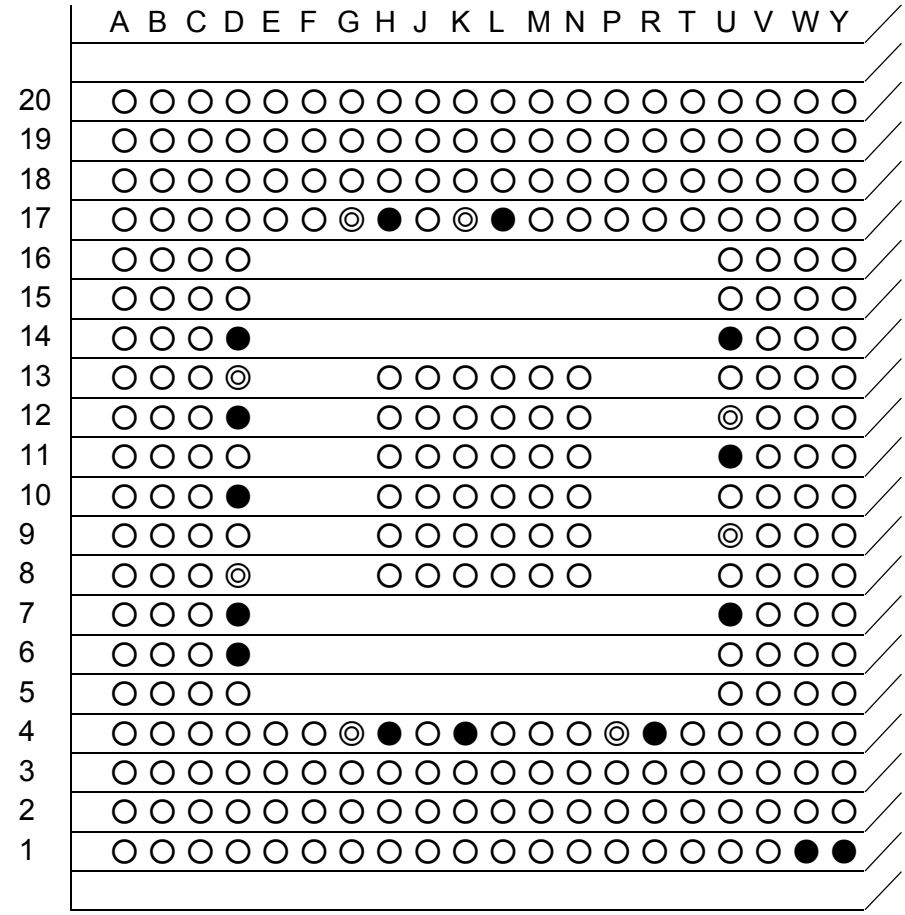


Figure 3 MN103E010HRA pin assignments

◎ 1.8 V-DD

● 3.3 V-DD(including AVDD, VDD, RVDD)

◎ VSS

–

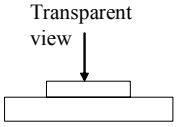


Table 2 MN103E010HRA pin assignments

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
A1	SA30	B1	SA29	C1	SA27	D1	SA24
A2	MA2	B2	MA3	C2	SA28	D2	SA25

A3	MA6	B3	MA5	C3	SA31	D3	SA26
A4	MA10	B4	MA0	C4	MA4	D4	VSS
A5	MA9	B5	MA8	C5	MA1	D5	VSS
A6	XMCS0	B6	MA14	C6	MA7	D6	VDD33
A7	XMRAS	B7	XMCS1	C7	MA13	D7	VDD33
A8	SDCLK	B8	SDCKE	C8	MA11	D8	VDD18
A9	XMBE1	B9	XMWE	C9	MA12	D9	VSS
A10	MDK	B10	XMCAS	C10	XMBE0	D10	VDD33
A11	MD7	B11	MD6	C11	MD9	D11	VSS
A12	MD8	B12	MD10	C12	MD4	D12	VDD33
A13	MD5	B13	MD11	C13	MD12	D13	VDD18
A14	SDCKI	B14	MD2	C14	MD1	D14	VDD33
A15	MD3	B15	MD14	C15	MD15	D15	VSS
A16	MD13	B16	TRCCLK	C16	TRCST	D16	VSS
A17	MD0	B17	TRCD2	C17	TRCD5	D17	VSS
A18	TRCD7	B18	TRCD4	C18	TRCD0	D18	TDI
A19	TRCD3	B19	EXTRG	C19	TCK	D19	PIO4[0]
A20	TRCD6	B20	TRCD1	C20	TDO	D20	TRSTMOD

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
E1	SA20	F1	SA17	G1	SA14	H1	SA11
E2	SA21	F2	SA18	G2	SA15	H2	SA12
E3	SA23	F3	SA19	G3	SA16	H3	SA13
E4	VSS	F4	SA22	G4	VDD18	H4	VDD33
E5	-	F5	-	G5	-	H5	-
E6	-	F6	-	G6	-	H6	-
E7	-	F7	-	G7	-	H7	-
E8	-	F8	-	G8	-	H8	VSS
E9	-	F9	-	G9	-	H9	VSS
E10	-	F10	-	G10	-	H10	VSS
E11	-	F11	-	G11	-	H11	VSS
E12	-	F12	-	G12	-	H12	VSS
E13	-	F13	-	G13	-	H13	VSS
E14	-	F14	-	G14	-	H14	-
E15	-	F15	-	G15	-	H15	-
E16	-	F16	-	G16	-	H16	-
E17	VSS	F17	PIO4[3]	G17	VDD18	H17	VDD33
E18	TMS	F18	PIO4[1]	G18	PIO3[1]	H18	PIO3[4]
E19	PIO3[0]	F19	PIO3[3]	G19	SBO0	H19	SBI1
E20	PIO4[2]	F20	PIO3[2]	G20	SBI0	H20	SBO1

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
J1	SA7	K1	SA5	L1	SA4	M1	SA3
J2	SA8	K2	SA6	L2	SA2	M2	SSZ0
J3	SA9	K3	SA10	L3	SA1	M3	XSBG
J4	VSS	K4	VDD33	L4	SA0	M4	XSBR
J5	-	K5	-	L5	-	M5	-
J6	-	K6	-	L6	-	M6	-
J7	-	K7	-	L7	-	M7	-
J8	VSS	K8	VSS	L8	VSS	M8	VSS
J9	VSS	K9	VSS	L9	VSS	M9	VSS
J10	VSS	K10	VSS	L10	VSS	M10	VSS
J11	VSS	K11	VSS	L11	VSS	M11	VSS
J12	VSS	K12	VSS	L12	VSS	M12	VSS
J13	VSS	K13	VSS	L13	VSS	M13	VSS
J14	-	K14	-	L14	-	M14	-
J15	-	K15	-	L15	-	M15	-
J16	-	K16	-	L16	-	M16	-
J17	VSS	K17	VDD18	L17	VDD33	M17	VSS
J18	SBT0	K18	SBO2	L18	XRESET	M18	XIRQ3
J19	SBI2	K19	XIRQ0	L19	XIRQ2	M19	XNMI
J20	SBT1	K20	SBT2	L20	XRSTOUT	M20	XIRQ1

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
N1	SSZ1	P1	XSDK	R1	SYSCLK	T1	OSCO
N2	SRXW	P2	PIO2[2]	R2	PIO2[1]	T2	TCPOUT
N3	PIO2[4]	P3	PIO2[3]	R3	PIO2[0]	T3	VSS
N4	XSRE	P4	VDD18	R4	VDD33	T4	VSS
N5	-	P5	-	R5	-	T5	-
N6	-	P6	-	R6	-	T6	-
N7	-	P7	-	R7	-	T7	-
N8	VSS	P8	-	R8	-	T8	-
N9	VSS	P9	-	R9	-	T9	-
N10	VSS	P10	-	R10	-	T10	-
N11	VSS	P11	-	R11	-	T11	-
N12	VSS	P12	-	R12	-	T12	-
N13	VSS	P13	-	R13	-	T13	-
N14	-	P14	-	R14	-	T14	-
N15	-	P15	-	R15	-	T15	-
N16	-	P16	-	R16	-	T16	-
N17	PIO0[3]	P17	VSS	R17	PIO0[7]	T17	AN5
N18	XIRQ6	P18	PIO0[1]	R18	PIO0[5]	T18	AN7
N19	XIRQ7	P19	PIO0[2]	R19	PIO0[6]	T19	AN6
N20	XIRQ4	P20	XIRQ5	R20	PIO0[0]	T20	PIO0[4]

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
U1	OSCI	V1	PVSS	W1	PVDD	Y1	RVDD
U2	PWROK	V2	XSWE2	W2	XSWE1	Y2	RCLKO
U3	XSWE3	V3	XSWE0	W3	XSCS7	Y3	RCLKI
U4	VSS	V4	XSCS6	W4	XSCS4	Y4	XSCS5
U5	VSS	V5	XSCS3	W5	XSCS0	Y5	XSCS1
U6	XSCS2	V6	XSAS	W6	SD30	Y6	SD31
U7	VDD33	V7	SD29	W7	SD27	Y7	SD28
U8	VSS	V8	SD25	W8	SD24	Y8	SD26
U9	VDD18	V9	SD21	W9	SD20	Y9	SD22
U10	VSS	V10	SD23	W10	SD18	Y10	SD19
U11	VDD33	V11	SD14	W11	SD17	Y11	SD16
U12	VDD18	V12	SD11	W12	SD15	Y12	SD13
U13	VSS	V13	SD8	W13	SD12	Y13	SD10
U14	VDD33	V14	SD5	W14	SD9	Y14	SD7
U15	SD2	V15	SD1	W15	SD6	Y15	SD4
U16	VSS	V16	PIO1[4]	W16	SD3	Y16	SD0
U17	PIO1[0]	V17	PIO1[3]	W17	PIO5[1]	Y17	CLK48
U18	AN3	V18	AN0	W18	PIO1[1]	Y18	PIO5[2]
U19	AN2	V19	AN1	W19	PIO1[2]	Y19	PIO5[0]
U20	AVDD	V20	AN4	W20	AVSS	Y20	VREFH

1.4.2. Pin Functions

Table 3 Pin functions of MN103E010/040

Category	Pin name	I/O	Number of pins	Also serves as	Description
Power supply/ground	VDD33	PWR	13	-	3.3V (I/O) digital power supplies
	VDD18	PWR	8	-	1.8V (core) digital power supplies
	VSS	PWR	21	-	Digital grounds
	VREFH	PWR	1	-	Reference power supply for A/D converter When the A/D conversion function is not used, connect to 3.3V digital power supplies.
	AVDD	PWR	1	-	3.3V analog power supply for A/D converter. If the A/D conversion function is not to be used, connect this pin to the 3.3V digital power supply.
	AVSS	PWR	1	-	Analog ground for A/D conversion If the A/D converter function is not to be used, connect this pin to the digital ground.
	PVDD	PWR	1	-	3.3V power supply for PLL circuit
	PVSS	PWR	1	-	Ground for PLL circuit

Category	Pin name	I/O	Number of pins	Also serves as	Description
	RVDD	PWR	1	-	3.3V power supply for real-time clock Connect the backup battery or VDD33 to this pin. The ground for the real-time clock is shared by the digital ground (VSS).
Clock	OSCI	I	1	-	Oscillation input
	OSCO	O	1	-	Oscillation output
	TCPOUT	O	1	-	PLL test output (analog output) This is the internal PLL test pin.
	SYSCLK	O	1	-	System bus clock output This is the reference clock for synchronous bus mode.
	CKIO	I	1	PIO2[3]	Test clock mode setting The test mode is set according to the state of the CKIO pin when the reset condition on the XRESET pin is released. Under normal use, this pin should be set to HIGH level when the reset condition on the XRESET pin is released.
	FRQS[1:0]	I	2	PIO1[4:3]	PLL MULTIPLIER MODE SETTING The PLL multiplier mode is switched according to the state of the FRQS pins when the reset condition on the XRESET pin is released. For details, refer to the chapter on the clock generator.
Mode control	CMOD	I	1	PIO2[4]	Test mode setting The CPU is set to test mode according to the state of the CMOD pin when the reset condition on the XRESET pin is released. Under normal use, this pin should be set to LOW level when the reset condition on the XRESET pin is released.
Reset	XRESET	I	1	-	Reset input This is the chip reset input pin. The chip is reset when this pin is low.
	XRSTOUT	O	1	-	Reset output This pin is driven low while the CPU is in the reset state in response to a reset through the XRESET pin or due to an internal source (including a software reset). Use this pin as a reset signal for external devices that should be reset by these reset sources. SYSCLK is output for at least 8 cycles before XRSTOUT is driven high.

Category	Pin name	I/O	Number of pins	Also serves as	Description
System bus interface	SA[31:0]	I/O	32	-	System bus address This is the 32-bit address bus.
	SD[31:0]	I/O	32	-	System data THIS IS THE 32-BIT DATA BUS.
	XSAS	I/O	1	-	Address strobe This is a negative logic address strobe. This signal is asserted when the master device initiates a bus access.
	XSCS[7:0]	O	8	-	CHIP SELECT THIS IS A NEGATIVE LOGIC CHIP SELECT SIGNAL.
	SSZ[1:0]	I/O	2	-	Data transfer size These pins indicate the data transfer size. 00: 1 byte 01: 2 bytes 10: 4 bytes 11: 16 bytes
	XSRE	O	1	-	Read enable This is a negative logic signal that is asserted during a read access.
	XSWE[3:0]	O	4	-	Write enable This is a negative logic signal that is asserted during a write access.
	SRXW	I/O	1	-	Read/write status signal This signal goes high for a read access, and low for a write access.
	XSDK	I/O	1	-	Data acknowledge This is a negative logic data acknowledge signal. When this LSI is the master: This is the acknowledge signal that the slave device asserts in handshake mode. (input) When an external master device is the bus master: This is the acknowledge signal that the LSI asserts. (output)

Category	Pin name	I/O	Number of pins	Also serves as	Description
System bus interface	XSBR	I	1	-	Bus request This signal is asserted low when an external master device requests authority to use the system bus.
	XSBG	O	1	-	Bus grant This is a negative logic bus grant signal for bus use authority request (XSBR) from an external master device.
	BOOTBW	I	1	PIO2[0]	Boot bus width selection This signal sets the bit width of the device that fetches an instruction immediately after the XRESET pin is reset. The following devices are set, depending on the status of the BOOTBW pin when the reset condition on the XRESET pin is released: High: 32 bits Low: 16 bits
	BOOTSEL	I	1	PIO2[1]	Boot device selection This signal sets the device that fetches an instruction immediately after the XRESET pin is reset. The following devices are set, depending on the status of the BOOTSEL pin when the reset condition on the XRESET pin is released: High: XSCS0 Low: XSCS1

Category	Pin name	I/O	Number of pins	Also serves as	Description
Memory interface	MA[14:0]	O	15	-	Memory bus address This is a 15-bit SDRAM address bus. The address that is driven on this bus is multiplexed with the RAS/CAS address.
	MD[15:0]	I/O	16	-	Memory bus data This is a 16-bit SDRAM data bus.
	XMCS[1:0]	O	2	-	Chip select These are the chip select signals for SDRAM.
	XMBE[1:0]	O	2	-	Data byte enable These are the byte enable signals for SDRAM.
	XMRAS	O	1	-	SDRAM RAS signal This is the RAS signal for SDRAM.
	XMCAS	O	1	-	SDRAM CAS signal This is the CAS signal for SDRAM.
	XMWE	O	1	-	SDRAM WE signal This is the write enable signal for SDRAM.
	MDK	I	1	-	Data acknowledge This is an acknowledge signal that is used when the memory interface is connected to a device that does not have a constant access cycle. In a normal SDRAM access, this signal should be kept high at all times.
	SDCLK	O	1	-	SDRAM clock output This is the clock output for SDRAM.
	SDCKE	O	1	-	SDRAM clock enable This is the clock enable signal for SDRAM.
	SDCKI	I	1	-	SDRAM data input clock This is used for sampling input data from SDRAM. Connect the input clock for SDRAM to this pin.
Interrupt controller	XIRQ[7:0]	I	8	-	External interrupt signal input This is an external interrupt pin. The trigger mode (edge/level) and the priority can be set through the control register.
	XNMI	I	1	-	External NMI signal input This is a negative logic nonmaskable interrupt pin.
8-bit timer	TM0IO	I/O	1	PIO0[0]	Timer 0 input/output
	TM1IO	I/O	1	PIO0[1]	Timer 1 input/output
	TM2IO	I/O	1	PIO0[2]	Timer 2 input/output
	TM3IO	I/O	1	PIO0[3]	Timer 3 input/output
16-bit timer	TM4IO	I/O	1	PIO0[4]	Timer 4 input/output
	TM5IO	I/O	1	PIO0[5]	Timer 5 input/output
	TM6IOA	I/O	1	PIO0[6]	Timer 6 input/output A
	TM6IOB	I/O	1	PIO0[7]	Timer 6 input/output B
	TM7IO	I/O	1	PIO1[0]	Timer 7 input/output
	TM8IO	I/O	1	PIO1[1]	Timer 8 input/output
	TM9IO	I/O	1	PIO1[2]	Timer 9 input/output

Category	Pin name	I/O	Number of pins	Also serves as	Description
	TM10IO	I/O	1	PIO1[3]	Timer 10 input/output
	TM11IO	I/O	1	PIO1[4]	Timer 11 input/output

Category	Pin name	I/O	Number of pins	Also serves as	Description
Serial interface	SBI[2:0]	I	3	-	Serial interface 0 to 2 data inputs
	SBO[1:0]	I/O	2	PIO4[5:4]	Serial interface 0 to 1 data input/outputs
	SBO2	O	1	-	Serial interface 2 data output
	SBT[1:0]	I/O	2	PIO4[7:6]	Serial interface 0 to 1 transfer clock input/outputs
	SBT2	I	1	-	Serial interface 2 transfer clock input
	XCTS	I	1	PIO0[4]	External interrupt signal input External interrupt pin for interrupting the serial interface 2 transmission
Analog front end interface	AFRXD	I	1	PIO3[0]	AFE data input
	AFTXD	O	1	PIO3[1]	AFE data output
	AFSCLK	I	1	PIO3[2]	AFE data clock
	AFFS	O	1	PIO3[3]	AFE frame sync signal
	AFEHC	O	1	PIO3[4]	AFE control signal
	EYED	O	1	PIO0[1]	EYE data output
	EYECLK	O	1	PIO0[0]	EYE data clock
Real-time clock	RCLKI	I	1	-	RTC oscillation input (32.768KHz)
	RCLKO	O	1	-	RTC oscillation output
	PWROK	I	1	-	Power supply confirmation input When this signal is high, it indicates that the supply of power from the VDD18 power supply has been confirmed. When using the backup battery to drive the real-time clock, it is necessary to negate (drive low) the PWROK input while VDD18 is not being supplied.
A/D converter	AN[7:0]	I	8	-	Analog input
	ADTRG	I	1	PIO1[0]	A/D conversion start trigger input
I2C controller	SCL[0]	I/O	1	PIO4[0]	I2C port 0 clock
	SDA[0]	I/O	1	PIO4[1]	I2C port 0 data
	SCL[1]	I/O	1	PIO4[2]	I2C port 1 clock
	SDA[1]	I/O	1	PIO4[3]	I2C port 1 data
IrDA controller	IRTXD	O	1	PIO5[0]	IrDA output data This is the serial output for SIR/FIR and UART mode.
	IRRXDS	I	1	PIO5[1]	IrDA SIR input data This is the serial input for SIR and UART mode.
	IRRXDF	I	1	PIO5[2]	IrDA FIR input data This is the serial input for FIR mode.
	SOUT	O	1	PIO5[0]	IrDA UART output data This is the serial output in the UART mode.
	SIN	I	1	PIO5[1]	IrDA UART input data This is the serial input in the UART mode.
	CLK48	I	1		IrDA clock input (48MHz) This is the 48MHz clock input for FIR. If FIR is not being used, fix this pin low.
DMA controller	XDMR[1:0]	I	2	PIO1[2:1]	DMA transfer request

Category	Pin name	I/O	Number of pins	Also serves as	Description
JTAG interface	TDI	I	1	-	JTAG data input
	TDO	O	1	-	JTAG data output
	TCK	I	1	-	JTAG clock input
	TMS	I	1	-	JTAG test mode selection
	TRSTMOD	I	1	-	JTAG test mode input
Debugging interface	TRCD[7:0]	O	8	-	Trace data output
	TRCST	O	1	-	Trace status output
	EXTRG	I/O	1	-	Debug trigger output
	TRCCLK	O	1	-	Trace clock output
I/O ports	PIO0[7:0]	I/O	8	-	General-purpose I/O port 0
	PIO1[4:0]	I/O	5	-	General-purpose I/O port 1
	PIO2[4:0]	I/O	5	-	General-purpose I/O port 2
	PIO3[4:0]	I/O	5	-	General-purpose I/O port 3
	PIO4[7:0]	I/O	8	-	General-purpose I/O port 4
	PIO5[2:0]	I/O	3	-	General-purpose I/O port 5

1.5. Register List

Table 4 CPU control registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0000040	CPUM	CPU mode register	16	0x0000	8, 6, 32
0xC0000020	CPUP	CPU pipeline control register	16	0x0000	8, 6, 32
0xC0000050	CPUREV	CPU revision register	32	Note	32

Table 5 Interrupt control registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0000000	IVAR0	Interrupt vector register 0	16	Undefined	16, 32
0xC0000004	IVAR1	Interrupt vector register 1	16	Undefined	16, 32
0xC0000008	IVAR2	Interrupt vector register 2	16	Undefined	16, 32
0xC000000C	IVAR3	Interrupt vector register 3	16	Undefined	16, 32
0xC0000010	IVAR4	Interrupt vector register 4	16	Undefined	16, 32
0xC0000014	IVAR5	Interrupt vector register 5	16	Undefined	16, 32
0xC0000018	IVAR6	Interrupt vector register 6	16	Undefined	16, 32
0xC0000044	SISR	Supervisor interrupt status register	32	0x00000000	32
0xD4000000	NMICR	NMI control register	16	0x0000	16, 32
0xC0000038	DEAR	Data access exception address register	32	Undefined	32
0xC0000024	TBR	Trap base register	32	0x40000000	32
0xC0000028	-	System reserve	-	-	-
0xC0000030	-	System reserve	-	-	-
0xC0000034	-	System reserve	-	-	-
0xC0000060	-	System reserve	-	-	-
0xC0000100	-	System reserve	-	-	-
0xC0000104	-	System reserve	-	-	-
0xC0000108	-	System reserve	-	-	-
0xC0000120	-	System reserve	-	-	-
0xC0000124	-	System reserve	-	-	-
0xC0000128	-	System reserve	-	-	-
0xC000012C	-	System reserve	-	-	-
0xC0000140	-	System reserve	-	-	-
0xC0000144	-	System reserve	-	-	-
0xC0000148	-	System reserve	-	-	-
0xC000014C	-	System reserve	-	-	-
0xC0000150	-	System reserve	-	-	-
0xC0000154	-	System reserve	-	-	-
0xC0000158	-	System reserve	-	-	-
0xC000015C	-	System reserve	-	-	-
0xC0000160	-	System reserve	-	-	-
0xC0000164	-	System reserve	-	-	-
0xC0000168	-	System reserve	-	-	-
0xC000016C	-	System reserve	-	-	-
0xC0000170	-	System reserve	-	-	-

Note: The operations are not guaranteed when writing to the system reserve.

Table 6 MMU control register

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0000090	MMUCTR	MMU control register	32	0x00000000	32
0xC0000094	PIDR	Process identification register	16	Undefined	16
0xC0000098	PTBR	Page table base register	32	Undefined	32
0xC00000A4	IPTEU	Instruction page table entry upper register	32	Undefined	32
0xC00000B4	DPTEU	Data page table entry upper register	32	Undefined	32
0xC00000A0	IPTEL	Instruction page table entry lower register	32	Undefined	32
0xC00000B0	DPTEL	Data page table entry lower register	32	Undefined	32
0xC00000A8	IPTEL2	Instruction page table entry lower register 2	32	Undefined	32
0xC00000B8	DPTEL2	Data page table entry lower register 2	32	Undefined	32
0xC000009C	MMUFCR	MMU fault cause register	32	Undefined	32

Table 7 Cache control registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0000070	CHCTR	Cache control register	16	0x0000	16

Table 8 Bus Controller Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0002000	BCCR	Bus controller control register	32	0x 12040000	8,16,32
0xC0002008	-	System reserve	-	-	-
0xC0002010	BCBERR	Bus error source register	32	0x00000000	8,16,32
0xC0002020	BCBEAR	Bus error address register	32	Undefined	8,16,32
0xC0002030	-	System reserve	-	-	-
0xC0002034	-	System reserve	-	-	-
0xC0002038	-	System reserve	-	-	-
0xC0002040	-	System reserve	-	-	-

Table 9 System Bus Controller Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8C00100	SBBASE0	Base address register 0	32	Note 1	8,16,32
0xD8C00110	SBBASE1	Base address register 1	32	Note 1	8,16,32
0xD8C00120	SBBASE2	Base address register 2	32	0x00000000	8,16,32

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8C00130	SBBASE3	Base address register 3	32	0x00000000	8,16,32
0xD8C00140	SBBASE4	Base address register 4	32	0x00000000	8,16,32
0xD8C00150	SBBASE5	Base address register 5	32	0x00000000	8,16,32
0xD8C00160	SBBASE6	Base address register 6	32	0x00000000	8,16,32
0xD8C00170	SBBASE7	Base address register 7	32	0x00000000	8,16,32
0xD8C00200	SBCTRL00	Bank control register 00	32	0x22100000	8,16,32
0xD8C00204	SBCTRL01	Bank control register 01	32	0x00001100	8,16,32
0xD8C00208	SBCTRL02	Bank control register 02	32	Note 2	8,16,32
0xD8C00210	SBCTRL10	Bank control register 10	32	0x22100000	8,16,32
0xD8C00214	SBCTRL11	Bank control register 11	32	0x00001100	8,16,32
0xD8C00218	SBCTRL12	Bank control register 12	32	Note 2	8,16,32
0xD8C00220	SBCTRL20	Bank control register 20	32	0x22100000	8,16,32
0xD8C00224	SBCTRL21	Bank control register 21	32	0x00001100	8,16,32
0xD8C00228	SBCTRL22	Bank control register 22	32	0x0000000F	8,16,32
0xD8C00230	SBCTRL30	Bank control register 30	32	0x22100000	8,16,32
0xD8C00234	SBCTRL31	Bank control register 31	32	0x00001100	8,16,32
0xD8C00238	SBCTRL32	Bank control register 32	32	0x0000000F	8,16,32
0xD8C00240	SBCTRL40	Bank control register 40	32	0x22100000	8,16,32
0xD8C00244	SBCTRL41	Bank control register 41	32	0x00001100	8,16,32
0xD8C00248	SBCTRL42	Bank control register 42	32	0x0000000F	8,16,32
0xD8C00250	SBCTRL50	Bank control register 50	32	0x22100000	8,16,32
0xD8C00254	SBCTRL51	Bank control register 51	32	0x00001100	8,16,32
0xD8C00258	SBCTRL52	Bank control register 52	32	0x0000000F	8,16,32
0xD8C00260	SBCTRL60	Bank control register 60	32	0x22100000	8,16,32
0xD8C00264	SBCTRL61	Bank control register 61	32	0x00001100	8,16,32
0xD8C00268	SBCTRL62	Bank control register 62	32	0x0000000F	8,16,32
0xD8C00270	SBCTRL70	Bank control register 70	32	0x22100000	8,16,32
0xD8C00274	SBCTRL71	Bank control register 71	32	0x00001100	8,16,32
0xD8C00278	SBCTRL72	Bank control register 72	32	0x0000000F	8,16,32

Table 10 Memory Bus Controller Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xDA000000	SDRAMBUS	Bus mode control register	32	0xAA96061C	8,16,32
0xDA000004	SDREFCNT	Refresh cycle register	32	0x00000C30	8,16,32
0xDA000008	SDBASE0	Base address register 0	32	0x0000F200	8,16,32
0xDA00000C	SDBASE1	Base address register 1	32	0x0000F200	8,16,32
0xDA000010	SDSHDW	SD shadow register	32	0x00000006	8, 16, 32

Table 11 DMA Controller Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD2000000	DM0CTR	DMA control register	32	0x80000000	8,16,32
0xD2000004	DM0SRC	DMA source address register	32	0x00000000	8,16,32
0xD2000008	DM0DST	DMA destination address register	32	0x00000000	8,16,32
0xD200000C	DM0SIZ	DMA transfer word count register	32	0x00000000	8,16,32
0xD2000010	DM0CYC	DMA intermittent transfer size register	32	0x00000000	8,16,32
0xD2000100	DM1CTR	DMA control register	32	0x80000000	8,16,32
0xD2000104	DM1SRC	DMA source address register	32	0x00000000	8,16,32
0xD2000108	DM1DST	DMA destination address register	32	0x00000000	8,16,32
0xD200010C	DM1SIZ	DMA transfer word count register	32	0x00000000	8,16,32
0xD2000110	DM1CYC	DMA intermittent transfer size register	32	0x00000000	8,16,32
0xD2000200	DM2CTR	DMA control register	32	0x80000000	8,16,32
0xD2000204	DM2SRC	DMA source address register	32	0x00000000	8,16,32
0xD2000208	DM2DST	DMA destination address register	32	0x00000000	8,16,32
0xD200020C	DM2SIZ	DMA transfer word count register	32	0x00000000	8,16,32
0xD2000210	DM2CYC	DMA intermittent transfer size register	32	0x00000000	8,16,32
0xD2000300	DM3CTR	DMA control register	32	0x80000000	8,16,32
0xD2000304	DM3SRC	DMA source address register	32	0x00000000	8,16,32
0xD2000308	DM3DST	DMA destination address register	32	0x00000000	8,16,32
0xD200030C	DM3SIZ	DMA transfer word count register	32	0x00000000	8,16,32
0xD2000310	DM3CYC	DMA intermittent transfer size register	32	0x00000000	8,16,32

Table 12 8-bit Timer Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
0xD4003000	TM0MD	Timer 0 mode register	8	0x00	8,16,32
0xD4003001	TM1MD	Timer 1 mode register	8	0x00	8
0xD4003002	TM2MD	Timer 2 mode register	8	0x00	8,16
0xD4003003	TM3MD	Timer 3 mode register	8	0x00	8
0xD4003010	TM0BR	Timer 0 base register	8	0x00	8,16,32
0xD4003011	TM1BR	Timer 1 base register	8	0x00	8
0xD4003012	TM2BR	Timer 2 base register	8	0x00	8,16
0xD4003013	TM3BR	Timer 3 base register	8	0x00	8
0xD4003020	TM0BC	Timer 0 binary counter	8	0x00	8,16,32
0xD4003021	TM1BC	Timer 1 binary counter	8	0x00	8
0xD4003022	TM2BC	Timer 2 binary counter	8	0x00	8,16
0xD4003023	TM3BC	Timer 3 binary counter	8	0x00	8
0xD4003071	TMPSCNT	Timer prescaler control register	8	0x00	8

Table 13 16-bit Timer Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
0xD4003080	TM4MD	Timer 4 mode register	8	0x00	8
0xD4003082	TM5MD	Timer 5 mode register	8	0x00	8
0xD4003084	TM6MD	Timer 6 mode register	16	0x0000	8,16
0xD4003086	TM7MD	Timer 7 mode register	8	0x00	8
0xD4003088	TM8MD	Timer 8 mode register	8	0x00	8
0xD400308A	TM9MD	Timer 9 mode register	8	0x00	8
0xD400308C	TM10MD	Timer 10 mode register	8	0x00	8
0xD400308E	TM11MD	Timer 11 mode register	8	0x00	8
0xD4003090	TM4BR	Timer 4 base register	16	0x0000	8,16,32
0xD4003092	TM5BR	Timer 5 base register	16	0x0000	8,16
0xD4003096	TM7BR	Timer 7 base register	16	0x0000	8,16
0xD4003098	TM8BR	Timer 8 base register	16	0x0000	8,16,32
0xD400309A	TM9BR	Timer 9 base register	16	0x0000	8,16
0xD400309C	TM10BR	Timer 10 base register	16	0x0000	8,16,32
0xD40030AE	TM11BR	Timer 11 base register	16	0x0000	8,16
0xD40030A0	TM4BC	Timer 4 binary counter	16	0x0000	8,16,32
0xD40030A2	TM5BC	Timer 5 binary counter	16	0x0000	8,16
0xD40030A4	TM6BC	Timer 6 binary counter	16	0x0000	8,16,32
0xD40030A6	TM7BC	Timer 7 binary counter	16	0x0000	8,16
0xD40030A8	TM8BC	Timer 8 binary counter	16	0x0000	8,16,32
0xD40030AA	TM9BC	Timer 9 binary counter	16	0x0000	8,16
0xD40030AC	TM10BC	Timer 10 binary counter	16	0x0000	8,16,32
0xD40030AE	TM11BC	Timer 11 binary counter	16	0x0000	8,16
0xD40030B4	TM6MDA	Timer 6 compare/capture A mode register	8	0x00	8,16
0xD40030B5	TM6MDB	Timer 6 compare/capture B mode register	8	0x00	8
0xD40030C4	TM6CA	Timer 6 compare/capture register A	16	0x0000	8,16

0xD40030D4	TM6CB	Timer 6 compare/capture register B	16	0x0000	8,16
0xD4003071	TMPSCNT	Timer prescaler control register	8	0x00	8

Table 14 Serial Controller Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
0xD4002000	SC0CTR	Serial 0 control register	16	0x0000	8, 16
0xD4002004	SC0ICR	Serial 0 interrupt mode register	8	0x00	8
0xD4002008	SC0TXB	Serial 0 transmit buffer	8	0x00	8
0xD4002009	SC0RXB	Serial 0 receive buffer	8	0x00	8
0xD400200C	SC0STR	Serial 0 status register	16	0x0000	8,16
0xD4002010	SC1CTR	Serial 1 control register	16	0x0000	8, 16
0xD4002014	SC1ICR	Serial 1 interrupt mode register	8	0x00	8
0xD4002018	SC1TXB	Serial 1 transmit buffer	8	0x00	8
0xD4002019	SC1RXB	Serial 1 receive buffer	8	0x00	8
0xD400201C	SC1STR	Serial 1 status register	16	0x0000	8,16
0xD4002020	SC2CTR	Serial 2 control register	16	0x0000	8, 16
0xD4002024	SC2ICR	Serial 2 interrupt mode register	8	0x00	8
0xD4002028	SC2TXB	Serial 2 transmit buffer	8	0x00	8
0xD4002029	SC2RXB	Serial 2 receive buffer	8	0x00	8
0xD400202C	SC2STR	Serial 2 status register	8	0x00	8
0xD400202D	SC2TIM	Serial 2 timer register	8	0x00	8

Table 15 Interrupt Controller Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD4000000	G0ICR	Nonmaskable interrupt control register	16	0x0000	8, 16
0xD4000004	G1ICR	Group 1 interrupt control register	16	0x0000	8, 16
0xD4000008	G2ICR	Group 2 interrupt control register	16	0x0000	8, 16
0xD400000C	G3ICR	Group 3 interrupt control register	16	0x0000	8, 16
0xD4000010	G4ICR	Group 4 interrupt control register	16	0x0000	8, 16
0xD4000014	G5ICR	Group 5 interrupt control register	16	0x0000	8, 16
0xD4000018	G6ICR	Group 6 interrupt control register	16	0x0000	8, 16
0xD400001C	G7ICR	Group 7 interrupt control register	16	0x0000	8, 16
0xD4000020	G8ICR	Group 8 interrupt control register	16	0x0000	8, 16
0xD4000024	G9ICR	Group 9 interrupt control register	16	0x0000	8, 16
0xD4000028	G10ICR	Group 10 interrupt control register	16	0x0000	8, 16
0xD400002C	G11ICR	Group 11 interrupt control register	16	0x0000	8, 16
0xD4000030	G12ICR	Group 12 interrupt control register	16	0x0000	8, 16
0xD4000034	G13ICR	Group 13 interrupt control register	16	0x0000	8, 16
0xD4000038	G14ICR	Group 14 interrupt control register	16	0x0000	8, 16
0xD400003C	G15ICR	Group 15 interrupt control register	16	0x0000	8, 16
0xD4000040	G16ICR	Group 16 interrupt control register	16	0x0000	8, 16
0xD4000044	G17ICR	Group 17 interrupt control register	16	0x0000	8, 16
0xD4000048	G18ICR	Group 18 interrupt control register	16	0x0000	8, 16
0xD400004C	G19ICR	Group 19 interrupt control register	16	0x0000	8, 16

0xD4000050	G20ICR	Group 20 interrupt control register	16	0x0000	8, 16
0xD4000054	G21ICR	Group 21 interrupt control register	16	0x0000	8, 16
0xD4000058	G22ICR	Group 22 interrupt control register	16	0x0000	8, 16
0xD400005C	G23ICR	Group 23 interrupt control register	16	0x0000	8, 16
0xD4000060	G24ICR	Group 24 interrupt control register	16	0x0000	8, 16
0xD4000064	G25ICR	Group 25 interrupt control register	16	0x0000	8, 16
0xD4000068	G26ICR	Group 26 interrupt control register	16	0x0000	8, 16
0xD400006C	G27ICR	Group 27 interrupt control register	16	0x0000	8, 16
0xD4000070	G28ICR	Group 28 interrupt control register	16	0x0000	8, 16
0xD4000074	G29ICR	Group 29 interrupt control register	16	0x0000	8, 16
0xD4000078	G30ICR	Group 30 interrupt control register	16	0x0000	8, 16
0xD400007C	G31ICR	Group 31 interrupt control register	16	0x0000	8, 16
0xD4000080	G32ICR	Group 32 interrupt control register	16	0x0000	8, 16
0xD4000084	G33ICR	Group 33 interrupt control register	16	0x0000	8, 16
0xD4000088	G34ICR	Group 34 interrupt control register	16	0x0000	8, 16
0xD400008C	G35ICR	Group 35 interrupt control register	16	0x0000	8, 16
0xD4000090	G36ICR	Group 36 interrupt control register	16	0x0000	8, 16
0xD4000094	G37ICR	Group 37 interrupt control register	16	0x0000	8, 16
0xD4000098	G38ICR	Group 38 interrupt control register	16	0x0000	8, 16
0xD400009C	G39ICR	Group 39 interrupt control register	16	0x0000	8, 16
0xD40000A0	G40ICR	Group 40 interrupt control register	16	0x0000	8, 16
0xD40000A4	G41ICR	Group 41 interrupt control register	16	0x0000	8, 16
0xD4000100	IAGR	Interrupt acceptance group register	16	0x0000	8, 16
0xD4000200	EXTDM	External pin interrupt conditional specification register	16	0x0000	8, 16

Table 16 Watchdog Timer Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0001000	WDBC	Watchdog binary counter	8	0x00	8, 16
0xC0001002	WDCTR	Watchdog timer control register	8	Note	8, 16
0xC0001004	RSTCTR	Reset control register	8	0x00	8, 16

Note: For the initial values, refer to chapter 14, watchdog timer.

Table 17 Analog Front-end Interface Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8300000	AFESYS	AFE system control register	16	0x0003	8, 16
0xD8300004	AFEINTM	AFE interrupt mask register	16	0x00FF	8, 16
0xD8300008	AFESTAT	AFE status register	16	0x0048	8, 16
0xD830000C	AFECTR	AFE control register	16	0x0300	8, 16
0xD8300010	AFETBUF	AFE transmit buffer register	16	Undefined	16
0xD8300014	AFERBUF	AFE receive buffer register	16	Undefined	16
0xD8300018	AFEFIFO	AFE FIFO size register	16	0x1100	8, 16
0xD830001C	AFEEYE	AFE eye pattern register	16	0x0000	8, 16
0xD8300020	AFESEC	AFE second source register	16	0x000C	8, 16

Table 18 A/D Converter Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8500000	ADCTR	A/D conversion control register	16	0x0000	8, 16
0xD8500010	AD0BUF	A/D 0 conversion data buffer	16	0x0000	8, 16
0xD8500012	AD1BUF	A/D 1 conversion data buffer	16	0x0000	8, 16
0xD8500014	AD2BUF	A/D 2 conversion data buffer	16	0x0000	8, 16
0xD8500016	AD3BUF	A/D 3 conversion data buffer	16	0x0000	8, 16
0xD8500018	AD4BUF	A/D 4 conversion data buffer	16	0x0000	8, 16
0xD850001A	AD5BUF	A/D 5 conversion data buffer	16	0x0000	8, 16
0xD850001C	AD6BUF	A/D 6 conversion data buffer	16	0x0000	8, 16
0xD850001E	AD7BUF	A/D 7 conversion data buffer	16	0x0000	8, 16

Table 19 Real-time Clock Registers

Address	Symbol	Name	Number of bits	Initial value as binary data x: undefined	Access size
0xD8600000	RTSCR	Seconds count register	8	Undefined	8
0xD8600001	RTSAR	Seconds alarm register	8	Undefined	8
0xD8600002	RTMCR	Minutes count register	8	Undefined	8
0xD8600003	RTMAR	Minutes alarm register	8	Undefined	8
0xD8600004	RTHCR	Hours count register	8	Undefined	8
0xD8600005	RTHAR	Hours alarm register	8	Undefined	8
0xD8600006	RTDWCR	Day of the week count register	8	Undefined	8
0xD8600007	RTDMCR	Days count register	8	Undefined	8
0xD8600008	RTMTCR	Months count register	8	Undefined	8

0xD8600009	RTYCR	Years count register	8	Undefined	8
0xD860000A	RTCRA	Control register A	8	xx10xxxx	8
0xD860000B	RTCRB	Control register B	8	Undefined	8
0xD860000C	RTSRC	Status register C	8	xxxx0000	8

Table 20 IrDA Controller Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080	-	Offset 0	8	Note	8
0xD8700081	-	Offset 1	8	Note	8
0xD8700082	-	Offset 2	8	Note	8
0xD8700083	-	Offset 3	8	Note	8
0xD8700084	-	Offset 4	8	Note	8
0xD8700085	-	Offset 5	8	Note	8
0xD8700086	-	Offset 6	8	Note	8
0xD8700087	-	Offset 7	8	Note	8

Note: For the initial values, refer to chapter 18, IrDA controller (IRC).

Table 21 I2C Controller Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8400000	IIC0DTRM	I2C TRANSMIT DATA REGISTER 0	32	0x00000000	32
0xD8400004	IIC0DREC	I2C receive data register 0	32	0x000009FF	32
0xD8400008	IIC0MYAD	I2C slave address register 0	32	0x00000000	32
0xD840000C	IIC0CLK	I2C clock register 0	32	0x00000000	32
0xD8400010	IIC0BRST	I2C bus reset register 0	32	0x00000001	32
0xD8400014	IIC0BSTS	I2C bus status register 0	32	Undefined	32
0xD8401000	IIC1DTRM	I2C transmit data register 1	32	0x00000000	32
0xD8401004	IIC1DREC	I2C receive data register 1	32	0x000009FF	32
0xD8401008	IIC1MYAD	I2C slave address register 1	32	0x00000000	32
0xD840100C	IIC1CLK	I2C clock register 1	32	0x00000000	32
0xD8401010	IIC1BRST	I2C bus reset register 1	32	0x00000001	32
0xD8401014	IIC1BSTS	I2C bus status register 1	32	Undefined	32

Table 22 I/O Port Registers

Address	Symbol	Name	Number of bits	Initial value	ACCESS SIZE
0xDB000000	P0MD	Port 0 mode register	16	0x0000	8, 16
0xDB000004	P0IN	Port 0 pin register	8	Undefined	8
0xDB000008	P0OUT	Port 0 output register	8	0x00	8
0xDB00000C	P0TMIO	Port 0TM pin input/output control register	8	0x00	8
0xDB000100	P1MD	Port 1 mode register	16	0x03C0	8, 16
0xDB000104	P1IN	Port 1 pin register	8	Undefined	8
0xDB000108	P1OUT	Port 1 output register	8	0x00	8
0xDB00010C	P1TMIO	Port 1TM pin input/output control register	8	0x00	8
0xDB000200	P2MD	Port 2 mode register	16	0x00FF	8, 16
0xDB000204	P2IN	Port 2 pin register	8	Undefined	8
0xDB000208	P2OUT	Port 2 output register	8	0x00	8
0xDB000300	P3MD	Port 3 mode register	16	0x0000	8, 16

0xDB000304	P3IN	Port 3 pin register	8	Undefined	8
0xDB000308	P3OUT	Port 3 output register	8	0x00	8
0xDB000400	P4MD	Port 4 mode register	16	0x0000	8, 16
0xDB000404	P4IN	Port 4 pin register	8	Undefined	8
0xDB000408	P4OUT	Port 4 output register	8	0x00	8
0xDB000500	P5MD	Port 5 mode register	16	0x0000	8, 16
0xDB000504	P5IN	Port 5 pin register	8	Undefined	8
0xDB000508	P5OUT	Port 5 output register	8	0x00	8

Chapter 2 CPU

2

2.1. Introduction

2.1.1. Overview

The AM33 is the top-of-the-line microcontroller core in Panasonic's AM30 Series of 32-bit microcontrollers. In addition to the original instruction set of the AM30 Series, the AM33 also has an extended instruction set that allows it to handle a wide range of information processing and signal processing applications. The heart of the AM33 is a compact 32-bit CPU core that has an instruction set with a basic instruction word length of one byte; in addition to this core, the AM33 also has the AM33 architecture extension, an MMU, cache memory, an FPU, and a debug unit.

2.1.2. Features

Compact, high-performance CPU core

- Full upward compatibility with the AM30/AM31/AM32 core
  AM30 series original instruction set + Am33 extended instruction set
- Instruction set with a basic instruction word length of one byte provides high coding efficiency
- Extended register model
  Eight extended general-purpose registers, four address registers, and four data registers
- Three privileged levels (user/supervisor/monitor)
- Sophisticated high-performance extended arithmetic operation unit on chip
  Implements extended instructions (including SIMD type), such as fast multiply instructions and multiply-and-accumulate instructions, that enhance signal processing capabilities.
- Adopts proprietary fast branch processing techniques
- Supports a linear address space of up to 4 gigabytes

MMU (Memory Management Unit)

- Memory protection function
  Access permission for logical address spaces can be set separately for the supervisor level and the user level.
- Address translation function.
  Address translation is implemented through the paging method (variable page sizes: 1KB/4KB/128KB/4MB)
- TLB (instruction/data separate type)
  32 entries each for instructions and data; association method: full associative

Cache memory

- Instruction cache
  16KB: 4KB x 4, 256 entries
  Line size: 16 bytes, 4-way set associative
  Pseudo LRU replacement algorithm for each way
  Entry lock function for each way (individual ways can be implemented through RAM)
- Data cache
  Maximum size: 16KB: 4KB x 4, 256 entries
  Line size: 16 bytes, 4-way set associative
  Pseudo LRU replacement algorithm for each way
  Write policy can be switched between write-through and write-back (write allocate/write nonallocate)
  Entry lock function for each way (individual ways can be implemented through RAM)

FPU (Floating-Point Operation Unit)

- Supports data types in compliance with the IEEE754 standard
- Supports rounding to the nearest value in compliance with the IEEE754 standard.
- 32 single-precision floating-point operation registers
  (These registers can also be loaded and stored as 16 double-precision floating-point operation registers.)
- Supports five floating-point operation exceptions in compliance with the IEEE754 standard and an unimplemented floating-point instruction exception.

On-chip debug unit

- Implements trace functions in conjunction with an external trace unit.

Low power consumption modes

- Three modes: HALT, SLEEP, and STOP

2.2. General Block Diagram

A general block diagram for the AM33 microcontroller core is shown below.

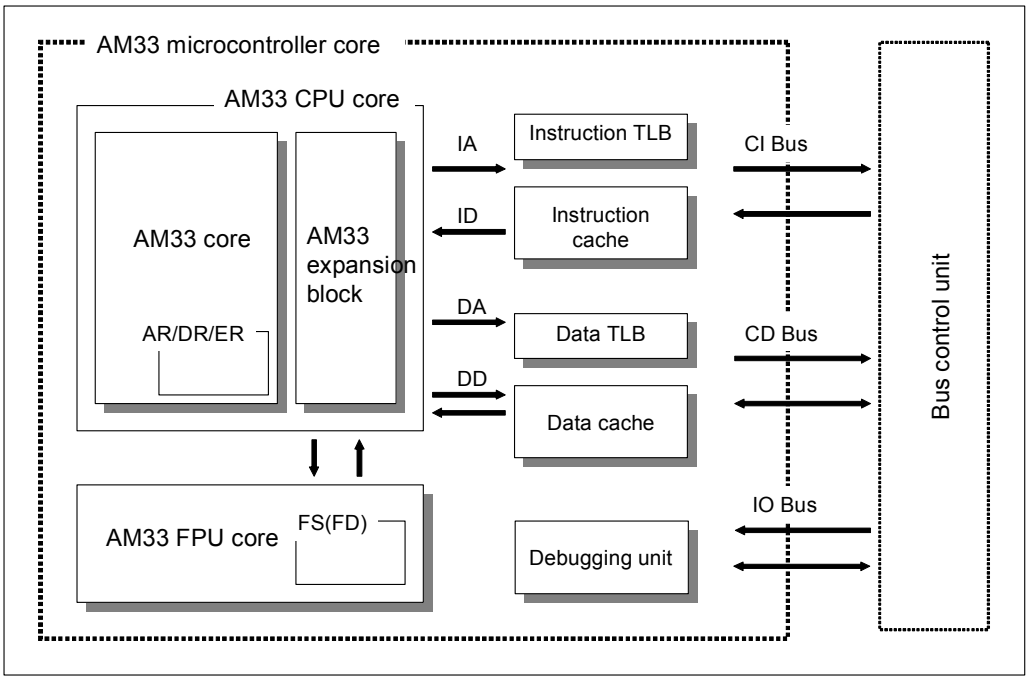


Figure 4 Block diagram of AM33 microcontroller core

The AM33 microcontroller core consists of five modules: an AM33 CPU core, an MMU, cache memory, an FPU, and a debug unit.

AM33 CPU core

This module functions as a central processing unit. This module consists of a data path unit comprised of an arithmetic operation unit and various registers, and a control unit comprised of an instruction decoder and a sequencer. The CPU core module is divided into the AM33 core block and the AM33 extended block. The former is the basic block from the AM30 architecture, while the latter is an add-on extension block for the architecture. The

AM31-compatible basic instructions and the extended basic instructions are implemented in the AM33 core block, while the AM31-compatible extended operation instructions and the LIW extended operation instructions are implemented in the AM33 extension block.

MMU (Memory management unit)

This module implements the memory protection functions and the address translation functions. Two levels of memory protection are supported. The paging method is used for address translation; four page sizes are supported (1KB, 4KB, 128KB, and 4MB). Independent instruction and data MMUs are provided.

Cache

This is a memory module that increases the average memory access speed by caching instructions and data. Independent instruction and data caches are provided. Each is a physical cache that is accessed through physical addresses.

FPU (Floating-point operation unit)

This module executes floating-point operation instructions. The module contains thirty-two 32-bit (single-precision) floating-point registers. These registers can handle floating-point data that complies with the IEEE754 standards.

Debug Unit

This unit implements the on-chip debug functions that are built into the AM33 microcontroller core. This unit can also be used to implement trace functions when used in conjunction with an external trace unit.

2.3. Programming Model

The AM33 microcontroller core programming models is upwardly compatible with the AM30/31/32 microcontroller core programming model. In particular, the AM33 offers two operating levels (user level and supervisor level), a built-in MMU, extended operation instructions with an enhanced multiply-and-accumulate operation, floating-point operation instructions, and a debug function.

2.3.1. Basic Register Set

The basic register set consists of data registers for arithmetic operations such as addition and subtraction, address registers for pointers, general-purpose registers that can be used for general purposes, stack pointers, a program counter, a processor status word, a multiply/divide register, a loop instruction register, and a loop address register. In addition to reducing the instruction code size and contributing greatly to improved performance, these registers make programming in high-level languages such as C language possible. In particular, the data registers and address registers have a bank configuration, and different registers can be designated as user-level or supervisor-level registers. In addition, the sixteen data, address, and general-purpose registers can be used as flat general-purpose registers (noted as "Rn" for the sake of convenience).
The basic register set is shown below.

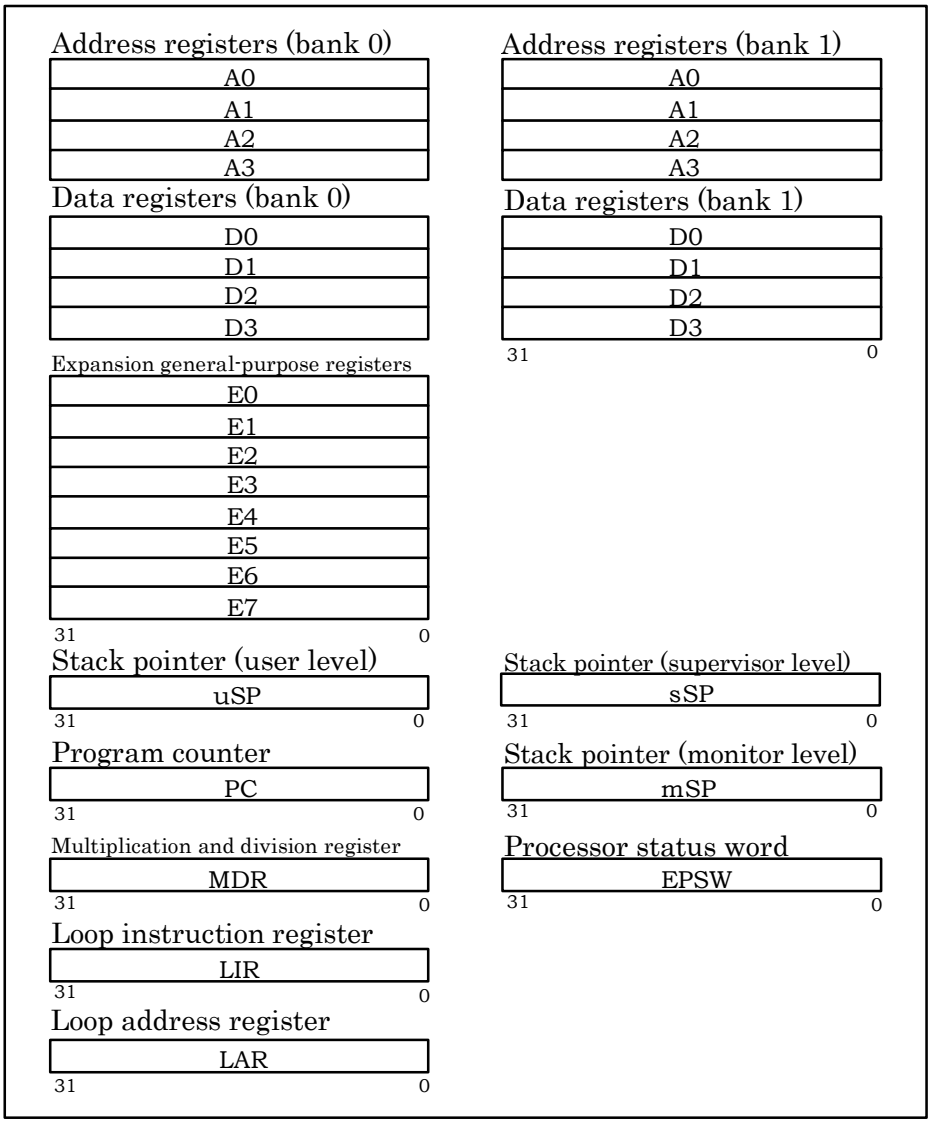
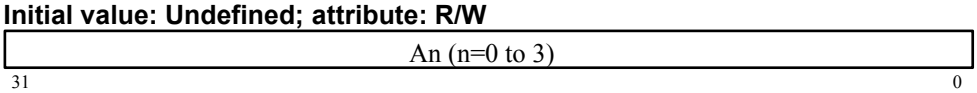


Figure 5 Basic register set list

Only one of the address/data register banks can be accessed at one time, depending on the state of the EPSW.nAR bit. If the EPSW.nAR bit is set to "0," the registers in bank 0 are accessed; if it is set to "1," the registers in bank 1 are accessed. A bank 0 register and a bank 1 register cannot be accessed simultaneously.
There are separate stack pointers for each privileged level. When in user level, the contents of uSP are used as the stack pointer. When in supervisor level, the contents of sSP are used as the stack pointer.

2.3.1.1. Address Registers

Register symbol: A0-A3
Initial value: Undefined; attribute: R/W

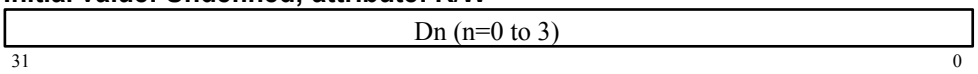


These registers are primarily used as address pointers. Among AM31-compatible operation instructions, these registers can only be used by instructions for address calculation

(addition/subtraction and compare). AM31-compatible transfer instructions use these registers as address pointers for data; transfers to memory are always performed in 32-bit lengths. Among extended basic instructions, these registers can be used as general-purpose registers.
The registers themselves are organized into two banks, which can be selected through the EPSW.nAR bit. For the sake of convenience, the register bank that is selected when the EPSW.nAR bit is set to "0" is designated as "bank 0," and the register bank that is selected when the EPSW.nAR bit is set to "1" is designated as "bank 1." In addition, bank 0 is also referred to as the "normal bank." The registers in bank 0 and bank 1 cannot be accessed simultaneously.

2.3.1.2. Data Registers

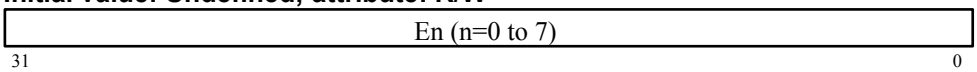
Register symbol: D0-D3
Initial value: Undefined; attribute: R/W



These are general-purpose operation registers that can be used for all operations. Operations are performed in 32-bit lengths. When handling 8-bit data and 16-bit data, the data size is converted by the transfer of data between the microcontroller and memory or by executing an EXTB/EXTH instruction. Among extended basic instructions, these registers can be used as general-purpose registers.
The registers themselves are organized into two banks, which can be selected through the EPSW.nAR bit. For the sake of convenience, the register bank that is selected when the EPSW.nAR bit is set to "0" is designated as "bank 0," and the register bank that is selected when the EPSW.nAR bit is set to "1" is designated as "bank 1." The registers in bank 0 and bank 1 cannot be accessed simultaneously.

2.3.1.3. Extended General-purpose Registers

Register symbol: E0-E7
Initial value: Undefined; attribute: R/W



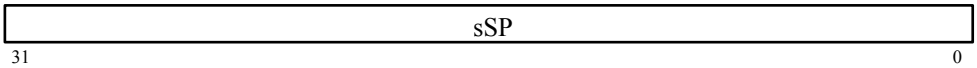
These are general-purpose registers that are used to store operation parameters and intermediate operation results. These registers are also positioned as extended registers for the basic register set, which consists of the data registers (D0 to D4) and the address registers (A0 to A3). These extended registers can be used by the extended basic instructions, the extended operation instructions, and the LIW extended operation instructions.

2.3.1.4. Stack Pointers

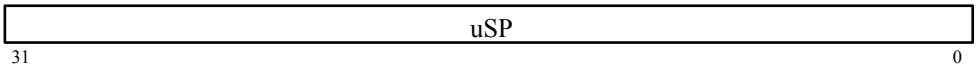
Register symbol: sSP, uSP
Initial value: Undefined; attribute: R/W

The register referred to as SP changes between sSP and uSP depending on privileged levels. The registers are the followings: sSP, uSP

sSP: Supervisor-level stack pointer (Initial value: undefined, Attribute: R/W)
When on supervisor level, this is referred to as SP. Executing an instruction that directly refers to and changes sSP on user level, a privileged instruction execution exception is generated. The instruction that directly refers to and changes uSP is allowed only on the supervisor level.



uSP: User-level stack pointer (Initial value: undefined, Attribute: R/W)
When on user level, this is referred to as SP. Executing an instruction that directly refers to and changes uSP on user level, a privileged instruction execution exception is generated. The instruction that directly refers to and changes uSP is allowed only on the supervisor level.

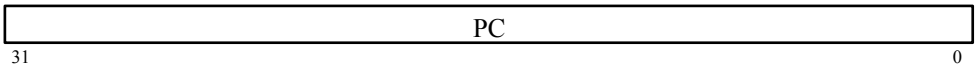


<Programming Note>
While the accesses to sSP and uSP on monitor level and uSP on supervisor level are possible, they cannot be accessed as SP. On user level, uSP can be accessed only as SP.

<Programming Note>
Saving of PC and EPSW through PC with change to sSP is carried out by the occurrence of the asynchronous and synchronous interrupts. Accordingly, sSP has to always be aligned at the boundary of 4 byte. Otherwise, double fault is caused as the result of the occurrence of misalignment by data access in the interrupt hardware sequence.

2.3.1.5. Program Counter

Register symbol: PC
Initial value: 0x40000000; attribute: R



This register stores the address of the instruction that is currently being executed by the CPU.

2.3.1.6. EPSW/PSW: Processor Status Word

Symbol: EPSW/PSW
Address: EPSW/PSW can be accessed by a MOVE instruction.
Purpose: The EPSW (Extended Processor Status Word) is a 32-bit register that was created by extending the 16-bit PSW (Processor Status Word) of AM30 Series to 32 bits. The EPSW includes bits concerning asynchronous interrupt/synchronous interrupt processing, bits that control the internal status of the CPU, and bits that indicate the results of integer operations.

In addition to the instructions that are used in the AM30 Series microcontroller cores, the AM33 microcontroller core has additional instructions that access the EPSW as a 32-bit register. These additional instructions are used to access the value in the upper 16 bits. When the EPSW is accessed as a 16-bit register (PSW), the upper 16 bits are read as "0," and any values that are written to the upper 16 bits are ignored. If these bits are read or written in user level, a privileged instruction execution exception is generated.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R/W							
Bit	23	22	21	20	19	18	17	16
Bit name	reserved			FE	ML	nAR	NMID	nSL
Initial value	0			0	0	0	0	0
R/W	R/W			R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Bit name	T	reserved	S		IE	IM		
Initial value	0	0	0		0	0		
R/W	R	R/W	R/W		R/W	R/W		
Bit	7	6	5	4	3	2	1	0
Bit name	reserved				V	C	N	Z
Initial value	0				0	0	0	0
R/W	R/W				R/W	R/W	R/W	R/W

Bit	Bit name	Description
31-21	reserved	These fields are reserved for future functional extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
20	FE	FPU enable This flags enables use of the FPU (floating-point unit) and enables the execution of floating-point operation instructions. When FE = 1, floating-point operation instructions can be executed. When FE = 0 and an attempt is made to execute a floating-point operation instruction, an FPU disable exception is generated. For details on the floating-point unit, refer to the chapter on the floating-point unit.

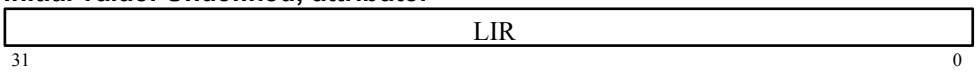
Bit	Bit name	Description
		<Programming Note> *If no floating-point unit has been implemented as hardware, this bit is fixed to "0." This bit can be overwritten by software only if the floating-point unit hardware has been implemented.* *<Programming Note>* Because the AM33 executes instructions through an instruction pipeline, delays will arise if software enables the FPU by setting the FE bit. An FPU instructions must be executed after executing at least three other instructions.
19	ML	Monitor level This bit indicates the current execution level. ML=1 is the monitor level and ML=0 is the non-monitor level. ML has priority over EPSW.nSL. This bit may be set to “1” in the case of the debugger hardware. EPSW.ML is fixed in a normal mode. Accordingly, it cannot become to the monitor level.
18	nAR	Register bank control This bit indicates that an alternate bank is to be used for the bank portion of the register file (A0 to A3, D0 to D3) when nAR is set to “1”. If an asynchronous interrupt/synchronous interrupt is accepted, nAR is set to "0" after the EPSW is saved. *<Programming Note>* *Because the AM33 executes instructions through an instruction pipeline, delays will arise if software sets the nAR bit. The bank portion of the register file must be accessed after executing at least three other instructions.* *<Programming Note>* *By having the kernel use registers in the normal bank, not an alternate bank, and having each task use alternate banks, the normal bank registers can be used right away without needing to save the bank registers, since the system switches to the normal bank registers immediately after an interrupt.* *<Programming Note>* *When switching tasks, it is necessary to switch the register bank and then save/restore the registers.* *<Programming Note>* *When using the bank registers to pass parameters for a system call, it is necessary to switch the register bank to access the parameters.*
17	NMID	Nonmaskable interrupt disable This flag disables the acceptance of nonmaskable interrupts. When NMID = 0, nonmaskable interrupts are enabled. When starting up after a reset, NMID = 0, that is, nonmaskable interrupts are accepted. If a nonmaskable interrupt is accepted, NMID is set to "1" (disabling interrupts). <Programming Note> If multiple nonmaskable interrupts are received within a nonmaskable interrupt processing program, NMID shall be set to "0." Note that if NMID is set to "1," it will not be possible to accept nonmaskable interrupts.
16	nSL	Supervisor level This bit indicates the current execution level. When nSL = 0, supervisor level is in effect; when nSL = 1, user level is in effect. When starting up after a reset in normal mode, nSL = 0 (supervisor level). If an asynchronous interrupt or a synchronous interrupt is generated, nSL is

Bit	Bit name	Description
		set to "0" after the EPSW is saved. After returning from the asynchronous interrupt/synchronous interrupt and restoring the saved contents to the EPSW, the execution level that was in effect when the asynchronous interrupt/synchronous interrupt was generated is restored. <*Programming Note*> *Because instructions are prefetched, the instruction fetch may be delayed if this bit is overwritten by an instruction other than an RTI instruction. Because the RTI instruction includes branching, any instruction that was prefetched is discarded. Therefore, instruction fetches reflect the setting of this bit.* <*Programming Note*> *Because supervisor level is set after a reset, it is possible to run programs that were created for the AM30/31/32 microcontroller core by leaving this bit unchanged so that the system remains in supervisor level at all times.*
15	T	Trace enable This bit is used by a debugger to realize single step function. This bit cannot normally be updated.
14	reserved	These fields are reserved for future functional extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
13-12	S	Software auxiliary bits These auxiliary bits can be used in any desired fashion by system software, etc. The application of these bits depends on the content of the software.
11	IE	Interrupt enable This flag is used to enable the acceptance of asynchronous interrupts, except for nonmaskable interrupts and reset interrupts. When IE = 1, maskable interrupts are accepted. If an asynchronous interrupt is accepted, IE is set to "0" (disabling maskable interrupts). <*Programming Note*> *When multiple interrupts are accepted by an interrupt processing program, IE is set to "1" after the resources have been saved. Because IE only enables the acceptance of asynchronous interrupts, any synchronous interrupt that is generated is accepted even if IE = 0 (interrupts disabled).*
10-8	IM	Interrupt mask These bits store the current asynchronous interrupt mask level. When IE = 1, maskable interrupts with a level higher than that indicated by the IM bits are accepted. If a maskable interrupt is accepted, the PSW is saved and then these bits are changed to the level of the maskable interrupt that was accepted.
7-4	reserved	These fields are reserved for future functional extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
3	V	Overflow flag This bit indicates whether the execution of an operation caused an overflow or not. For details, refer to the descriptions of the individual instructions.
2	C	Carry flag This bit indicates whether the execution of an operation caused a carry out of the MSB/borrow to the MSB or not. For details, refer to the descriptions of the individual instructions.

Bit	Bit name	Description
1	N	Negative flag This bit indicates whether the result of an operation was "negative" or not. For details, refer to the descriptions of the individual instructions.
0	Z	Zero flag This bit indicates whether the result of an operation was “zero” or not. For details, refer to the descriptions of the individual instructions.

2.3.1.7. Loop Instruction Register

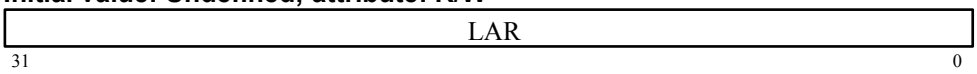
Register symbol: LIR
Initial value: Undefined; attribute: -



This register stores the first four bytes of the branch destination instruction stream when executing a LOOP instruction. This register is used to speed up the execution of LOOP instructions. This register is set by the SETLB instruction. This register can also be saved to and restored from the stack area by using the MOVM instruction.

2.3.1.8. Loop Address Register

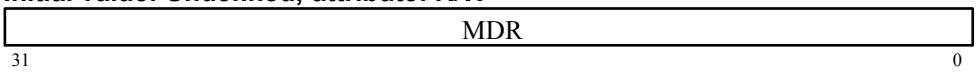
Register symbol: LAR
Initial value: Undefined; attribute: R/W



This register stores the start address of the instruction stream that follows the instruction stream that is set in the LIR; in other words, the address that is four greater than the destination address of the loop instruction. This register is used to speed up the execution of LOOP instructions. This register is set by the SETLB instruction. This register can also be saved to and restored from the stack area by using the MOVM instruction.

2.3.1.9. Multiply/divide Register

Register symbol: MDR
Initial value: Undefined; attribute: R/W



This register is provided for multiply/divide instructions. The register stores the upper 32 bits of a 64-bit multiplication result. For a division operation, the register stores the upper 32 bits of the divided and the remainder (32 bits). For details, refer to the description of operation of individual instructions.

2.3.2. Extended Operation Register Set

The extended operation register set consists of registers that support the extended operation instructions that are provided in the AM33.

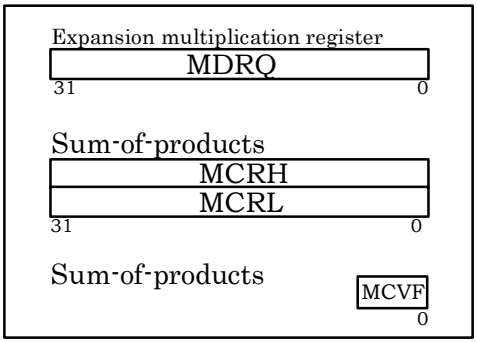
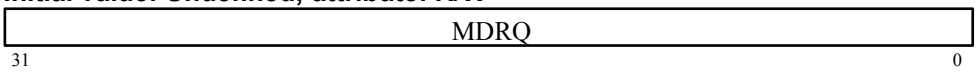


Figure 6 Extended Operation Register Set

2.3.2.1. Multiply/Divide Register

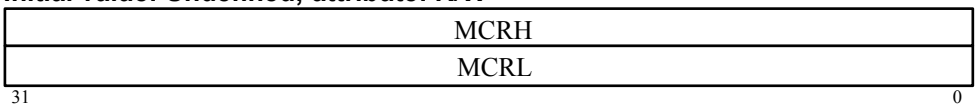
Register symbol: MDRQ
Initial value: Undefined; attribute: R/W



This register is used in the multiply/divide instructions. In the case of multiplication, this stores the upper 32 bits of 64-bit multiplication result. In the case of division, this stores the 32-bit surplus.

2.3.2.2. Multiply-and-accumulate Operation Registers

Register symbol: MCRH, MCRL
Initial value: Undefined; attribute: R/W



These registers are provided as accumulators for the multiply-and-accumulate operation that is performed by the extended operation unit. MCRH stores the upper 32 bits of the 64-bit multiply-and-accumulate operation result, and MCRL stores the lower 32 bits of the 64-bit multiply-and-accumulate operation result. For details, refer to the description of operation of individual instructions.

2.3.2.3. Multiply-and-accumulate Overflow Flag

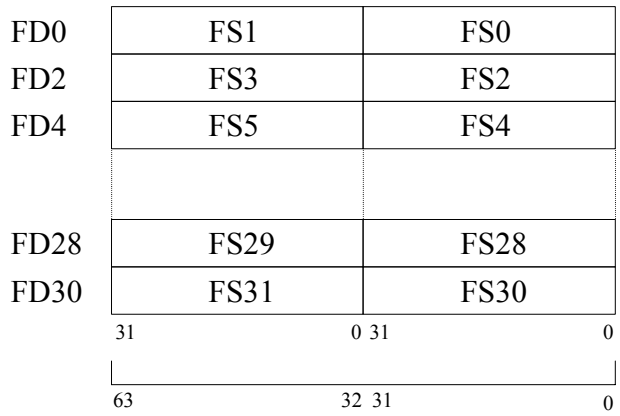
Register symbol: MCVF
Initial value: Undefined; attribute: R/W

This register is used to store the result of overflow detection for the result of executing the multiply-and-accumulate operation instruction. For details, refer to the description of operation of individual instructions.

2.3.3. Floating-point Register Set

2.3.3.1. Floating-point Registers

Initial value: undefined; attribute: R/W



The floating-point register consists basically of the thirty-two 32-bit single-precision floating-point registers, FS0 to FS31. These floating-point registers can be treated as sixteen 64-bit double-precision floating-point registers, FD0 to FD30.

When treated as double-precision floating-point register, it is treated as 64-bit double-precision floating-point register through connecting the two 32-bit single-precision floating-point registers to each other. Concretely, in the case of accessing to the double-precision floating-point register FDn (n:0 to 30, only the multiples of 2), the upper 32 bits of FDn (Bit 63-bit 32) are allocated to FSn+1, and the lower 32 bits (Bit 31-bit 0) of FDn to FSn.

The single-precision floating-point instruction treats the floating-point register as thirty-two 32-bit single-precision floating-point registers, FS0 to FS31, and the double-precision floating-point instruction treats the floating-point register as sixteen 64-bit double-precision floating-point registers, FD0 to FD30.

2.3.3.2. Floating-point Unit Control Register

Register symbol:	FPCR
Address:	FPCR can be accessed by MOVE instruction.
Purpose:	FPCR is a 32-bit register. FPCR includes bits that control the operation of the floating-point control unit, bits that control the operation of floating-point operation exceptions, and bits that indicate the status of floating-point operation exceptions. FPCR can be accessed as a 32-bit register by using a special MOV instruction (FMOV FPCR,Rn/ FMOV Rm,FPCR/ FMOV imm32,FPCR). An FPU disable exception is generated if an attempt is made to access FPCR while the FPSW.FE (FE: FPU Enable) bit has been reset to "0." If the FPSW.FE bit has been set to "1," FPCR can be accessed in

both user level and supervisor level.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	Reserved		FCC.L	FCC.G	FCC.E	FCC.U	RM	
Initial value	0		Undefined	Undefined	Undefined	Undefined	00	
R/W	R		R/W	R/W	R/W	R/W	R	
Bit	15	14	13	12	11	10	9	8
Bit name	reserved	EC.V	EC.Z	EC.O	EC.U	EC.I	EE.V	EE.Z
Initial value	0	Undefined	Undefined	Undefined	Undefined	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Bit name	EE.O	EE.U	EE.I	EF.V	EF.Z	EF.O	EF.U	EF.I
Initial value	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Description
31-22	reserved	These fields are reserved for future expansion. Reading these fields always returns a value of "0." When writing these fields, always write "0."
21	FCC.L	FPU floating-point condition code (When the comparison result is negative) This bit indicates that the comparison result of a floating-point compare operation being executed by the FCMP instruction is "negative."
20	FCC.G	FPU floating-point condition code (When the comparison result is positive) This bit indicates that the comparison result of a floating-point compare operation being executed by the FCMP instruction is "positive."
19	FCC.E	FPU floating-point condition code (When the comparison result is zero) This bit indicates that the comparison result of a floating-point compare operation being executed by the FCMP instruction is "0."
18	FCC.U	FPU floating-point condition code (When the comparison is impossible) This bit indicates that it is impossible to compare the floating-point value that is the source operand for an FCMP instruction. This bit is set to "1" if at least one of the source operands that is a floating-point number in an FCMP instruction is a quiet non-number. In all other cases, this bit is reset to "0."
17-16	RM	Rounding mode This sets the rounding mode of FPU operation. RM = 00: Round to nearest value RM = 01: reserved RM = 10: reserved RM = 11: reserved
15	reserved	These fields are reserved for future expansion. Reading these fields always returns a value of "0." When writing these fields, always write "0."
14	EC.V	FPU exception cause (in the case of the invalid operation cause) This bit is set to "1" if an FPU exception is generated due to an invalid operand. If an FPU exception due to any other cause is generated, this bit is reset to "0."
13	EC.Z	FPU exception cause (in the case of the divide cause by 0)

Bit	Bit name	Description
		This bit is set to "1" if an FPU exception is generated due to a division by zero. If an FPU exception due to any other cause is generated, this bit is reset to "0."
12	EC.O	FPU exception cause (in the case of an overflow cause) This bit is set to "1" if an FPU exception is generated due to an overflow. If an FPU exception due to any other cause is generated, this bit is reset to "0."
11	EC.U	FPU exception cause (in the case of an underflow cause) This bit is set to "1" if an FPU exception is generated due to an underflow. If an FPU exception due to any other cause is generated, this bit is reset to "0."
10	EC.I	FPU exception cause (in the case of an undefined cause) This bit is set to "1" if an FPU exception is generated due to an inexact value. If an FPU exception due to any other cause is generated, this bit is reset to "0."
9	EE.V	FPU exception cause (in the case of an invalid operation cause) This bit enables FPU exceptions due to an invalid operand. When EE.V = 0, FPU exceptions due to an invalid operand are not generated.
8	EE.Z	FPU exception enable (in the case of a divide cause by zero) This bit enables FPU exceptions due to a division by zero. When EE.Z = 0, FPU exceptions due to a division by zero are not generated.
7	EE.O	FPU exception enable (in the case of an overflow cause) This bit enables FPU exceptions due to an overflow. When EE.O = 0, FPU exceptions due to an overflow are not generated.
6	EE.U	FPU exception enable (in the case of an underflow cause) This bit enables FPU exceptions due to an underflow. When EE.U = 0, FPU exceptions due to an underflow are not generated.
5	EE.I	FPU exception enable (in the case of an undefined cause) This bit enables FPU exceptions due to an inexact value. When EE.I = 0, FPU exceptions due to an inexact value are not generated.
4	EF.V	FPU exception flag (in the case of an invalid operation cause) This bit is set to "1" if an FPU exception is generated due to an invalid operand.
3	EF.Z	FPU exception flag (in the case of a divide cause by zero) This bit is set to "1" if an FPU exception is generated due to a division by zero.
2	EF.O	FPU exception flag (in the case of an overflow cause) This bit is set to "1" if an FPU exception is generated due to an overflow.
1	EF.U	FPU exception flag (in the case of an underflow cause) This bit is set to "1" if an FPU exception is generated due to an underflow.
0	EF.I	FPU exception flag (in the case of an undefined cause) This bit is set to "1" if an FPU exception is generated due to an inexact value.

EF: FPU exception flags

These flags retain the FPU exception generation status. These flags are not cleared until they are explicitly cleared by software.

These flags can retain the generation status of individual FPU exception causes.

<Programming Note>

When an FPU exception flag is already set to "1," the flag is cleared by writing a "1" to that flag. The flag does not change if any other value is written to it. Therefore, it is not possible to set a flag to "1" through software.

EE: FPU exception enable
This field controls the generation of an FPU exception when an FPU exception cause is generated as a result of the execution of a floating-point operation instruction.
This field can be used to control individual FPU exception causes.
EC: FPU exception cause
This field retains the cause of an FPU exception when an FPU exception is generated.
This field has separate bits for each individual FPU exception cause.
RM: Rounding mode
This field sets the FPU rounding mode
<Programming Note>
FPCR.RM (rounding mode field) for floating-point unit in the AM33/2.0 microcontroller core can only be set to "00". If any other value is written to this field, that value is ignored. When read, the value that is returned is always "00".
FCC: FPU condition code
This field retains the status of the execution results of a floating-point operation instruction.

2.3.4. System Register Set

The system register set consists of registers that are used to control the operation and detect the status of the CPU and various interfaces. In addition to system registers that are compatible with the AM30/31/32 microcontroller core, the registers in this register set that concern the MMU have been extended and enhanced. This entire register set is located in the internal I/O space, and the registers are accessed as I/O registers in reading and writing. The system register set can be accessed only in supervisor level. An attempt to access the system register set at the user level will generate an illegal memory access exception.

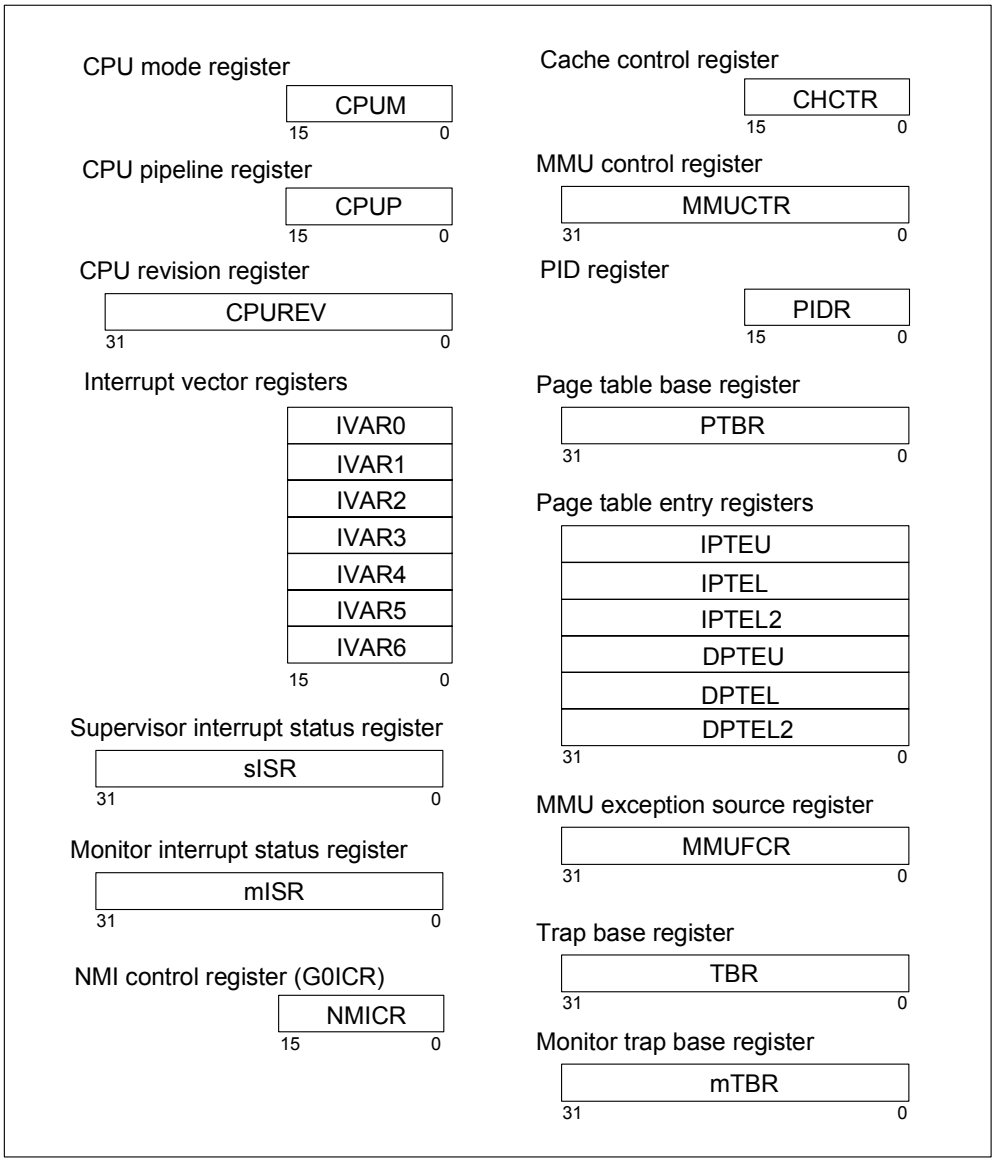


Figure 7 System register set

2.3.4.1. Register List

Table 23 CPU control register

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0000040	CPUM	CPU mode register	16	0x0000	8, 16, 32
0xC0000020	CPUP	CPU pipeline control register	16	0x0000	8, 16, 32
0xC0000050	CPUREV	CPU revision register	32	Note	32

Note: For the initial values, refer to page 95, 2.3.4.4 CPU Revision Register.

Table 24 Interrupt control register

Address	Symbol	Name	Number of bits	Initial value	Access size
0XC0000000	IVAR0	Interrupt vector register 0	16	Undefined	16, 32
0xC0000004	IVAR1	Interrupt vector register 1	16	Undefined	16, 32
0xC0000008	IVAR2	Interrupt vector register 2	16	Undefined	16, 32
0xC000000C	IVAR3	Interrupt vector register 3	16	Undefined	16, 32
0xC0000010	IVAR4	Interrupt vector register 4	16	Undefined	16, 32
0xC0000014	IVAR5	Interrupt vector register 5	16	Undefined	16, 32
0XC0000018	IVAR6	Interrupt vector register 6	16	Undefined	16, 32
0XC0000044	SISR	Supervisor interrupt status register	32	0x00000000	32
0XD4000000	NMICR	NMI control register	16	0x0000	16, 32
0XC0000038	DEAR	Data access exception address register	32	Undefined	32
0XC0000024	TBR	Trap base register	32	0x40000000	32
0XC0000028	-	System reserve	-	-	-
0XC0000030	-	System reserve	-	-	-
0XC0000034	-	System reserve	-	-	-
0XC0000060	-	System reserve	-	-	-
0XC0000100	-	System reserve	-	-	-
0XC0000104	-	System reserve	-	-	-
0XC0000108	-	System reserve	-	-	-
0XC0000120	-	System reserve	-	-	-
0XC0000124	-	System reserve	-	-	-
0XC0000128	-	System reserve	-	-	-
0XC000012C	-	System reserve	-	-	-
0XC0000140	-	System reserve	-	-	-
0XC0000144	-	System reserve	-	-	-
0XC0000148	-	System reserve	-	-	-
0XC000014C	-	System reserve	-	-	-
0XC0000150	-	System reserve	-	-	-
0XC0000154	-	System reserve	-	-	-
0XC0000158	-	System reserve	-	-	-
0XC000015C	-	System reserve	-	-	-
0XC0000160	-	System reserve	-	-	-
0XC0000164	-	System reserve	-	-	-
0XC0000168	-	System reserve	-	-	-

Address	Symbol	Name	Number of bits	Initial value	Access size
0XC000016C	-	System reserve	-	-	-
0XC0000170	-	System reserve	-	-	-

Note: The operations are not guaranteed when writing to the system reserve.

MMU Control registers

There are 8 registers related to MMU operation. These registers are located in the control register space and can be accessed only at the supervisor level or higher through specifying an address. Note that the space from S2 to S4 should be used for register updating and the space from SU0 to SU1 should be used after confirming that updating has been completed. If the space from SU0 to SU1 is used before updating registers is completed, the proper functioning of subsequent operations will not be assured.

For details, refer to page 155, 2.7.1 Address Space.

Table 25 MMU control register

Address	Symbol	Name	Number of bits	Initial value	Access size
0XC0000090	MMUCTR	MMU control register	32	0x00000000	32
0XC0000094	PIDR	Process identification register	16	Undefined	16
0XC0000098	PTBR	Page table base register	32	Undefined	32
0XC00000A4	IPTEU	Instruction page table entry upper register	32	Undefined	32
0XC00000B4	DPTEU	Data page table entry upper register	32	Undefined	32
0XC00000A0	IPTEL	Instruction page table entry lower register	32	Undefined	32
0XC00000B0	DPTEL	Data page table entry lower register	32	Undefined	32
0XC00000A8	IPTEL2	Instruction page table entry lower register 2	32	Undefined	32
0XC00000B8	DPTEL2	Data page table entry lower register 2	32	Undefined	32
0xC000009C	MMUFCR	MMU fault cause register	32	Undefined	32

Table 26 *Cache control register*

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0000070	CHCTR	Cache control register	16	0x0000	16

2.3.4.2. CPU Mode Register

Symbol: CPUM
Address: 0xC0000040
Purpose: This register sets the clock operation mode for the CPU core and its peripheral circuits. The flag settings that are used to transition to the different operation modes are shown in the table on the following page.
When the CPU core is not in the monitor level, the clock operation mode can be changed from NORMAL mode to either SLEEP, HALT, or STOP mode by setting an arbitrary bit in the CPUM to "1." In any of these cases, the clock that is supplied to the CPU stops.
The above operation is performed by writing of an arbitrary bit in CPUM, but the timing with which the clock signal stops depends on the internal status of the CPU and the status of the cache. The CPU finishes executing the instruction to be executed until the clock stopped, and then stops.
If a maskable interrupt or nonmaskable interrupt is generated while the clock operation mode is either SLEEP, HALT, or STOP, the clock operation mode changes to NORMAL mode and the CPU begins operating.
If interrupt is generated, the CPUM register value is cleared to "0" and the clock operation mode changes to NORMAL mode.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	reserved			STOP	HALT	SLEEP	reserved	
Initial value	0			0	0	0	0	
R/W	R			R/W	R/W	R/W	R	

Bit	Bit name	Description
15-5	reserved	These bits are reserved for future functional extension. A "0" is always returned when these bits are read. When writing this register, always write a "0" to these bits.
4	STOP	STOP mode change request flag This flag is set in order to change to stop mode. The CPU changes back to NORMAL mode by an interrupt after confirming the operational stabilization of the clock generator.
3	HALT	HALT mode change request flag This flag is set in order to change to HALT mode. The CPU changes back to NORMAL mode by an interrupt.
2	SLEEP	SLEEP mode change request flag This flag is set in order to change to SLEEP mode. The CPU changes back to NORMAL mode by an interrupt.
1-0	Reserved	These bits are reserved for future functional extension. A "0" is always returned when these bits are read. When writing this register, always write a "0" to these bits.

<Programming mote>
Set “1” by using the bset instruction when writing to STOP, HALT, and SLEEP.

2.3.4.3. CPU Pipeline Control Register

Symbol: CPUP
Address: 0xC0000020
Purpose: This register is reserved for the implementation of additional functions for CPU pipeline operations.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	EXM	IPFD	DWBD	reserved				
Initial value	0	0	0	0				
R/W	R/W	R/W	R/W	R				

Bit	Bit name	Description
15-8	reserved	These bits are reserved for future functional extension. A "0" is always returned when these bits are read. When writing this register, always write a "0" to these bits.
7	EXM	Exception operation mode There are two exception operation modes implemented in the AM33 microcontroller core. The exception operation mode bits are used in order to set this exception operation mode. Set this bit as shown below for the corresponding exception operation modes: 0: AM33/1.0 mode 1: AM33/2.0 mode
6	IPFD	Instruction pre-fetch disable flag This bit is set in order to disable the instruction pre-fetch operation. If the instruction pre-fetch operation is disabled, the instruction fetch operation is not performed until either the instruction buffer becomes empty or there are not enough instructions for execution.
5	DWBD	Write buffer disable flag Normally, the CPU does not wait until writes are completed. This write operation mode can be changed by setting this bit. However, the specific effects of this bit on CPU operation will depend on the implementation and vary among the kind.
4-0	reserved	These bits are reserved for future functional extension. A "0" is always returned when these bits are read. When writing this register, always write a "0" to these bits.

2.3.4.4. CPU Revision Register

Symbol: CPUREV (IDCODE)
Address: 0xC0000050
Purpose: This register indicates the CPU core revision. The data stored in this register includes the CPU type, the CPU version number, the core release type, the number of instruction cache ways, the instruction cache size, the number of data cache ways, the data cache size, and the on-chip debug function category. This register can be referenced when the system boots up, and the contents can be used to configure the system.

Bit	31	30	29	28	27	26	25	24
Bit name	DTYPE				CRTYPE			
Initial value	0000				0001			
R/W	R				R			
Bit	23	22	21	20	19	18	17	16
Bit name	DCS				DCW			
Initial value	0100				0100			
R/W	R				R			
Bit	15	14	13	12	11	10	9	8
Bit name	ICS				ICW			
Initial value	0100				0100			
R/W	R				R			
Bit	7	6	5	4	3	2	1	0
Bit name	CREV				CTYPE			
Initial value	NOTE				0001			
R/W	R				R			

NOTE : For the initial values, refer to the description of the following CPUREV[7:4].

Bit	Bit name	Description
31-28	DTYPE	On-chip debug category This field indicates the category of the on-chip debug function.
27-24	CRTYPE	Core release type This field indicates the name of the CPU core release This LSI reads out 0001.
23-20	DCS	Data cache size This field indicates the size of an individual data cache way.
19-16	DCW	Number of data cache ways This field indicates the number of data cache ways.
15-12	ICS	Instruction cache size This field indicates the size of an individual instruction cache way.
11-8	ICW	Number of instruction cache ways This field indicates the size of an individual instruction cache ways.
7-4	CREV	CPU revision This field indicates the revision number of the CPU core.
3-0	CTYPE	CPU type This field indicates the name of the CPU core release type.

CTYPE[3:0]	Core type name	Architecture
0000	AM33-1	AM33/1.00
0001	AM33-2	AM33/2.00
0010	AM34-1	AM33/2.00

Bit	Bit name	Description		
		Others	Reserved	-

2.3.4.5. Interrupt Vector Registers

Symbol: IVARn
Address: 0xC0000000(n=0), 0xC0000004(n=1), 0xC0000008(n=2) 0xC000000C(n=3), 0xC0000010(n=4), 0xC0000014(n=5), 0xC0000018(n=6),
Purpose: Interrupt Vector Address Registers IVAR0 to IVAR6 store the offset address from TBA[31:0] that corresponds to each interrupt level for the maskable interrupts. The entry points for maskable interrupt levels 0 through 6 correspond to IVAR0 to IVAR6. These registers can be accessed only in supervisor level.

If a maskable interrupt is generated, control shifts to a 32-bit address of which the upper 16 bits are the upper 16bits of TBA [31:0] and the lower 16 bits are IVARn (n=0 to 6) corresponding to the level of the maskable interrupt. These registers are not initialized by a reset.

Bit	15	14	13	12	11	10	9	8
Bit name	IVARn							
Initial value	undefined							
R/W	R/W							
Bit	7	6	5	4	3	2	1	0
Bit name	IVARn							
Initial value	undefined							
R/W	R/W							

Registers name	Description
IVAR0	This register stores interrupt vector 0. The interrupt code that corresponds to this vector is 0x280
IVAR1	This register stores interrupt vector 1. The interrupt code that corresponds to this vector is 0x288.
IVAR2	This register stores interrupt vector 2. The interrupt code that corresponds to this vector is 0x290.
IVAR3	This register stores interrupt vector 3. The interrupt code that corresponds to this vector is 0x298.
IVAR4	This register stores interrupt vector 4. The interrupt code that corresponds to this vector is 0x2A0.
IVAR5	This register stores interrupt vector 5. The interrupt code that corresponds to this vector is 0x2A8.
IVAR6	This register stores interrupt vector 6. The interrupt code that corresponds to this vector is 0x2B0.

<Programming Note>
For maskable interrupts, it is possible to set an interrupt vector entry for each interrupt level by setting IVAR. Therefore, it is not necessary to use the interrupt code in order to distinguish the interrupt level. The interrupt code can be used by the software to identify the interrupt that is currently being processed, etc.
<Programming Note>
This register is accessed through half-word (16-bit) access. It cannot be accessed through word (32-bit) or byte (8-bit) access.

2.3.4.6. Supervisor Interrupt Status Register

Symbol: sISR
Address: 0xC0000044
Purpose: The sISR (supervisor interrupt status register) stores the generation status of system exceptions. This register can be accessed only in monitor level or supervisor level.

Bit	31	30	29	28	27	26	25	24
Bit name	NE	reserved	FPUOP	FPUUI	FPUD	SYSC	DSIA	PRIDA
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Bit name	PRIVA	IOIA	ILGDA	ILGIA	DTEX	ITEX	DTMISS	ITMISS
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	15	14	13	12	11	10	9	8
Bit name	DBG	reserved	DBLFT	BUSERR	PRIV	reserved	EXUNIMP	reserved
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R	R/W	R
Bit	7	6	5	4	3	2	1	0
Bit name	reserved				UNIMP	MISSA	Reserved	
Initial value	0				0	0	0	
R/W	R				R/W	R/W	R	

Bit	Bit name	Description
31	NE	Multi-synchronous exception This bit is set to "1" if a synchronous exception is generated while EPSW.NMID is already set to "1."
30	reserved	These fields are reserved for future extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
29	FPUOP	FPU operation exception This bit is set to "1" if an FPU operation exception is generated.
28	FPUUI	FPU unimplemented instruction exception This bit is set to "1" if an FPU unimplemented instruction exception is generated.
27	FPUD	FPU disable exception This bit is set to "1" if an FPU disable exception is generated.
26	SYSC	System call instruction exception This bit is set to "1" if a system call instruction exception is generated.
25	DSIA	Data space instruction access exception This bit is set to "1" if a data space instruction access exception is generated.
24	PRIDA	Privileged space data access exception This bit is set to "1" if a privileged space TLB access exception is generated.
23	PRIVA	Privileged space instruction access exception This bit is set to "1" if a privileged space instruction access exception is generated.
22	IOIA	Internal IO space instruction access exception This bit is set to "1" if an internal I/O space instruction access exception

Bit	Bit name	Description
		is generated.
21	ILGDA	Illegal data access exception This bit is set to "1" if an illegal data access exception is generated.
20	ILGIA	Illegal instruction access exception This bit is set to "1" if an illegal instruction access exception is generated.
19	DTEX	Data TLB access exception This bit is set to "1" if a data TLB access exception is generated.
18	ITEX	Instruction TLB access exception This bit is set to "1" if an instruction TLB access exception is generated.
17	DTMISS	Data TLB miss This bit is set to "1" if a data TLB miss exception is generated.
16	ITMISS	Instruction TLB miss This bit is set to "1" if an instruction TLB miss exception is generated.
15	DBG	Debug reserved interrupt (monitor reserved interrupt) This bit is set to "1" if an interrupt from the interrupt group reserved for a debugger is generated. Writing to this bit does not change the setting of the bit. This bit can be cleared by clearing all of the flags by debugging.
14	Reserved	These fields are reserved for future extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
13	DBLFT	Double fault This bit is set to "1" if a nonrecoverable synchronous interrupt is generated during the hardware interrupt sequence (double fault).
12	BUSERR	Bus error This bit is set to "1" if a synchronous bus error is generated during bus access. Those bus errors that will cause the CPU to stall if they are not processed (in other words, bus errors that are triggered as a result of reading an instruction, reading data, or writing to the CPU control register space) are called "synchronous bus errors."
11	PRIV	Privileged instruction execution exception This bit is set to "1" if a privileged instruction execution exception is generated.
10	Reserved	These fields are reserved for future extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
9	EXUNIMP	Extended operation unit exception/ unimplemented extended instruction exception flag This bit is set to "1" if an exception is generated in an extended operation instruction.
8-4	Reserved	These fields are reserved for future extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
3	UNIMP	Unimplemented instruction exception flag This bit is set to "1" if an unimplemented instruction exception is generated.
2	MISSA	Misalignment flag This bit is set to "1" if a misalignment interrupt is generated.
1-0	Reserved	These fields are reserved for future extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."

<Programming Note>
A flag that is already set to "1" is cleared to "0" by writing a "1" to the flag. No other writes will change the setting of a flag. Therefore, it is not possible to set a flag to "1" through software.

2.3.4.7. NMI Control Register

Symbol: NMICR/G0ICR
Address: 0xD4000000
Purpose: NMICR/G0ICR (NMI Control Register) is the nonmaskable interrupt control register. This register can be accessed in monitor level and supervisor level only. When an NMI pin interrupt, a WDT overflow interrupt, or an asynchronous bus error (those bus error exceptions that are not synchronous bus errors) is generated, the flags are set as shown below.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	reserved				ABUSERR	reserved	WDIF	NMIF
Initial value	0				0	0	0	0
R/W	R				R/W	R	R/W	R/W

Bit	Bit name	Description
15-4	reserved	These fields are reserved for future functional expansion. Reading these fields always returns a value of "0." When writing these fields, always write "0."
3	ABUSERR	Asynchronous bus error This bit is set to "1" if an asynchronous bus error (those bus errors that are not synchronous bus errors) is generated.
2	reserved	These fields are reserved for future functional expansion. Reading these fields always returns a value of "0." When writing these fields, always write "0."
1	WDIF	WDT overflow flag This bit is set to "1" if a WDT overflow interrupt request is generated.
0	NMIF	NMI pin interrupt flag This bit is set to "1" if an NMI pin interrupt request is generated.

<Programming Note>
The corresponding flags are cleared by software within the interrupt processing program.
<Programming Note>
A flag that is already set to "1" is cleared to "0" by writing a "1" to the flag. No other writes will change the setting of a flag. Therefore, it is not possible to set a flag to "1" through software.
<Programming Note>
The flag that was set in the interrupt-processing program must be cleared for clearing the generated interrupt request. In the case of returning from the interrupt without clearing the interrupt request, there is a possibility of reaccepting the same interrupt after returning from the interrupt-processing program.
<Programming Note>
This register can be accessed through half-word (16-bit) access and byte (8-bit) access. It cannot be accessed through word (32-bit) access.

2.3.4.8. Data Access Exception Address Register

Symbol: DEAR
Address: 0xC0000038
Purpose: This register stores the address of the data that generated a data access exception.

Bit	31	30	29	28	27	26	25	24
Bit name	DEAR							
Initial value	undefined							
R/W	R/W							
Bit	23	22	21	20	19	18	17	16
Bit name	DEAR							
Initial value	undefined							
R/W	R/W							
Bit	15	14	13	12	11	10	9	8
Bit name	DEAR							
Initial value	undefined							
R/W	R/W							
Bit	7	6	5	4	3	2	1	0
Bit name	DEAR							
Initial value	undefined							
R/W	R/W							

Bit	Bit name	Description
31-0	DEAR	Data access exception address If any of the following data access exceptions are generated, this register stores the address of that data. - Illegal data access exception - Privileged space data access exception - Misalignment exception - Synchronous bus error during data access

2.3.4.9. Trap Base Register

Symbol: TBR
Address: 0xC0000024
Purpose: The TBR stores the base address and interrupt codes for all interrupt vectors. This register can be accessed only in supervisor level.

Bit	31	30	29	28	27	26	25	24
Bit name	TB							
Initial value	0x40							
R/W	R/W							
Bit	23	22	21	20	19	18	17	16
Bit name	INT_CODE							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	INT_CODE							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	INT_CODE							
Initial value	0							
R/W	R							

Bit	Bit name	Description
31-24	TB	Interrupt vector base This is the MSB of the interrupt vector base address. The initial value is 0x40 in order to ensure compatibility with the AM30/31/32 microcontroller core. The interrupt vector start address (interrupt base address TBA[31:0]) is the address that is generated when the interrupt code portion of TBR is "0." This field can be overwritten only in supervisor level.
23-0	INT_CODE	Interrupt code When an asynchronous interrupt or a synchronous interrupt is accepted, the corresponding interrupt code is set in this field.

<Programming Note>
In the AM33, when an asynchronous interrupt or a synchronous interrupt is accepted, control jumps to the interrupt vector that corresponds to the asynchronous interrupt or synchronous interrupt type.
<Programming Note>
Because multiple asynchronous interrupts and synchronous interrupts share the same interrupt vector, the interrupt code is used to indicate the specific cause. The following approaches are possible:
(1) Jump to the address indicated by the TBR.
(2) Load the jump address from the address indicated by the TBR.

2.3.4.10. MMU Registers

Symbol: MMUCTR
Address: 0xC0000090
Purpose: The MMU control register (MMUCTR) controls the operation of the MMUs. This register is written in units of 2 bytes or 4 bytes. This register can be read as individual bytes.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved					DTL		
Initial value	0					0		
R/W	R					R/W		
Bit	23	22	21	20	19	18	17	16
Bit name	DIV	DME	DRP					
Initial value	0	0	0					
R/W	R/W	R/W	R/W					
Bit	15	14	13	12	11	10	9	8
Bit name	CE	reserved				ITL		
Initial value	0	0				0		
R/W	R/W	R				R/W		
Bit	7	6	5	4	3	2	1	0
Bit name	IIV	IME	IRP					
Initial value	0	0	0					
R/W	R/W	R/W	R/W					

Bit	Bit name	Description
31-27	reserved	These fields are reserved for future functional extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
26-24	DTL	Data TLB entry lock pointer This field sets the data TLB entry lock space. The following settings are possible. 0x0: No lock 0x1:Entry 0 is locked 0x2:Entry 0 and 1 are locked 0x3:Entry 0 through 3 are locked 0x4:Entry 0 through 7 are locked 0x5:Entry 0 through 15 are locked 0x6:Setting prohibited 0x7:Setting prohibited
23	DIV	Data TLB invalid bit Setting this bit to "1" (by writing a "1" to this bit) invalidates all entries in the data TLB. When this bit is read, a "0" is output.
22	DME	Data MMU enable bit This bit controls whether the data TLB is enabled or disabled. When this bit is "1," the data TLB is enabled; when this bit is "0," the data TLB is disabled.
21-16	DRP	Data TLB replace pointer This field indicates the data TLB replace entry address. Although this field is set automatically by the hardware, it can be set as desired by the

Bit	Bit name	Description
		software. *<Programming Note>* Replacement by the hardware is performed on an FIFO basis.
15	CE	Cacheable bit enable This bit enables the setting of the "C" (cacheable) bit in each page table entry of the instruction/data TLB. If the cacheable bit enable is set to "0," the SU0 space is cached according to the settings in the cache control register, with no regard for the "C" bit. Caching is not performed at all for the SU1 space. If cacheable bit enable is set to "1," caching is controlled for individual pages in the SU0 and SU1 spaces according to the C bit.
14-11	reserved	These fields are reserved for future functional extension. Reading these fields always returns a value of "0." When writing these fields, always write "0."
10-8	ITL	Instruction TLB lock pointer This field sets the instruction TLB entry lock space. The following settings are possible. 0x0: No lock 0x1:Entry 0 is locked 0x2: Entry 0 and 1 are locked 0x3:Entry 0 through 3 are locked 0x4:Entry 0 through 7 are locked 0x5:Entry 0 through 15 are locked 0x6:Setting prohibited 0x7:Setting prohibited
7	IIV	Instruction TLB invalid bit Setting this bit to "1" (by writing a "1" to this bit) invalidates all entries in the instruction TLB. When this bit is read, a "0" is output.
6	IME	Instruction MMU enable bit This bit controls whether the instruction TLB is enabled or disabled. When this bit is "1," the instruction TLB is enabled; when this bit is "0," the instruction TLB is disabled.
5-0	IRP	Instruction TLB replace pointer This field shows the instruction TLB replace entry address. Although this field is set automatically by the hardware, it can also be set as desired by the software. *<Programming Note>* *Replacement by the hardware is performed on a FIFO basis.*

2.3.4.11. Process Identifier Register

Symbol: PIDR
Address: 0xC0000094
Purpose: The PID register (PIDR) stores the 8-bit identification number that is assigned to each process. This register is written in units of 2 bytes. This register can be read as individual bytes.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	PID							
Initial value	undefined							
R/W	R/W							

Bit	Bit name	Description
15-8	reserved	This field is reserved for future functional extension. Reading this field always returns a value of "0." When writing this field, always write "0."
7-0	PID	Process identifier This field shows the current process number.

2.3.4.12. Page Table Base Register

Symbol: PTBR
Address: 0xC0000098
Purpose: The page table base register (PTBR) stores the base address for the page table that is currently being used. This register is written in units of 4 bytes. (This register can be read as individual bytes.)
The contents of PTBR are not updated unless so instructed by the software.

Bit	31	30	29	28	27	26	25	24
Bit name	PTB							
Initial value	undefined							
R/W	R/W							
Bit	23	22	21	20	19	18	17	16
Bit name	PTB							
Initial value	undefined							
R/W	R/W							
Bit	15	14	13	12	11	10	9	8
Bit name	PTB							
Initial value	undefined							
R/W	R/W							
Bit	7	6	5	4	3	2	1	0
Bit name	PTB							
Initial value	undefined							
R/W	R/W							

Bit	Bit name	Description
31-0	PTB	Page table base address This stores a base address of the page table being currently used.

2.3.4.13. Page Table Entry Upper Register

Symbol: IPTEU/DTPEU
Address: 0xC00000A4/0xC00000B4
Purpose: The page table entry upper register (IPTEU/DPTEU) consists of a virtual page number (VPN) and the PID. This register is written in units of 4 bytes. This register can be read as individual bytes. If a TLB miss exception or an access exception is generated, the hardware sets the VPN and PID of the logical address where the exception was generated in the VPN and PID fields.

Bit	31	30	29	28	27	26	25	24
Bit name	VPN							
Initial value	undefined							
R/W	R/W							
Bit	23	22	21	20	19	18	17	16
Bit name	VPN							
Initial value	undefined							
R/W	R/W							
Bit	15	14	13	12	11	10	9	8
Bit name	VPN						reserved	
Initial value	undefined						0	
R/W	R/W						R	
Bit	7	6	5	4	3	2	1	0
Bit name	PID							
Initial value	undefined							
R/W	R/W							

Bit	Bit name	Description
31-10	VPN	Virtual page number This field stores the upper 22 bits of the logical address.
9-8	reserved	This field is reserved for future functional extension. Reading this field always returns a value of "0." When writing this field, always write "0."
7-0	PID	Process identifier This field shows the number of the process that can allocate the virtual page.

2.3.4.14. Page Table Entry Lower Register

Symbol: IPTEL/DPTEL
Address: 0xC00000A0/0xC00000B0
Purpose: The page table entry lower register (IPTEL/DPTEL) consists of a physical page number (PPN) and the page attributes. TLB replacement is performed by writing to this register. The TLB entry lookup operation is performed by reading this register. This register is written in units of 4 bytes. This register can be read as individual bytes.

Bit	31	30	29	28	27	26	25	24
Bit name	PPN[31:24]							
Initial value	undefined							
R/W	R/W							
Bit	23	22	21	20	19	18	17	16
Bit name	PPN[23:16]							
Initial value	undefined							
R/W	R/W							
Bit	15	14	13	12	11	10	9	8
Bit name	PPN[15:12]				PS		G	PR
Initial value	undefined				undefined		undefined	undefined
R/W	R/W				R/W		R/W	R/W
Bit	7	6	5	4	3	2	1	0
Bit name	PR		D	PV	C	PPN[11:10]		V
Initial value	undefined		undefined	undefined	undefined	undefined		undefined
R/W	R/W		R/W	R/W	R/W	R/W		R/W

Bit	Bit name	Description
31-12	PPN[31:12]	Physical page number These bits store the upper 20 bits of the physical address.
11-10	PS	Page size bit This field indicates the size of the page that was accessed. The meanings of these bits are defined below. 00 : 4Kbytes 01 : 128 Kbytes 10 : 1Kbytes 11 : 4Mbytes
9	G	Shared bit This bit indicates whether the PID is compared during tag comparison. If this bit is "1," the PID is not compared. If this bit is "0," the PID is compared.
8-6	PR	Page protection bit This bit indicates the page access permission. The meanings of the value of this field is described below: 000 : Can be read (R) only in privileged level 001 : Setting prohibited 010 : Can be read (R) in privileged level and user level 011 : Setting prohibited 100 : Can be read/written (R/W) only in privileged level 101 : Setting prohibited 110 : Can be read/written (R/W) in privileged level, and can only be read (R) in user level

Bit	Bit name	Description
		111 : Can be read/written (R/W) in privileged level and user level
5	D	Dirty bit This bit indicates whether the page that was accessed was written or not. A value of "1" indicates that the page was written, and a value of "0" indicates that the page was not written.
4	PV	Page valid bit This bit indicates whether the page that was read is valid or not. A value of "1" indicates that the page is valid, and a value of "0" indicates that the page is invalid.
3	C	Cacheable bit This bit indicates whether the page is cacheable or uncacheable. A value of "1" indicates that the page is cacheable, and a value of "0" indicates that the page is uncacheable. The cacheable bit setting is valid only when the cacheable bit enable in MMUCTR is set to "1."
2-1	PPN[11:10]	Physical page number These bits store the 11^{th} and 10^{th} bits of the physical address.
0	V	TLB entry valid bit When the IPTEL or DPTEL is read, this bit indicates whether the data that was read is valid or not. If the value of this bit is "1," the data is valid; if the value is "0," the data is not valid.

2.3.4.15. Page Table Entry Lower Register 2

Symbol: PTEL2/DPTEL2
Address: 0xC00000A8/0xC00000B8
Purpose: The page table entry lower register 2(IPTEL2/DPTEL2) consists of a physical page number (PPN) and the page attributes.
TLB replacement is performed by writing to this register. The TLB entry lookup operation is performed by reading this register. This register is written in units of 4 bytes. This register can be read as individual bytes.
<Programming Note>
Page table entry lower register 2 has a completely equivalent function to page table entry lower register, except that the bit assignments are different.

Bit	31	30	29	28	27	26	25	24
Bit name	PPN[31:24]							
Initial value	undefined							
R/W	R/W							
Bit	23	22	21	20	19	18	17	16
Bit name	PPN[23:16]							
Initial value	undefined							
R/W	R/W							
Bit	15	14	13	12	11	10	9	8
Bit name	PPN[15:10]						PS	
Initial value	undefined						undefined	
R/W	R/W						R/W	
Bit	7	6	5	4	3	2	1	0
Bit name	G	PR			D	PV	C	V
Initial value	undefined	undefined			undefined	undefined	Undefined	undefined
R/W	R/W	R/W			R/W	R/W	R/W	R/W

Bit	Bit name	Description
31-10	PPN[31:10]	Physical page number These bits store the upper 22 bits of the physical address.
9-8	PS	Page size bit This field indicates the size of the page that was accessed. The meanings of these bits are defined below. 00 : 4Kbytes 01 : 128 Kbytes 10 : 1Kbytes 11 : 4Mbytes
7	G	Shared bit This bit indicates whether the PID is compared during tag comparison. If this bit is "1," the PID is not compared. If this bit is "0," the PID is compared.
6-4	PR	Page protection bit This bit indicates the page access permission. The meanings of the value of this field is described below: 000 : Can be read (R) only in privileged level 001 : Setting prohibited 010 : Can be read (R) in privileged level and user level

Bit	Bit name	Description
		011 : Setting prohibited 100 : Can be read/written (R/W) only in privileged level 101 : Setting prohibited 110 : Can be read/written (R/W) in privileged level, and can only be read (R) in user level 111 : Can be read/written (R/W) in privileged level and user level
3	D	Dirty bit This bit indicates whether the page that was accessed was written or not. A value of "1" indicates that the page was written, and a value of "0" indicates that the page was not written.
2	PV	Page valid bit This bit indicates whether the page that was read is valid or not. A value of "1" indicates that the page is valid, and a value of "0" indicates that the page is invalid.
1	C	Cacheable bit This bit indicates whether the page is cacheable or uncacheable. A value of “1” indicates that the page is cacheable, and a value of “1” indicates that the page is uncacheable. The cacheable bit setting is valid only when the cacheable bit enable in MMUCTR is set to “1”.
0	V	TLB entry valid bit When the IPTEL or DPTEL is read, this bit indicates whether the data that was read is valid or not. If the value of this bit is "1," the data is valid; if the value is "0," the data is not valid.

2.3.4.16. MMU Exception Cause Register (MMUFCR)

Symbol: MMUFCR
Address: 0xC000009C
Purpose: The MMU exception cause register (MMUFCR) stores the cause flags for exceptions that are generated due to the instruction TLB and the data TLB. During exception processing, the cause of an exception can be determined by referencing this register.
This register can be read as individual bytes.

Bit	31	30	29	28	27	26	25	24
Bit name	Reserved						DFC	
Initial value	0						0	
R/W	R						R	
Bit	23	22	21	20	19	18	17	16
Bit name	DFC							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	Reserved						IFC	
Initial value	0						0	
R/W	R						R	
Bit	7	6	5	4	3	2	1	0
Bit name	IFC							
Initial value	0							
R/W	R							

Bit	Bit name	Description
31-26	reserved	These fields are reserved for future functional extension. Reading this field always returns a value of “0”. When writing this field, always write “0”.
25-16	DFC	Data access exception cause code This field stores the code that indicates the exception cause that generated a data access exception. For the details about the exception cause codes, refer to page 162, 2.7.5 Exceptions.
15-10	reserved	These fields are reserved for future functional extension. Reading this field always returns a value of "0." When writing this field, always write "0."
9-0	IFC	Instruction access exception This field stores the code that indicates the exception cause that generated an instruction access exception. For the details about the exception cause codes, refer to page 162, 2.7.5 Exceptions.

2.3.4.17. Cache Control Register

Symbol: CHCTR
Address: 0xC0000070
Purpose: The cache control register (CHCTR) is used to make various settings concerning cache operation. In order to use a cache, the cache control register (CHCTR) settings and the cache need to be initialized. Reads and writes are performed in 2-byte units.

Bit	15	14	13	12	11	10	9	8
Bit name	DCWMD				ICWMD			
Initial value	0				0			
R/W	R/W				R/W			
Bit	7	6	5	4	3	2	1	0
Bit name	DCALMD	DCWTMD	DCINV	ICINV	DCBUSY	ICBUSY	DCEN	ICEN
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit name	Description
15-12	DCWMD	Data cache way mode Each bit from CHCTR[12] to CHCTR[15] sets the way operation mode for way 0 through way 3, respectively. A "0" indicates normal operation, while "1" indicates that the refill operation is not to be performed, even if a cache miss occurs.
11-8	ICWMD	Instruction cache way mode Each bit from CHCTR[8] to CHCTR[11] sets the way operation mode for way 0 through way 3, respectively. A "0" indicates normal operation, while "1" indicates that the refill operation is not to be performed, even if a cache miss occurs.
7	DCALMD	Data cache allocate mode When write-back mode is selected, this bit specifies the allocation method that is used when a data cache write miss occurs. A "0" indicates "allocate" (write only in the cache, not in external memory), a "1"indicates "do not allocate" (write only in external memory, do not write in the cache). When write-through mode is selected, "do not allocate" (write only in external memory, do not write in the cache) is selected, regardless of the setting of this bit.
6	DCWTMD	Data cache write mode This bit specifies the data cache writing mode. A "0" indicates write-back mode, a "1" indicates write-through mode.
5	DCINV	Data cache invalidate All ways and entries in the data cache can be invalidated by writing a "1" to this bit. This operation is accomplished by clearing the valid bit (V) for all entries in tag memory. When this bit is read, a "0" is always returned.
4	ICINV	Instruction cache invalidate All ways and entries in the instruction cache can be invalidated by writing a "1" to this bit. This operation is accomplished by clearing the valid bit (V) for all entries in tag memory. When this bit is read, a "0" is always returned.
3	DCBUSY	Data cache busy This bit indicates whether the data cache is busy or not. This bit

Bit	Bit name	Description
		must be checked when directly accessing the contents of data memory and tag memory. If this bit is "0," the data cache is idle; if this bit is "1," the data cache is busy.
2	ICBUSY	Instruction cache busy This bit indicates whether the instruction cache is busy or not. This bit must be checked when directly accessing the contents of data memory and tag memory. If this bit is "0," the instruction cache is idle; if this bit is "1," the instruction cache is busy.
1	DCEN	Data cache enable This bit sets whether the data cache is to be used or not. When this bit is "0," the data cache is disabled; when this bit is "1," the data cache is enabled.
0	ICEN	Instruction cache enable This bit sets whether the instruction cache is to be used or not. When this bit is "0," the instruction cache is disabled; when this bit is "1," the instruction cache is enabled.

2.4. Data Formats

There are four data types for integer data: bit, byte, half-word, and word. Byte data, half-word data, and word data can all be handled as either signed data or unsigned data. In the case of signed data, the MSB is the sign bit. For floating-point values, the following two formats can be handled: single-precision floating-point values and double-precision floating-point values.

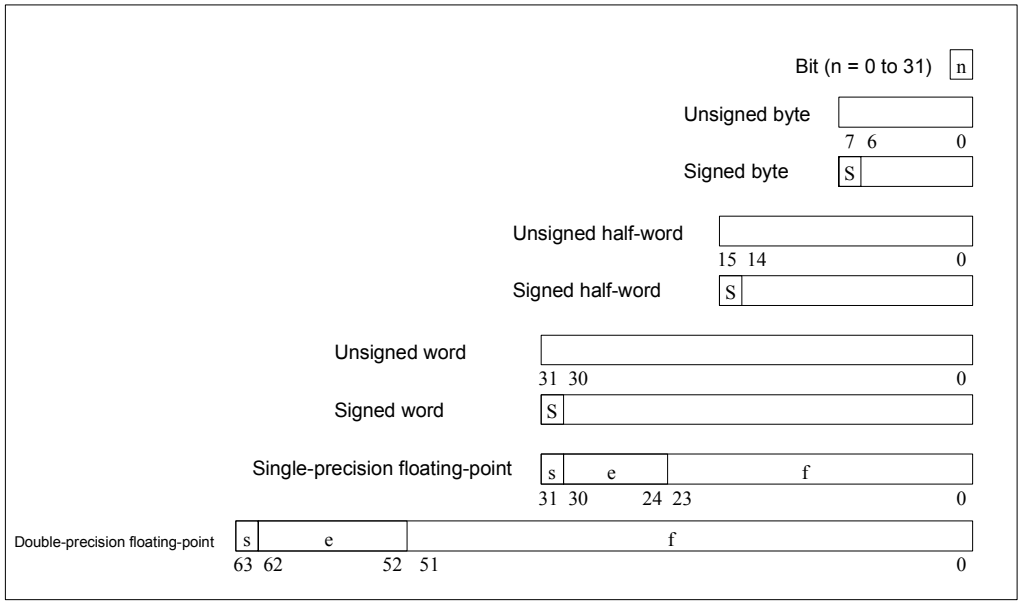


Figure 8 Data formats

Integer data in memory must be properly aligned. In short, the two bits on the LSB side of an address where word data is stored must be "00" (so that the address is a multiple of 4), and the one bit on the LSB side of an address where half-word data is stored must be "0" (so that the address is a multiple of 2). If data that is not properly aligned is accessed, a misalignment exception is generated. (For details, refer to page 125, 2.6 Interrupt System.) Floating-point data in memory must also be aligned in a similar fashion. In short, the two bits on the LSB side of an address where single-precision floating-point data is stored must be "00" (so that the address is a multiple of 4). When double-precision floating-point data is stored in memory, in the case that the double-precision floating-point load/store instruction (FMOV) is being used to access memory, the three bits on the LSB side of the address must be "000" (so that the address is a multiple of 8). When using two single-precision floating-point load/store instructions to access data, the two bits on the LSB side of the address must be "00" (so that the address is a multiple of 4). If data that is not properly aligned is accessed, a misalignment exception is generated. (For details, refer to page 125, 2.6 Interrupt System.)

Bit No Memory address	31 24 (4n+3)	23 16 (4n+2)	15 8 (4n+1)	7 0 (4n)
Word data	Address : 4n			
Half word data	Address : 4n+2		Address : 4n	
Byte date	Address : 4n+3	Address : 4n+2	Address : 4n+1	Address : 4n

Table 27 Data assignment

The bytes are positioned according to Little Endian format. Therefore, the address of the byte

data that is on the MSB side of half-word data is one greater than the address of the byte data that is on the LSB side of the half-word data. The address of the byte data that is on the MSB side of word data is three greater than the address of the byte data that is on the LSB side of the word data. Regarding bit numbers, the bit on the LSB side is "bit 0," and the bit numbers increase sequentially towards the MSB side.

2.5. Instructions

2.5.1. Instruction Formats

There are 16 instruction formats. The instruction set has a variable word length in which the basic word length is one byte, and the instruction length can vary in units of one byte. The shortest format is one-byte-long S0 format. The longest formats are seven-byte-long S6 format, D5 format, and T4 format.

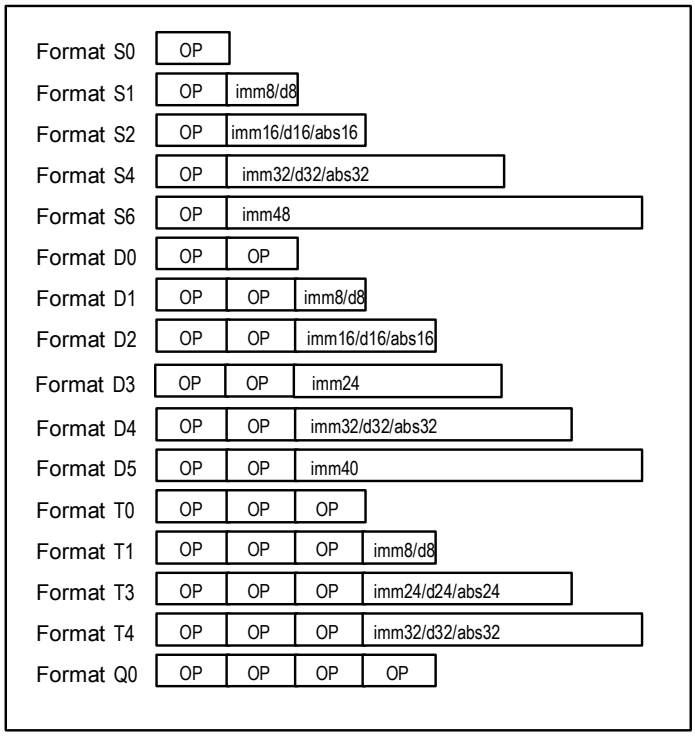


Figure 9 Instruction format

Normally, the opcode is followed by an 8-, 16-, or 32-bit immediate value, displacement value, or absolute value. However, in the instructions with formats S2, S4, S6, D2, and D5, the opcode is followed by two or more immediate values, displacement values, or absolute values; these are noted as a whole as 16-bit immediate values (imm16), 24-bit immediate values (imm24), 32-bit immediate values (imm32), 40-bit immediate values (imm40), and 48-bit immediate values (imm48). According to these notations, the following instructions accept 16-, 24-, 32-, 40-, or 48-bit immediate values:

imm16:	RET	regs,imm8
	RETF	regs,imm8
	BTST	imm8,(d8,An)
	BSET	imm8,(d8,An)
	BCLR	imm8,(d8,An)
imm24:	BTST	imm8,(abs16)
	BSET	imm8,(abs16)
	BCLR	imm8,(abs16)
imm32:	CALL	(d16,PC),regs,imm8
imm40:	BTST	imm8,(abs32)

```
                BSET    imm8,(abs32)
                BCLR    imm8,(abs32)
imm48:          CALL    (d32,PC),regs,imm8
```

2.5.2. Addressing Modes

There are six addressing modes that are frequently used by the compiler:

Addressing mode		Address calculation	Logical address
Register direct	Rm/Rn MDR/PSW/EPSW/SP MDRQ/MCRH/MCRL/MCVF FSm/FSn/FDm/FDn	-	-
Immediate	imm8/regs, imm16, imm24, imm32, imm40, imm48	-	-
Register indirect	(Rm)/(Rn) (Rm+)/(Rn+)	Rm,Rm+/Rn,Rn+ 31 0	32-bit address 31 0
Register relative indirect	(d8,Rm)/(d8,Rn) (d16,Am)/(d16,An) (d24,Rm)/(d24,Rn) (d32,Rm)/(d32,Rn) d8,d16,d24:sign extension	Rm,Am/Rn,An 31 + 0 d32/d24/d16/d8 31 23 15 7 0	32-bit address 31 0
	(d8,PC) (d16,PC) (d32,PC) d8,d16:sign extension (used only in branch instructions)	PC 31 + 0 d32/d16/d8 31 15 7 0	32-bit address 31 0
	(d8,SP) (d16,SP) (d24,SP) (d32,SP) d8,d16,d24:zero extension	SP 31 + 0 d32/d24/d16/d8 31 23 15 7 0	32-bit address 31 0
Absolute	(abs8), (abs16), (abs24),(abs32) abs8, abs16, abs24:zero extension	abs32/24/16/8 31 23 15 0	32-bit address 31 0
Register indirect relative with indexed addressing mode	(Ri,Rm)/(Ri,Rn)	Rm/Rn 31 + 0 Ri 31 0	32-bit address 31 0

Table 28 Addressing Mode Types

Data transfer instructions permit the use of six different addressing modes: register direct, immediate, register indirect, register relative indirect, absolute, and register indirect with indexed addressing mode. Register operation instructions permit the use of two addressing modes: register direct and immediate. Register indirect with indexed addressing mode is used to efficiently access data in an array, etc.

2.5.3. Instruction Set

The instruction set is based on a simple instruction set. A C compiler will produce a code that is compact and optimized from this instruction set.

Although the basic instruction word length is one byte and the instruction set is a simple one that limits data transfers with memory to load and store operations, it is possible to minimize the increase in code size due to the assembler program. Furthermore, because the generated code is compact, more instructions can be placed in the limited cache memory space, resulting in an improved cache hit ratio and making it possible to minimize the degradation of performance that results from accessing external memory in the event of a miss-hit.
The AM33 microcontroller core instruction set consists of five instruction categories: basic instructions, extended basic instructions, extended operation instructions, LIW extended operation instructions, and floating-point operation instructions. The basic instructions are common throughout the entire AM30 Series; these instructions maintain compatibility between the different microcontrollers in the series. The extended basic instructions are an extension of the basic instructions for the AM33 microcontroller core and are intended for the enhancement of the interrupt functions and the effective use of the extended registers. Functionally, these instruction are equivalent to basic instructions. The extended operation instructions provide compatibility with the extended instructions that were implemented in the AM31 microcontroller core. The LIW extended operation instructions support parallel operations on data. The floating-point operation instructions provide basic floating-point operations, such as arithmetic functions handling floating-point numbers and multiply-and-accumulate operations.
The instructions are all listed in the following chart. There are 47 basic instructions, 18 extended operation instructions, 70 LIW extended operation instructions, and 15 floating-point operation instructions. (Almost all of the extended basic instructions were implemented as operand extensions, so they have been included in the count of the basic instructions.)

Table 29 Instruction types

Basic instructions/extended basic instructions
Transfer instructions
MOV, MOVU, MOVHU, MOVBU, MOVM, EXT, EXTH, EXTHU, EXTB, EXTBU, CLR, DCPF
Arithmetic operation instructions
ADD, ADDC, INC, INC4, SUB, SUBC, MUL, MULU, DIV, DIVU
Compare instruction
CMP
Logic operation instructions
AND, OR, XOR, NOT
Shift instructions
ASR, LSR, ASL, ASL2, ROR, ROL
Bit manipulation instructions
BTST, BSET, BCLR
Branch instructions
Bcc, Lcc, SETLB, JMP, CALL, CALLS, TRAP, RET, RETF, RETS, RTI, SYSCALL
NOP instruction
NOP

Extended operation instructions
DMULH, DMULHU, MAC, MACU, MACH, MACHU, MACB, MACBU,
DMACH, DMACHU, SAT16, SAT24, MCSTE
BSCH, SWAP, SWAPH, SWHW

LIW extended operation instructions
ADD_ADD, ADD_SUB, ADD_CMP, ADD_MOV, ADD_ASR, ADD_LSR, ADD_ASL
SUB_ADD, SUB_SUB, SUB_CMP, SUB_MOV, SUB_ASR, SUB_LSR, SUB_ASL
CMP_ADD, CMP_SUB, CMP_MOV, CMP_ASR, CMP_LSR, CMP_ASL
MOV_ADD, MOV_SUB, MOV_CMP, MOV_MOV, MOV_ASR, MOV_LSR, MOV_ASL

AND_ADD, AND_SUB, AND_CMP, AND_MOV, AND_ASR, AND_LSR, AND_ASL
OR_ADD, OR_SUB, OR_CMP, OR_MOV, OR_ASR, OR_LSR, OR_ASL
XOR_ADD, XOR_SUB, XOR_CMP, XOR_MOV, XOR_ASR, XOR_LSR, XOR_ASL
DMACH_ADD, DMACH_SUB, DMACH_CMP, DMACH_MOV,
DMACH_ASR, DMACH_LSR, DMACH_ASL
SAT16_ADD, SAT16_SUB, SAT16_CMP, SAT16_MOV,
SAT16_ASR, SAT16_LSR, SAT16_ASL
SWHW_ADD, SWHW_SUB, SWHW_CMP, SWHW_MOV,
SWHW_ASR, SWHW_LSR, SWHW_ASL
MOV_Lcc

Floating-point operation instructions
FMOV, FABS, FNEG, FCMP, FRSQRT, FADD, FSUB, FMUL, FDIV,
FMADD, FMSUB, FNMADD, FNMSUB
FBcc, FLcc

Debug instruction
PI

2.5.3.1. Transfer Instructions

Transfer instructions are used to transfer data between registers, between memory and registers, and between memory and the data cache. Transfer instructions are grouped as either MOV-type instructions, EXT-type instructions, CLR-type instructions, and DCPF-type instructions.
MOV-type instructions provide data transfer functions using various addressing modes. Depending on the operation, displacement and immediate values also carry out a sign extension.
EXT-type instructions provide transfer functions between registers with a sign extension. The CLR-type instruction provides a function that clears the contents of registers (a function that transfers "0" into the registers). The DCPF-type instruction provides a function that pre-fetches data into the data cache. Except for the CLR-type instruction, none of these instructions generates any changes among flags.

Table 30 Transfer Instruction list

Instruction	Operation
MOV	Word transfer between registers, immediate value transfer between registers Word transfer between a register and memory (load/store) Word transfer with post-incrementing between a register and memory (load/store)
MOVU	Immediate value transfer to register (immediate value is with zero extension)
MOVHU	Zero extension half-word transfer between a register and memory (load/store) Zero extension half-word transfer with post-incrementing between a register and memory (load/store)
MOVBU	Zero extension byte transfer between a register and memory (load/store)
MOVM	Block transfer between multiple registers and memory (load/store)
EXT	64-bit signed extension for word data (processing between registers)
EXTH	32-bit signed extension for half-word data (processing between registers)
EXTHU	32-bit zero extension for half-word data (processing between registers)
EXTB	32-bit signed extension for byte data (processing between registers)
EXTBU	32-bit zero extension for byte data (processing between registers)
CLR	Data clear (transfers 0 to register)
DCPF	Data cache line data pre-fetch (transfers from memory to data cache)

2.5.3.2. Arithmetic Operation Instructions

The arithmetic operation instructions perform arithmetic operations on source operands and store the results in a register. These instructions may cause changes in flags. The "+1" and "+4" addition instructions, which are frequently used for address calculation, have been established as separate instructions.

Table 31 Arithmetic instructions

Instruction	Description
ADD	Addition between registers, addition between an immediate value and a register
ADDC	Addition with carry between registers, addition with carry between an immediate value and a register
SUB	Subtraction between registers, subtraction between an immediate value and a register
SUBC	Subtraction with borrow between registers, subtraction with borrow between an immediate value and a register
MUL	Signed multiplication between registers, signed multiplication between an immediate value and a register
MULU	Unsigned multiplication between registers, unsigned multiplication between an immediate value and a register
DIV	Signed division between registers, signed division between an immediate value and a register
DIVU	Unsigned division between registers, unsigned division between an immediate value and a register
INC	Adds ”1” to the content of a register
INC4	Adds “4” to the content of a register

2.5.3.3. Compare Instruction

The compare instruction compares the contents of two registers, or compares an immediate value with the content of a register. This instruction is primarily used ahead of a condition branch instruction. This instruction may cause changes in flags.

Table 32 Compare Instruction

Instruction	Description
CMP	Comparison of the contents of two registers, or of an immediate value and the content of a register

2.5.3.4. Logic Operation Instructions

The logic operation instructions perform logic operations on source operands and store the results in a register. These instructions may cause changes in flags.

Table 33 Logical operation instructions

Instruction	Description
AND	AND operation between registers, or AND operation between an immediate value and a register
OR	OR operation between registers, or OR operation between an immediate value and a register
XOR	Exclusive OR operation between registers, or exclusive OR operation between an immediate value and a register
NOT	Inversion of all bits in a register (one’s complement processing)

2.5.3.5. Bit Manipulation Instructions

The bit manipulation instructions perform bit manipulation operations between immediate values and the contents of registers, between immediate values and the contents of memory, or between the contents of registers and the contents of memory. These instructions may cause changes in flags.

Table 34 Bit Manipulation instructions

Instruction	Description
BTST	Multiple bit test (between an immediate value and a register, between an immediate value and memory)
BSET	Multiple bit test and set (between a register and memory, between an immediate value and memory)
BCLR	Multiple bit test and clear (between a register and memory, between an immediate value and memory)

2.5.3.6. Shift Instructions

The shift instructions perform bit shifts of the specified amount. Regardless of the amount of the shift, the instructions can be performed in one cycle. These instructions may cause changes in flags.

Table 35 Shift instructions

Instruction	Description
ASR	Arithmetic right shift of any number of bits
LSR	Logic right shift of any number of bits
ASL	Arithmetic left shift of any number of bits
ASL2	Arithmetic left shift of two bits
ROR	Rotate right one bit
ROL	Rotate left one bit

2.5.3.7. NOP Instruction

The NOP instruction performs no operation.

Table 36 NOP instruciton

Instruction	Description
NOP	No operation

2.5.3.8. Branch Instructions

Branch instructions are instructions that change the flow of program execution according to some conditions. There are two types of conditional branch instructions: normal conditional branch instructions, and loop-only conditional branch instructions. The loop-only conditional branch instructions minimize the branching penalty and permit fast loop execution by using dedicated registers. Subroutine calls and returns are a highly functional method of manipulating the PC, saving (restoring) multiple registers to (from) the stack, and allocating/releasing stack area.

Table 37 Branch Instructions

Instruction	Description
Bcc	Conditional branch (branches to PC-relative address)
Lcc	Loop-dedicated conditional branch (branches to start of loop set by SETLB)
SETLB	Registration of loop start information
JMP	Unconditional branch (PC relative, register indirect)
CALL	Subroutine call (saves next PC and multiple registers to stack, and allocates stack area)
CALLS	Subroutine call (saves next PC only)
RET	Return from subroutine (restores stack contents and releases stack area)
RETF	Return from subroutine (restores stack contents and releases stack area)
RETS	Return from subroutine (restores PC only)
RTI	Return from interrupt program
TRAP	Subroutine call to a specific address
SYSCALL	System call

2.5.3.9. Extended Operation Instructions

Extended operation instructions are defined for an add-on type extended operation unit. The instruction formats are predefined and the instruction map is also reserved. In addition to maintaining compatibility with the extended instructions that were implemented in the AM31 microcontroller core, the AM33 microcontroller core also supports dual multiplication/dual multiply-and-accumulate operations that multiply, in parallel, two 16-bit data values that are packed in word data. These functions accelerate audio data signal processing.

Table 38 Extended operation instructions

Instruction	Description
DMULH	Signed dual multiplication between registers, signed dual multiplication between an immediate value and a register
DMULHU	Unsigned dual multiplication between registers, unsigned dual multiplication between an immediate value and a register
DMACH	Signed dual sum-of-products operation between registers, signed dual sum-of-products operation between an immediate value and a register
DMACHU	Unsigned dual sum-of-products operation between registers, unsigned dual sum-of-products operation between an immediate value and a register
MAC	Signed sum-of-products operation between registers, signed sum-of-products operation between an immediate value and a register
MACU	Unsigned sum-of-products operation between registers, unsigned sum-of-products operation between an immediate value and a register
MACH	Signed half-word sum-of-products operation between registers, signed half-word sum-of-products operation between an immediate value and a register
MACHU	Unsigned half-word sum-of products operation between registers, unsigned half-word sum-of-products operation between an immediate value and a register
MACB	Signed byte sum-of-products operation between registers, signed byte sum-of-products operation between an immediate value and a register
MACBU	Unsigned byte sum-of-products operation between registers, unsigned byte sum-of-products operation between an immediate value and a register
SWHW	Data ordering swap (Swap half-word ordering within a word)
SWAP	Data ordering swap (Swap byte ordering within a word)
SWAPH	Data ordering swap (Swap byte ordering within a half-word)
SAT16	16-bit saturation processing

SAT24	24-bit saturation processing
MCSTE	Sum-of-products operation result saturation processing
BSCH	Bit search

2.5.3.10. LIW Extended Operation Instructions

The LIW extended operation instructions perform two operations in a single instruction. For details on each instruction, refer to the MN103E-series instruction specifications.

Table 39 LIW extended operation instructions

Instruction	Description
ADD_OP2	Parallel execution of addition between registers, and OP2 ADD_ADD, ADD_SUB, ADD_CMP, ADD_MOV, ADD_ASR, ADD_LSR, ADD_ASL
SUB_OP2	Parallel execution of subtraction between registers, and OP2 SUB_ADD, SUB_SUB, SUB_CMP, SUB_MOV, SUB_ASR, SUB_LSR, SUB_ASL
CMP_OP2	Parallel execution of comparison between registers, and OP2 CMP_ADD, CMP_SUB, CMP_MOV, CMP_ASR, CMP_LSR, CMP_ASL
MOV_OP2	Parallel execution of transfer between registers, and OP2 MOV_ADD, MOV_SUB, MOV_CMP, MOV_MOV, MOV_ASR, MOV_LSR, MOV_ASL, MOV_Lcc
AND_OP2	Parallel execution of AND operation between registers, and OP2 AND_ADD, AND_SUB, AND_CMP, AND_MOV, AND_ASR, AND_LSR, AND_ASL
OR_OP2	Parallel execution of OR operation between registers, and OP2 OR_ADD, OR_SUB, OR_CMP, OR_MOV, OR_ASR, OR_LSR, OR_ASL
XOR_OP2	Parallel execution of XOR operation between registers, and OP2 XOR_ADD, XOR_SUB, XOR_CMP, XOR_MOV, XOR_ASR, XOR_LSR, XOR_ASL
DMACH_OP2	Parallel execution of signed dual sum-of-products operation between registers, and OP2 DMACH_ADD, DMACH_SUB, DMACH_CMP, DMACH_MOV, DMACH_ASR, DMACH_LSR, DMACH_ASL
SAT16_OP2	Parallel execution of 16-bit saturation processing, and OP2 SAT16_ADD, SAT16_SUB, SAT16_CMP, SAT16_MOV, SAT16_ASR, SAT16_LSR, SAT16_ASL
SWHW_OP2	Parallel execution of swapping half-word ordering within a word, and OP2 SWHW_ADD, SWHW_SUB, SWHW_CMP, SWHW_MOV, SWHW_ASR, SWHW_LSR, SWHW_ASL

2.5.3.11. Floating-point Operation Instructions

Floating-point operation instructions are executed using the floating-point operation unit. For details on the floating-point unit, refer to the chapter on the floating-point operation unit. For details on each instruction, refer to the instruction specifications.

Table 40 Floating-point operation instructions

Instruction	Description
FMOV	Data transfer between floating-point registers, transfer of immediate value to a floating-point register Data transfer between a floating-point register and memory (load/store) Data transfer with post-increment between a floating-point register and memory (load/store) Data transfer between a floating-point register and a register
FABS	Absolute value operation between floating-point registers
FNEG	Sign inversion operation between floating-point registers

Instruction	Description
FRSQRT	Square root reciprocal operation between floating-point registers
FCMP	Compare operation between floating-point registers
FADD	Addition between floating-point registers, addition between an immediate value and a floating-point register
FSUB	Subtraction between floating-point registers, subtraction between an immediate value and a floating-point register
FMUL	Multiplication between floating-point registers, multiplication between an immediate value and a floating-point register
FDIV	Division between floating-point registers, division between an immediate value and a floating-point register
FMADD FNMADD FMSUB FNMSUB	Compound operation of multiplication and addition between floating-point registers Compound operation of multiplication and subtraction between floating-point registers
FBcc	Conditional branch on floating-point condition flag (branches to PC-relative address)
FLcc	Loop-dedicated conditional branch on floating-point condition flag (branches to start of loop set by SETLB)

2.5.3.12. Debug Instruction

The debug instruction is used and reserved by a debugger.

Table 41 Debug instruction

Instruction	Description
PI	This instruction is reserved by a debugger. Normally, an unimplemented instruction exception occurs when this instruction is executed.

2.6. Interrupt System

The AM33 microcontroller core uses asynchronous interrupts and synchronous interrupts as interrupt systems for interrupting the program sequence in response to the occurrence of certain events. These interrupts are defined below:

Asynchronous interrupts

These interrupts are generated in response to events that do not occur in synchronization with a specific instruction. The asynchronous interrupts in the AM33 microcontroller core are listed below.

- Reset interrupts
  - Reset pin interrupts (user reset interrupts)
- Nonmaskable interrupts
- Maskable interrupts
- Asynchronous bus errors

Synchronous interrupts

These interrupts (also called "exceptions") are generated in response to events that do occur in synchronization with a specific instruction. The synchronous interrupts in the AM33 microcontroller core are listed below.

- MMU exceptions
- System exceptions, excluding asynchronous bus errors

- System call instruction exceptions
- FPU exceptions

Asynchronous and synchronous interrupts are further categorized according to whether they are precise or imprecise and whether they permit program execution to resume.

Precise interrupts

Execution of all instructions prior to the instruction that was interrupted has been completed. The instruction that was interrupted is in either the "not yet executed," "partially executed," or "completed" state.

Execution (including operations that are "side effects") has not yet begun for any instruction that follows the instruction that was interrupted.

Imprecise interrupts

Execution of all instructions prior to the instruction that was interrupted may not be completed. The instruction that was interrupted is in either the "not yet executed," "partially executed," or "completed" state.

Execution (including operations that are "side effects") has not yet begun for any instruction that follows the instruction that was interrupted.

Execution can be resumed

Execution can be resumed after an asynchronous interrupt or a synchronous interrupt under the following conditions:

When the instruction that was interrupted has not been executed (prior to execution)

When the instruction that was interrupted has only been partially executed and any information that is needed in order to resume execution has been retained

When the instruction that was interrupted has only been partially executed and re-executing any operations that occur as side-effects of that instruction will not have an effect on the system

When execution of the instruction that was interrupted has been completed

When all information that is needed in order to resume execution of instructions that have not yet been executed but which are before the instruction that was interrupted has been retained (in the case of an imprecise interrupt)

Execution cannot be resumed (errors)

An asynchronous interrupt or a synchronous interrupt that does not permit program execution to resume is an "error," and the interrupted program cannot be re-executed. This applies in the following cases:

When the instruction that was interrupted has been partially executed, and the information that is needed in order to resume execution has not been retained

When the instruction that was interrupted has been partially executed, and re-executing any operations that occur as side-effects of that instruction will have an effect on the system

2.6.1. Overview of Interrupts

In the AM33 microcontroller core, the speed of interrupt processing and the flexibility of software control has been improved by limiting the resources, which are saved to memory when an interrupt occurs, to the eight bytes of the PC and the EPSW in order to minimize to the greatest extent possible the resources that are saved. In addition, fast response and optimal program placement can be made by placing the interrupt processing program at an address that varies for each interrupt level. In the AM33 microcontroller core, seven levels of maskable level interrupts and nonmaskable interrupts (NMI) can be established. Each interrupt can be grouped into these levels by the LSI interrupt controller.

In addition to the interrupt functions that are supported by the AM30/31/32 microcontroller core, the AM33 microcontroller core also expands interrupts concerning the MMU, interrupts concerning the FPU, and interrupts concerning the privileged level (supervisor level).

The addition of the Trap Base Register (TBR) makes it possible to change the base address for

interrupt vectors through the software to any desired address. There is the Trap Base Register for the supervisor level (TBR).
The MMU support includes support for synchronous interrupts that are generated by the MMU when memory is accessed. The FPU support includes support for synchronous interrupts that are generated when floating-point operation instructions are executed. The privileged level support includes support for synchronous interrupts such as privileged violations that concern the newly adapted supervisor level. In addition, by including a separate SP (Stack Pointer: sSP) for the supervisor level and converting some of the registers to a bank configuration, it is now possible to have separate registers for user level and supervisor level.

2.6.1.1. CPU Mode

Normal mode

In this mode, the CPU moves to supervisor level if a synchronous interrupt or an asynchronous interrupt is generated.
In normal mode, the privileged level can be in any of three states: supervisor level, user level, or monitor level.
The monitor level is a privileged level which is used by a debugger. This specification describes only about the supervisor level and user level.

Debug mode

This mode is used by a debugger. This specification describes only in the scope of normal mode.

2.6.1.2. Exception Operation Modes

There are two AM33/2.0 microcontroller core exception operation modes: AM33/2.0 mode and AM33/1.0 mode.
The exception operation mode is determined by the EXM (exception operation mode) bit of CPUP (CPU pipeline control register). When CPUP.EXM = 0, AM33/1.0 mode is in effect. When CPUP.EXM = 1, AM33/2.0 mode is in effect.

AM33/2.0 mode

This is the normal operation mode for the AM33/2.0 microcontroller core.

AM33/1.0 mode

This mode is compatible with the exception operation of the AM33/1.0 microcontroller core.

2.6.1.3. Privileged Level

There are two states defined for the privileged level. The memory space that can be accessed depends on the privileged level. Furthermore, because the internal resources that can be controlled also depend on the privileged level, the instructions that can be executed in each level differ. There are privileged instructions that can be executed only in a certain privileged level. For details on the privileged instructions, refer to the description of the instructions.

Supervisor level

Supervisor level permits access to all of memory spaces except for the monitor spaces (0xE0000000 to 0xFFFFFFFF).
Supervisor level permits access to all internal resources other than monitor resources, and all privileged instructions other than those that can be executed only in monitor level can be executed in supervisor level.

User level

User level permits use of areas in memory that have been allocated by the memory protection facility.
User level permits access only to user resources, and only normal instructions other than privileged instructions can be executed in user level.

2.6.2. Interrupt Types (AM33/1,0 Mode)

When CPUP.EXM = 0, the AM33 microcontroller core enters AM33/1.0 mode.
In AM33/1.0 mode, the AM33 microcontroller core supports reset interrupts, nonmaskable interrupts, maskable interrupts (level interrupts), MMU exceptions, system exceptions, SYSCALL instruction exceptions, monitor interrupts, and FPU exceptions.

2.6.2.1. Reset Interrupts

Reset interrupts have the highest priority. There are two types of reset interrupts: reset pin interrupts and debug reset interrupts.

2.6.2.1.1. Reset Pin Interrupts

Reset pin interrupts are generated when the CPU detects that the reset pin has been asserted. The interrupt type is "asynchronous/imprecise/error." The interrupt code is 0x000. The following interrupt processing sequence is executed by the hardware :

(1) The SP is switched to the sSP.
(2) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] <- 0 (initializes the maskable interrupt mask)
- EPSW.NMID <- 0 (enables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML <- 0 (non-monitor level)
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE <- 0 (disables the FPU)

(3) The interrupt code is set.
(4) Control shifts to address (TBA[31:0] + 0x000). The TBR is initialized to 0x40000000 by the reset, so the entry address of the reset pin interrupt is always 0x40000000.

2.6.2.2. Nonmaskable Interrupts

Nonmaskable interrupts are interrupts that are accepted regardless of the values of EPSW.IE and EPSW.IM, and include NMI pin interrupts and WDT overflow interrupts. If a nonmaskable interrupt occurs, control jumps to the NMI entry (TBA[31:0] + 0x008) regardless of the CPU mode. When EPSW.ML = 0, the CPU shifts to the supervisor level. Nonmaskable interrupts are masked by setting the EPSW.NMID bit to "1." When a nonmaskable interrupt is generated, the following interrupt processing sequence is executed by the hardware:

(1) The SP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID <- 1 (disables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + 0x008).

<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

2.6.2.2.1. WDT Overflow Interrupt

This interrupt is generated when a WDT (watchdog timer) overflow is detected. The interrupt type is “asynchronous/re-executable”. The interrupt code is 0x240. The address of the instruction where the WDT overflow was generated (detected) is stored as the return address.

2.6.2.2.2. NMI Pin Interrupts

An NMI pin interrupt is generated when the NMI pin is asserted. The interrupt type is "asynchronous/precise/re-executable." The interrupt code is 0x248. The address of the instruction where the NMI pin interrupt was generated (detected) is stored as the return address.

2.6.2.3. Maskable Interrupts

A maskable interrupt is requested if the level interrupt pin for the microcontroller core is asserted. Whether the maskable interrupt is generated or not depends on the level of the maskable interrupt that was requested and the status of the EPSW.IE and EPSW.IM[2:0] bits. The interrupt type is "asynchronous/precise/re-executable." When EPSW.IE = 0, maskable interrupts are masked. When EPSW.IE = 1, masking depends on the status of the level interrupt pin that was asserted and on the status of EPSW.IM[2:0]. There are seven levels for maskable interrupts, levels 0 through 6. Level 0 is the highest interrupt level, and level 6 is the lowest interrupt level. The interrupt level from the level interrupt pin and the mask level indicated by EPSW.IM[2:0] are compared to each other, and interrupts with a higher level are accepted.
The following table indicates whether a maskable interrupt is masked or not, based on the status of EPSW.IM[2:0] and the interrupt level from the level interrupt pins. The "O" mark indicates that a maskable interrupt is accepted.

Table 42 Mask of interrupt level and maskable interrupt

	Level interrupt pin						
EPSW.IM[2:0]	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
000	-	-	-	-	-	-	-
001	O	-	-	-	-	-	-
010	O	O	-	-	-	-	-
011	O	O	O	-	-	-	-
100	O	O	O	O	-	-	-
101	O	O	O	O	O	-	-
110	O	O	O	O	O	O	-
111	O	O	O	O	O	O	O

If a maskable interrupt is generated, the interrupt processing sequence shown below is executed by the hardware, regardless of the CPU mode.

(1) The SP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).

(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] <- interrupt level
- EPSW.NMID is left unchanged.
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + IVAR[level]).

<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

2.6.2.4. MMU Exceptions

These are synchronous interrupts that are generated by the memory protection functions. These interrupts are generated only when address translation is enabled by MMUCTR.ITE and MMUCTR.DTE. The following interrupt processing sequence is executed by the hardware.

(1) The SP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID <- 1 (disables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + interrupt code).

<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

<Programming Note>
Synchronous interrupts concerning an instruction fetch address are caused during the instruction pre-fetch cycle, but the interruption does not actually occur until the instruction that was supposed to be pre-fetched is executed.

<Programming Note>
If an instruction is positioned so that it crosses a page boundary, an MMU exception will be generated while the instruction is being processed. In this case, the return address that is

saved is the start address of that instruction. The virtual page number where the MMU exception was generated is recorded in MMU.iPTEU.VPN.

<Programming Note>

When executing the instruction that performs multiple instruction fetches (SETLB), the SETLB instruction, which itself is only a 1-byte instruction, is not executed until the four bytes that follow it are fetched. If an MMU exception is generated while the bytes following SETLB are being fetched, the return address that is saved is the address of the SETLB instruction.

<Programming Note>

If an MMU exception caused by a data access is generated while executing an instruction that performs multiple data accesses (MOVM, CALL, RET, etc.), the registers may have already been updated by data accesses and read accesses that were performed prior to the data access that generated the exception. (The SP, however, will not have been updated.) Therefore, caution is required when using these instructions that perform multiple data accesses with devices that produce side effects other than reading/writing the contents of memory/registers.

2.6.2.4.1. Instruction TLB Miss Exception

This interrupt is generated when MMUCTR.ITE = 1 and the page information for an instruction fetch logical address in the address translation spaces is not cached in the instruction TLB. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x100. ITMISS is set in sISR. The virtual page number where the synchronous interrupt was generated is recorded in MMU.IPTEU.VPN.

2.6.2.4.2. Data TLB Miss Exception

This interrupt is generated when MMUCTR.DTE = 1 and the page information for a data access logical address in the address translation spaces is not cached in the data TLB. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x108. DTMISS is set in sISR. The virtual page number where the synchronous interrupt was generated is recorded in MMU.DPTEU.VPN. The address of the instruction where the TLB miss was generated (detected) is stored as the return address.

2.6.2.4.3. Instruction Access Exception

This interrupt is generated when MMUCTR.ITE = 1 and the page protection information that is cached in the instruction TLB for an instruction fetch logical address in the address translation area conflicts with the execution mode. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x110. ITEX is set in sISR. The information on the protection conflict is recorded in MMUFCR.IFCR (Instruction Fault Cause Register). The virtual page number where the synchronous interrupt was generated is recorded in MMU.IPTEU.VPN.

2.6.2.4.4. Data Access Exception

This interrupt is generated when MMUCTR.DTE = 1 and the page protection information that is cached in the data TLB for a data fetch logical address in the address translation spaces conflicts with the execution mode. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x118. DTEX is set in sISR. The information on the protection conflict is recorded in MMUFCR.DFCR (Data Fault Cause Register). The virtual page number where the synchronous interrupt was generated is recorded in MMU.DPTEU.VPN. The address of the instruction where the data access exception was generated (detected) is stored as the return address.

2.6.2.5. System Exceptions

If a system exception is generated, the following operations are performed according to the CPU mode.

Normal mode

Control jumps to the NMI entry 0x008, and the CPU enters supervisor level. At this point, the following interrupt processing sequence is executed by the hardware.

(1) The SP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID <- 1 (disables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + 0x008).

<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

2.6.2.5.1. Privileged Instruction Execution Exception

This interrupt is generated when executing an instruction that is not permitted in a given execution mode. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x160. PRIV is set in sISR. The address of the instruction where the privileged instruction execution exception was generated (detected) is stored as the return address.

2.6.2.5.2. Unimplemented Instruction Exception

This interrupt is generated when executing an instruction that has not been implemented. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x168. UNIMP is set in sISR. The address of the instruction where the unimplemented instruction exception was generated (detected) is stored as the return address.

2.6.2.5.3. Extended operation unit exception/unimplemented extended instruction exception

This interrupt is generated when an exception has occurred in an extended operation instruction or when an unimplemented extended operation instruction is to be executed. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x170. EXUNIMP is set in sISR. The address of the instruction where the unimplemented extended instruction exception was generated (detected) is stored as the return address.

<Programming Note>
In the AM33, this interrupt is generated only when the extended operation instructions have not been implemented.

2.6.2.5.4. Illegal Memory Access Exceptions

Exceptions that are generated in regards to illegal memory accesses are generically called "illegal memory access exceptions." There are six types of illegal memory access exceptions:

(1) Illegal instruction access exceptions
(2) Illegal data access exceptions
(3) I/O space instruction access exceptions
(4) Privileged space instruction access exceptions
(5) Privileged space data access exceptions
(6) Data space instruction access exceptions

The following charts show the different combinations of TLB status (on/off) with instruction and data accesses, and the relationship of illegal memory access exceptions with each privileged level (USER: user level; SUP: supervisor level) and the upper four bits of each access address.

a) Instruction access/MMU On

Access address Upper four bits of access address	USER	SUP
0x0...	-	-
0x1...	-	-
0x2...	-	-
0x3...	-	-
0x4-7...	-	-
0x8-b...	(4)	-
0xC...	(4)	(3)
0xd...	(4)	(3)
0xe...	(1)	(1)
0xf...	(1)	(1)

b) Data access/Data MMU enable

Access address Upper four bits of access address	USER	SUP
0x0...	-	-
0x1...	-	-
0x2...	-	-
0x3...	-	-
0x4-7...	-	-
0x8-b...	(5)	-
0xC...	(5)	-
0xd...	(5)	-
0xe...	(2)	(2)
0xf...	(2)	(2)

c) Instruction access/Instruction MMU Off

Access address Upper four bits of access address	USER	SUP
0x0...	(6)	(6)
0x1...	(6)	(6)
0x2...	(3)	(3)
0x3...	(3)	(3)
0x4-b...	-	-
0xC...	(3)	(3)
0xd...	(3)	(3)

0xe...	(1)	(1)
0xf...	(1)	(1)

d)Data access/Data MMU Off

Access address Upper four bits of access address	USER	SUP
0x0...	-	-
0x1...	-	-
0x2...	(5)	-
0x3...	-	-
0x4-b...	-	-
0xC...	(5)	-
0xd...	-	-
0xe...	(2)	(2)
0xf...	(2)	(2)

2.6.2.5.4.1. Illegal Instruction Access Exceptions

This interrupt is generated in response to an instruction access to the monitor area in a non-monitor level. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x178. ILGIA is set in sISR. The address of the instruction where the illegal instruction access was generated (detected) is stored as the return address.

2.6.2.5.4.2. Illegal Data Access Exceptions

This interrupt is generated in response to a data access to the monitor area in a non-monitor level. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x178. ILGDA is set in sISR. The address of the instruction where the illegal data access was generated (detected) is stored as the return address. The address of the data where the illegal data access was generated (detected) is stored in DEAR.

2.6.2.5.4.3. I/O Space Instruction Access Exceptions

This interrupt is generated in response to an instruction access to the CPU's internal I/O space. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x178. IOIA is set in sISR. The address of the instruction where the I/O space instruction access was generated (detected) is stored as the return address.

2.6.2.5.4.4. Privileged Space Instruction Access Exceptions

This interrupt is generated in response to an instruction access to the privileged space in user level. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x178. PRVIA is set in sISR. The address of the instruction where the privileged space instruction access was generated (detected) is stored as the return address.

2.6.2.5.4.5. Privileged Space Data Access Exceptions

This interrupt is generated in response to a data access to the privileged space in user level. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x178. PRVDA is set in sISR. The address of the instruction where the privileged space data access was generated (detected) is stored as the return address. The address of the data where the privileged space data access was generated (detected) is stored in DEAR.

2.6.2.5.4.6. Data Space Instruction Access Exceptions

This interrupt is generated in response to an instruction access to the data space while data MMU is disabled. The interrupt type is "synchronous/precise/recoverable." The interrupt code is

0x178. DSIA is set in sISR. The address of the instruction where the data space instruction access was generated (detected) is stored as the return address.

2.6.2.5.5. Misalignment Exception

This interrupt is generated when the accessed address does not conform with the address boundary conditions assumed by the data access instruction. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x180. MISSA is set in sISR. The address of the instruction where the misalignment was generated (detected) is stored as the return address. The address of the data where the misalignment was generated (detected) is stored in DEAR.

2.6.2.5.6. Double Fault

A double fault is generated when a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence. In other words, one of the following synchronous interrupts was generated during a stack access that used the sSP:

- Misalignment (four-byte boundary) exception
- Illegal memory access exception
- MMU exception

The interrupt type is "synchronous/imprecise/error." The interrupt code is 0x200. DBLFLT is set in sISR.

<Programming Note>

If a synchronous interrupt is detected by referencing an instruction using an interrupt vector, a user reset interrupt is generated by the hardware.

<Programming Note>

If a double fault occurs during the hardware sequence for an interrupt that sets NMID, the sISR.NE bit is set to "1."

In the event of a double fault, the PC and EPSW are saved to the registers A0 and D0 , respectively, without using the SP. The address of the instruction where the double fault was generated (detected) is stored as the return address. However, if the double fault was generated as the result of a system call instruction exception, the address of the instruction that follows the system call instruction is stored as the return address.

2.6.2.5.7. Bus Errors

If an error occurs during a bus access, an interrupt is generated. The interrupt type is "asynchronous/imprecise/error." The interrupt code is 0x188. If the error was detected as an asynchronous error, the interrupt is masked by EPSW.ML and EPSW.NMID. If the error was detected as a synchronous bus error, BUSERR is set in sISR. If the error was detected as a synchronous bus error in a data access, the data address where the synchronous bus error was generated is stored in DEAR. If the error was detected as an asynchronous error, ABUSERR is set in NMICR.

<Programming Note>

When a bus error interrupt is generated during an instruction fetch that the CPU requested, the interrupt occurs during the instruction that was to have been fetched in response to the CPU's request. This bus error is a synchronous bus error.

<Programming Note>

When a bus error interrupt is generated during a data read that the CPU requested, or while writing the CPU's internal I/O space, the interrupt occurs during the instruction that executed the data access. This bus error is a synchronous bus error.

<Programming Note>

- *When a bus error interrupt is generated during an instruction fetch that the CPU cancelled*
- *When a bus error interrupt is generated during a cache refill for a portion that the CPU did not request*
- *When a bus error interrupt is generated during a data access*

- *When a bus error interrupt is generated during an access by a bus master other than the CPU*

Once the bus error interrupt is generated, register updates are halted until the interrupt processing begins. If a data access was being performed at the moment that the interrupt was generated (regardless of whether the data access was the cause of the bus error or not), the data access is aborted. In this case, the registers are not updated due to the data access instruction.

2.6.2.6. System Call Instruction Exceptions

A system call instruction exception is generated when the SYSCALL instruction is executed. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x300 + imm4 x 8, where imm4 is four-bit immediate data from the immediate value field in the SYSCALL instruction. SYSC is set in sISR. The interrupt processing sequence shown below is executed by the hardware, regardless of the CPU mode.

(1) The SP is switched to the sSP.
(2) The PC for the instruction that follows the SYSCALL instruction (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE is left unchanged.
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID is left unchanged.
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + 0x300 + imm4 x 8).

<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

<Programming Note>
When a system call is performed using the SYSCALL instruction, in the case that the stack is being used to pass parameters, it is necessary to access the parameters through the user stack with using the user stack pointer (uSP).

<Programming Note>
Because EPSW.NMID is not set and EPSW.IE is not cleared for a system call instruction exception, there is a possibility that multiple interrupts could occur. Accordingly, in the system call instruction exception handler, it is possible that the TBR could be damaged as a result of the multiple interrupts. Therefore, programs must be written without the use of the resources that could be damaged as a result of multiple interrupts in the system call instruction exception handler.

2.6.2.7. FPU Exceptions

FPU exceptions are synchronous interrupts that are generated in conjunction with the execution of floating-point operation instructions. FPU unimplemented instruction exceptions and FPU operation exceptions are generated only when the FPU is enabled according to the EPSW.FE bit. If a floating-point operation instruction is executed while the FPU is disabled according to the

EPSW.FE bit, an "FPU disabled" exception is generated. The following interrupt processing sequence is executed by the hardware.

Normal mode

(1) The SP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID <- 1 (disables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + interrupt code).

<Programming Note>

If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

2.6.2.8. "FPU Disabled" Exception

This exception is generated if a floating-point operation instruction is to be executed while the EPSW.FE bit is "0." The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1C0. FPUD is set in sISR.

2.6.2.9. FPU Unimplemented Instruction Exception

This exception is generated if an unimplemented floating-point operation instruction is to be executed while the EPSW.FE bit is "1." The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1C8. FPUUI is set in sISR.

2.6.2.10. FPU Operation Exception

If any interrupt cause of the followings: imprecise, underflow, overflow, division by zero, invalid operation, is generated while the EPSW.FE bit is "1," an FPU operation exception is generated. Each cause of an FPU operation exception can be individually enabled by setting the corresponding bit in the FPCR.FE field. For details, refer to Page 183, 2.9.4.3 "FPU Operation Exceptions." The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1D0. FPUOP is set in sISR.

2.6.3. List of Interrupt Codes and Interrupt Vectors (AM33/1.0 Mode)

The following table shows the followings: the interrupt codes for each interrupt source in AM33/1.0 mode, the transition levels and interrupt vectors for each CPU mode.
The control shifts to the address (TBA + interrupt vector) when the interrupt occurs. Basically, the interrupt code is written in the TBR interrupt code field.

Table 43 Transition level and interrupt vector list (AM33/1.0 mode)

Interrupt type		Interrupt code	CPU mode	
			Normal mode	
			Transition level	Interrupt vector
Reset interrupt	Reset pin interrupt	0x000	SL	0x000
MMU exceptions	Instruction TLB miss exception	0x100	SL	0x100
	Data TLB miss exception	0x108	SL	0x108
	Instruction access exception	0x110	SL	0x110
	Data access exception	0x118	SL	0x118
System exceptions	Privileged instruction execution exception	0x160	SL	0x008
	Unimplemented instruction exception	0x168	SL	0x008
	Unimplemented extended instruction exception	0x170	SL	0x008
	Misalignment exception	0x180	SL	0x008
	Illegal instruction access exception	0x178	SL	0x008
	Illegal data access exception	0x178	SL	0x008
	I/O space instruction access exception	0x178	SL	0x008
	Privileged space instruction access exception	0x178	SL	0x008
	Privileged space data access exception	0x178	SL	0x008
	Data space instruction access exception	0x178	SL	0x008
System exception	Bus error	0x188	SL	0x008
FPU exceptions	"FPU disabled" exception	0x1C0	SL	0x1c0
	"FPU unimplemented" exception	0x1C8	SL	0x1c8
	FPU operation exception	0x1D0	SL	0x1d0
System exception	Double fault	0x200	SL	0x008
Nonmaskable interrupts	WDT overflow	0x240	SL	0x008
	NMI pin interrupt	0x248	SL	0x008
Maskable interrupts	Level 0	0x280	SL	ivar0
	Level 1	0x288	SL	ivar1
	Level 2	0x290	SL	ivar2
	Level 3	0x298	SL	ivar3
	Level 4	0x2A0	SL	ivar4
	Level 5	0x2A8	SL	ivar5
	Level 6	0x2B0	SL	ivar6
System call instruction exception		0x300 +imm4	SL	0x300 +imm4

The table below shows the status of the EPSW corresponding to each exception cause for each CPU mode. (A “-“ indicates “no change”.)

Table 44 States of the bits associated with EPSW

CPU mode	Normal mode							
EPSW	IE	IM	NMID	nSL	ML	T	nAR	FE
Reset pin interrupt	0	0	0	0	0	0	0	0
Nonmaskable interrupt	0	-	1	0	-	0	0	-
Maskable interrupt	0	IM	-	0	-	0	0	-
MMU exception	0	-	1	0	-	0	0	-
System exception	0	-	1	0	-	0	0	-
SYSCALL instruction exception	-	-	-	0	-	0	0	-
FPU exception	0	-	1	0	-	0	0	-

2.6.4. Interrupt Types (AM33/2.0 Mode)

When CPUP.EXM = 1, the AM33 microcontroller core enters AM33/2.0 mode.
In AM33/2.0 mode, the AM33 microcontroller core supports reset interrupts, nonmaskable interrupts, maskable interrupts (level interrupts), MMU exceptions, system exceptions, SYSCALL instruction exceptions, and FPU exceptions.

2.6.4.1. Reset Interrupts

Reset interrupts have the highest priority and it includes reset pin interrupts.

2.6.4.1.1. Reset Pin Interrupts

Reset pin interrupts are generated when the CPU detects that the reset pin has been asserted. The interrupt type is "asynchronous/imprecise/error." The interrupt code is 0x000. The following interrupt processing sequence is executed by the hardware:

(1) The SP is switched to the sSP.
(2) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] <- 0 (initializes the maskable interrupt mask)
- EPSW.NMID <- 0 (enables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML <- 0 (non-monitor level)
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE <- 0 (disables the FPU)

(3) The interrupt code is set.
(4) Control shifts to address (TBA[31:0] + 0x000). Because TBR is initialized to 0x40000000 through the reset, the entry address for reset pin interrupts in normal mode is always 0x40000000.

2.6.4.2. Nonmaskable Interrupts

Nonmaskable interrupts are accepted regardless of the values of EPSW.IE and EPSW.IM. There are two types of nonmaskable interrupts: external pin nonmaskable interrupts and watchdog timer overflow interrupts. If a nonmaskable interrupt occurs, control jumps to the NMI entry (TBA[31:0] + 0x240 or 0x248) regardless of the CPU mode. When EPSW.ML = 0, the CPU shifts to the supervisor level. Nonmaskable interrupts are masked by setting the EPSW.NMID bit to "1." When a nonmaskable interrupt is generated, the following interrupt processing sequence is executed by the hardware:

(1) TheSP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID <- 1 (disables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.

(7) Control shifts to address (TBA[31:0] + interrupt code).
<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

2.6.4.2.1. WDT Overflow Interrupt

This interrupt is generated when a WDT (Watchdog Timer) overflow is detected. The interrupt type is "asynchronous/precise/recoverable." The interrupt code is 0x240. The address of the instruction that was being executed when the WDT overflow was generated (detected) is saved as the return address.

2.6.4.2.2. NMI Pin Interrupts

An NMI pin interrupt is generated when the NMI pin is asserted. The interrupt type is "asynchronous/precise/recoverable." The interrupt code is 0x248. The address of the instruction that was being executed when the NMI pin interrupt was generated (detected) is saved as the return address.

2.6.4.3. Maskable Interrupts

A maskable interrupt is requested if the level interrupt pin for the microcontroller core is asserted. Whether the maskable interrupt is generated or not depends on the level of the maskable interrupt that was requested and the status of the EPSW.IE and EPSW.IM[2:0] bits. The interrupt type is "asynchronous/precise/re-executable." When EPSW.IE = 0, maskable interrupts are masked. When EPSW.IE = 1, masking depends on the status of the level interrupt pin that was asserted and on the status of EPSW.IM[2:0]. There are seven levels for maskable interrupts, levels 0 through 6. Level 0 is the highest interrupt level, and level 6 is the lowest interrupt level. The interrupt level from the level interrupt pin and the mask level indicated by EPSW.IM[2:0] are compared to each other, and interrupts with a higher level are accepted.
The following table indicates whether a maskable interrupt is masked or not, based on the status of EPSW.IM[2:0] and the interrupt level from the level interrupt pins. The "O" mark indicates that a maskable interrupt is accepted.

Table 45 Interrupt level and maskable interrupt mask

	Level Interrupt pin						
EPSW.IM[2:0]	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
000	-	-	-	-	-	-	-
001	O	-	-	-	-	-	-
010	O	O	-	-	-	-	-
011	O	O	O	-	-	-	-
100	O	O	O	O	-	-	-
101	O	O	O	O	O	-	-
110	O	O	O	O	O	O	-
111	O	O	O	O	O	O	O

If a maskable interrupt is generated, the interrupt processing sequence shown below is executed by the hardware, regardless of the CPU mode.

(1) TheSP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)

- EPSW.IM[2:0] <- interrupt level
- EPSW.NMID is left unchanged.
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control transfers to address (TBA[31:0] + IVAR[level]).

<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

2.6.4.4. MMU Exceptions

These are synchronous interrupts that are generated by the memory protection functions. These interrupts are generated only when address translation is enabled by MMUCTR.ITE and MMUCTR.DTE. The following interrupt processing sequence is executed by the hardware.

(1) The SP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID <- 1 (disables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + interrupt code).

<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

<Programming Note>
Synchronous interrupts concerning an instruction fetch address are caused during the instruction pre-fetch cycle, but the interruption does not actually occur until the instruction that was supposed to be pre-fetched is executed.

<Programming Note>
If an instruction is positioned so that it crosses a page boundary, an MMU exception will be generated while the instruction is being processed. In this case, the return address that is saved is the starting address of that instruction. The virtual page number where the MMU exception was generated is recorded in MMU.iPTEU.VPN.

<Programming Note>
When executing the instruction that performs multiple instruction fetches (SETLB), the SETLB instruction is not executed until the four bytes that follow it are fetched although it is only a 1-byte instruction. If an MMU exception is generated while the bytes that follow SETLB are being fetched, the return address that is saved is the address of the SETLB instruction.

<Programming Note>
If an MMU exception caused by a data access is generated while executing an instruction that performs multiple data accesses (MOVM, CALL, RET, etc.), the registers may have already been updated by data accesses and read accesses that were performed prior to the data access which generated the exception. (The SP, however, will not have been updated.) Therefore, caution is necessary for using these instructions that perform multiple data accesses with devices that produce side effects other than reading/writing the contents of memory/registers.

2.6.4.4.1. Instruction TLB Miss Exception

This interrupt is generated when MMUCTR.ITE = 1 and the page information for an instruction fetch logical address in the address translation spaces is not cached in the instruction TLB. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x100. ITMISS is set in sISR. The virtual page number where the synchronous interrupt was generated is recorded in MMU.IPTEU.VPN.

2.6.4.4.2. Data TLB Miss Exception

This interrupt is generated when MMUCTR.DTE = 1 and the page information for a data access logical address in the address translation spaces is not cached in the data TLB. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x108. DTMISS is set in sISR. The virtual page number where the synchronous interrupt was generated is recorded in MMU.DPTEU.VPN. The address of the instruction where the data TLB miss was generated (detected) is stored as the return address.

2.6.4.4.3. Instruction Access Exception

This interrupt is generated when MMUCTR.ITE = 1 and the page protection information that is cached in the instruction TLB for an instruction fetch logical address in the address translation spaces conflicts with the execution mode. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x110. ITEX is set in sISR. The information on the protection conflict is recorded in MMUFCR.IFCR (Instruction Fault Cause Register). The virtual page number where the synchronous interrupt was generated is recorded in MMU.IPTEU.VPN.

2.6.4.4.4. Data Access Exception

This interrupt is generated when MMUCTR.DTE = 1 and the page protection information that is cached in the data TLB for a data fetch logical address in the address translation spaces conflicts with the execution mode. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x118. DTEX is set in sISR. The information on the protection conflict is recorded in MMUFCR.DFCR (Data Fault Cause Register). The virtual page number where the synchronous interrupt was generated is recorded in MMU.DPTEU.VPN. The address of the instruction where the data access exception was generated (detected) is stored as the return address.

2.6.4.5. System Exceptions

If a system exception is generated, the following operations are performed.

Normal mode

Control jumps to the interrupt entry that corresponds to the interrupt cause (TBA[31:0] + interrupt code), and the CPU enters supervisor level. At this point, the following interrupt processing sequence is executed by the hardware.

(1) The SP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).

(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID <- 1 (disables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + interrupt code).

<Programming Note>

If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

2.6.4.5.1. Privileged Instruction Execution Exception

This interrupt is generated when executing an instruction that is not permitted in a given execution mode. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x160. PRIV is set in sISR. The address of the instruction where the privileged instruction execution exception was generated (detected) is stored as the return address.

2.6.4.5.2. Unimplemented Instruction Exception

This interrupt is generated if executing an instruction that has not been implemented. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x168. UNIMP is set in sISR. The address of the instruction where the unimplemented instruction exception was generated (detected) is stored as the return address.

2.6.4.5.3. Extended operation unit exception/unimplemented extended instruction exception

This interrupt is generated when an exception has occurred in an extended operation instruction or when an unimplemented extended operation instruction has been to be executed. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x170. EXUNIMP is set in sISR. The address of the instruction where the unimplemented extended instruction exception was generated (detected) is stored as the return address.

<Programming Note>

In the AM33, this interrupt is generated only when the extended operation instructions have not been implemented.

2.6.4.5.4. Illegal Memory Access Exceptions

Exceptions that are generated in regards to illegal memory accesses are generically called "illegal memory access exceptions." There are six types of illegal memory access exceptions:

(1) Illegal instruction access exceptions
(2) Illegal data access exceptions
(3) I/O space instruction access exceptions
(4) Privileged space instruction access exceptions
(5) Privileged space data access exceptions
(6) Data space instruction access exceptions

The following charts show the different combinations of MMU status (enable/disable) with instruction and data accesses, and the relationship of illegal memory access exceptions with each privileged level (USER: user level; and SUP: supervisor level) and the upper four bits of each access address.

a) Instruction access/MMU enable

Access address

Upper four bits of access address	USER	SUP
0x0...	-	-
0x1...	-	-
0x2...	-	-
0x3...	-	-
0x4-7...	-	-
0x8-b...	(4)	-
0xC...	(4)	(3)
0xd...	(4)	(3)
0xe...	(1)	(1)
0xf...	(1)	(1)

b) Data access/Data MMU enable

Access address

Upper four bits of access address	USER	SUP
0x0...	-	-
0x1...	-	-
0x2...	-	-
0x3...	-	-
0x4-7...	-	-
0x8-b...	(5)	-
0xC...	(5)	-
0xd...	(5)	-
0xe...	(2)	(2)
0xf...	(2)	(2)

c) Instruction access/Instruction MMU disable

Access address

Upper four bits of access address	USER	SUP
0x0...	(6)	(6)
0x1...	(6)	(6)
0x2...	(3)	(3)
0x3...	(3)	(3)
0x4-b...	-	-
0xC...	(3)	(3)
0xd...	(3)	(3)
0xe...	(1)	(1)
0xf...	(1)	(1)

d) Data access/Data MMU disable

Access address

Upper four bits of access address	USER	SUP
0x0...	-	-
0x1...	-	-
0x2...	(5)	-
0x3...	-	-

0x4-b...	-	-
0xC...	(5)	-
0xd...	-	-
0xe...	(2)	(2)
0xf...	(2)	(2)

2.6.4.5.4.1. Illegal Instruction Access Exceptions

This interrupt is generated in response to an instruction access to the monitor spaces in a non-monitor level. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x190. ILGIA is set in sISR. The address of the instruction where the illegal instruction access was generated (detected) is stored as the return address.

2.6.4.5.4.2. Illegal Data Access Exceptions

This interrupt is generated in response to a data access to the monitor spaces in a non-monitor level. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x198. ILGDA is set in sISR. The address of the instruction where the illegal data access was generated (detected) is stored as the return address. The address of the data where the illegal data access was generated (detected) is stored in DEAR.

2.6.4.5.4.3. I/O Space Instruction Access Exceptions

This interrupt is generated in response to an instruction access to the CPU's internal I/O space. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1A0. IOIA is set in sISR. The address of the instruction where the I/O space instruction access was generated (detected) is stored as the return address.

2.6.4.5.4.4. Privileged Space Instruction Access Exceptions

This interrupt is generated in response to an instruction access to the privileged space in user level. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1A8. PRVIA is set in sISR. The address of the instruction where the privileged space instruction access was generated (detected) is stored as the return address.

2.6.4.5.4.5. Privileged Space Data Access Exceptions

This interrupt is generated in response to a data access to the privileged space in user level. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1B0. PRVDA is set in sISR. The address of the instruction where the privileged space data access was generated (detected) is stored as the return address. The address of the data where the privileged space data access was generated (detected) is stored in DEAR.

2.6.4.5.4.6. Data Space Instruction Access Exceptions

This interrupt is generated in response to an instruction access to the data space while MMU is disabled. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x1B8. DSIA is set in sISR. The address of the instruction where the data space instruction access was generated (detected) is stored as the return address.

2.6.4.5.5. Misalignment Exception

This interrupt is generated when the accessed address does not conform with the address boundary conditions assumed by the data access instruction. The interrupt type is "synchronous/precise/re-executable." The interrupt code is 0x180. MISSA is set in sISR. The address of the instruction where the misalignment was generated (detected) is stored as the return address. The address of the data where the misalignment was generated (detected) is stored in DEAR.

2.6.4.5.6. Double Fault

A double fault is generated when a nonrecoverable synchronous interrupt is generated during the

interrupt hardware sequence. In other words, one of the following synchronous interrupts was generated during a stack access using the sSP:

- Misalignment (four-byte boundary) exception
- Illegal memory access exception
- MMU exception

The interrupt type is "synchronous/imprecise/error." The interrupt code is 0x200. DBLFLT is set in sISR.

<Programming Note>
If a synchronous interrupt is detected by referencing an instruction using an interrupt vector, a user reset interrupt is generated by the hardware.

<Programming Note>
If a double fault occurs during the hardware sequence for an interrupt that sets NMID, the sISR.NE bit is set to "1."

In the event of a double fault, the PC and EPSW are saved to the registers A0 and D0 , respectively, without using the SP. The address of the instruction where the double fault was generated (detected) is stored as the return address. However, if the double fault was generated as the result of a system call instruction exception, the address of the instruction that follows the system call instruction is stored as the return address.

2.6.4.5.7. Bus Errors

If an error occurs during a bus access, an interrupt is generated. The interrupt type is "asynchronous/imprecise/error." The interrupt code is 0x188. If the error was detected as an asynchronous error, the interrupt is masked by EPSW.ML and EPSW.NMID. If the error was detected as a synchronous bus error, BUSERR is set in sISR. If the error was detected as a synchronous bus error in a data access, the data address where the synchronous bus error was generated is stored in DEAR. If the error was detected as an asynchronous error, ABUSERR is set in NMICR.

<Programming Note>
When a bus error interrupt is generated during an instruction fetch that the CPU requested The interrupt occurs during the instruction that was to have been fetched in response to the CPU's request. This bus error is a synchronous bus error.

<Programming Note>
When a bus error interrupt is generated during a data read that the CPU requested, or while writing the CPU's internal I/O space, the interrupt occurs during the instruction that executed the data access. This bus error is a synchronous bus error.

<Programming Note>

- *When a bus error interrupt is generated during an instruction fetch that the CPU cancelled*
- *When a bus error interrupt is generated during a cache refill for a portion that the CPU did not request*
- *When a bus error interrupt is generated during a data access*
- *When a bus error interrupt is generated during an access by a bus master other than the CPU*

Once the bus error interrupt is generated, register updates are halted until the interrupt processing begins. If a data access was being performed at the moment that the interrupt was generated (regardless of whether the data access was the cause of the bus error or not), the data access is aborted. In this case, the registers are not updated due to the data access instruction.

2.6.4.6. System Call Instruction Exceptions

A system call instruction exception is generated when the SYSCALL instruction is executed. The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x300 + imm4 x 8, where imm4 is four-bit immediate data from the immediate value field in the SYSCALL instruction.

SYSC is set in sISR. The interrupt processing sequence shown below is executed by the hardware.

(1) The SP is switched to the sSP.
(2) The PC for the instruction that follows the SYSCALL instruction (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE IS LEFT UNCHANGED.
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID is left unchanged.
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + 0x300 + imm4 x 8).

<Programming Note>
If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

<Programming Note>
When a system call is performed using the SYSCALL instruction, in the case that the stack is being used to pass parameters, it is necessary to access the parameters through the user stack with the use of the user stack pointer (uSP).

<Programming Note>
Because EPSW.NMID is not set and EPSW.IE is not cleared for a system call instruction exception, there is a possibility that multiple interrupts could occur. Accordingly, in the system call instruction exception handler, it is possible that the TBR could be damaged as a result of the multiple interrupts. Therefore, programs must be written without the use of the resources that could be damaged as a result of multiple interrupts in the system call instruction exception handler.

<Programming Note>
If single-step is detected by SYSCALL (if the SYSCALL instruction was executed with EPSW.T = 1), a single-step interrupt is generated in the instruction that is executed after the SYSCALL instruction (in other words, the first instruction in the system call instruction exception handler). In normal mode, PSW.T is cleared by the system call instruction exception, so when control returns from the single-step handler without any additional changes, the system call instruction exception handler is normally executed, and when control returns from the system call instruction exception handler, step execution is resumed.

2.6.4.7. FPU Exceptions

FPU exceptions are synchronous interrupts that are generated in conjunction with the execution of floating-point operation instructions. FPU unimplemented instruction exceptions and FPU operation exceptions are generated only when the FPU is enabled according to the EPSW.FE bit. If a floating-point operation instruction is executed while the FPU is disabled according to the EPSW.FE bit, an "FPU disabled" exception is generated. The following interrupt processing sequence is executed by the hardware.

Normal mode

(1) The SP is switched to the sSP.
(2) The PC (the return address) is saved to the stack (SP-4).
(3) EPSW is saved to the stack (SP-8).
(4) The contents of EPSW are updated:
- EPSW.IE <- 0 (disables maskable interrupts)
- EPSW.IM[2:0] is left unchanged.
- EPSW.NMID <- 1 (disables nonmaskable interrupts)
- EPSW.nSL <- 0 (supervisor level)
- EPSW.ML is left unchanged.
- EPSW.T <- 0 (disables single-step operation)
- EPSW.nAR <- 0 (uses the normal bank registers)
- EPSW.FE is left unchanged.

(5) The contents of the stack pointer SP are updated (SP <- SP-8)
(6) The interrupt code is set.
(7) Control shifts to address (TBA[31:0] + interrupt code).

<Programming Note>

If a nonrecoverable synchronous interrupt is generated during the interrupt hardware sequence, a double fault occurs. This condition is reset when a new synchronous interrupt is generated by referencing the interrupt vector.

2.6.4.8. "FPU Disabled" Exception

This exception is generated if a floating-point operation instruction is to be executed while the EPSW.FE bit is "0." The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1C0. FPUD is set in sISR.

2.6.4.9. FPU Unimplemented Instruction Exception

This exception is generated if an unimplemented floating-point operation instruction is to be executed while the EPSW.FE bit is "1." The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1C8. FPUUI is set in sISR.

2.6.4.10. FPU Operation Exception

If any of the following interrupt causes: imprecise, underflow, overflow, division by zero, or invalid operation, is generated while the EPSW.FE bit is "1," an FPU operation exception is generated. Each cause of an FPU operation exception can be individually enabled by setting the corresponding bit in the FPCR.FE field. For details, refer to page 183, 2.9.4.3, "FPU Operation Exceptions." The interrupt type is "synchronous/precise/recoverable." The interrupt code is 0x1D0. FPUOP is set in sISR.

2.6.5. List of Interrupt Codes and Interrupt Vectors (AM33/2.0 Mode)

The following table lists the interrupt codes for each interrupt source in AM33/2.0 mode, and the transition levels and interrupt vectors for each CPU mode.
The control shifts to the address (TBA + interrupt vector). Basically, the interrupt code is written in the TBR interrupt code field. It is written in the mTBR interrupt code field when the transition level is ML.

Table 46 Transition level and interrupt vector list (AM33/2.0 mode)

Interrupt type		Interrupt code	CPU mode	
			Normal mode	
			Transition level	Interrupt vector
Reset interrupt	Reset pin interrupt	0x000	SL	0x000
MMU exceptions	Instruction TLB miss exception	0x100	SL	0x100
	Data TLB miss exception	0x108	SL	0x108
	Instruction access exception	0x110	SL	0x110
	Data access exception	0x118	SL	0x118
System exceptions	Privileged instruction execution exception	0x160	SL	0x160
	Unimplemented instruction exception	0x168	SL	0x168
	Unimplemented extended instruction exception	0x170	SL	0x170
	Misalignment exception	0x180	SL	0x180
	Illegal instruction access exception	0x190	SL	0x190
	Illegal data access exception	0x198	SL	0x198
	I/O space instruction access exception	0x1A0	SL	0x1A0
	Privileged space instruction access exception	0x1A8	SL	0x1A8
		0x1B0	SL	0x1B0
	Privileged space data access exception	0x1B8	SL	0x1B8
	Data space instruction access exception			
System exception	Bus error	0x188	SL	0x188
FPU exceptions	"FPU disabled" exception	0x1C0	SL	0x1C0
	"FPU unimplemented" exception	0x1C8	SL	0x1C8
	FPU operation exception	0x1D0	SL	0x1D0
System exception	Double fault	0x200	SL	0x200
Nonmaskable interrupts	WDT overflow	0x240	SL	0x240
	NMI pin interrupt	0x248	SL	0x248
Maskable interrupts	Level 0	0x280	SL	Ivar0
	Level 1	0x288	SL	Ivar1
	Level 2	0x290	SL	Ivar2
	Level 3	0x298	SL	Ivar3
	Level 4	0x2A0	SL	Ivar4
	Level 5	0x2A8	SL	Ivar5
	Level 6	0x2B0	SL	Ivar6
System call instruction exception		0x300 +imm4	SL	0x300 +imm4

The table below shows the status of the EPSW bits corresponding to each exception cause for each CPU mode.

Table 47 States of the bits associated with EPSW

(A "-" indicates "no change.")

CPU mode	Normal mode							
EPSW	IE	IM	NMID	nSL	ML	T	nAR	FE
Reset pin interrupt	0	0	0	0	0	0	0	0
Nonmaskable interrupt	0	-	1	0	-	0	0	-
Maskable interrupt	0	IM	-	0	-	0	0	-
MMU exception	0	-	1	0	-	0	0	-
System exception	0	-	1	0	-	0	0	-
SYSCALL instruction exception	-	-	-	0	-	0	0	-
FPU exception	0	-	1	0	-	0	0	-

2.6.6. Returning from an Interrupt

The procedure for returning from an asynchronous interrupt/synchronous interrupt is described below.

Supervisor level

When an asynchronous interrupt/synchronous interrupt is generated or an asynchronous interrupt/synchronous interrupt in normal mode is generated, the asynchronous interrupt/synchronous interrupt processing is executed in supervisor level. The return from supervisor level is accomplished by executing the RTI (Return from Interrupt) instruction. In supervisor level, the RTI (Return from Interrupt) instruction performs the following processing:

(1) Returns the contents of EPSW from the stack (sSP).
(2) When EPSW.nSL = 0, the SP switches to sSP. (SP <- sSP)
(3) When EPSW.nSL = 1, the SP switches to uSP. (SP <- uSP)
(4) Returns the PC (return address) from the stack (sSP + 4).
(5) Updates the contents of the stack pointer. (sSP + 8 -> sSP)

2.6.7. Priority Ranking

The interrupt priority ranking is as shown below. Exceptions with ranking 1 have the highest priority. If two interrupts have the same priority ranking, the one that is listed upper has higher priority.

Table 48 Interrupt priority ranking

Ranking	Interrupt type		Detection stage
1	Reset interrupts	1. Reset pin interrupt	Async
2	System exception	1. Double fault	E, M
3	System exceptions	1. Synchronous instruction bus error (bus error)	D
		2. Asynchronous bus error (buss error)	Async (D)
4	Nonmaskable interrupts	1. WDT overflow	Async (D)
		2. NMI pin interrupt	Async (D)
5	Maskable interrupts		Async (D)
6	MMU exceptions	1. Instruction TLB miss exception	D
		2. Instruction access exception	D
7	System exceptions	1. Privileged instruction execution exception	D
		2. Unimplemented instruction exception	D
		3. Extended operation unit exception/ Unimplemented extended instruction exception	D
		4. Illegal instruction access exception	D
		5. I/O space instruction access exception	D
		6. Privileged space instruction access exception	D
		7. Data space instruction access exception	D
8	System call instruction exceptions		D
9	FPU exception	1. "FPU disabled" exception	D
10	System exceptions	1. Misalignment exception	E
		2. Illegal data access exception	M
		3. Privileged space data access exception	M
11	System exception	1. Synchronous data bus error (bus error)	M
12	MMU exceptions	1. Data TLB miss exception	M
		2. Data access exception	M
13	FPU exceptions	1. "FPU unimplemented" instruction exception	M
		2. FPU operation exception	M

2.7. Memory Management

The AM33 microcontroller core has two independent built-in Memory Management Units (MMUs), one for instructions and one for data. By using the built-in address translation buffer (TLB: Translation Lookaside Buffer) and caching the data from the address translation table that was created in external memory, logical addresses can be quickly converted into physical addresses. The paging method is used for address translation, and page sizes of 1KB, 4KB, 128KB, and 4MB are supported. Memory can be protected by setting access authority to logical spaces in units of individual pages, according to the privileged level/user level. Note that the instruction and data caches are both physical address caches.

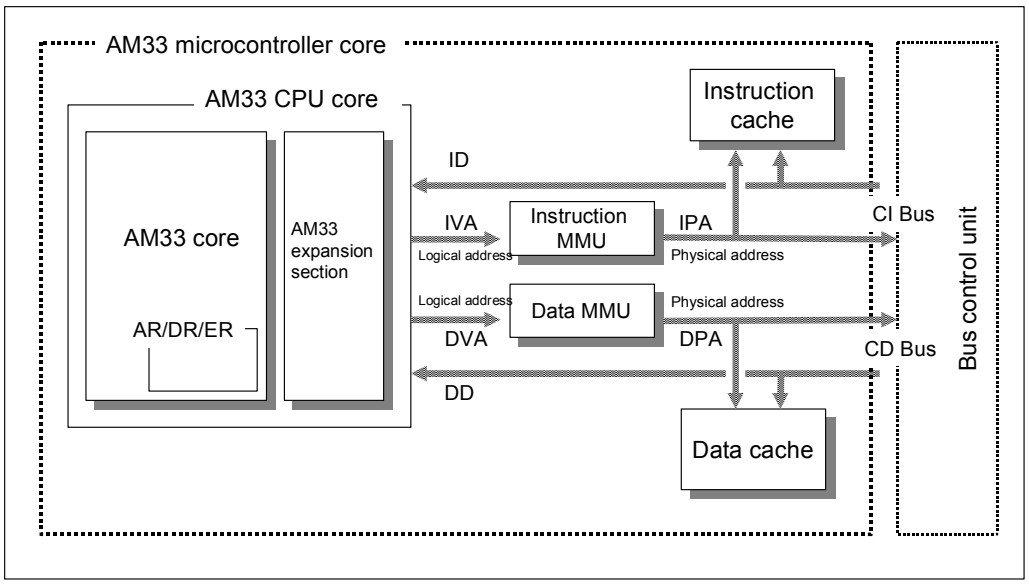


Figure 10 Block diagram

2.7.1. Address Space

Address space when using an MMU (logical address space)

The AM33 microcontroller core supports a 32-bit logical address space, and can access a 4GB logical address space. The logical address space is divided into six spaces: SU0, SU1, S2, S3, S4, and M0.

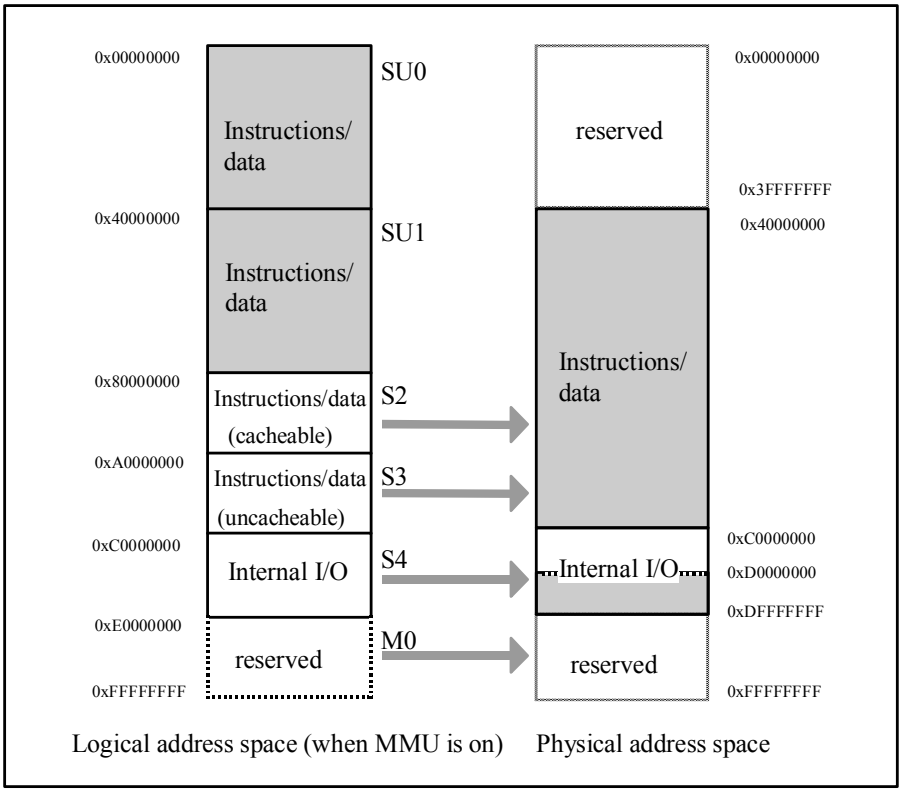


Figure 11 Address space when using an MMU

The SU0 space (0x00000000 to 0x3FFFFFFF: 1GB) and the SU1 space (0x40000000 to 0x7FFFFFFF: 1GB) can both be used at the user level and the supervisor level, and are mapped onto the physical address space in page units according to the content of the address translation table. Caching control for the SU0 space and the SU1 space is controlled as follows, according to the value of the MMUCTR.CE (Cacheable bit Enable) bit:

(1) When the MMUCTR.CE bit is set to "0"
SU0 space: Cacheable
SU1 space: Uncacheable

(2) When the MMUCTR.CE bit is set to "1"
SU0 space: Whether each page is cacheable or uncacheable can be individually controlled through the status of the PTE.C bit.
SU1 space: Whether each page is cacheable or uncacheable can be individually controlled through the status of the PTE.C bit.

The physical address can be mapped to SU0 and SU1 spaces are 0x40000000～0xBFFFFFFF in external memory for an instruction access, and 0x40000000～0xBFFFFFFF in external memory and 0xD0000000～0xDFFFFFFF in I/O space for a data access.
The S2 (0x80000000 to 0x9FFFFFFF: 0.5GB) and S3 (0xA0000000 to 0xBFFFFFFF: 0.5GB) spaces are supervisor level-only spaces; the logical addresses are mapped to physical

addresses 0x80000000 to 0x9FFFFFFF in a fixed manner.
The S4 (0xC0000000 to 0xDFFFFFFF: 0.5GB) space is a supervisor level-only space; the logical addresses are mapped to the internal I/O space 0xC0000000 to 0xDFFFFFFF in a fixed manner.
The M0 (0xE0000000 to 0xFFFFFFFF: 0.5GB) space is a reserved space and cannot be accessed.

<Programming Note>
Logical addresses cannot be mapped to the control register space, which is the space with addresses from 0xC0000000 to 0xCFFFFFFF in the internal I/O space. This space can be accessed through the S4 space at privileged level or higher.

<Programming Note>
Pages must be aligned by page size. The hardware cannot convert addresses properly if page sizes other than those described above are defined.

<Programming Note>
The instruction MMU cannot map the internal I/O space to the SU0/SU1 spaces.

Address space when not using an MMU

The AM33 microcontroller core supports a 32-bit address space, and is upwardly compatible with the AM30/AM31/AM32 microcontroller core and memory map. The addresses are mapped to the physical address space in a fixed manner.

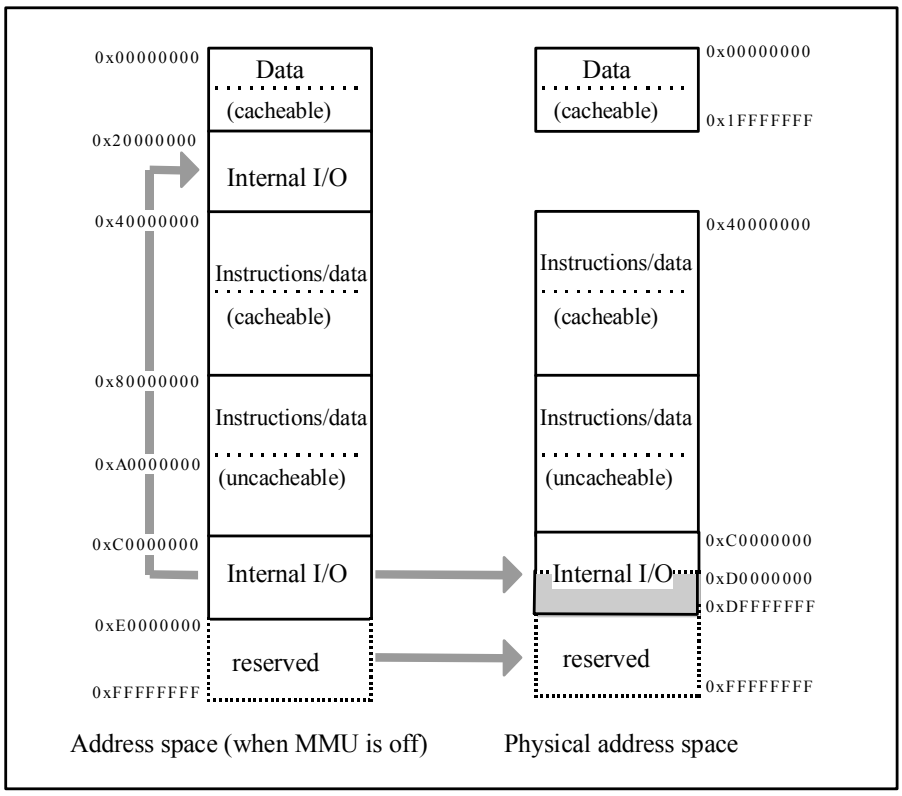


Figure 12 Address space when not using MMU

In order to maintain compatibility with the AM30/AM31/AM32 microcontroller core, the internal I/O space 0x20000000 to 0x3FFFFFFF is mirrored on 0xC0000000 to 0xDFFFFFFF.

Physical Space

The AM33 microcontroller core supports a 32-bit physical address space. The mapping of the addresses in the physical address space onto the physical memory that is implemented in the system depends on the BCU (Bus Control Unit), which is external to the microcontroller core.

2.7.2. Address Translation and Process Identifiers

Address translation

When an MMU is enabled, the logical address space is divided into units called pages, and those page units are converted into physical addresses. The address translation table in external memory stores the physical addresses that correspond to the logical addresses and the memory protection codes, etc. as additional information. In order to speed up the address translation processing, the TLB caches the contents of the address translation table in external memory.

If a memory access is generated and the address belongs to either SU0 or SU1, the TLB is searched according to the logical address. If the logical address is registered in the TLB, a TLB hit is generated and the corresponding physical address and protective attributes are read from the TLB. If there is no violation of protection, the physical address is determined. If there is a violation of protection, a TLB access exception is generated and processing shifts to the TLB access exception processing routine. However, if the accessed physical address is not registered in the TLB, a TLB miss exception is generated and processing shifts to the TLB miss exception processing routine. In the TLB miss processing routine, the address translation table in external memory is searched and other additional information such as the corresponding logical address and memory protection code is registered in the TLB. After returning from the exception processing routine, the instruction fetch or instruction that generated the TLB miss is re-executed.

If the address belongs to S2 or S4, or if the MMU is off, the logical addresses are mapped to physical addresses in a fixed manner.

Process Identifier (PID)

The TLB includes process identifiers (PIDs) that are used to distinguish multiple processes running simultaneously while sharing logical addresses. A PID consists of 8 bits. The software can set the PID of the process that is currently running in PIDR, one of the MMU registers. By using PIDs, there is no need to purge the TLB when switching processes.

2.7.3. TLB (Translation Lookaside Buffer)

2.7.3.1. TLB Configuration

The TLB caches the address translation table that is located in external memory. The address translation table stores virtual page numbers (VPN), their corresponding physical page numbers (PPN), process identifiers (PID), and additional information such as memory protection codes (PR).

The following figure

shows the overall configuration of the TLB. The association method for this TLB is full associative, and the replacement algorithm is FIFO.

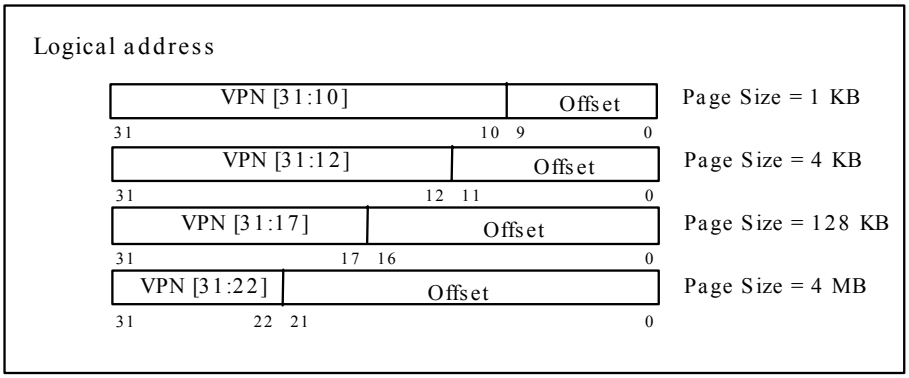


Figure 13 TLB configuration

<Programming Note>
When the power is turned on, the TLB is not initialized. Before using the TLB, all entries must be invalidated by using the MMUCTR.IIV and MMUCTR.DIV bits.

2.7.3.2. TLB Address Comparison

TLB address comparison is performed when accessing external memory (for an instruction fetch or data reference from a program). In this case, the targets of the comparison differ according to the combination of the PS bits and the G bit. If the result of the comparison is "match" and the entry is valid (V = 1), a TLB hit occurs. If the result of the comparison is "no match" or the entry is invalid (V = 0), a TLB miss occurs.
The VPN that is the target of comparison differs according to the page size, as shown in the following table:

Table 49 TLB Address Comparison

Page size	PS	Target of comparison
1 Kbytes	10	Entire VPN (22 bits)
4 Kbytes	00	Upper 20 bits of VPN
128 Kbytes	01	Upper 15 bits of VPN
4 Mbytes	11	Upper 10 bits of VPN

When G = 0, the target VPNs and PIDs are compared. When G = 1, only the target VPNs are compared.

<Programming Note>
Do not establish settings that will result in multiple entries generating hits simultaneously. Hardware operation is not guaranteed in such circumstances.

The page management information within a TLB entry includes the PV, D, PR, and C bits in addition to G.
The PV bit indicates whether that page resides in memory or not. If the PV bit is "0," an instruction access exception or a data access exception is generated.
The D bit indicates whether there was a write to the page corresponding to the entry. If writing to page is to be carried out while the D bit is "0," a data access exception is generated.
The PR bit indicates the access permission for that page in both supervisor level and user level, and is used for memory protection. If an access violating the access permission is to be executed, an instruction access exception or a data access exception is generated.
The C bit controls caching for that page. However, the bit information is valid only when the CE bit in MMUCTR has been set to "1." If the CE bit in MMUCTR is "1" and the C bit is "1," the page

is cacheable. If the CE bit in MMUCTR is "1" and the C bit is "0," the page is uncacheable. If the CE bit in MMUCTR is "0," whether the page is cacheable or uncacheable depends on the space in which that page resides, regardless of the C bit. In other words, if the CE bit in MMUCTR is "0," the SU0 space is cacheable and the SU1 space is uncacheable.

2.7.3.3. TLB Entry Lock

The initial value for the TLB replacement entry is "0," and this value is automatically incremented by "1" by the hardware each time when the replacement operation is completed. (After the count reaches "31," the next value is "0.") If MMUCTR.ITL/MMUCTR.DTL is a value other than "000," a predetermined entry lock can be applied. When the lock is set, in the case that the incremented entry reaches the lock space, the smallest entry outside of the lock space becomes the next replacement entry. (When MMUCTR.ITL = 100, RP = 8.)
If setting the lock for MMUCTR.ITL/MMUCTR.DTL, set the value of MMUCTR.IRP/MMUCTR.DRP to the nonlocked space simultaneously. The RP value before setting the lock is used for the first replacement after the lock was set, and replacement occurs even in the lock space. Note that even if the lock has been set, replacement to the lock space is possible when the RP was specified by software or when the RP was updated by the TLB entry lookup operation.

2.7.4. MMU Functions

2.7.4.1. MMU Hardware Management

There are two types of MMU hardware management.

(1) Converting logical addresses into physical addresses by controlling the TLB, all in accordance with the settings of the MMU control register (MMUCTR).
(2) Accepting page management information from the TLB and hit information during address translation, and evaluating MMU exceptions.

2.7.4.2. MMU Software Management

2.7.4.2.1. MMU Register Settings

The MMU register settings are made in the S2/S4 space in supervisor level. After confirming that the register contents have been changed, instruction fetches and data accesses are made to the SU0/SU1 space.

2.7.4.2.2. TLB Entry Registration

To register a TLB entry, set the virtual page number (VPN) and PID in the page entry upper register (IPTEU or DPTEU). (in the event of a TLB miss, the hardware automatically sets the missed virtual page number (VPN) and PID of the page in PTEU. The hardware simultaneously sets the replacement page entry number in MMUCTR.IRP/MMUCTR.DRP) After loading the contents of the page table entry into the CPU's internal registers, it is possible to register a page in the entry indicated by MMUCTR.IRP/MMUCTR.DRP by transferring the contents of the CPU's internal registers to page entry lower register 2 (IPTEL2 or DPTEL2). (This access is a dummy access; nothing is written in IPTEL/DPTEL or in IPTEL2/DPTEL2.) Furthermore, a page can be registered under a specific entry by overwriting the value in MMUCTR.IRP/MMUCTR.DRP.
Note that when registration is accomplished by writing to the page entry upper register (IPTEU or DPTEU), the hardware guarantees that the contents of the PS bit and G bit in the TLB tag will match the contents of the same bits in the data section.

<*Programming Note*>
The TLB entry registration operation must be performed in the S2/S4 spaces.

2.7.4.2.3. TLB Entry Read

By setting the VPN and PID in the page entry upper register (IPTEU or DPTEU) and transferring the contents of the page entry lower register (IPTEL or DPTEL) or the contents of the page entry lower register 2 (IPTEL2 or DPTEL2) to the CPU's internal registers, the entry with the same VPN and PID can be read. (This is called the "TLB lookup operation.")
The read value has the format shown in page 157, 2.7.3.1, TLB Configuration and includes the G bit and PS bit from the TAG array and the PPN, PR, D, and PV from the data array. When PTEL.V = 1, it indicates that the matching entry was found, and the read value is valid. In addition, the entry number that generated the hit is set in MMUCTR.IRP/MMUCTR.DRP. When PTEL.V = 0, it indicates that the matching entry was not found, and the read value is undefined. In addition, the existing value is maintained in MMUCTR.IRP/MMUCTR.DRP.

<*Programming Note*>
The TLB entry read operation must be performed in the S2 through S4 spaces.

2.7.4.2.4. TLB Entry Deletion

A TLB entry can be deleted by reading and writing the page entry lower register (IPTEL or DPTEL) or the page entry lower register 2 (IPTEL2 or DPTEL2).

In addition, when the power is turned on, the TLB is not initialized. Before turning an MMU on from its initial status, all of the entries must be invalidated by using MMUCTR.IIV/MMUCTR.DIV.

<Programming Note>

The TLB entry deletion operation must be performed in the S2 through S4 spaces.

2.7.4.3. Table Work

If a TLB miss occurs (data), the software references the page table and refills the TLB according to the procedure described below.

First, the base address is referenced in memory for the page table that is indicated in PTBR (page table base register). Next, the PTD (page table descriptor) is referenced in the address that is derived from the above base address with an offset of the upper bits of VPN of the logical address x 4 bytes.

Afterwards, the necessary information is read/written through referencing the PTE in the address that is derived from PTB2 (secondary base address) indicated in PTD with an offset of the lower bits of VPN x 4 bytes.

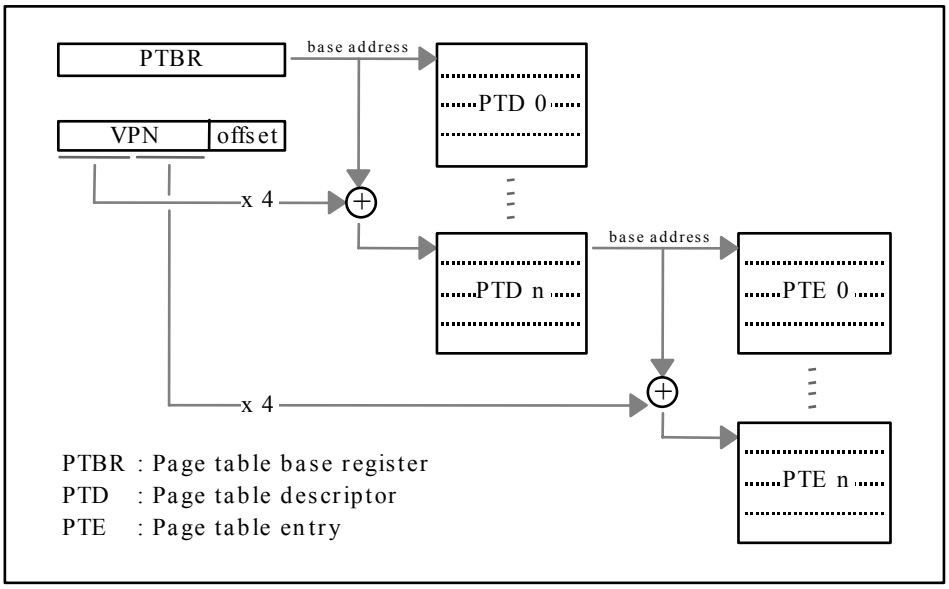


Figure 14 TLB Refill referencing the page table

<Programming Note>

The MMUs are turned on/off by writing to MMUCTR.ITE and MMUCTR.DTE. Therefore, there is a delay between the time when the instruction turning an MMU on or off is executed and the time when the MMU actually turns on or off. Accordingly, when using the data MMU, overwrite MMUCTR.DTE in the S2 through S4 areas, and perform a memory access in the SU0 or SU1 spaces after confirming that the overwrite has been completed. Similarly, when using the instruction MMU, branch to an instruction in the SU0 or SU1 spaces by using a jump instruction, etc., after confirming that the MMUCTR.ITE overwrite has been completed. This is also applied when turning an MMU off. If SU0 or SU1 is used before the overwrite is completed, subsequent operation is not guaranteed.

2.7.5. Exceptions

There are four exceptions that are generated in an MMU:

(1) Instruction TLB miss exception
(2) Data TLB miss exception
(3) Instruction access exception
(4) Data access exception

2.7.5.1. Instruction TLB Miss Exception

An instruction TLB miss exception is generated under the following circumstances:

(1) When no matching entry was found in the instruction TLB tag comparison
(2) When the entry was invalid although a matching entry was found in the instruction TLB tag comparison

The following processing is performed by the MMU if an instruction TLB miss exception is generated:

- The MMU writes the virtual page number and PIDR.PID for the logical address where the exception was generated in the virtual page entry upper register (IPTEU).
- The MMU writes the exception cause flag in the MMUFCR.IFC (MMUFCR[3:0] = 0001).
- The MMU determines the entry number for the refill destination and writes it in MMUCTR.IRP.
- The MMU notifies the CPU of the generation of the instruction TLB miss exception.

2.7.5.2. Data TLB Miss Exception

A data TLB miss exception is generated under the following circumstances:

(1) When no matching entry was found in the data TLB tag comparison
(2) When the entry was invalid although a matching entry was found in the data TLB tag comparison

The following processing is performed by the MMU if a data TLB miss exception is generated:
The MMU writes the virtual page number and PIDR.PID for the logical address where the exception was generated in the virtual page entry upper register (DPTEU).

- The MMU writes the exception cause flag in the MMUFCR.DFC (MMUFCR[19:16] = 0001).
- The MMU determines the entry number for the refill destination and writes it in MMUCTR.DRP.
- The MMU notifies the CPU of the generation of the data TLB miss exception.

2.7.5.3. Instruction Access Exception

An instruction access exception occurs under the following circumstances:

(1) When the actual access type is not permitted under the access permission specified by the PR bit (MMUFCR[3] = 1) although the instruction TLB tag comparison results in a match and the entry is valid
(2) When the PV bit is "0" (MMUFCR[2] = 1) although the instruction TLB tag comparison results in a match and the entry is valid,
(3) When the mapped physical address is illegal (0x00000000 to 0x3FFFFFFF, or 0xC0000000 to 0xFFFFFFFF) (MMUFCR[9] = 1) although the instruction TLB tag comparison results in a match and the entry is valid

The following processing is performed by the MMU if an instruction access exception is generated:

- The MMU writes the virtual page number and PIDR.PID for the logical address where the exception was generated in the virtual page entry upper register (IPTEU).
- The MMU writes the exception cause flag in the MMUFCR.IFC.

- The MMU writes the entry number where the exception occurred in MMUCTR.IRP.
- The MMU notifies the CPU of the generation of the instruction access exception.

2.7.5.4. Data Access Exception

A data access exception occurs under the following circumstances:

(1) When the actual access type was not permitted under the access permission specified by the PR bit (MMUFCR[19] = 1) although the data TLB tag comparison results in a match and the entry is valid
(2) When the PV bit was "0" (MMUFCR[18] = 1) although the data TLB tag comparison results in a match and the entry is valid
(3) When the D bit is "0" (MMUFCR[17] = 1) although the data TLB tag comparison results in a match, the entry is valid, the page is valid, and the write access permission is valid
(4) When the mapped physical address is illegal (0x00000000 to 0x3FFFFFFF, or 0xC0000000 to 0xCFFFFFFF, 0xE0000000 to 0xFFFFFFFF) (MMUFCR[25] = 1) although the data TLB tag comparison results in a match and the entry is valid

The following processing is performed by the MMU if a data access exception is generated:

- The MMU writes the virtual page number and PIDR.PID for the logical address where the exception was generated in the virtual page entry upper register (DPTEU).
- The MMU writes the exception cause flag in the MMUFCR.DFC.
- The MMU writes the entry number where the exception occurred in MMUCTR.DRP.
- The MMU notifies the CPU of the generation of the data access exception.

2.7.5.5. Exception Cause Codes

The exception cause flags are located in the MMU exception cause register (MMUFCR), and consist of 10 bits for instruction flags and 10 bits for data flags (IFC[9:0] and DFC[9:0]). These bits are defined as follows:

Table 50 Exception cause flag

IFC[0], DFC[0]	TLB miss flag
IFC[1],DFC[1]	Initial write exception flag
IFC[2],DFC[2]	Page invalid exception flag
IFC[3],DFC[3]	Protection violation exception flag
IFC[4],DFC[4]	Access level flag (0:usr level; 1: supervisor level)
IFC[5],DFC[5]	Access type flag (0: read; 1: write)
IFC[8:6],DFC[8:6]	Protection flag (PR)
IFC[9],DFC[9]	Illegal address exception flag

A list if the exception cause codes is shown below.

Table 51 Exception cause code list

Exception type	Description	Level	Operation details	IFC[9:0]	DFC[9:0]
TLB miss	-	-	-	xxxxxxxxx1	xxxxxxxxx1
Access exceptions	Initial write	-	-	-	xxxxxxxx1x
	Page invalid	-	-	xxxxxxx1xx	xxxxxxx1xx
	Protection violation	supervisor	Write to supervisor read-only page	-	x000111xxx
			Write to supervisor/user read-only page	-	x010111xxx
		User	Write to supervisor read-only page	-	x000101xxx
			Write to supervisor read/write page	-	x100101xxx
			Write to supervisor/user read-only page	-	x010101xxx
			Write to supervisor read/write page or user read-only page	-	x110101xxx
			Read from supervisor read-only page	x000001xxx	x000001xxx
	Illegal address	-	Read from supervisor read/write page	x100001xxx	x100001xxx
			-	1xxxxxxxxx	1xxxxxxxxx

2.7.5.6. Flow of Processing When an Exception Is Generated

The flow of MMU exception processing is shown below.

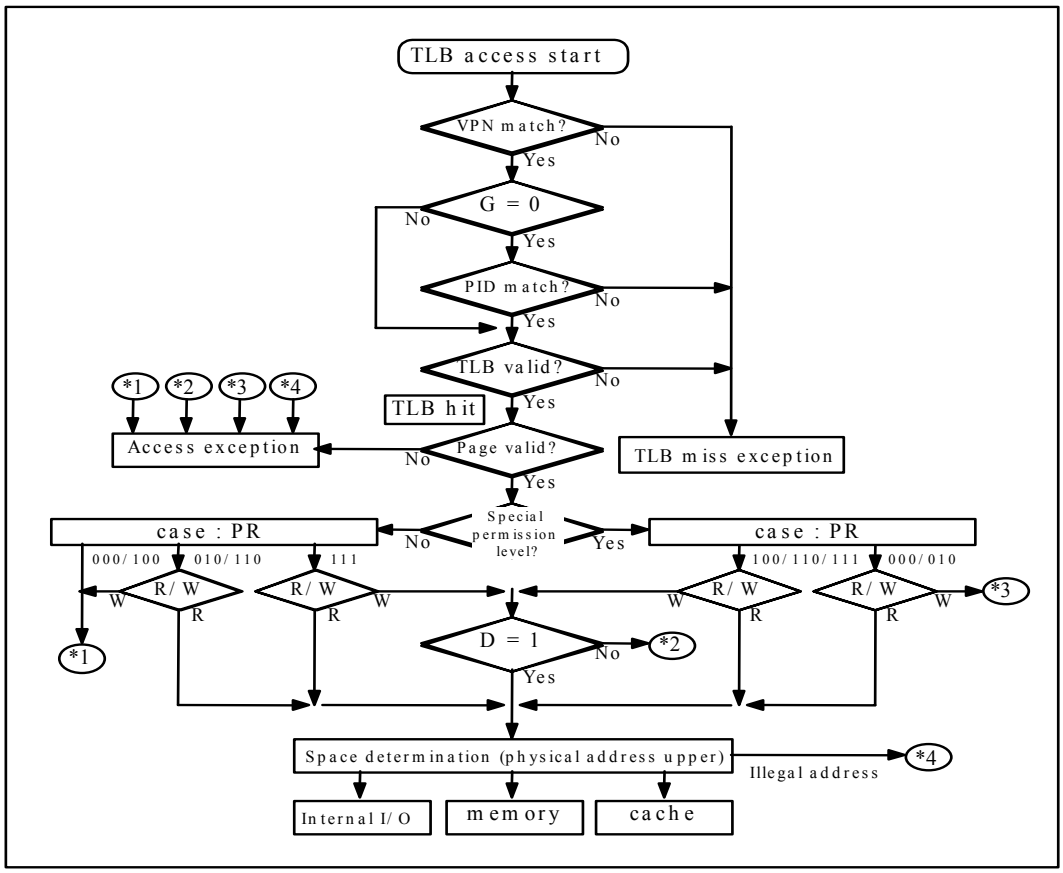


Figure 15 MMU Exception process flow

2.8. Cache

The AM33 microcontroller core is equipped with Harvard-type caches in which instructions and data are separated. Each cache is a physical cache that is accessed through physical addresses, and the association method is four-way set associative. Furthermore, the size of the instruction cache and the data cache is 16K each.
The purpose of a cache is to absorb the difference between the operating speed of the external memory and that of the CPU core, resulting in a faster apparent memory access speed. The instruction cache stores instructions that were requested by the CPU core in line units (16 bytes), and the data cache stores data that was requested by the CPU core in line units (16 bytes). If the cache is enabled, all instruction fetches and data accesses by a load/store instruction in a cacheable area are cached. Both the instruction cache and the data cache are physical caches that handle cache operations through physical addresses.
The followings are the features of the AM33 microcontroller core's internal caches:

- Separate caches for instructions and data is adopted and avoids conflicts between instruction accesses and data accesses
- 4-way set associative association method is adopted for both the instruction cache and the data cache
- Instruction cache size: 16KB; data cache size: 16KB
- Memory can be accessed during the refill operation (nonblocking cache)
- Way-unit operation can be selected for both instruction cache and data cache (cache or RAM)
- Refill begins from the missed word in order to minimize the cache miss penalty
- The way unit LRU method is used as the replace algorithm for both the instruction cache and the data cache
- Write-back and write-through can be selected as the data cache writing method
- In write-back mode, it is possible to select whether to allocate or not to allocate lines in which a write miss occurred
- The data cache permits batch invalidation of its contents
- Permits entry batch purge that searches for tags in all four ways by specifying only an entry without specifying ways.

2.8.1. Instruction Cache

The instruction cache has a size of 16K, employs the four-way set associative method of association, and is a physical cache that is accessed through physical addresses after address translation. The cache consists of four data memories and four tag memories. The cache can be used through enabling it after initializing it in the disabled state. When an instruction is fetched from a cacheable area, the instruction cache is read and written under the control of hardware, with no intervention by the software. In addition, the memories are mapped in the internal I/O space, and can be directly read and written by software.

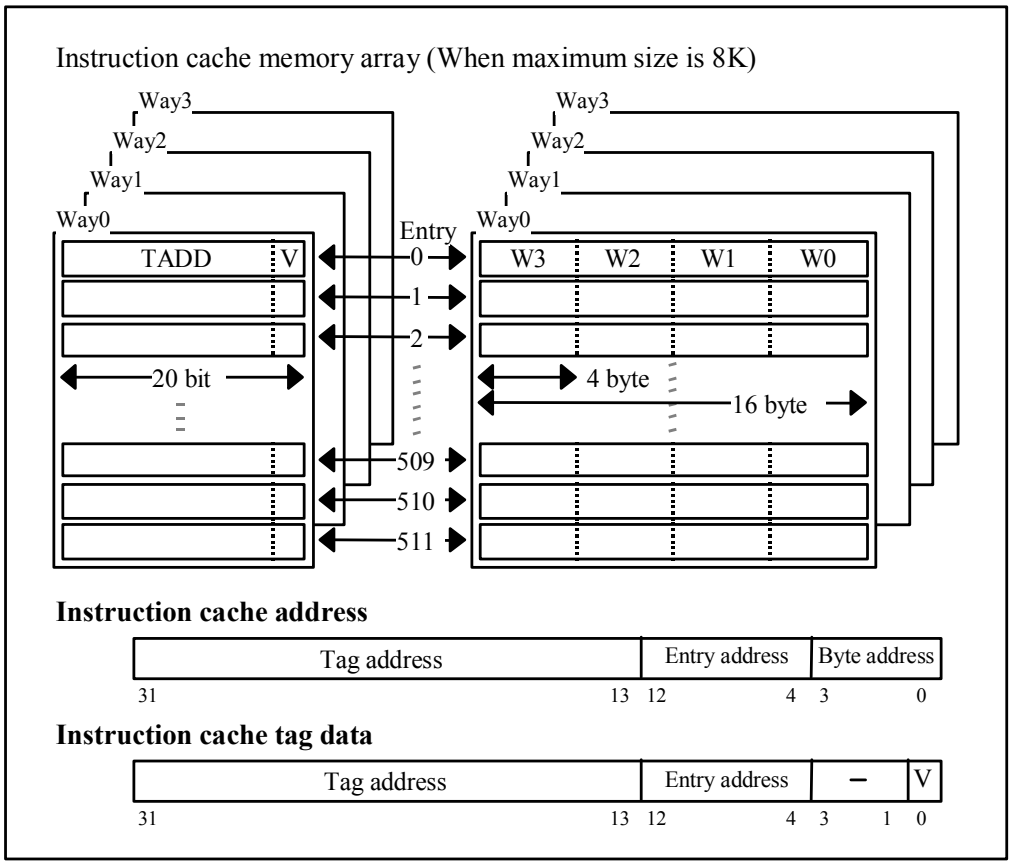


Figure 16 Instruction cache

Data memory

The data memory stores instructions in 16-byte units. The size of the data memory for one way is 4KB, for a total size of 16KB. The line size in a data memory is 16 bytes, and the number of entries is 256. Instruction transfers from external memory to the instruction cache are performed in units of 16 bytes (128 bits), and instruction transfers from the instruction cache to the CPU core are performed in units of 64 bits. The contents of the data memory are not initialized at a reset.

Tag memory

When the instruction cache is at its size of 16KB, tag memory has a maximum of 256 entries. When the size of the instruction cache is 32KB, each entry consists of a tag address field (TADD) that stores bits 31 through 12 (20 bits) of the instruction address, and a valid bit (V) that indicates whether the entry is a valid entry or not. The tag address field (TADD) and the valid bit (V) are not initialized at a reset. The valid bit (V) is initialized by manipulating the instruction cache invalidate bit in the cache control register (CHCTR).

2.8.2. Data Cache

The data cache has a size of 16KB, employs the four-way set associative method of association, and is a physical cache that is accessed through physical addresses after address translation. The cache consists of four data memories, four tag memories, and a write-back buffer. A load/store instruction performs a data access in a cacheable area, accordingly the data cache is read and written under the control of hardware, with no intervention by the software. In addition, the memories are mapped in the internal I/O space, and can be directly read and written by software.

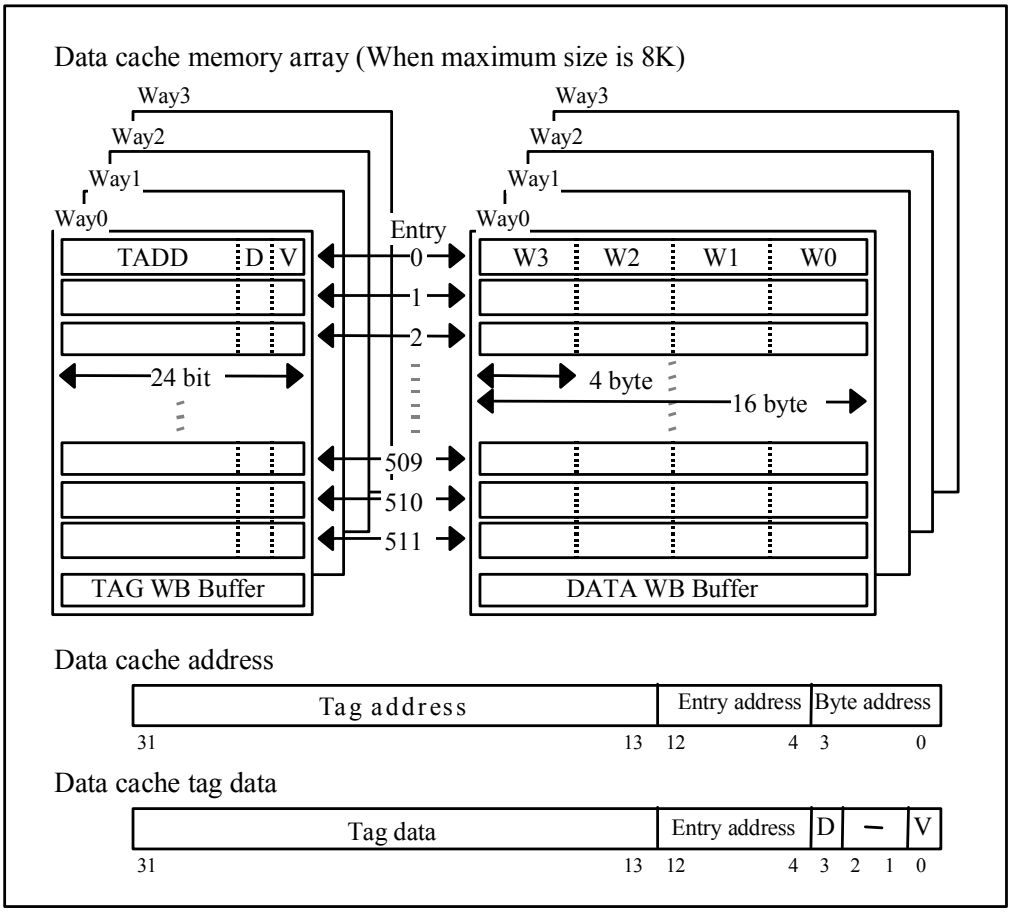


Figure 17 Data cache

Data memory

The data memory stores data in 16-byte units. The size of the data memory for one way is 4K, for a total size of 16K. The line size in a data memory is 16 bytes, and the maximum number of lines is 256. Data transfers from external memory to the data cache are performed in units of 16 bytes (128 bits), and data transfers from the data cache to the CPU core are performed in units of 32 bits. The contents of the data memory are not initialized at a reset.

Tag memory

When the data cache is at its size of 16K, tag memory has a of 256 entries. When the size of the data cache is 16K, each entry consists of a tag address field (TADD) that stores bits 31 through 12 (20 bits) of the data address, a valid bit (V) that indicates whether the entry is a valid entry or not, and a dirty bit (D) that indicates whether there has been a write to the

corresponding entry in write-back mode. The tag address field (TADD), the valid bit (V), and the dirty bit (D) are not initialized at a reset. The valid bit (V) and the dirty bit (D) are initialized by manipulating the data cache invalidate bit in the cache control register (CHCTR).

Write-back buffer

There is a write-back buffer for one line (16 bytes) in the data array and for one entry in the tag array. When the data cache is used in write-back mode, the write-back buffer is used to temporarily hold the write-back data.

2.8.3. Operation

2.8.3.1. Instruction Cache

2.8.3.1.1. Initialization

The instruction cache is disabled when the CPU is reset. To enable the instruction cache, first set the instruction cache invalidate bit (ICINV) in the cache control register (CHCTR) to invalidate all entries set the instruction cache enable bit (ICEN) after setting the instruction cache invalidate bit (ICINV) in the cache control register (CHCTR) to invalidate all entries. Note that control through the cache control register (CHCTR) becomes valid only when an instruction fetch is generated subsequent to the point at which the instruction writing to the cache control register (CHCTR) reaches the CPU pipeline writing stage.

Examples of initialization routines are shown below.

When initializing immediately after a reset

```
mov     0xC0000070,a0
mov     0x0010,d0
movhu   d0,(a0)     ; Invalidates the cache (initialization)
mov     0x0001,d0
movhu   d0,(a0)     ; Enables the instruction cache
```

When initializing the instruction cache in operation

```
mov     0xC0000070,a0
movhu   (a0),d0     ; Reads the current contents of the control register
and     0xFFFFFFFE,d0
movhu   d0,(a0)     ; Disables the instruction cache
setlb
movhu   (a0),d0
btst    0x04,d0     ; Checks whether the instruction cache is busy
lne
or      0x0010,d0
movhu   d0,(a0)     ; Invalidates the instruction cache (initialization)
```

<Programming Note>

The operation which changes the way mode must be carried out after disabling the cache, checking the busy bit, and confirming that the cache is not in operation.

2.8.3.1.2. Reading Operation

Cache hit operation

If an instruction fetch in a cacheable space is executed when the instruction cache is enabled, the instruction cache tag array is accessed with using the tag entry address portion of the instruction fetch address as the address. If the value in the tag address field (TADD) of the accessed entry matches the value in the tag field of the instruction fetch address, and the valid bit (V) for the entry has been set (i.e., is "1"), it is called an "instruction cache hit".

If an instruction cache hit occurs, the instruction in the corresponding entry (line) in the data memory is sent to the CPU. It takes one cycle from the tag array access to the instruction read. The bit width of an instruction that can be supplied in one access depends on the core implementation, but the core guarantees enough bus bandwidth to be able to supply at least a 32-bit instruction in a single access.

Cache miss operation

If the instruction cache tag array is accessed with using the tag entry address portion of the instruction fetch address as the address, and the value in the tag address field (TADD) of the accessed entry does not match the value in the tag field of the instruction fetch address, or if

they match each other and the valid bit (V) for the entry has not been set (i.e., is "0"), it is called a "instruction cache miss".
If an instruction cache miss occurs, the external memory becomes the target of the instruction fetch operation. At the same time that the instruction is fetched from the external memory, it is necessary to allocate in the cache an entry (line) for caching the instruction.
First, the way for the refill target is selected on the basis of the value of the valid bit (V) in the accessed tag entry, the way operation mode that is set in the cache control register (CHCTR), and information on the way that was selected at the last data accessing.
A flowchart for selecting the refill target way for the instruction cache is shown below.

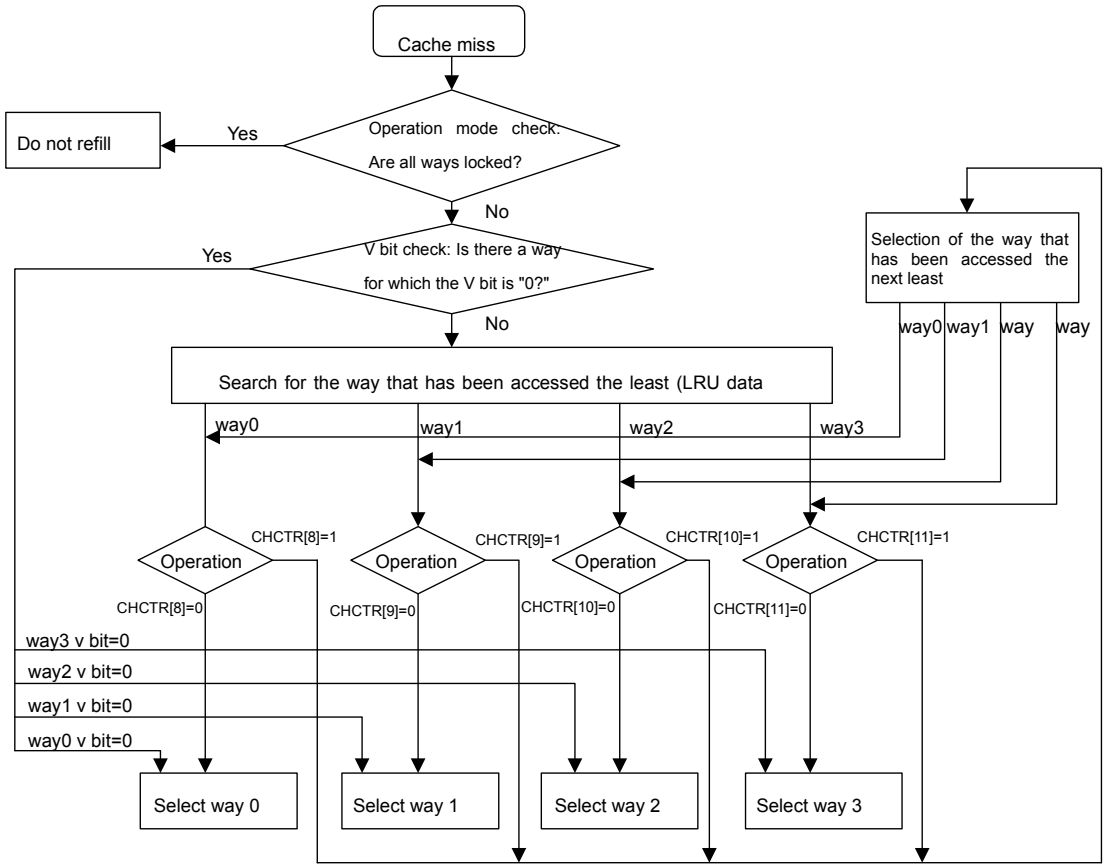


Figure 18 Flowchart for selecting the refill target way for the instruction cache

Next, an external bus access (refill) is initiated in order to load one line of instructions from the external memory into the cache memory in the selected way. A refill begins from the word (4 bytes) including the access address, and consists of a burst transfer of four words (16 bytes) for the data transfer for one line of data. During the refill sequence, the tag address field (TADD) in the tag array entry is updated and the valid bit (V) is set, and then the target line within the data array is updated. The instruction is simultaneously supplied to the CPU. The instruction is transferred in units of at least four bytes. The CPU resumes operations at the moment that the needed instruction is transferred.

2.8.3.2. Data Cache

2.8.3.2.1. Initialization

The data cache is disabled when the CPU is reset. To enable the data cache, the data cache

enable bit (DCEN) must be set after setting the data cache invalidate bit (DCINV) in the cache control register (CHCTR) and invalidating all entries. Note that control through the cache control register (CHCTR) becomes valid only when a data access is generated subsequent to the point at which the instruction writing to the cache control register (CHCTR) reaches the CPU pipeline writing stage.
Examples of initialization routines are shown below.
When initializing immediately after a reset

```
mov     0xC0000070,a0
mov     0x0020,d0
movhu   d0,(a0)     ; Invalidates the cache (initialization)
mov     0x0002,d0
movhu   d0,(a0)     ; Enables the data cache
```

When initializing the data cache while in operation

```
mov     0xC0000070,a0
movhu   (a0),d0     ; Reads the current contents of the control register
and     0xFFFFFFFD,d0
movhu   d0,(a0)     ; Disables the data cache
setlb
movhu   (a0),d0
btst    0x08,d0     ; Checks whether the data cache is busy
lne
or      0x0010,d0
movhu   d0,(a0)     ; Invalidates the data cache (initialization)
```

<Programming Note>
An operation that invalidates the cache, switches the writing mode, or changes the way mode must be performed after disabling the cache, checking the busy bit, and confirming that the cache is not in operation.

2.8.3.2.2. Reading Operation

Cache hit operation

If a data read access in a cacheable space is executed when the data cache is enabled, the data cache tag array is accessed with using the tag entry address portion of the data address as the address. If the value in the tag address field (TADD) of the accessed entry matches the value in the tag field of the data address, in the case that the valid bit (V) for the entry has been set (i.e., is "1"), it is called a "data cache read access hit".
If a data cache read access hit occurs, the data in the corresponding entry (line) in the data memory is sent to the CPU. It takes one cycle from the tag array access to reading the data. When writing data, it takes at least one cycle to access the tag array.

Cache miss operation

If the data cache tag array is accessed with using the tag entry address portion of the data address as the address, and the value in the tag address field (TADD) of the accessed entry does not match the value in the tag field of the data address, or if they match but the valid bit (V) for that entry has not been set (i.e., is "0"), it is called a "data cache read access miss".
If a data cache read access miss occurs, the external memory becomes the target of the data access operation. At the same time that the data is read from external memory, it is necessary to allocate in the cache an entry (line) for caching that instructions. At this point, it is necessary to perform an operation that confirms the relationship between the data that is being pushed out of the entry that is going to be allocated and the data in external memory, and maintains the consistency of the data.
First, the way that is to be the refill target is selected on the basis of the value of the valid bit (V) in the accessed tag entry, the way operation mode that is set in the cache control register (CHCTR), and information on the way that was selected the last time a data access was executed. A flowchart for selecting the refill target way for the data cache is shown below.

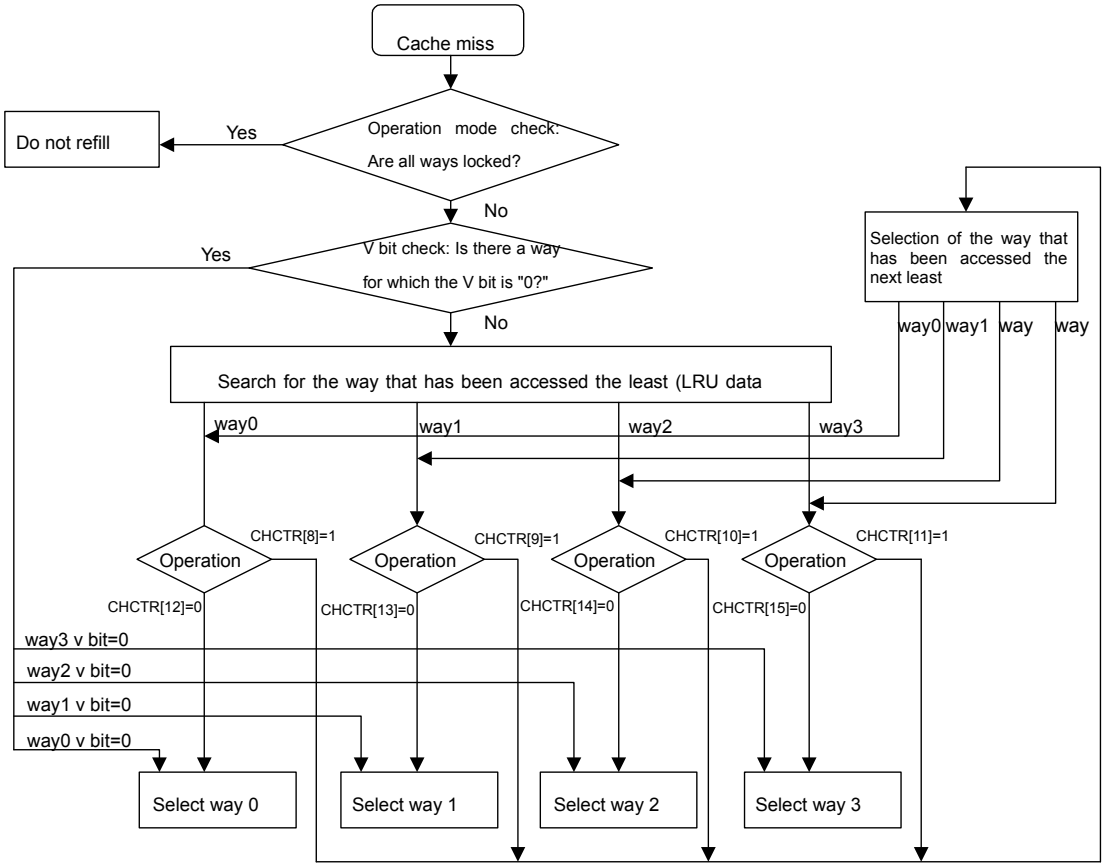


Figure 19 Flowchart for selecting the refill target way for the data cache

Next, the dirty bit (D) in the tag entry for the selected way, the value in the tag address field (TADD), and the one line of data that is in the access entry within the data array are saved in the write-back buffer. This save sequence is performed regardless of the data cache write method.
Then, an external bus access (refill) is initiated in order to load one line of data from external memory into cache memory. A refill begins from the word (4 bytes) including the access address, and consists of a burst transfer of four words (16 bytes) for one line of data. During the refill sequence, the tag address field (TADD) in the tag array entry is updated, the valid bit

(V) is set, the dirty bit (D) is cleared, and then the target line within the data array is updated. The data is simultaneously supplied to the CPU. The CPU resumes operations at the moment that the needed data is transferred.
When the data cache write method is write-back mode, in the case that the dirty bit (D) in the accessed entry has been set (i.e., is "1"), an external bus access is initiated in order to perform the write-back operation after the refill operation is completed, and the data that was saved in the write-back buffer is written to external memory. The purpose of this operation is to maintain consistency between the data in the data cache and the data in external memory.

2.8.3.2.3. Write Operation

There are two modes for writing the data cache: write-back and write-through. The writing operation differs according to these two writing modes.

Write-back mode

Cache hit operation

If a data write access in a cacheable space is executed when the data cache is enabled, the data cache tag array is accessed with using the tag entry address portion of the data address as the address. If the value in the tag address field (TADD) of the accessed entry matches the value in the tag field of the data address, and the valid bit (V) for that entry has been set (i.e., is "1"), it called a "data cache write access hit."
If a data cache write access hit occurs, data from the CPU is written in the corresponding entry (line) in the data memory. The line in which the data is written at this point will have data that is newer than the data in the same address in external memory, so data consistency is not being maintained. This state is called the "dirty state," and the dirty bit (D) in the corresponding tag entry is set to "1." This dirty line will be written back to external memory when the line is refilled. It is essential to note that until that happens, consistency is not being maintained between the data in the data cache and the data in external memory.
If write access hits continue to be generated, consecutive tag array accesses and data writes can be executed each cycle with no waiting.

Cache miss operation

If the data cache tag array is accessed with using the tag entry address portion of the data address as the address, and the value in the tag address field (TADD) of the accessed entry does not match the value in the tag field of the data address, or if the valid bit (V) for that entry has not been set (i.e., is "0") although they match, it is called a "data cache write access miss."
In write-back mode, the location of the data write can be controlled through the setting of the data cache allocate mode bit (DCALMD) in the cache control register (CHCTR).
When allocate mode is set (DCALMD = 0), the data is written only in the cache, and not in external memory. In other words, even if a data cache write access miss is generated, the location of the data write is in the data cache. As a result, it becomes necessary to allocate an entry (line) for writing that data in the cache. At this point, it is necessary to perform an operation that confirms the relationship between the data that is being pushed out of the entry that is going to be allocated and the data in external memory, and maintains the consistency of the data.
First, the way that is to be the refill target is selected on the basis of the value of the valid bit (V) in the tag entry that was accessed, the way operation mode that is set in the cache control register (CHCTR), and information on the way that was selected at the last data access. The flowchart for selecting the refill target way is identical to the one that is used in the event of a data cache read miss.
Next, the dirty bit (D) in the tag entry for the selected way, the value in the tag address field (TADD), and the one line of data that is in the access entry within the data array are saved in the write-back buffer. This save sequence is performed regardless of the data cache write method.
Then, an external bus access (refill) is initiated in order to load one line of data from external

memory into cache memory. A refill begins from the word (4 bytes) including the access address, and consists of a burst transfer of four words (16 bytes) for one line of data. During the refill sequence, the tag address field (TADD) in the tag array entry is updated, the valid bit (V) is set, the dirty bit (D) is set, and then the target line within the data array is updated. If the dirty bit (D) in the accessed entry has already been set (i.e., is set to "1"), an external bus access is initiated in order to perform the write-back operation after the refill operation is completed. The data that was saved in the write-back buffer is written to external memory. The purpose of this operation is to maintain consistency between the data in the data cache and the data in external memory.

In "do not allocate" mode (DCALMD = 1), data is only written to external memory, and is not written in the cache. Therefore, the cache line replacement operation is not performed.

If the data cache way operation mode settings are set to "Do not refill even in the event of a cache miss" mode for all ways, the write operation is not performed, and the CPU executes the next instruction.

Write-through mode

Cache hit operation

If a data write access in a cacheable space is executed when the data cache is enabled, the data cache tag array is accessed using the tag entry address portion of the data address as the address. If the value in the tag address field (TADD) of the accessed entry matches the value in the tag field of the data address, and the valid bit (V) for that entry has been set (i.e., is "1"), it is called a "data cache write access hit".

If a data cache write access hit occurs, data from the CPU is written in the corresponding entry (line) in the data memory, and the data is also written to the corresponding address in external memory at the same time. Because the write to the data cache and the write to external memory are performed simultaneously, consistency is always maintained between the data in the data cache and the data in external memory.

Operation during a data cache write access in write-through mode is closely related to the data cache way operation mode setting. The write to the data cache is performed in any operation mode, but the write to external memory is not performed if the operation mode that is set for the way that was hit does not call for the replacement operation to be performed even if a cache miss occurs (i.e., the value of the way operation mode bit in CHCTR is "1"). In write-through mode, the CPU halts operations until the write of data in external memory is completed.

Cache miss operation

If the data cache tag array is accessed with using the tag entry address portion of the data address as the address, and the value in the tag address field (TADD) of the accessed entry does not match the value in the tag field of the data address, or if the valid bit (V) for that entry has not been set (i.e., is "0") although they match, it is called a "data cache write access miss" .

In write-through mode, if a data cache write access miss is generated, the data will be written in external memory only, and will not be written in the data cache ("non-allocate for writing"). The refill operation is also not performed.

<Programming Note>

Switch between write-back mode and write-through mode while the CPU is operating must be carried out after invalidating all of the contents of the data cache through the cache control register (CHCTR).

2.8.3.2.4. Consistency between Caches and External Memory

If write-back mode has been selected as the writing mode for a data cache, situations will occur where consistency is not being maintained between the data in the cache memory and the data in external memory. In order to maintain consistency between the data in these two different locations, it is necessary for the software to write back to external memory the data that is being kept in the data cache. This operation is called "purging;" in the AM33 microcontroller core, specific entries can be purged by executing a data access at a specific address. The table below

shows an example of address allocation when the cache size is 16KB (256 entries for each way). The allocation of addresses for purging depends on the maximum size of the cache and the implementation of the core in the LSI.

Way	Entry	Purge address
0	0	0xC8400000
	1	0xC8400010
	..	……
	254	0xC8400FE0
	255	0xC8400FF0
1	0	0xC8401000
	1	0xC8401010
	..	……
	254	0xC8401FE0
	255	0xC8401FF0
2	0	0xC8402000
	1	0xC8402010
	..	……
	254	0xC8402FE0
	255	0xC8402FF0
3	0	0xC8403000
	1	0xC8403010
	..	……
	254	0xC8403FE0
	255	0xC8403FF0

There are two types of purge operations: unconditional purging and purging with address comparison. Unconditional purging is performed by executing a read access in the address space used for purging as described previously. Purging with address comparison is performed by executing a write access to the address in question.

When performing an unconditional purge (a read access to the purge address space), the purge is performed in external memory if the entry in question of the way in question is valid and dirty (V = 1 and D = 1). After the purge is executed, that entry becomes invalid (V = 0).

When performing a purge with an address comparison (a write access to the purge address space), bits 31 through 12 of the data specified as the operand are used as the address that is compared with the tag data in the entry in all of the ways; if the address matches, and the entry is valid and dirty (V = 1 and D = 1), then the entry is purged in external memory. The status of the valid bit (V) after the purge is executed can be specified in the least significant bit (LSB) of the data specified in the operand.

If the addresses do not match or the corresponding entry is not valid (V=0), nothing happens. If the addresses match, the corresponding entry is valid, and not-dirty (V=1, D=0), the purge is not performed and the least significant bit (LSB) of the operand determines the state of valid bit.

For example, code for a purge with an address comparison versus entry 1 would be written in the following manner:

```
mov     0x44444401,d0
mov     d0,(0xC8400010)
```

The tag data for entry 1 from all ways is compared with 0x444444. If the tag data matches, a purge is executed and after the purge is executed, the valid bit (V) is set to "1." If the address does not match, nothing happens.

A purge that is initiated by one data access is performed only on the entry that meets the above conditions.

A data access that is made to a specific address in order to initiate the purge operation is a dummy access. If the data access is a read access, the data that is read is undefined. If the data access is a write access, no data is written anywhere.

2.8.3.3. Way Operation Mode

For both the instruction cache and the data cache, the operation mode can be set for each way by manipulating the cache control register (CHCTR). In normal operation mode, a way operates as a cache, but it is possible to set the way so that the contents of the cache are not updated by selecting a mode that does not perform the refill operation in the event of a cache miss. This function can be used to make it possible to use a cache in the same manner as on-chip RAM.
A way for which a mode that does not perform the refill operation even in the event of a cache miss retains its various statuses as they were when the mode was set. (This is equivalent to locking the cache.) A way for which this mode has been set is not selected as a target for the refill operation when a cache miss occurs.
If this mode has been set for all ways, the results of an access that results in a cache miss are not guaranteed. Therefore, it is essential to set data that will definitely result in a cache hit in the tag data beforehand. The tag data can be set either by setting the mode after the cache is already in operation, or else by disabling the cache and then accessing the I/O space in order to write the data in the tag array and the data in the data array.

2.8.3.4. Cache Entry Address Allocation

The cache memory tag array and data array are mapped onto the internal I/O space, and can be read/written directly through I/O access. The access size is "word" (32 bits). The following charts are an example for a 16K cache.
It is important to note that if the contents of the tag array and the data array are overwritten while they are being used as a cache, the contents of external memory and the cache will no longer match.

Instruction cache (tag)

Way	Entry	Address
0	0	0xC8100000
	1	0xC8100010
	..	…….
	255	0xC8100FF0
1	0	0xC8101000
	1	0xC8101010
	..	…….
	255	0xC8101FF0
2	0	0xC8102000
	1	0xC8102010
	..	…….
	255	0xC8102FF0
3	0	0xC8103000
	1	0xC8103010
	..	…….
	255	0xC8103FF0

Instruction cache (data)

Way	Entry	Offset 3	Offset 2	Offset 1	Offset 0
0	0	0xC800000C	0xC8000008	0xC8000004	0xC8000000
	1	0xC800001C	0xC8000018	0xC8000014	0xC8000010
	..	…….	…….	…….	…….
	255	0xC8000FFC	0xC8000FF8	0xC8000FF4	0xC8000FF0
1	0	0xC800100C	0xC8001008	0xC8001004	0xC8001000
	1	0xC800101C	0xC8001018	0xC8001014	0xC8001010
	..	…….	…….	…….	…….
	255	0xC800100C	0xC8001008	0xC8001004	0xC8001000
2	0	0xC800200C	0xC8002008	0xC8002004	0xC8002000
	1	0xC800201C	0xC8002018	0xC8002014	0xC8002010
	..	…….	…….	…….	…….
	255	0xC8002FFC	0xC8002FF8	0xC8002FF4	0xC8002FF0
3	0	0xC800300C	0xC8003008	0xC8003004	0xC8003000
	1	0xC800301C	0xC8003018	0xC8003014	0xC8003010
	..	…….	…….	…….	…….
	255	0xC8003FFC	0xC8003FF8	0xC8003FF4	0xC8003FF0

Data cache (tag)

Way	Entry	Address
0	0	0xC8300000
	1	0xC8300010
	..	…….
	255	0xC8300FF0
1	0	0xC8301000
	1	0xC8301010
	..	…….
	255	0xC8301FF0
	0	0xC8302000

Data cache (data)

Way	Entry	Offset 3	Offset 2	Offset 1	Offset 0
0	0	0xC820000C	0xC8200008	0xC8200004	0xC8200000
	1	0xC820001C	0xC8200018	0xC8200014	0xC8200010
	..	…….	…….	……. .	……. .
	255	0xC8200FFC	0xC8200FF8	0xC8200FF4	0xC8200FF0
1	0	0xC820100C	0xC8201008	0xC8201004	0xC8201000
	1	0xC820101C	0xC8201018	0xC8201014	0xC8201010
	..	…….	…….	……. .	……. .
	255	0xC8201FFC	0xC8201FF8	0xC8201FF4	0xC8201FF0
	0	0xC820200C	0xC8202008	0xC8202004	0xC8202000

2	1	0xC8302010
	..	…….
	255	0xC8302FF0
3	0	0xC8303000
	1	0xC8303010
	..	…….
	255	0xC8303FF0

2	1	0xC820201C	0xC8202018	0xC8202014	0xC8202010
	..	…….	…….	……. .	……. .
	255	0xC8202FFC	0xC8202FF8	0xC8202FF4	0xC8202FF0
3	0	0xC820300C	0xC8203008	0xC8203004	0xC8203000
	1	0xC820301C	0xC8203018	0xC8203014	0xC8203010
	..	…….	…….	……. .	……. .
	255	0xC8203FFC	0xC8203FF8	0xC8203FF4	0xC8203FF0

2.9. Floating-point Unit

2.9.1. Overview

The features of the AM33/2.0 microcontroller core floating-point unit are listed below:

- Supports data types in compliance with the IEEE754 standard.
- Supports rounding to the nearest value in compliance with the IEEE754 standard.
- 32 single-precision floating-point registers
  (These registers can also be referenced as 16 double-precision floating-point registers.)
- Supports five floating-point operation exceptions in compliance with the IEEE754 standard and an unimplemented floating-point instruction exception.

The floating-point unit cannot be used when the EPSW.FE (FPU Enable) bit has been reset to "0." If a floating-point instruction is to be executed while the EPSW.FE bit has been reset to "0," an FPU disable exception is generated.

2.9.2. Data Format

2.9.2.1. Floating-point Format

2.9.2.1.1. Floating-point Numbers

Floating-point numbers format is expressed through the following three parameters:

(1) p : Number of bits in the mantissa (precision)
(2) Emax : Maximum exponent
(3) Emin : Minimum exponent

The floating-point number based on these three parameters would be as follows:

$(-1)^S 2^E (B_0.B_1B_2...B_{P-1})$

where:

and the following can be expressed:

s=0 or 1
$Emin \leq E \leq Emax$
b_1=0 or 1
Two infinite values: $+\infty$ and $-\infty$
At least one signaling NaN (sNaN)
At least one quiet NaN (qNaN)

In addition, a floating-point number that is expressed as follows:

$\pm 2^{Emin}(0.\ b_1b_2...b_{p-1})$

is called an denormalized number.

2.9.2.1.2. Floating-point Format

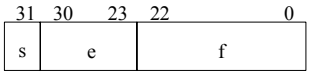
Single-precision floating-point format and double-precision floating-point format consist of the following three fields:

(1) s : One-bit sign
(2) e = E+bias : Exponent with bias
(3) f = b1b2...bp-1 : Mantissa

The range for an exponent with no bias is the entire range of integers between Emin and Emax:

(Emin≤E≤Emax)

Emin - 1 represents either "0" with a positive or negative sign (+/-0), or an denormalized number.
Emax + 1 represents either infinity with a positive or negative sign (+/-), or a NaN.(Not a Number)
In the floating-point unit for the AM33/2.00 microcontroller core, single-precision and double-precision floating-point numbers can be handled in either single-precision floating-point format or double-precision floating-point format, as shown below.



Single-precision floating-point format

Double-precision floating-point format

Figure 20 Floating-point format

The floating-point formats and parameters are listed below.

Parameter	Single-precision	Double-precision
p: Number of bits in the	24	53

mantissa (precision)		
Emax: Maximum exponent	+127	+1023
Emin: Minimum exponent	-126	-1022
bias: Exponent bias	+127	+1023
Fraction field	23	52
Exponent field	8	11
Total number of bits	32	64

2.9.2.1.3. Single-precision Floating-point Format

The floating-point value "v" according to single-precision floating-point format is determined as follows:

(1) e = 255 and f ≠ 0: "v" is a non-number (NaN), regardless of the sign "s."
(2) e = 255 and f = 0: "v" is $(-1)^s\infty$ (positive or negative infinity)
(3) 0 < e < 255: "v" is $(-1)^s 2^{e-127}(1.b_1b_2...b_{p-1})$ (normalized number)
(4) e = 0 and f ≠ 0: "v" is $(-1)^s 2^{e-126}(0.b_1b_2...b_{p-1})$ (denormalized number)
(5) e = 0 and f = 0: "v" is $(-1)^s 0$ (positive or negative zero)

The following chart shows different floating-point number values in single-precision floating-point format expressed in hexadecimal.

Type	Single-precision floating-point format
+∞ [positive infinity]	0x7F800000
-∞ [negative infinity]	0xFF800000
Positive regular numbers	0x00800000 - 0x7F7FFFFF
Negative regular numbers	0xFF7FFFFF - 0x80800000
Positive irregular numbers	0x00000001 - 0x007FFFFF
Negative irregular numbers	0x80000001 - 0x807FFFFF
+ 0 [positive zero]	0x00000000
- 0 [negative zero]	0x80000000
Signaling NaN (sNaN)	0x7FFFFFFF - 0x7fC00000 0xFFC00000 - 0xFFFFFFFF
Quiet NaN (qNaN)	0x7FBFFFFF - 0x7F800001 0xFF80001 - 0xFFBFFFFF

2.9.2.1.4. Double-precision Floating-point Format

The floating-point value "v" according to double-precision floating-point format is determined as follows:

(1) e = 2047 and f ≠ 0: "v" is a NaN, regardless of the sign "s."
(2) e = 2047 and f = 0: "v" is $(-1)^s\infty$ (positive or negative infinity)
(3) 0 < e < 2047: "v" is $(-1)^s 2^{e-1023}(1.b_1b_2...b_{p-1})$ (normalized number)
(4) e = 0 and f ≠ 0:"v" is $(-1)^s 2^{e-1022}(0.b_1b_2...b_{p-1})$ (denormalized number)
(5) e = 0 and f = 0: "v" is $(-1)^s 0$ (positive or negative zero)

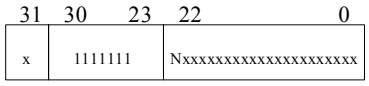
The following chart shows different floating-point number values in double-precision floating-point format expressed in hexadecimal.

Type	Double-point floating-point format
+∞ [positive infinity]	0x7FF0000000000000
-∞ [negative infinity]	0xFFF0000000000000
Positive normalized numbers	0x0010000000000000 - 0x7FEFFFFFFFFFFFFF
Negative normalized numbers	0xFFEFFFFFFFFFFFFF - 0x8010000000000000
Positive denormalized numbers	0x0000000000000001 - 0x000FFFFFFFFFFFFF
Negative denormalized numbers	0x8000000000000001 - 0x800FFFFFFFFFFFFF
+ 0 [positive zero]	0x0000000000000000
- 0 [negative zero]	0x8000000000000000

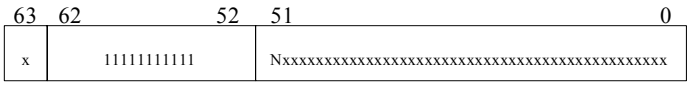
Signaling NaN (sNaN)	0x7FFFFFFFFFFFFFFF - 0x7FF8000000000000 0xFFF8000000000000 - 0xFFFFFFFFFFFFFFFF
Quiet NaN (qNaN)	0x7FBFFFFF ～ 0x7F800001 0xFFF0000000000001 ～ 0xFFF7FFFFFFFFFFF

2.9.2.2. NaN (Not-a-Number)

The bit pattern for NaN (Not a Number) is shown below.



NaN pattern in single-precision floating-point format



NaN pattern in double-precision floating-point format

N = 1: Signaling NaN (sNaN)
N = 0: Quiet NaN (qNaN)

Bit pattern of NaN: Not a Number

When each of the fields in floating-point format has the values shown below, the value is a NaN.

- Sign bit: 0 or 1
- Exponent field: All are "1"
- Fraction field: At least one bit is "1"

NaNs are further categorized by whether the MSB of the fraction field is a "1" or a "0." If the MSB is a "1," the value is a signaling NaN (sNaN); if the MSB is "0," the value is a quiet NaN (qNaN). In other words, when each of the fields in floating-point format has the values shown below, the value is a signaling NaN (sNaN).

- Sign bit: 0 or 1
- Exponent field: All are "1"
- Fraction field: MSB is "0" and at least one of the other bits is "1"

When each of the fields in floating-point format has the values shown below, the value is a quiet Nan (qNaN).

- Sign bit: 0 or 1
- Exponent field: All are "1"
- Fraction field: MSB is "1"

sNaN is input in an operation that generates floating-point values other than FMOV, FABS, and FNEG. If sNaN is input in these operations:

- The result of the operation becomes qNaN if the EE.V bit in the FPCR register is "0."
- An invalid operation exception (from among the floating-point operation exceptions) is generated if the EE.V bit in the FPCR register is "1." In this case, the contents of the operation destination register do not change.

If qNaN is input in an operation that generates floating-point values and sNaN is not input in that operation, the output is practically always qNaN, regardless of the setting of the EE.V bit in the FPCR register. In this case, no exception is generated.
For details on floating-point operations when a NaN is input, refer to the description of the individual instruction.
The qNaN value that the AM33/2.0 microcontroller core floating-point unit generates as an operation result is always as follows:

- Single-precision qNaN: 0x7FBFFFFF

The AM33/2.0 microcontroller core floating-point unit does not output sNaN as an operation result.

2.9.2.3. Denormalized Numbers

When each of the fields in floating-point format has the values shown below, the value is an denormalized number.

- Sign bit: 0 or 1
- Exponent field: All are "0"
- Fraction field: At least one bit is "1"

If an denormalized number is input as a source operand in an operation that generates floating-point values other than FMOV, FABS, and FNEG, the input is considered as the “0” . FMOV, FABS, and FNEG operations process the value as it is.
For details on floating-point operations when an denormalized number is input, refer to the description of the individual instruction.

2.9.3. Rounding

The floating-point unit in the AM33/2.0 microcontroller core supports a IEEE754-compliant rounding mode.
Rounding occurs when the final result of an operation is being calculated from the intermediate solution of the operation.
The rounding mode is defined by the RM filed in the FPCR register.

RM = 00: Round to nearest value
RM = 01: reserved
RM = 10: reserved
RM = 11: reserved

(The floating-point operation unit in the AM33/2.0 microcontroller core supports only rounding to the nearest value.)

- Rounding to the nearest value

Floating-point values are rounded to the nearest value that can be expressed by 32 bits for a single-precision value, and 64 bits for a double-precision value.
If there are two expressible values that are closest to the intermediate solution, the value is rounded so that the LSB is "0." (If the intermediate solution is equidistant between two expressible values, the intermediate value is rounded so that the LSB is "0.")

<Programming Note>

In floating-point operations, rounding occurs when generating the final result from an intermediate solution. Therefore, the results of a combination operation such as FMADD, FMSUB, FNMADD, or FNMSUB will differ from the result of the same combination operation that is performed by using multiple combinations of FADD, FSUB, and FMUL.

When executing FMADD through a combination of instructions, FMUL and FADD are used in combination. However, when FMADD is executed, rounding only occurs once; when FMUL and FADD are executed in combination, rounding occurs twice.

2.9.4. Exceptions

There are three exceptions that are related to the floating-point unit in the AM33/2.0 microcontroller core:

(1) FPU disable exception
(2) FPU unimplemented instruction exception
(3) FPU operation exception

For details on operation when a floating-point unit exception is generated, refer to page 125, 2.6 Interrupt System.

2.9.4.1. FPU Disable Exception

This exception is generated if a floating-point operation instruction is to be executed while the EPSW.FE (FPU enable) bit is "0."

2.9.4.2. FPU Unimplemented Instruction Exception

This exception is generated if an unimplemented floating-point operation instruction is to be executed while the EPSW.FE bit is "1."
The following floating-point operation instructions are unimplemented for the floating-point operation unit in the AM33/2.0 microcontroller core:

Type	Mnemonic	Function
Single-precision floating-point operations	FSQRT	Floating-point square root
Double-precision floating-point operations	FMOV	TRANSFER BETWEEN FLOATING-POINT REGISTERS
	FABS	Floating-point absolute value operation
	FNEG	Floating-point sign reversal
	FSQRT	Floating-point square root
	FRSQRT	Floating-point reciprocal square root
	FADD	Floating-point addition
	FSUB	Floating-point subtraction
	FMUL	Floating-point multiplication
	FDIV	Floating-point division
Type conversion	FTOI	Single-precision -> integer conversion
	ITOF	Integer -> single-precision conversion
	DTOF	Double-precision -> single-precision conversion
	FTOD	single-precision -> double-precision conversion

2.9.4.3. FPU Operation Exceptions

2.9.4.3.1. Exception Causes

There are five causes of FPU operation exceptions:

- I: Inexact
  When the result of an operation is inexact due to overflow, underflow, or rounding

- U: Underflow
  When an underflow occurs in an operation result
- O: Overflow
  When an overflow occurs in an operation result
- Z: Zero divide
  When a floating-point division operation was executed with "0" as the divisor
- V: Invalid Operand
  When an operation was attempted with an invalid input such as NaN

The FPCR EF (FPU exception flags) field, EE (FPU exception enable) field, and EC (FPU exception cause) field each include bits that correspond to I, U, O, Z, and V above.
If an exception which was caused by an FPU operation exception is generated due to the execution of a floating-point operation instruction, the corresponding bit in the FPCR EC field is set to "1" and a "1" is also stored in the corresponding bit in the FPCR EF field.
If no FPU exception cause is generated during the execution of a floating-point operation instruction, the corresponding bit in the FPCR EC field is reset to "0" and the corresponding bit in the FPCR EF field is not changed.
For details on FPU exception causes, refer to the descriptions of the individual functions.

2.9.4.3.2. FPU Exception Enable

FPU exception processing can be enabled/disabled for each individual FPU exception cause.
Exception processing can be enabled for each FPU exception cause by setting the corresponding bit in the FPCR EE field to "1."

- I: Inexact
  When FPCR.EE.I = 1 and the operation result is inexact
- U: Underflow
  When FPCR.EE.U = 1 and an underflow was generated in the operation result
- O: Overflow
  When FPCR.EE.O = 1 and an overflow was generated in the operation result
- Z: Zero divide
  When FPCR.EE.Z = 1 and a floating-point division operation was executed with “0” as the divisor
- V: Invalid operand
  When FPCR.EE.V = 1 and an operation was attempted with an invalid input such as NaN

An FPU exception is generated in all of the above cases.

<Programming Note>
If an FPU exception is generated, the cause of the exception can be determined through software by reading the contents of FPCR and then interpreting the information contained within.

Even if an FPU exception occurs due to any of these causes, the contents of the destination register are not changed.
Exception processing for FPU exceptions due to specific individual causes can be disabled by setting the appropriate bit in the FPCR EE field to "0."
If exception processing for an FPU exception has been disabled, the processing described below is performed when the corresponding cause is generated. No FPU exception is generated.

- I: Inexact
  When FPCR.EE.I = 0 and the operation result is inexact
  -> The inexact value is generated as the result.
- U: Underflow
  When FPCR.EE.U = 0 and an underflow was generated in the operation result
  -> "0" with the same sign as before rounding is generated.
- O: Overflow
  When FPCR.EE.O = 0 and an overflow was generated in the operation result
  -> Infinity with the same sign as before rounding is generated.
- Z: Zero divide

When FPCR.EE.Z = 0 and a floating-point division operation was executed with “0” as the divisor
-> Infinity with the same sign as before rounding is generated.

- V: Invalid operand
  When FPCR.EE.V = 0 and an operation was attempted with an invalid input such as NaN
  -> qNaN is generated as the result.

Chapter 3 Operating Modes

3

CHAPTER 3

Operating Modes

3.1. General

There are three operating modes as listed below. In order to reduce power consumption, these modes control whether the oscillator is running or stopped, and whether the CPU or peripheral circuits are running or stopped.

Operating Modes

1. Reset mode (RESET)
2. Normal operating mode (NORMAL)
3. Low power consumption modes
   - Stop mode (STOP)
   - Halt mode (HALT)
   - Sleep mode (SLEEP)

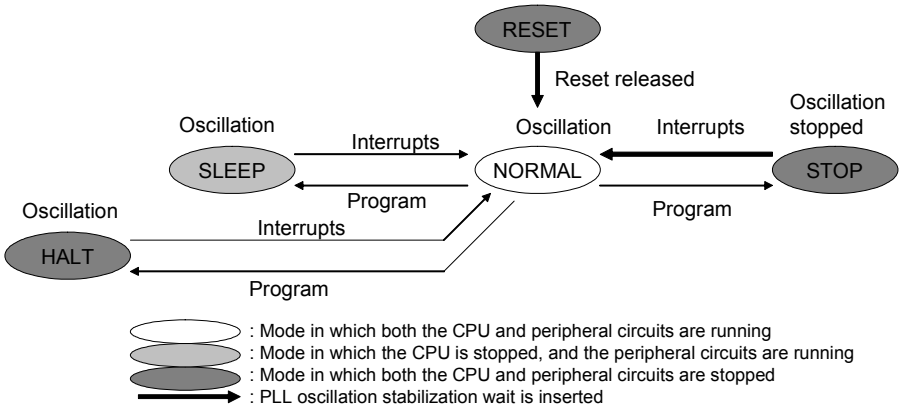


Figure 21 State Transition Diagram

3.2. Low power consumption modes

Power consumption is reduced by stopping the oscillation of the oscillator and the clock generator (CKG) and by stopping the clock that is supplied to the CPU and the peripheral circuits. There are three low power consumption modes. The microcontroller can be put into any of these modes by software.

Stop mode (STOP)

In this mode, the oscillation of the oscillator and the PLL is stopped.

After the oscillator and the PLL has begun running and then its oscillation has stabilized, the microcontroller then shifts to the normal operating mode (NORMAL).

Halt mode (HALT)

In this mode, although the oscillator and the PLL are oscillating, the clock is not supplied to the CPU or the peripheral circuits, and the CPU and peripheral circuits are not running.

When an interrupt is received during HALT mode, the microcontroller immediately shifts to the normal operating mode (NORMAL).10

Sleep mode (SLEEP)

In this mode, although the oscillator and the PLL are oscillating, the clock is not supplied to the CPU, the CPU is not running, and the peripheral circuits are running.

When an interrupt is received during SLEEP mode, the microcontroller immediately shifts to the normal operating mode (NORMAL).

The operations of the peripheral circuits in the SLEEP, HALT, and STOP modes are shown in the table below.

Table 52 Operating Modes of Internal Blocks

	SLEEP	HALT	STOP
CPU core	Stopped	Stopped	Stopped
Clock generator (CKG)	Running	Running	Stopped
Bus controller (BCU)	Running	Stopped	Stopped
System bus controller	Running	Stopped	Stopped
Memory bus controller	Running	Stopped	Stopped
Interrupt controller	Running	Running	Running
8-bit timer	Running	Stopped	Stopped
16-bit timer	Running	Stopped	Stopped
Watchdog timer	Running	Stopped	Stopped
Serial interface	Running	Stopped	Stopped
Real-time clock	Running	Running	Running
A/D converter	Running	Stopped	Stopped
IrDA	Running	Stopped	Stopped
I2C	Running	Stopped	Stopped
External resonator oscillation cell	Running	Running	Stopped

3.3. Oscillation Stabilization Wait Operation

When recovering from the reset state or from STOP mode to the normal operating mode (NORMAL), the watchdog timer operates as an oscillation stabilization wait timer.
While waiting for oscillation stabilization after a reset, the watchdog timer operates as an 18-bit binary counter. The relationship between the oscillation stabilization wait time t_{OSCW} and the oscillating frequency f_{OSC} [MHz] is as follows:

$t_{OSCW} = 2^n/(f_{OSC} \times 10^3)$[ms] (n = 18)

In other words, when f_{OSC} = 25[MHz], t_{OSCW} = 10.485[ms].
Once the oscillation stabilization wait time has elapsed after a reset, and then the internal reset is released and the microcontroller returns to the normal operating mode.
While waiting for oscillation stabilization after returning from STOP mode, the watchdog timer functions as a binary counter of the number of bits needed for the value that was set in the watchdog timer when the microcontroller entered STOP mode.

Chapter 4 Clock generator (CKG)

4

4.1. General

The clock generator (CKG) has a built-in PLL circuit, and supplies frequencies that are multiple of the input clock in this microcontroller. The clock generator also supplies a clock pulse that has the equal or half frequencies as compared with the oscillation frequencies to external devices.

4.2. Features

The features of the clock generator are described below:

- Flexible clock control
- Supports self-excited oscillation.
- Supplies a twofold or fourfold input frequency (FRQS-pin setting) as the CPU clock (MCLK).
- Supplies a pulse that is 1/2 of MCLK as the internal I/O bus clock (IOBCLK).
- Supplies a pulse that is 1/4 of MCLK as the peripheral clock (IOCLK) and the system bus clock (SYSCLK).
- Supplies a pulse that is 1/1 of MCLK as the memory bus clock (SDCLK).
- Supplies a pulse that is 1/1 of MCLK as the asynchronous mode system bus clock (ASYSCLK).
- Phase synchronous clock generation
- Generates the IOBCLK, IOCLK, SDCLK, SYSCLK, and ASYSCLK as clocks that are phase synchronous with the CPU clock (MCLK).

4.3. Block Diagram

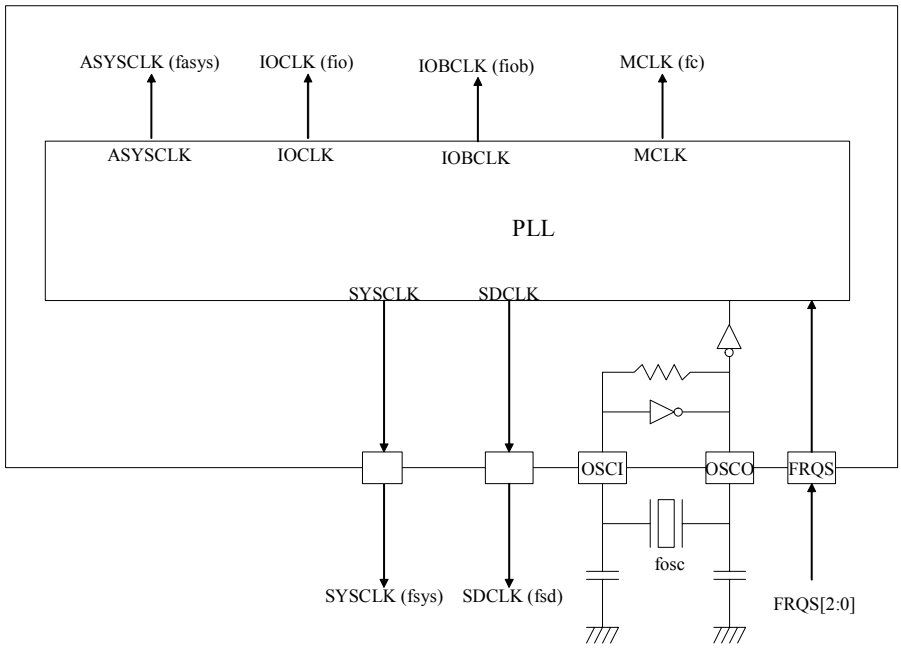


Figure 22 Clock Generator Block Diagram

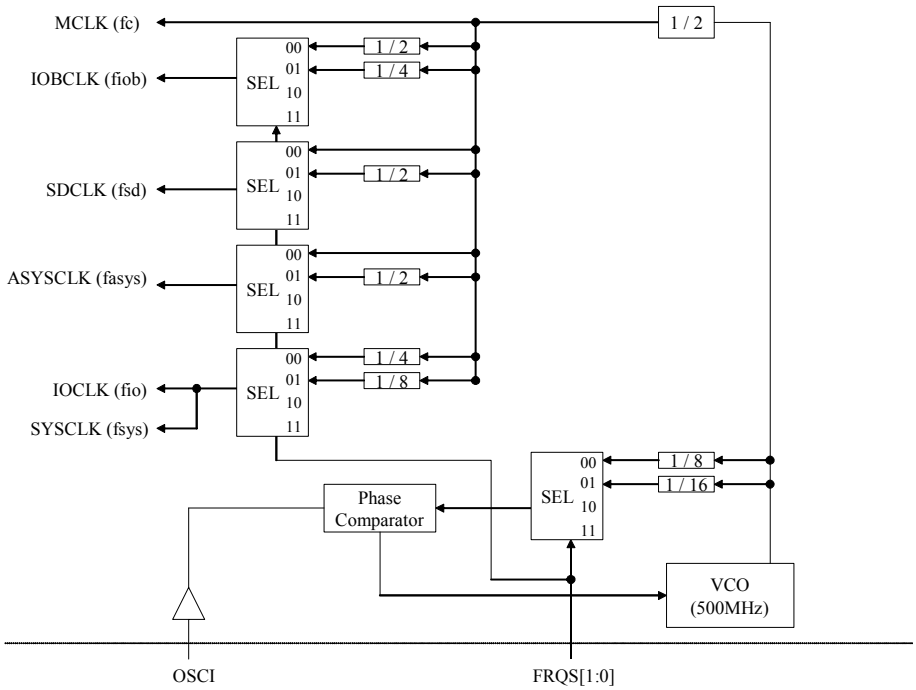


Figure 23 PLL Block Diagram

4.4. Description of Operation

4.4.1. Input Frequency

The input frequency range for the clock generator is 25 to 40MHz. An operation at setting the input frequency beyond the CPU operation frequency is not guaranteed. Moreover, the following conditions need to be satisfied.
The eightfold input frequency at FRQS[1:0]=00 and four input frequency at FRQS[1:0]=01 need to be set within the oscillation range, 115 to 270MHz.
Accordingly, the input frequency (fosci) and CPU operation frequency (fc) are as follows:
When FRQS[1:0]=00,

- Input frequency (fosci): 25MHz to 33.33MHz
- MCLK: 100MHz to 133MHz

When FRQS[1:0]=01,

- Input frequency (fosci): 28.75MHz to 40MHz
- MCLK: 57.5MHz to 80MHz

Use the table below as a reference, and use the oscillator with the frequency in accordance with the internal operating frequency.

4.4.2. Clock Supply

The relationship between the setting of the FRQS pins and the multipliers (versus the input frequency fosci) for the various clocks (SYSCLK, MCLK, IOBCLK, IOCLK, SDCLK, and ASYSCLK) is shown in the table below.

Table 53 Relationship between the FRQS Mode Pins and the Supply Clock Frequency

FRQS[1:0]	OSCI (fosc)	MCLK (fc)	IOBCLK (fiob)	IOCLK (fio)	SDCLK (fsd)	SYSCLK (fsys)	ASYSCLK (fasys)
00	Fosci	Fosci x 4	fc / 2	fc / 4	fc	fc / 4	fc
01	fosci	Fosci x 2	fc / 2	fc / 4	fc	fc / 4	fc
10	Setting prohibited						
11	Setting prohibited						

The following table shows the frequency of each clock at inputting fosci=30MHz.

Table 2 Frequency of each clock at the operation of the CPU 120MHz

FRQS[1:0]	OSCI (fosci)	MCLK (fc)	IOBCLK (fiob)	IOCLK (fio)	SDCLK (fsd)	SYSCLK (fsys)	ASYSCLK (fasys)
00	30	120	60	30	120	30	120
01	30	60	30	15	60	15	60
10	Setting prohibited						
11	Setting prohibited						

4.4.3. Cautions

FRQS[1:0] = 10 and 11 in the clock mode (FRQS inputs) settings are prohibited; operation is not guaranteed if these settings are used.
An operation at inputting OSCI beyond the guaranteed operation frequency of the CPU clock (MCLK) is not guaranteed.

Chapter 5 Bus controller (BCU)

5

CHAPTER 5

Bus controller (BCU)

5.1. General

The bus controller handles access arbitration and data transfer for the four slave buses (EX bus, OpEX bus, OCM bus, and IO bus) for the three master devices (CPU instructions, CPU data, and DMA).

5.2. Features

- Arbitration
- Programmable slave bus address allocation
- Timeout error detection
- Retains the sources and error access addresses.
- Support for parallel access between multiple buses.

5.3. Configuration

The diagram below shows the internal bus configuration for this LSI. The bus controller handles access to the four slave buses (EX bus, OpEX bus, IO bus, and OCM bus) for the three master buses (CI bus, CD bus, and DMA bus), and handles arbitration control on the DMA bus between the DMAC on the DMA bus and the EMC. Each master device can access different slave devices simultaneously. The bus controller performs arbitration for conflicting accesses to the same slave through the arbitration method which is set in the bus controller control register (BCCR).

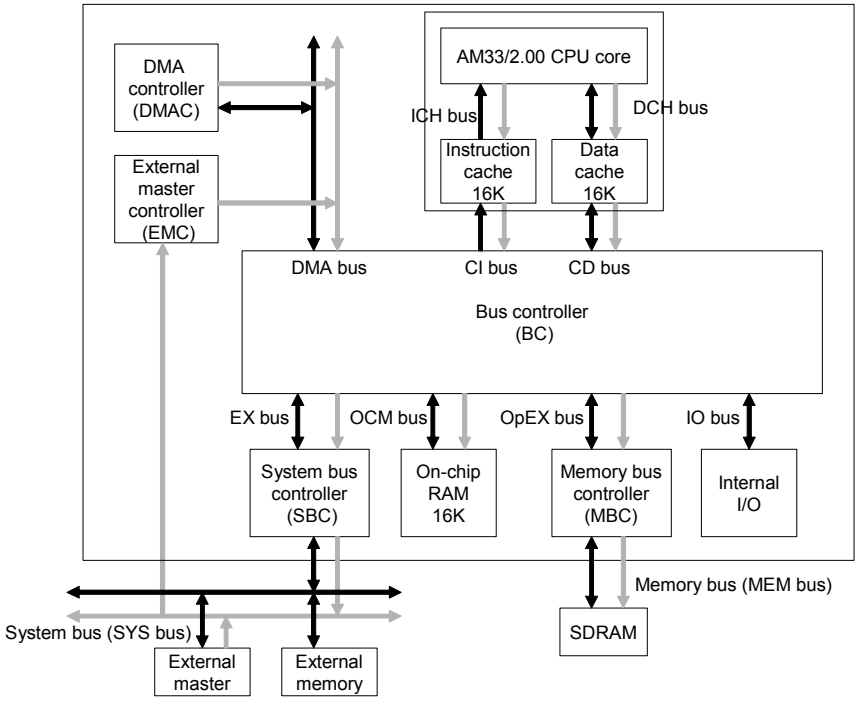


Figure 24 Internal Bus Configuration

The following diagram shows the connections between the master devices and the slave devices in the bus controller. The CPU instruction master cannot access the on-chip I/O bus.

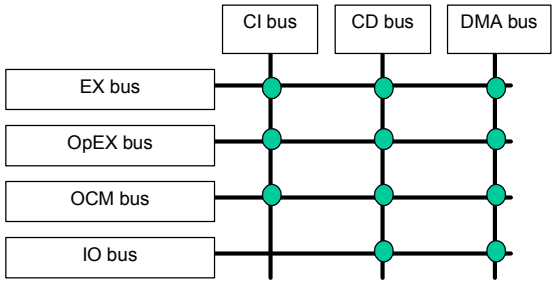


Figure 25 Connection Relationships between the Master Bus and Slave Bus

Table 54 Characteristics of Each Bus

Bus Name	Block	Bus Width	Operation Clock
CI bus	Instruction cache - BCU	32	MCLK
CD bus	Data cache - BCU	32	MCLK
DMA bus	DMA/EMC - BCU	32	MCLK
EX bus	BCU - System bus controller	32	MCLK
OCM bus	BCU – On-chip memory	32	MCLK
OpEX bus	BCU - Memory bus controller	32	MCLK
IO bus	BCU - Internal peripheral host interface	32	IOBCLK
System bus	System bus controller - External memory/external device	16/32	SYSCLK (synchronous mode) ASYSCLK (asynchronous mode)
Memory bus	Memory bus controller - External memory	16	SDCLK

5.4. Description of Registers

Table 55 Description of Bus Controller Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xC0002000	BCCR	Bus controller control register	32	0x12040000	8,16,32
0xC0002000	-	System reserve	-	-	-
0xC0002010	BCBERR	Bus error source register	32	0x00000000	8,16,32
0xC0002020	BCBEAR	Bus error address register	32	Undefined	8,16,32
0xC0002030	-	System reserve	-	-	-
0xC0002034	-	System reserve	-	-	-
0xC0002038	-	System reserve	-	-	-
0xC0002040	-	System reserve	-	-	-

5.4.1. Bus controller control register

Register symbol: BCCR
Address: 0x C0002000
Purpose: This register allocates individual address blocks to slave devices.

This register also sets the arbitration priority among master devices.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved			TMOE	reserved		TMON	
Initial value	000			1	00		10	
R/W	R			RW	R		RW	
Bit	23	22	21	20	19	18	17	16
Bit name	BEPRI		reserved			API		
Initial value	00		000			100		
R/W	RW		R			RW		
Bit	15	14	13	12	11	10	9	8
Bit name	B7AD[1:0]		B6AD[1:0]		B5AD[1:0]		B4AD[1:0]	
Initial value	00		00		00		00	
R/W	RW		RW		RW		RW	
Bit	7	6	5	4	3	2	1	0
Bit name	B3AD[1:0]		B2AD[1:0]		B1AD[1:0]		B0AD[1:0]	
Initial value	00		00		00		00	
R/W	RW		RW		RW		RW	

Bit	Bit name	Description
31-29	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
28	TMOE	Timeout detection enable 0 : Disabled 1 : Enabled
27-26	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
25-24	TMON	Timeout value setting 00 : 16 IOCLK cycles 01 : 256 IOCLK cycles 10 : 4096 IOCLK cycles 11 : 65536 IOCLK cycles
23-22	BEPRI	Bus error address priority setting If multiple bus errors occur simultaneously, this field sets the priority among the masters for which error address is to be saved. 00 : DMA, CI, CD 01 : DMA, CD, CI 10 : CI, CD, DMA 11 : CD, CI, DMA
21-19	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
18-16	API	Sets the bus arbitration priority. The followings show the highest priority is on the left. 000 : DMA, CI, CD 001 : DMA, CD, CI 010 : CI, CD, DMA 011 : CD, CI, DMA 100 : Round robin 101 –111 : Setting prohibited
15-14	B7AD[1:0]	Sets the bus allocation for block 7 (0x9C000000-0x9FFFFFFF). 00 : EX BUS 01 : OpEX bus 10 : OCM bus

Bit	Bit name	Description
		11 : Setting prohibited
13-12	B6AD[1:0]	Sets the bus allocation for block 6 (0x98000000-0x9BFFFFFF). 00 : EX BUS 01 : OpEX bus 10 : OCM bus 11 : Setting prohibited
11-10	B5AD[1:0]	Sets the bus allocation for block 5 (0x94000000-0x97FFFFFF). 00 : EX BUS 01 : OpEX bus 10 : OCM bus 11 : Setting prohibited
9-8	B4AD[1:0]	Sets the bus allocation for block 4 (0x90000000-0x93FFFFFF). 00 : EX BUS 01 : OpEX bus 10 : OCM bus 11 : Setting prohibited
7-6	B3AD[1:0]	Sets the bus allocation for block 3 (0x8C000000-0x8FFFFFFF). 00 : EX bus 01 : OpEX bus 10 : OCM bus 11 : Setting prohibited
5-4	B2AD[1:0]	Sets the bus allocation for block 2 (0x88000000-0x8BFFFFFF). 00 : EX bus 01 : OpEX bus 10 : OCM bus 11 : Setting prohibited
3-2	B1AD[1:0]	Sets the bus allocation for block 1 (0x84000000-0x87FFFFFF). 00 : EX BUS 01 : OpEX bus 10 : OCM bus 11 : Setting prohibited
1-0	B0AD[1:0]	Sets the bus allocation for block 0 (0x80000000-0x83FFFFFF). 00 : EX bus 01 : OpEX bus 10 : OCM bus 11 : Setting prohibited

5.4.2. Bus error source register

Register symbol: BCBERR
Address: 0xC0002010
Purpose: This register indicates information concerning the bus cycle in which an error occurred.
Once information on an error has been stored, the information on a new error cannot be stored until a "0" is written to the BEMR[2:0] field of the BCBERR register.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16

Bit name	reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	-	BEMR[2:0]			BEME	BEBST	BESD	BERW
Initial value	0	0			0	0	0	0
R/W	R	RW			R	R	R	R
Bit	7	6	5	4	3	2	1	0
Bit name	reserved			BESB[4:0]				
Initial value	0			0				
R/W	R			R				

Bit	Bit name	Description
31-15	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
14-12	BEMR[2:0]	This field indicates the master bus that caused the error. These are writable bits. An error information is stored only when these bits are “0”. 000 : No error has occurred. 001 : CI bus master 010 : CD BUS MASTER 100 : DMA bus master
11	BEME	Multi-bus error bit This flag indicates that multiple errors occurred. If a new bus error occurs while there is already error information stored in this register (i.e., BEMR[2:0] is not "0"), the BEME bit is set to "1." Once this bit is set to "1," it is not cleared to "0" until "0" is written to BEMR[2:0].
10	BEBST	This field indicates the type of access in which the bus error occurred. 0 : Single access 1 : Burst access
9	BESD	This field indicates the device that detected the bus error. 0 : BCU 1 : Slave bus
8	BERW	This field indicates whether the access in which the bus error occurred was a read or a write. 0 : WRITE 1 : Read
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4-0	BESB[4:0]	This field indicates the destination of the access in which the bus error occurred. 00001 : Monitor space (0xE0000000-0xFFFFFFFF) 00010 : IO bus 00100 : EX bus 01000 : OpEX bus 10000 : OCM

5.4.3. Bus error address register

Register symbol: BCBEAR
Address: 0xC0002020

Purpose: This register indicates the address of the bus cycle in which the error occurred.
Once an error address is stored in this register, a new error can be recorded in this register until "0" is written to the BCBERR register.

Bit	31	30	29	28	27	26	25	24
Bit name	BEA[31:24]							
Initial value	Undefined							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	BEA[23:16]							
Initial value	Undefined							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	BEA[15:8]							
Initial value	Undefined							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	BEA[7:0]							
Initial value	Undefined							
R/W	R							

Bit	Bit name	Description
31-0	BEA[31:0]	Bus error address This field indicates the address of the access that was the source of the bus error. This register stores the address of an error that occurred while the BEMR[2:0] field in the BCBERR register was "0." If the error occurs in the state that the BEMR[2:0] field is not "0" (indicating that a error has previously occurred), the address of the error is not stored in this register.

5.5. Description of Operation

5.5.1. Bus Error Detection

If an access to a slave bus does not end even beyond the timeout setting value, the BCU reports a bus error to the master side. When a bus error occurs, the address of the access that was the cause of the bus error is stored in the bus error address register, and the source is stored in the bus error source register. After the values are set in these registers, they are retained until the software writes the bus error source register.

5.5.2. Burst Transfer

The CI bus, the CD bus, and the DMA bus can issue a request for a 16-byte burst transfer. The transfer sequence for 16-byte burst transfers from the CI and the CD buses is always in wraparound mode.

5.5.3. Bus Lock

When the CPU has executed a BSET or BCLR instruction, the master that accessed a slave bus locks up the slave bus, and all other masters are prohibited from accessing the bus. If the BCU requests bus lock for the EX bus, the system bus controller access implements the bus lock by not releasing the bus to the bus request from the external bus master devices.

5.5.4. Write Buffer

The BCU has a write buffer for each slave bus, and returns an acknowledge signal in response to an access from a master bus before the write access on the slave bus is completed. Accordingly, if it is necessary to confirm the completion of a write, a read access must be performed to memory on the same slave bus.

5.6. Memory Space

The physical addresses that are generated by the AM33/2.00 CPU core are partitioned into 8 blocks of 64MB by the BCU. Each block is allocated to a physical memory address (BCU slave bus address) according to the settings in the BCCR register. The physical addresses, 0x20000000 to 0x3FFFFFFF and 0xC0000000 to 0xDFFFFFFF are allocated to internal I/O. The relationship between the physical addresses and the physical memory addresses in each block is shown in the following table.

Table 56 Block Partitions in the Physical Address Space and Allocation to Physical Memory Addresses

Block	Physical addresses	Physical Memory Addresses
Block 0	0X40000000 - 0X43FFFFFF	0x80000000 - 0x83FFFFFF
	0x60000000 - 0x63FFFFFF	
	0x80000000 - 0x83FFFFFF	
	0xA0000000 - 0xA3FFFFFF	
Block 1	0x44000000 - 0x47FFFFFF	0x84000000 - 0x87FFFFFF
	0x64000000 - 0x67FFFFFF	
	0x84000000 - 0x87FFFFFF	
	0xA4000000 - 0xA7FFFFFF	
Block 2	0x48000000 - 0x4BFFFFFF	0x88000000 - 0x8BFFFFFF
	0x68000000 - 0x6BFFFFFF	
	0x88000000 - 0x8BFFFFFF	
	0xA8000000 - 0xABFFFFFF	
Block 3	0x4C000000 - 0x4FFFFFFF	0x8C000000 - 0x8FFFFFFF
	0x6C000000 - 0x6FFFFFFF	
	0x8C000000 - 0x8FFFFFFF	
	0xAC000000 - 0xAFFFFFFF	
Block 4	0x00000000 - 0x03FFFFFF	0x90000000 - 0x93FFFFFF
	0x10000000 - 0x13FFFFFF	
	0x50000000 - 0x53FFFFFF	
	0x70000000 - 0x73FFFFFF	
	0x90000000 - 0x93FFFFFF	
	0xB0000000 - 0xB3FFFFF	
Block 5	0x04000000 - 0x07FFFFFF	0x94000000 - 0x97FFFFFF
	0x14000000 - 0x17FFFFFF	
	0x54000000 - 0x57FFFFFF	
	0x74000000 - 0x77FFFFFF	
	0x94000000 - 0x97FFFFFF	
	0xB4000000 - 0xB7FFFFF	
Block 6	0x08000000 - 0x0BFFFFFF	0x98000000 - 0x9BFFFFFF
	0x18000000 - 0x1BFFFFFF	
	0x58000000 - 0x5BFFFFFF	
	0x78000000 - 0x7BFFFFFF	

	0x98000000 - 0x9BFFFFFF	
	0xB8000000 - 0xBBFFFFF	
Block 7	0x0C000000 - 0x0FFFFFFF	0x9C000000 - 0x9FFFFFFF
	0x1C000000 - 0x1FFFFFFF	
	0x5C000000 - 0x5FFFFFFF	
	0x7C000000 - 0x7FFFFFFF	
	0x9C000000 - 0x9FFFFFFF	
	0xBC000000 - 0xBFFFFFFF	

Chapter 6 On-Chip RAM (OCR)

6

CHAPTER 6

On-Chip RAM (OCR)

6.1. General

This LSI has 16KB of on-chip instruction and data RAM. This on-chip RAM is connected to the CPU core through the DMA controller and the bus controller by a 32-bit bus.
Physical addresses in on-chip RAM are allocated to blocks specified by the bus controller's BCCR register. It is possible to use an instruction cache or a data cache by accessing the allocated physical addresses from a cacheable address space.
On-chip RAM can also be accessed from an external bus master device on the system bus.

6.2. Features

On-chip memory has the following features:

- On-chip instruction/data RAM
  - Size 16K bytes
  - Bus Width 32 bits
  - Access size 1-/2-/4-byte single access, 16-byte burst access
  - Access cycle (when there is no wait for bus arbitration)
    - During a single read : 4 MCLK cycles
    - During a single write : 1 MCLK cycle (when using the write buffer)
    - During a 16-byte burst read : 7 MCLK cycles
    - During a 16-byte burst write : 4 MCLK cycles (when using the write buffer)

Chapter 7 System Bus Controller (SBC)

7

7.1. General

The System Bus Controller (SBC) is capable of connecting directly to ROM, SRAM, burst ROM, peripheral LSIs, etc., without any external circuitry through eight chip selections.

7.2. Features

- Supports synchronous mode (synchronized with SYSCLK) and asynchronous mode (synchronized with ASYSCLK).
- Controls the external memory space by partitioning it into 8 banks.
- Outputs a separate chip select signal for each bank.
- Permits setting the bus width to either 16 or 32 bits for each bank.
- Banks 0 through 7 can be allocated to SRAM, ROM, flash RAM, and burst ROM.
- Burst ROM Interface
  Performs a 16-byte burst transfer.

7.3. Description of Registers

Address:	Symbol	Name	NUMBER OF BITS	Initial value	Access size
0xD8C00100	SBBASE0	Base address register 0	32	Note 1	8,16,32
0xD8C00110	SBBASE1	Base address register 1	32	Note 1	8,16,32
0xD8C00120	SBBASE2	Base address register 2	32	0x00000000	8,16,32
0xD8C00130	SBBASE3	Base address register 3	32	0x00000000	8,16,32
0xD8C00140	SBBASE4	Base address register 4	32	0x00000000	8,16,32
0xD8C00150	SBBASE5	Base address register 5	32	0x00000000	8,16,32
0xD8C00160	SBBASE6	Base address register 6	32	0x00000000	8,16,32
0xD8C00170	SBBASE7	Base address register 7	32	0x00000000	8,16,32
0xD8C00200	SBCTRL00	BANK CONTROL REGISTER 00	32	0x22100000	8,16,32
0xD8C00204	SBCTRL01	Bank control register 01	32	0x00001100	8,16,32
0xD8C00208	SBCTRL02	Bank control register 02	32	Note 2	8,16,32
0xD8C00210	SBCTRL10	Bank control register 10	32	0x22100000	8,16,32
0xD8C00214	SBCTRL11	Bank control register 11	32	0x00001100	8,16,32
0xD8C00218	SBCTRL12	BANK CONTROL REGISTER 12	32	Note 2	8,16,32
0xD8C00220	SBCTRL20	Bank control register 20	32	0x22100000	8,16,32
0xD8C00224	SBCTRL21	Bank control register 21	32	0x00001100	8,16,32
0xD8C00228	SBCTRL22	Bank control register 22	32	0x0000000F	8,16,32
0xD8C00230	SBCTRL30	Bank control register 30	32	0x22100000	8,16,32
0xD8C00234	SBCTRL31	Bank control register 31	32	0x00001100	8,16,32
0xD8C00238	SBCTRL32	Bank control register 32	32	0x0000000F	8,16,32
0xD8C00240	SBCTRL40	Bank control register 40	32	0x22100000	8,16,32
0xD8C00244	SBCTRL41	Bank control register 41	32	0x00001100	8,16,32
0xD8C00248	SBCTRL42	Bank control register 42	32	0x0000000F	8,16,32
0xD8C00250	SBCTRL50	Bank control register 50	32	0x22100000	8,16,32
0xD8C00254	SBCTRL51	Bank control register 51	32	0x00001100	8,16,32
0xD8C00258	SBCTRL52	Bank control register 52	32	0x0000000F	8,16,32
0xD8C00260	SBCTRL60	Bank control register 60	32	0x22100000	8,16,32
0xD8C00264	SBCTRL61	Bank control register 61	32	0x00001100	8,16,32
0xD8C00268	SBCTRL62	Bank control register 62	32	0x0000000F	8,16,32
0xD8C00270	SBCTRL70	Bank control register 70	32	0x22100000	8,16,32
0xD8C00274	SBCTRL71	Bank control register 71	32	0x00001100	8,16,32
0xD8C00278	SBCTRL72	Bank control register 72	32	0x0000000F	8,16,32

Note 1: For the initial values, refer to page 209, 7.3.1 Base Address Register.
Note 2: For the initial values, refer to page 210, 7.3.2 Bank control register 0.

7.3.1. Base Address Register

Register symbol: SBBASEn (n= 0 – 7)
Address: 0xD8C001n0 (SBBASEn)
PURPOSE: This register specifies the range of addresses that can be allocated to bank n (XSCSn).
This register enables comparison with the specified address range.

Bit	31	30	29	28	27	26	25	24
BIT NAME	BBA[31:24]							
Initial value	0							
R/W	RW							
Bit	23	22	21	20	19	18	17	16
BIT NAME	BBA[23:17]							reserved
Initial value	0							0
R/W	RW							R
Bit	15	14	13	12	11	10	9	8
BIT NAME	BAM[31:24]							
Initial value	0							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
BIT NAME	BAM[23:17]							BE
Initial value	0							Note
R/W	RW							RW

Note : Refer to the descriptions below for the initial values.

Bit	Bit name	Description
31-17	BBA[31:17]	Bank Base Address This field (BBA[31:17]) sets the base address for the addresses that are allocated to the bank in question.
16	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
15-1	BAM[31:17]	Bank Address Comparison Mask Setting This field sets the comparison mask for the address that was accessed and the address that was specified in BBA[31:17].
0	BE	Bank Enable This bit enables/disables the generation of the chip select signal for the bank in question. 0 : Disabled 1 : Enabled The initial value for this bit is explained below: SBBASE0: When BOOTSEL = H, 1; when BOOTSEL = L, 0 SBBASE1: When BOOTSEL = H, 0; when BOOTSEL = L, 1 SBBASE2 - SBBASE7: Always 0

7.3.2. Bank control register 0

Register symbol: SBCTRLn0 (n= 0 – 7)
Address: 0xD8C002n0 (SBCTRLn0)
Purpose: THIS REGISTER SETS THE CONTROL MODE FOR BANK N (XSCSN)

Bit	31	30	29	28	27	26	25	24
BIT NAME	ADH[3:0]				DAH[3:0]			
Initial value	2				2			
R/W	RW				RW			

Bit	23	22	21	20	19	18	17	16
BIT NAME	CSH[3:0]				RWH[3:0]			
Initial value	1				0			
R/W	RW				RW			
Bit	15	14	13	12	11	10	9	8
BIT NAME	REH[3:0]				WEH[3:0]			
Initial value	0				0			
R/W	RW				RW			
Bit	7	6	5	4	3	2	1	0
BIT NAME	reserved				RESERVED			
Initial value	0				0			
R/W	R				RW			

Bit	Bit name	Description
31-28	ADH[3:0]	Address hold This field sets the number of cycles that the address (SA) is held after the count that is set in the WC parameter (SBCTRLn2) (in fixed wait mode), or after XSDK is asserted (in handshake mode).
27-24	DAH[3:0]	Data hold This field sets the number of cycles that the data (SD) is held after the count that is set in the WC parameter (SBCTRLn2) (in fixed wait mode), or after XSDK is asserted (in handshake mode).
23-20	CSH[3:0]	Chip select hold This field sets the number of cycles that the chip select signal (XSCSn) is held after the count that is set in the WC parameter (SBCTRLn2) (in fixed wait mode), or after XSDK is asserted (in handshake mode).
19-14	RWH[3:0]	SRW signal hold This field sets the number of cycles from the point when the read/write status signal (SRXW) is asserted until the access is completed.
15-12	REH[3:0]	Read enable hold This field sets the number of cycles that the read enable signal (XSRE) is held after the count that is set in the WC parameter (SBCTRLn2) (in fixed wait mode), or after XSDK is asserted (in handshake mode).
11-8	WEH[3:0]	Write enable hold This field sets the number of cycles that the write enable signal (XSWE) is held from the falling edge of the clock after the count that is set in the WC parameter (SBCTRLn2) (in fixed wait mode), or after XSDK is asserted (in handshake mode).
7-0	reserved	THESE ARE RESERVED BITS. "0" IS ALWAYS RETURNED WHEN THESE BITS ARE READ. ALWAYS WRITE A "0" TO THESE BITS.

There are limit conditions for setting.

7.3.3. Bank control register 1

Register symbol: SBCTRLn1 (n= 0 – 7)
Address: 0xD8C002n4 (SBCTRLn1)
Purpose: This register sets the control mode for bank n (XSCSn)

Bit	31	30	29	28	27	26	25	24
Bit name	ASD[3:0]				CSD[3:0]			
Initial value	0				0			
R/W	RW				RW			
Bit	23	22	21	20	19	18	17	16
Bit name	ASW[3:0]				RWD[3:0]			
Initial value	0				0			
R/W	RW				RW			
Bit	15	14	13	12	11	10	9	8
Bit name	RED[3:0]				WED[3:0]			
Initial value	1				1			
R/W	RW				RW			
Bit	7	6	5	4	3	2	1	0
Bit name	reserved				reserved			
Initial value	0				0			
R/W	R				R			

Bit	Bit name	Description
31-28	ASD[3:0]	Address strobe delay This field sets the number of cycles until the address strobe signal (XSAS) is asserted since the address is driven.
27-24	CSD[3:0]	Chip select delay This field sets the number of cycles until the chip select signal (XSCSn) is asserted since the address is driven.
23-20	ASW[3:0]	Address strobe width This field sets the number of cycles that the address strobe signal (XSAS) is asserted. When ASW=0, the address strobe signal is asserted for one cycle.
19-16	RWD[3:0]	Read/write status signal delay This field sets the number of cycles until the read/write status signal (SRXW) is asserted since the address is driven.
15-12	RED[3:0]	Read enable delay This field sets the number of cycles until the read enable signal (XSRE) is asserted since the address is driven.
11-8	WED[3:0]	Write enable delay This field sets the number of cycles until the write enable signal (XSWE) is asserted since the falling edge of the clock for the cycle when the address is driven.
7-0	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.

There are limit conditions. Refer to page 235, 7.5 Cautions.

7.3.4. Bank control register 2

Register symbol: SBCTRLn2 (n= 0 – 7)
Address: 0xD8C002n8 (SBCTRLn2)
Purpose: This register sets the control mode for bank n (XSCSn)

Bit	31	30	29	28	27	26	25	24
Bit name	BSTE	BT[2:0]			RWINV	BW	BM	WM
Initial value	0	000			0	Note	0	0
R/W	RW	RW			RW	RW	RW	RW
Bit	23	22	21	20	19	18	17	16
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	RESERVED				BWC[3:0]			
Initial value	0				0			
R/W	R				R			
Bit	7	6	5	4	3	2	1	0
Bit name	WC[7:0]							
Initial value	00001111							
R/W	RW							

Note : Refer to the descriptions shown below for the initial values.

Bit	Bit name	Description
31	BSTE	Burst enable This bit enables burst access for a 16-byte access. This setting is only valid in the handshake mode. 0 : Perform four accesses (under the 32-bit bus setting) or eight accesses (under the 16-bit bus setting) for a 16-byte access. 1 : Perform one burst data transfer for a 16-byte access.
30-28	BT[2:0]	Bus type setting This field sets the bus type. 000 : SRAM interface 001 : Address/data multiplexed interface 010 : Burst ROM interface 011 - 111 : Setting prohibited
27	RWINV	Read/write signal polarity This bit sets the polarity of the SRXW signal. 0 : Normal (read = H, write = L) 1 : Inverted (read = L, write = H)
26	BW	Bus width setting This bit sets the bus width. 0 : 32bits 1 : 16BITS The initial value for this bit is explained below: SBCTRL02: WHEN BOOTSEL = L: “0” When BOOTSEL = H/BOOTBW = H: “0” When BOOTSEL = H/BOOTBW = L: “1” SBCTRL12: When BOOTSEL = H: “0” When BOOTSEL = L/BOOTBW = H: “0” When BOOTSEL = L/BOOTBW = L: “1” SBCTRL22-SBCTRL72: Always "0"
25	BM	Synchronization mode setting This bit sets synchronous mode/asynchronous mode. 0 : Synchronous mode (SYSCLK) 1 : Asynchronous mode (ASYSCLK)
24	WM	Wait mode setting

Bit	Bit name	Description
		This bit sets either fixed wait access or handshake access. 0 : Fixed wait access 1 : Handshake access
23-12	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
11-8	BWC[3:0]	Burst wait count This field sets the number of access cycles after the first data in burst ROM mode. When BWC=0, the number of wait cycles is set to 0 and the number of access cycles is set to one cycle.
7-0	WC[7:0]	Wait count If fixed wait access is set (WM=0), this field sets the number of wait cycles after the chip select signal (XSCSn) is asserted. If WC=0, the number of wait cycles is set to "0" and the number of access cycles is set to one cycle. If handshake access is set (WM=1), this field sets the number of cycles until sampling for the acknowledge signal begins after the chip select signal is asserted. If WC=0, the number of wait cycles is set to "0" and sampling begins after one cycle.

7.4. Description of Operation

7.4.1. Access Data Alignment

This LSI supports Little Endian format for the byte data arrangement method, in which the least significant byte (LSB) is address 0.

Bit number	31 24	23 16	15 8	7 0
Memory address	(address 4n+3)	(address 4n+2)	(address 4n+1)	(address 4n)
Word data	Address : 4n			
Halfword data	Address : 4n+2		Address : 4n	
Byte data	Address : 4n+3	Address : 4n+2	Address : 4n+1	Address : 4n

Figure 26 Access data alignment

The width of the system bus can be either 16 bits or 32 bits. The data alignment and the status of the write enable signals (XSWE[3:0]) during an access are shown in the tables below.

Table 57 Data Alignment and Write Enable Signals When a 32-bit Bus Is Set

Access	Data bus SD[31:0]				Write enable XSWE[3:0]			
	31-24	23-16	15-8	7-0	3	2	1	0
Address 0 8-bit access	-	-	-	7-0	-	-	-	○
Address 1 8-bit access	-	-	7-0	-	-	-	○	-
Address 2 8-bit access	-	7-0	-	-	-	○	-	-
Address 3 8-bit access	7-0	-	-	-	○	-	-	-
Address 0 16-bit access			15-8	7-0	-	-	○	○
Address 2 16-bit access	15-8	7-0			○	○	-	-
Address 0 32-bit access	31-24	23-16	15-8	7-0	○	○	○	○

Table 58 Data Alignment and Write Enable Signals When a 16-bit Bus Is Set

Access	Data bus SD[31:0]				Write enable XSWE[3:0]			
	31-24	23-16	15-8	7-0	3	2	1	0
Address 0 8-bit access	-	7-0	-	-	-	○	-	-
Address 1 8-bit access	7-0	-	-	-	○	-	-	-
Address 2 8-bit access	-	7-0	-	-	-	○	-	-
Address 3 8-bit access	7-0	-	-	-	○	-	-	-
Address 0 16-bit access	15-8	7-0	-	-	○	○	-	-

Address 2 16-bit access		15-8	7-0	-	-	○	○	-	-
Address 0 32-bit access	First time	15-8	7-0	-	-	○	○	-	-
	Second time	31-24	23-16	-	-	○	○	-	-

7.4.2. Transfer Size

The system bus supports 1-byte, 2-byte, 4-byte, and 16-byte transfers. The transfer size is specified by the value of the size signals (SSZ[1:0]). SSZ[1:0] is valid during the same interval as the system bus address (SA[31:0]).

Table 59 System Bus Transfer Size

SSZ[1:0]	Transfer Size
00	1 byte
01	2 bytes
10	4 bytes
11	16 bytes

7.4.3. Chip select

The bus controller makes requests for addresses in a 1GB space from 0x80000000 to 0x9FFFFFFF to the system bus controller. As shown in the diagram below, if the value of the bit product between the requested address and SBBASE.BAM[31:17] is equal to the value of the bit product between SBBASE.BBA[31:17] and SBBASE.BAM[31:17], the chip select signal for the corresponding bank is asserted.

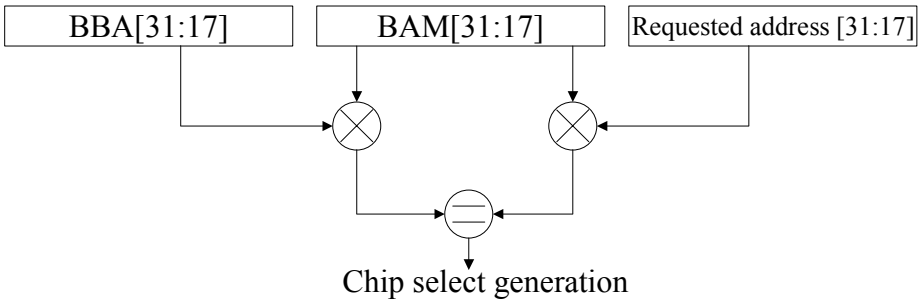


Figure 27 Chip select

7.4.4. SRAM Interface

When the BT[2:0] field in bank control register 2 (SBCTRLn2.BT[2:0]) is "000", the SRAM interface can be set for banks 0 to 7. The bus cycles are synchronized with SYSCLK in a synchronous mode and with ASYSCLK in an asynchronous mode. The bus cycles are generated in accordance with the parameters that are set in bank control registers 0,1,and 2. In the asynchronous mode, handshake mode access is not supported.

7.4.4.1. 32-bit Bus Fixed Wait Access

The following chart is a timing chart for a read access when the 32-bit bus fixed wait settings have been made.
In a read access, after the address (SA[31:0]) and transfer size SSZ[1:0]) are output, the chip select signal (XSCSn) is asserted after a number of cycles that are set by CSD. Once the chip select signal is asserted, the data is sampled at the rising edge of the clock that follows the number of cycles set by WC (WC + 1). Once the data has been sampled, the read cycle ends after the cycles set at the largest value in ADH, CSH, RWH, and REH.
In a write access, after the address (SA[31:0]), transfer size (SSZ[1:0]), and the write data (SD[31:0]) are output, the chip select signal (XSCSn) is asserted after the cycles set by

CSD. Once the chip select signal is asserted, the write cycle ends after the cycles set at the largest value in ADH, DAH, CSH, RWH, and WEH after the rising edge of the clock that follows the number of cycles set by WC (WC+1).

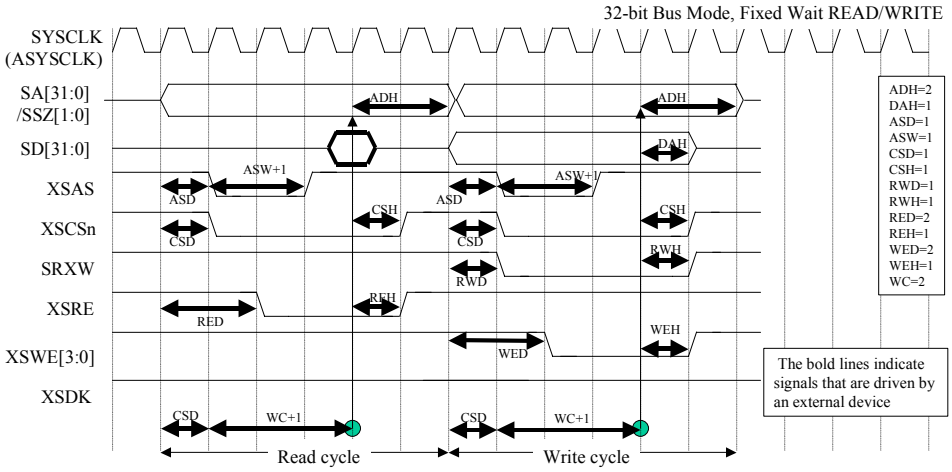


Figure 28 Timing Chart for 32-bit Bus Fixed Wait Access

The ADH parameter must be set to a suitable value, giving consideration to the read data tristate time in order to avoid a bus conflict between read data and write data on the data bus (SD[31:0]) in a write access following a read access.

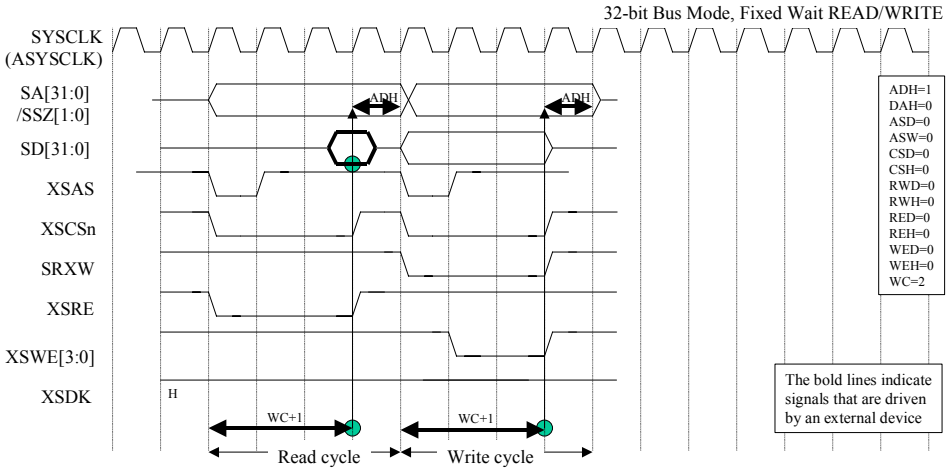


Figure 29 Timing Chart for 32-bit Bus Fixed Wait Access

7.4.4.2. 16-bit Bus Fixed Wait Access

A 4-byte access to a 16-bit bus is performed by conducting two 2-byte accesses. In the address for the first access, the two lowest bits are "00;" in the address for the second access, the two lowest bits are "10."
In 16-bit bus mode, the data transfer is performed through using the 16 bits of the data bus SD[31:16].

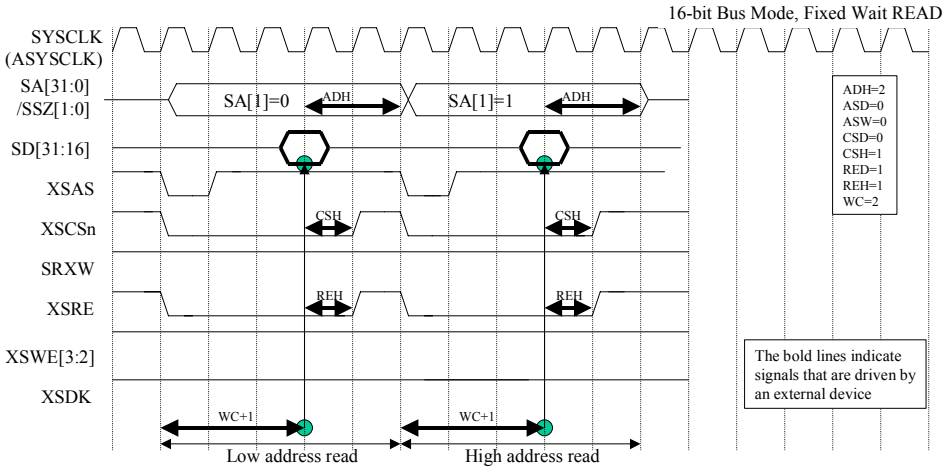


Figure 30 Timing Chart for a 16-bit Bus Fixed Wait Read Access

In 16-bit bus mode, XSWE[3:2] is asserted as a write enable signal according to the address and access size.

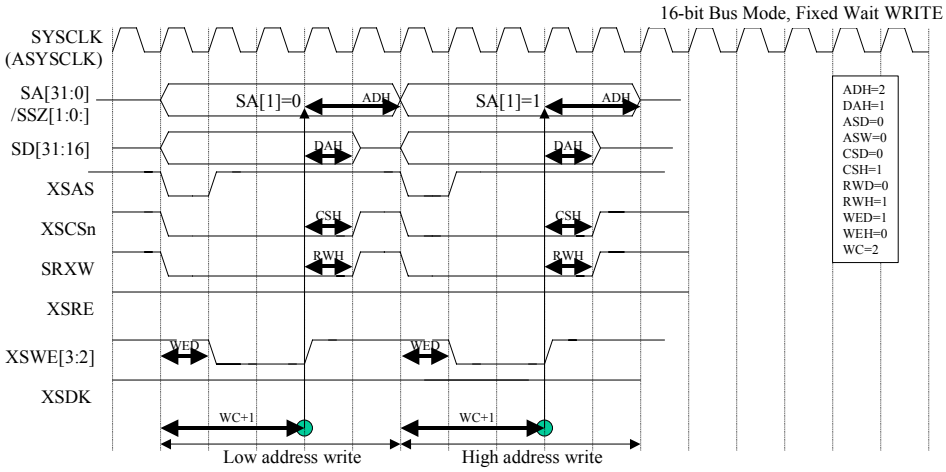


Figure 31 Timing Chart for a 16-bit Bus Fixed Wait Access

7.4.4.3. 32-bit Bus Handshake Access

Handshake access supports synchronous mode (synchronized with SYSCLK) only.
In a read access, the data becomes valid at the rising edge of the clock with the asserted acknowledge signal (XSDK).
In a write access, the write cycle ends at the edge of the clock with the asserted XSDK , and the control signals are negated according to the various parameter settings.

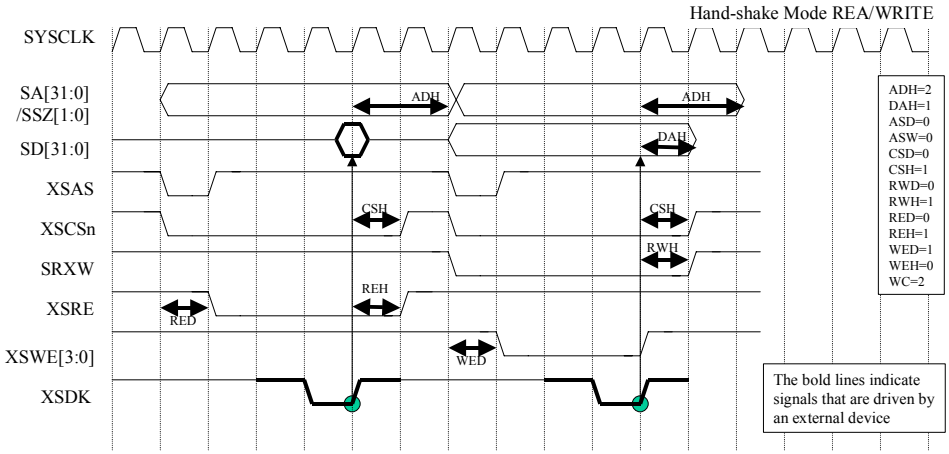


Figure 32 Timing Chart for 32-bit Bus Handshake Access

If the BSTE bit in bank control register 2 (SBCTRLn2) is "0," four 32-bit accesses are performed in response to a 16-byte read access request in a handshake access mode. If the BSTE bit is "1," a burst data access is performed as indicated in the below chart. An external device asserts the acknowledge signal (XSDK) in each cycle in which the data becomes valid.
The external device supplies the data in sequence from word 0 (byte addresses 0 to 3) to word 3 (byte addresses 12 to 15).
In the example in the below chart, the acknowledge signal (XSDK) is asserted at one-cycle intervals, but the interval at which XSDK is asserted can be any number of cycles, depending on the external slave device.

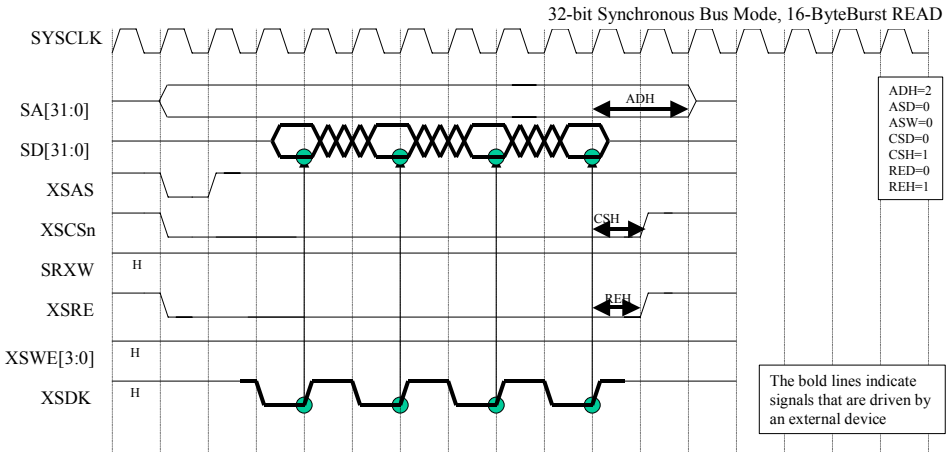


Figure 33 Timing Chart for 16-byte Burst Read Access

Regarding write accesses as well, a burst access is performed if the BSTE bit is "1." The external device requests the next data from the LSI by asserting the acknowledge signal (XSDK).
The system bus controller supplies the data in sequence from word 0 (byte addresses 0 to 3) to word 3 (byte addresses 12 to 15).
In the example in the below chart, the acknowledge signal (XSDK) is asserted at one-cycle intervals, but the interval at which XSDK is asserted can be any number of cycles,

depending on the external slave device.

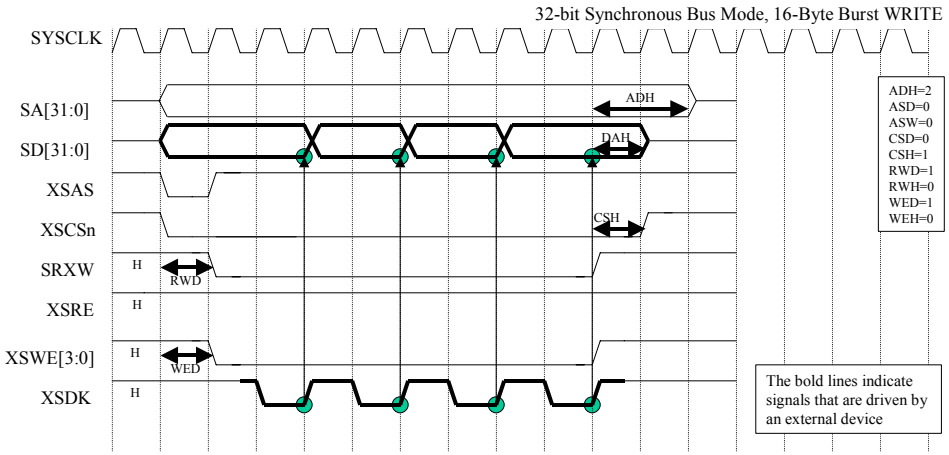


Figure 34 Timing Chart for 16-byte Burst Write Access

7.4.4.4. 16-bit Bus Handshake Access

A 4-byte access to a 16-bit bus is performed by conducting two 2-byte accesses. In the address for the first access, the two lowest bits are "00;" in the address for the second access, the two lowest bits are "10."

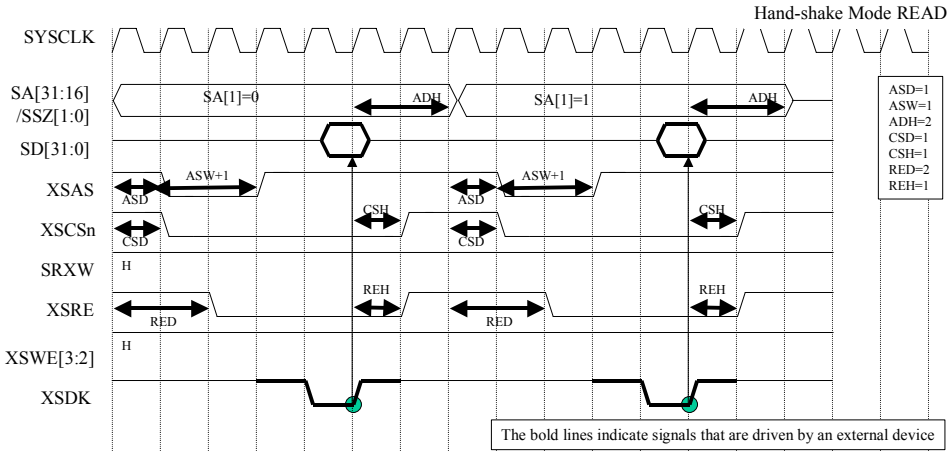


Figure 35 Timing Chart for a 16-bit Bus Handshake Read Access

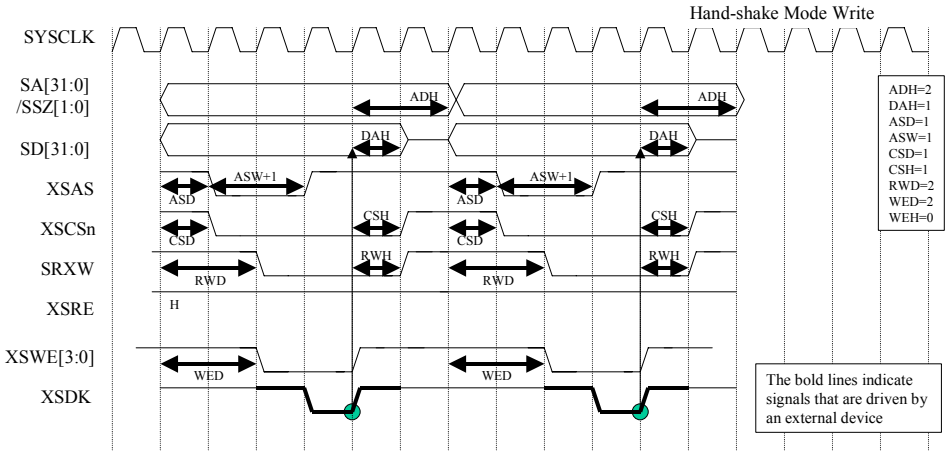


Figure 36 Timing Chart for a 16-bit Bus Handshake Write Access

If the BSTE bit in bank control register 2 (SBCTRLn2) is "0," eight 16-bit accesses are performed in response to a 16-byte read access request. If the BSTE bit is "1," a burst data access is performed as indicated in the below chart. An external device asserts the acknowledge signal (XSDK) in each cycle in which the data becomes valid.
The external device supplies the data in sequence from 16-bit word 0 (byte addresses 0 to 1) to 16-bit word 7 (byte addresses 14 to 15).
In the example in the below chart, the acknowledge signal (XSDK) is asserted continuously for several cycles, but the interval at which XSDK is asserted can be any number of cycles, depending on the external slave device.

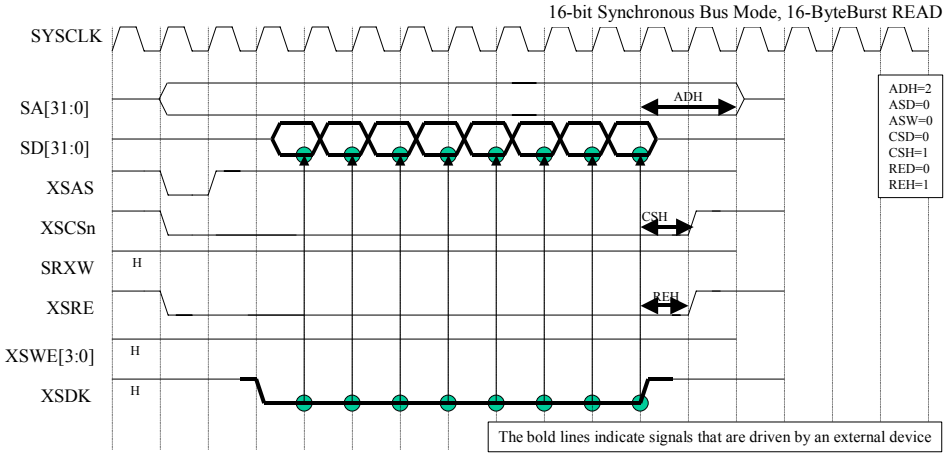


Figure 37 Timing Chart for 16-bit Bus 16-byte Handshake Read Access

Regarding write accesses as well, a burst access is performed if the BSTE bit is "1." The external device requests the next data from the system bus controller by asserting the acknowledge signal (XSDK).
The system bus controller supplies the data in sequence from 16-bit word 0 (byte addresses 0 to 1) to 16-bit word 7 (byte addresses 14 to 15).
In the example in the below chart, the acknowledge signal (XSDK) is asserted continuously for several cycles, but the interval at which XSDK is asserted can be any number of cycles, depending on the external slave device.

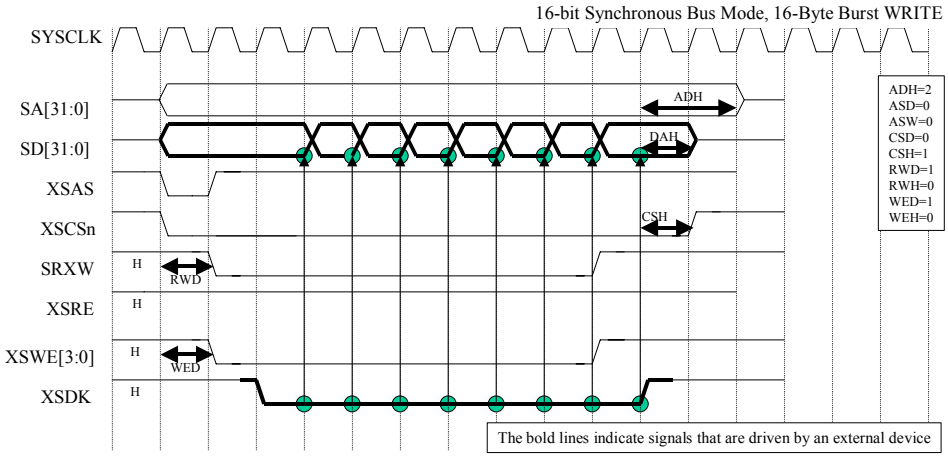


Figure 38 Timing Chart for 16-bit Bus 16-byte Handshake Write Access

7.4.5. Address/Data Multiplexed Interface

Banks 0 through 7 can use the SD[31:0] data bus as both an address and data bus by setting the BT field in SBCTRLn2 to the address/data multiplexed interface. Address/data multiplexing only supports synchronous fixed access and synchronous handshake access.

7.4.5.1. 32-bit Bus Fixed Wait Access

If an address is driven on the address bus (SA[31:0]), the address is driven in the same cycle on the data bus (SD[31:0]). The address is driven once the address strobe signal (XSAS) is asserted in accordance with the timing set in ASD, and is continuously driven until the cycle following the cycle in which XSAS is de-asserted in accordance with the timing set in ASW. The data bus is set to the tristate condition.
In a write access, the address is driven continuously until the cycle following the cycle in which XSAS is de-asserted in accordance with the timing set in ASW, and then the write data is driven on the data bus.

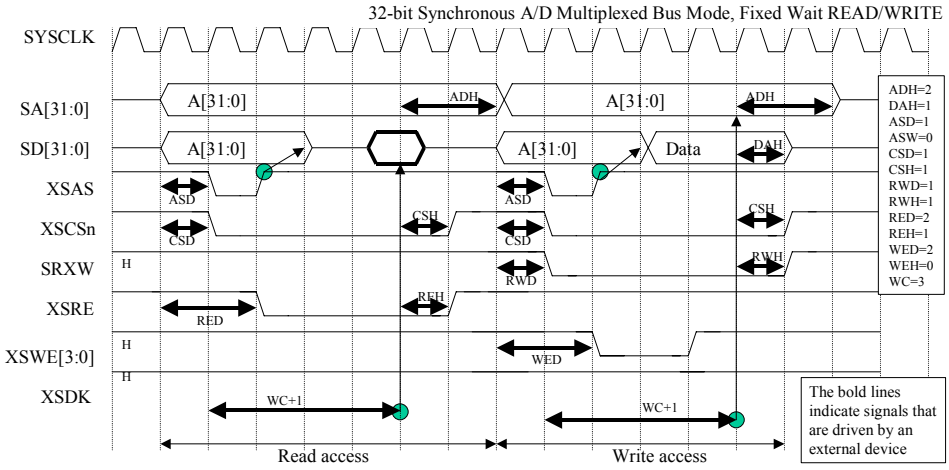


Figure 39 Timing Chart for 32-bit Bus Address Data Multiplexed/Fixed Wait Access

7.4.5.2. 16-bit Bus Fixed Wait Access

The chart shown below is a timing chart for a read access through address/data multiplexed interface access in 16-bit bus mode. If an address is driven on the address bus (SA[31:0]), the lower 16 bits of the address (SA[15:0]) are driven on the data bus (SD[31:16]) in the same cycle. If it is necessary to know the upper portion of the address, refer to the upper 16 bits of the address bus (SA[31:16]). The address is driven once the address strobe signal (XSAS) is asserted in accordance with the timing set in ASD, and is continuously driven until the cycle following the cycle in which XSAS is de-asserted in accordance with the timing set in ASW. The data bus is set to the tristate condition.
A 4-byte access to a 16-bit bus is performed by conducting two 2-byte accesses. In the address for the first access, the two lowest bits are "00;" in the address for the second access, the two lowest bits are "10."

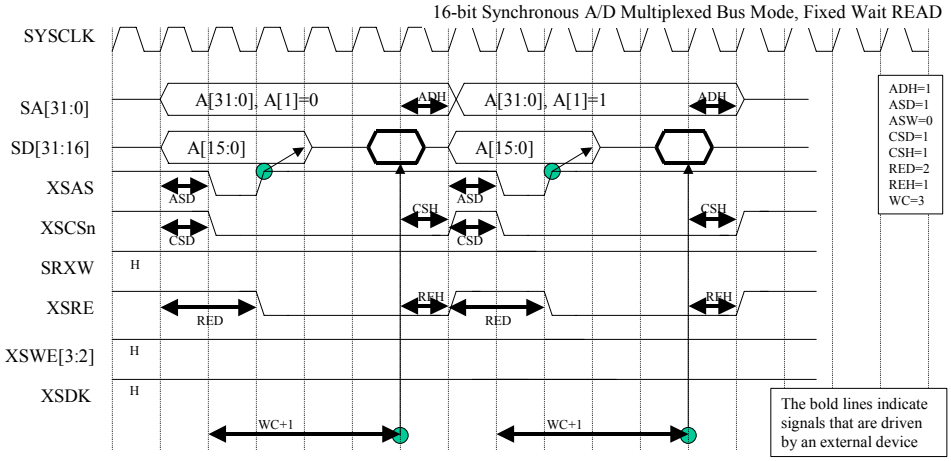


Figure 40 Timing Chart for 16-bit Address Data Multiplxed/Fixed Wait Read Access

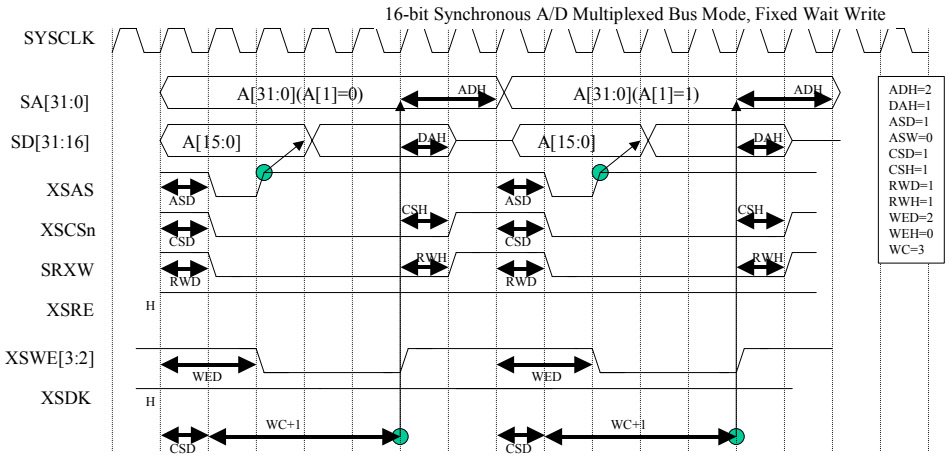


Figure 41 Timing Chart for 16-bit Bus Address Data Multiplxed/Fixed Wait Write Access

7.4.5.3. 32-bit Bus Handshake Access

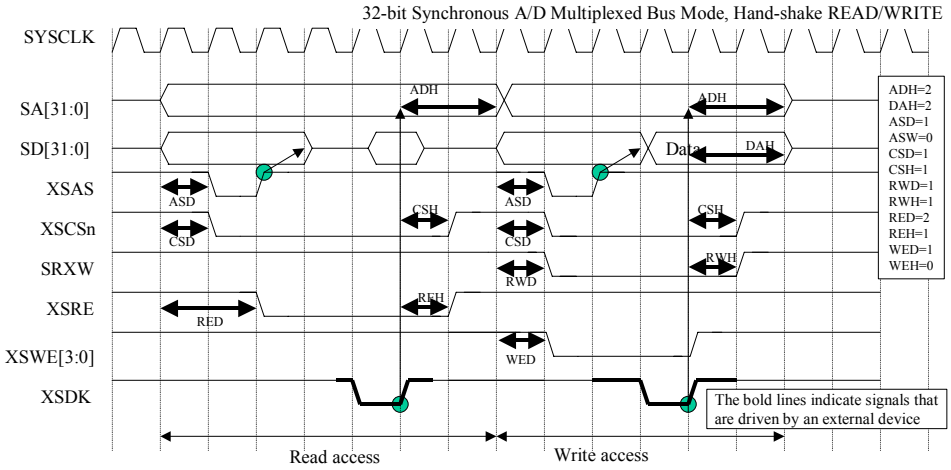


Figure 42Timing Chart for 32-bit Bus Address Data Multiplexed/Handshake Access

If the BSTE bit in bank control register 2 (SBCTRLn2) is "0," then four 32-bit accesses are performed in response to a 16-byte read access request. If the BSTE bit is "1," then a burst data access is performed as indicated in the chart shown below. An external device asserts the acknowledge signal (XSDK) in each cycle in which the data becomes valid.
The external device supplies the data in sequence, from word 0 (byte addresses 0 to 3) to word 3 (byte addresses 12 to 15).

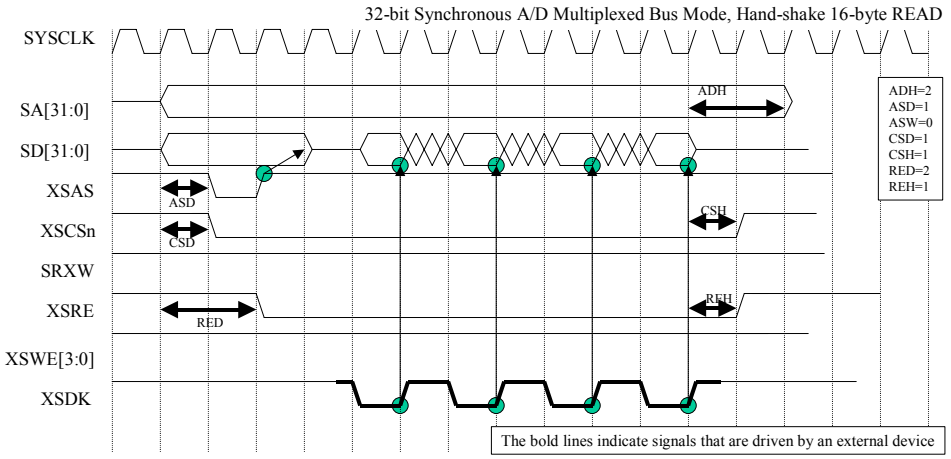


Figure 43 Timing Chart for 16-byte Handshake Read Access

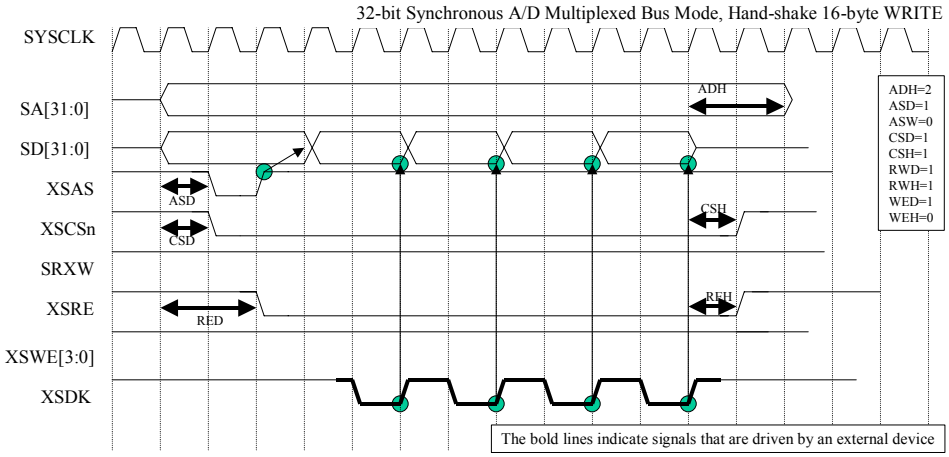


Figure 44 Timing Chart for 16-byte Handshake Write Access

7.4.5.4. 16-bit Bus Handshake Access

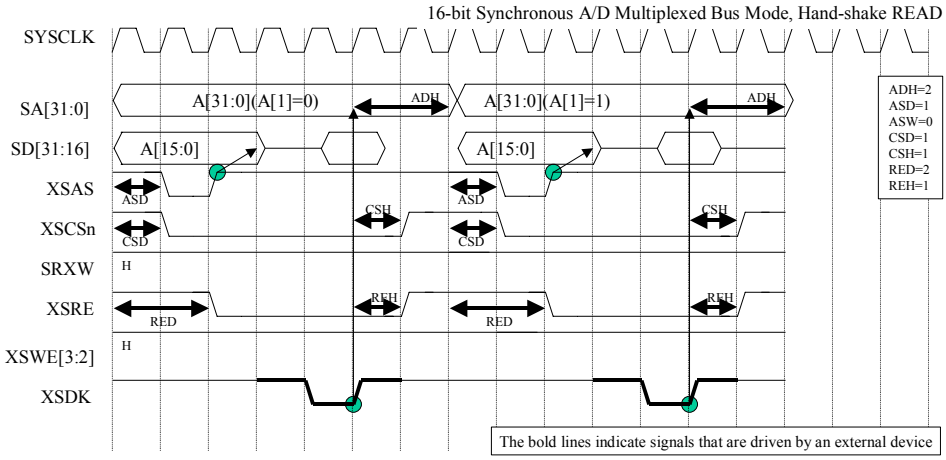


Figure 45 Timing Chart for 16-bit Bus Handshake Read Access

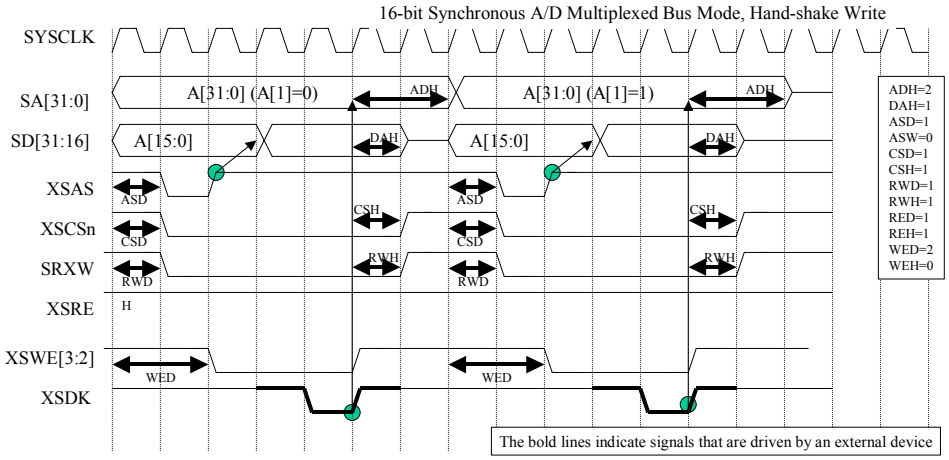


Figure 46 Timing Chart for 16-bit Bus Handshake Write Access

If the BSTE bit in bank control register 2 (SBCTRLn2) is "0," then eight 16-bit accesses are performed in response to a 16-byte read access request. If the BSTE bit is "1," then a burst data access is performed as indicated in the chart shown below. An external device asserts the acknowledge signal (XSDK) in each cycle in which the data becomes valid.
The external device supplies the data in sequence, from 16-bit word 0 (byte addresses 0 to 1) to 16-bit word 7 (byte addresses 14 to 15).

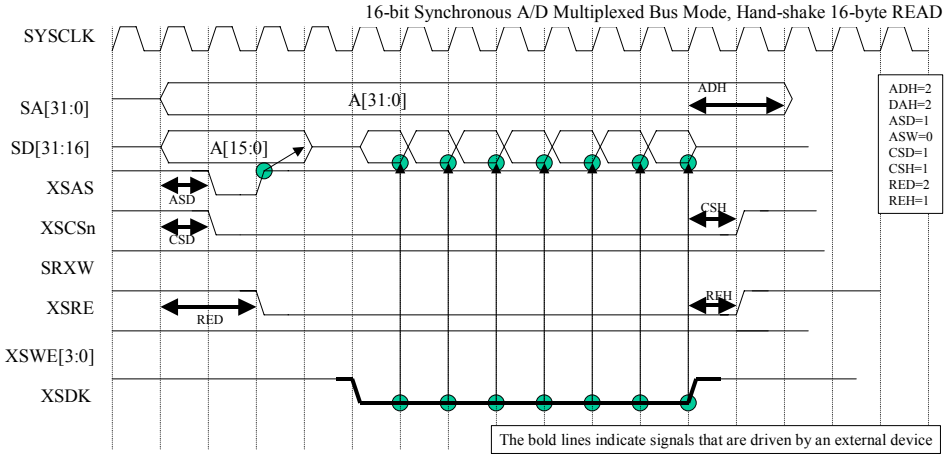


Figure 47 Timing Chart for 16-bit Bus 16-byte Handshake Read Access

Regarding write accesses as well, a burst access is performed if the BSTE bit is "1." The external device requests the next data from the LSI by asserting the acknowledge signal (XSDK).
The LSI supplies the data in sequence, from 16-bit word 0 (byte addresses 0 to 1) to 16-bit word 7 (byte addresses 14 to 15).

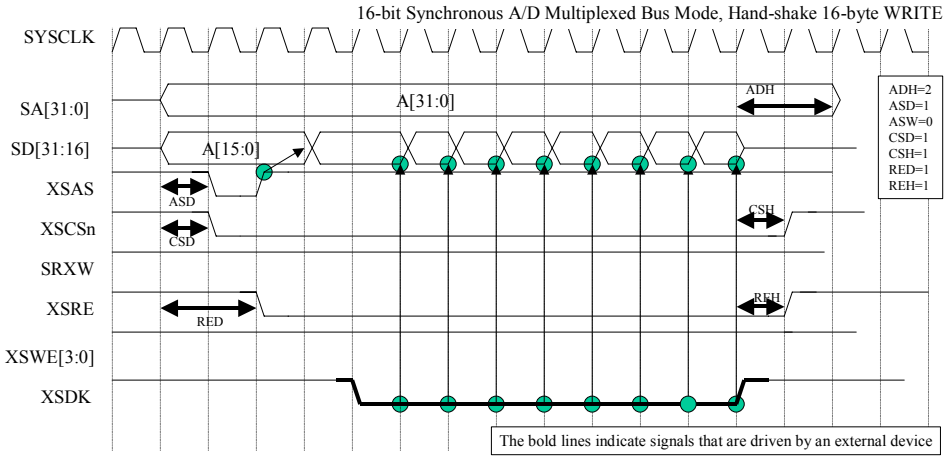


Figure 48 Timing Chart for 16-bit Bus 16-byte Handshake Access

7.4.6. Burst ROM Interface

When the bank is set to the burst ROM interface in the bus type field (BT) in SBCTRLn2, a 16-byte burst access is performed. Either synchronous mode or asynchronous mode can be set.

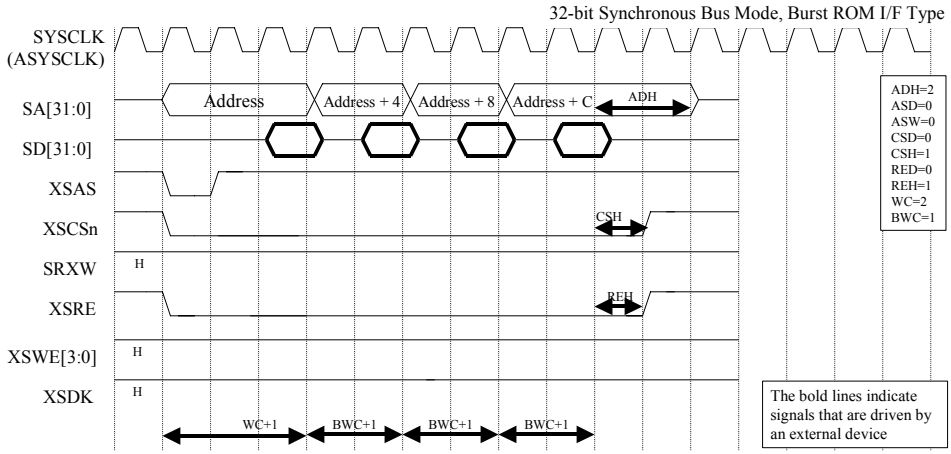


Figure 49 Timing Chart for Burst ROM 16-byte Read Access

7.4.7. External Master Device Support

The system bus controller supports access by external master devices on the system bus. The system bus controller interfaces with the external master device over the 32-bit data bus in synchronization with SYSCLK.

7.4.7.1. Bus Arbitration

Bus arbitration on the system bus is implemented by using the bus request signal (XSBR) and the bus grant signal (XSBG). If the external master device asserts the XSBR signal, the LSI asserts XSBG after the current bus access ends, releases the bus authority to the external master device, and then sets the address bus (SA[31:0], SSZ[1:0]), the address strobe signal (XSAS), and the read/write signal (SRXW) to high impedance.

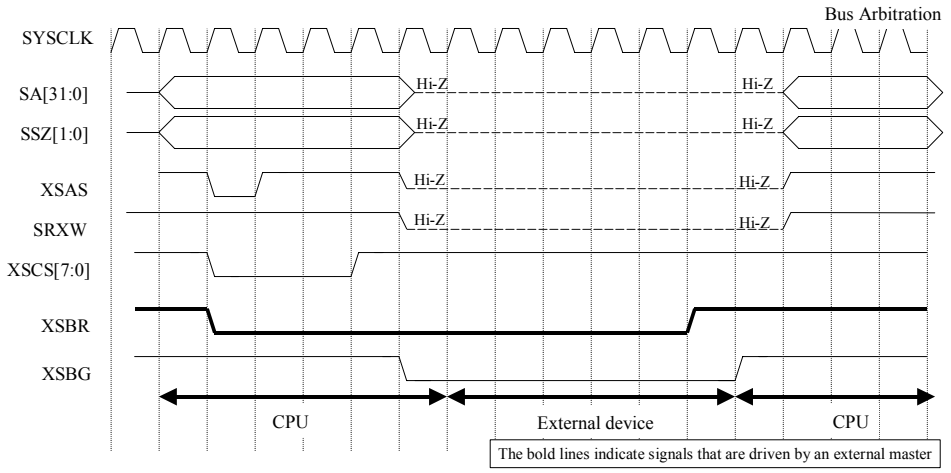


Figure 50 System bus arbitration

The external master device can assert the address strobe signal (XSAS) and start the bus cycle in a cycle in which the bus request signal (XSBR) and the bus grant signal (XSBG) are both asserted. The LSI conducts a read or write access (read access when SRXW is

high, write access when SRXW is low), of the size specified by SSZ[1:0] at the address specified by the address bus (SA[31:0]), which becomes valid at the rising edge of the clock in which the address strobe signal (XSAS) was asserted. The external master device asserts the bus request signal (XSBR) until the bus cycle ends.
By continuing to assert the bus request even after the bus cycle ends, the external master device is able to continuously generate multiple bus cycles, but doing so will interfere with bus access by the CPU. Therefore, it is recommended that the external device not continue to occupy the bus for longer than is necessary.
If there is a bus access request with a higher priority while the external master device is using the bus, the system bus controller requests the release of the bus by negating the XSBG signal. If the XSBG signal is negated while a bus cycle is in progress, the external master device must negate the XSBR signal and release the bus after the current bus cycle ends.

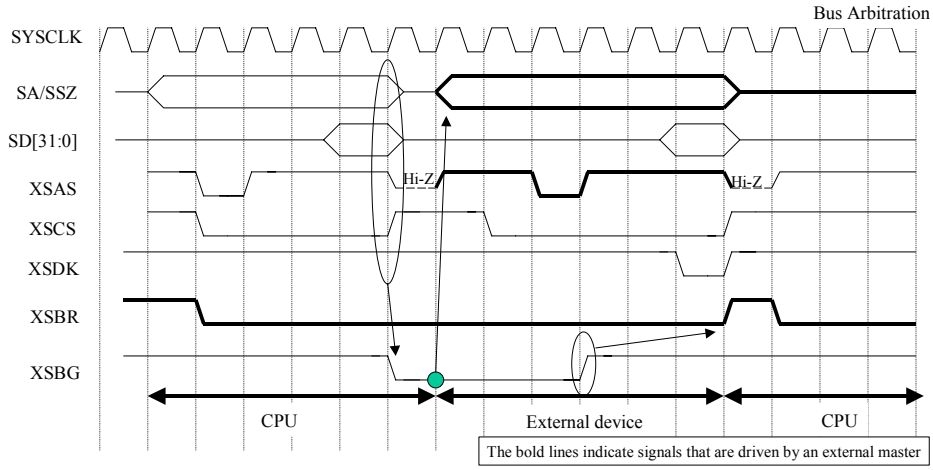


Figure 51 When a bus access with high priority is performed during using the bus of the external master device.

In cases where multiple bus masters reside on the system bus, a bus arbitration function is necessary in order to handle arbitration among the external bus master devices.

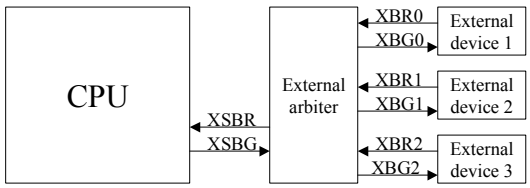


Figure 52 Bus arbitration function

7.4.7.2. External Master Device Access

An external master can begin a read access by asserting the address strobe signal (XSAS) while the read/write status signal (SRXW) is driven high. The system bus controller can notify the external master device that there is valid data on the data bus (SD[31:0]) by asserting the acknowledge signal (XSDK). Furthermore, it can begin a write access by asserting the address strobe signal (XSAS) while the read/write status signal (SRXW) is driven low. In the case of a write access, the system bus controller ends the write access cycle by asserting the acknowledge signal (XSDK).

7.4.7.2.1. Internal Memory, Memory Bus Access

When the request is to read internal memory (on-chip SRAM or an internal peripheral register) or SDRAM that is connected to the memory interface, the external master controller asserts the acknowledge signal (XSDK) after the read is completed and drives the data that was read on the system bus. In the case of a write, the data that is to be written is sent to the slave bus along with the address, and after the write is completed, the write access is ended by driving the acknowledge signal (XSDK).

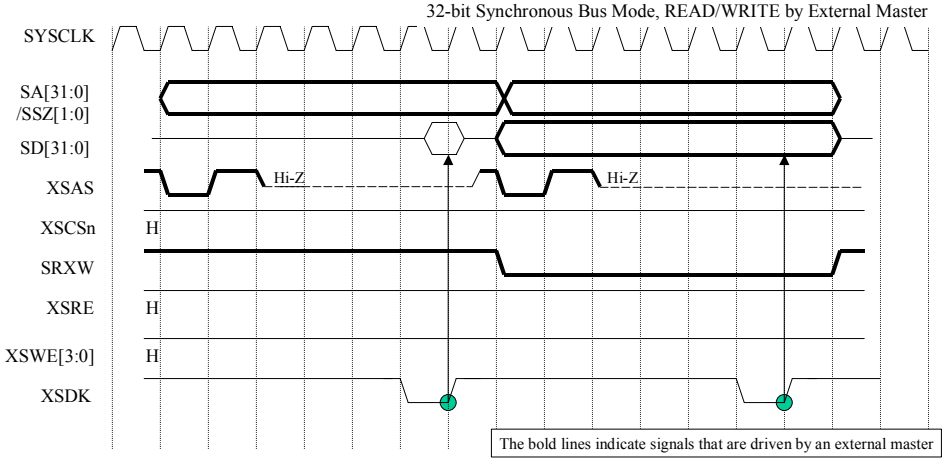


Figure 53 Timing Chart for External Master Device Access (Read/Write Access)

When an external master device requests a 16-byte read access, the system bus controller indicates that the first word data on the data bus (SD[31:0]) is valid by asserting the acknowledge signal (XSDK), and then continuously drives the second and subsequent words of data on the data bus (SD[31:0]).

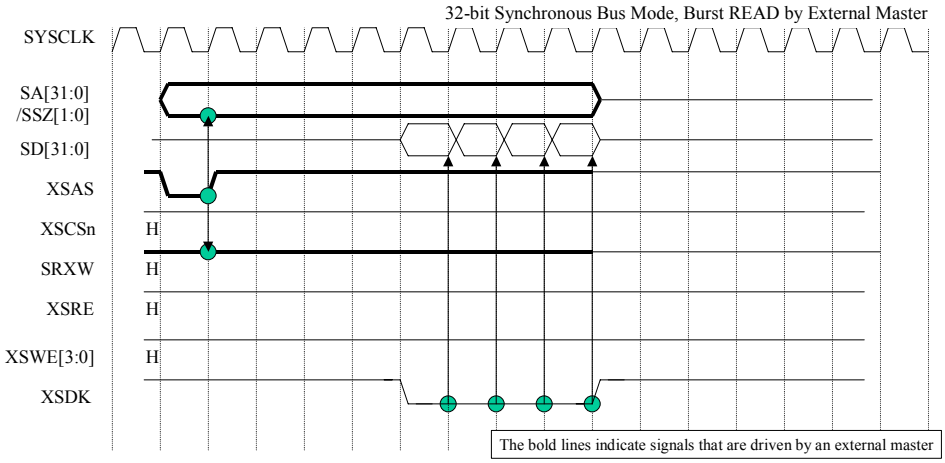


Figure 54 Timing Chart for External Master Device Access (16-byte Burst Read Access)

When an external master device requests a 16-byte write access, the system bus controller continuously requests the second and subsequent words of data by asserting the acknowledge signal (XSDK) and then samples the three words following the second word as the data to be written.

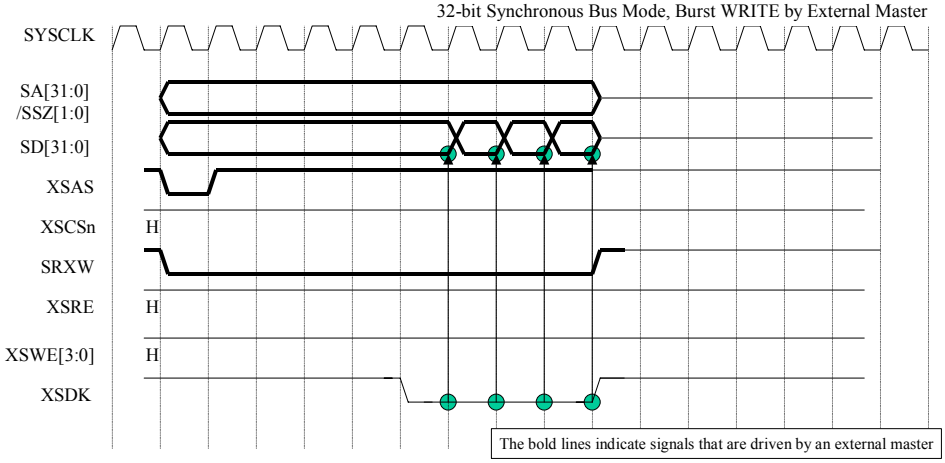


Figure 55 Timing Chart for External Master Device Access (16-byte Burst Write Access)

7.4.7.2.2. System Bus Access

When the request is for access to a device on the system bus, the system bus controller generates system bus control signals according to the access mode and timing that is set for the corresponding bank (indicated by the chip select signal).
In the case of a read, the data is read once from the slave device and then driven on the data bus (SD[31:0]), and the acknowledge signal (XSDK) is asserted. This also applied to 16-byte burst access.

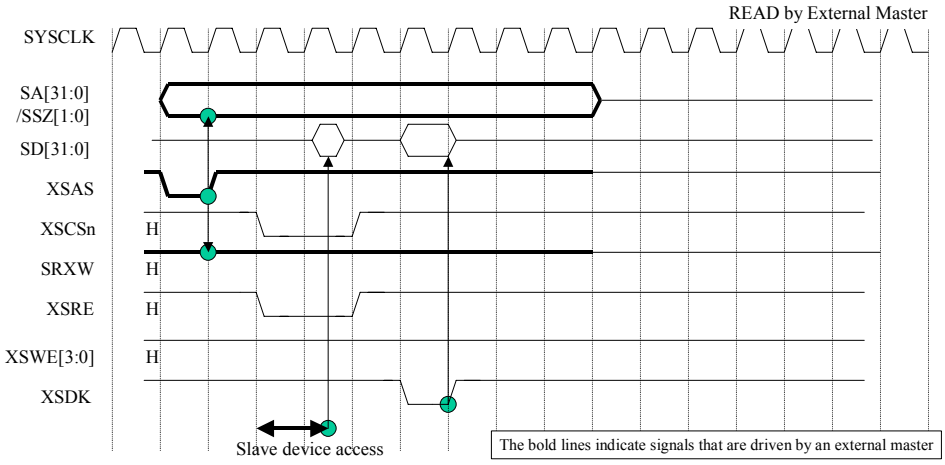


Figure 56 External Master Device Read (Read of a Slave Device on the System Bus)

In the case of a write, the system bus controller asserts the acknowledge signal (XSDK) and gets the write data according to the access size. In this case, the system bus can negate XSBR through the slave device and acquire the authority to use the system bus after the cycle ends by negating the bus grant signal (XSBG) and writing to the slave device. This also applies to 16-byte burst access.

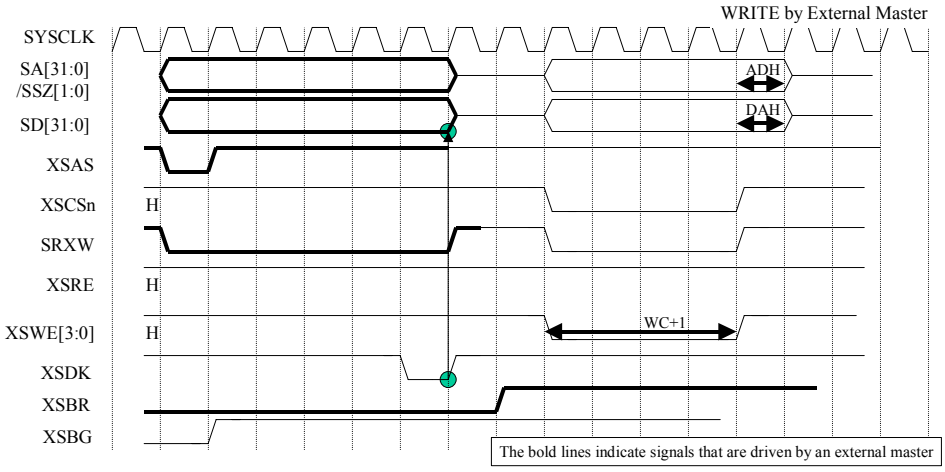


Figure 57 External Master Device Write (Write to a Slave Device on the System Bus)

7.5. Cautions

7.5.1. Limitations concerning the bus timing setting

- Bank control register 0 and bank control register 1 can set bus timings. This LSI has the limitations for the bus timing setting.

  The setting that one is not “0” must not carried out when a bank sets ADH, DAH, CSH, RWH, REH of bank control register 0 to “0” and when one of RFD and WED in bank control register 1 of the same bank and different bank is “0”.

- Do not set a higher value for ASD than CSD+WC.

Chapter 8 Memory Bus Controller (MBC)

8

CHAPTER 8

Memory Bus Controller (MBC)

8.1. General

Memory bus controller (MBC) can connect directly to SDRAM max. 2 chips through RAS/CAS signals and others without an external circuit.

Moreover, this can connect directly to a device with undefined CAS Latency through the control of data acknowledge (MDK) signals.

8.2. Features

- SDRAM interface
- 16-bit data bus
- Supports one or two 64Mbit/128Mbit/256Mbit (x 16-bit width) SDRAM
  - 64 Mbit (configuration: 1M x 16, four banks)
  - 128 Mbit (configuration: 2M x 16, four banks)
  - 256 Mbit (configuration: 4M x 16, four banks)

  When using two SDRAM, they must be of the same size.
- Support a maximum of 64 MB. (256Mbit x 2)

8.3. Description of Registers

Table 60 Memory bus controller register

Address:	Symbol	Name	NUMBER OF BITS	Initial value	Access size
0x DA000000	SDRAMBUS	Bus mode control register	32	0x AA96061C	8,16,32
0x DA000004	SDREFCNT	Refresh cycle register	32	0x 00000C30	8,16,32
0x DA000008	SDBASE0	Base address register 0	32	0x 0000F200	8,16,32
0x DA00000C	SDBASE1	Base address register 1	32	0x 0000F200	8,16,32
0x DA000010	SDSHDW	Test register	32	0x 00000006	8,16,32

8.3.1. Bus mode control register

Register symbol: SDRAMBUS
Address: 0xDA000000
Purpose: SDRAMBUS is used for the SDRAM I/F mode control settings. SDRAMBUS must be changed while the refresh operation is disabled. Operation is not guaranteed if the register is changed while SDRAM is being accessed.

Bit	31	30	29	28	27	26	25	24
Bit name	CASLATE		RASLATE		PREWAIT		SETWAIT	
Initial value	10		10		10		10	
R/W	R/W		R/W		R/W		R/W	
Bit	23	22	21	20	19	18	17	16
Bit name	BSTWAIT		REFNUM		TRANSWAIT		SIZE	
Initial value	10		01		11		11	
R/W	R/W		R/W		R/W		R/W	
Bit	15	14	13	12	11	10	9	8
Bit name	reserved		WTPREWAIT		BL			SELFON
Initial value	00		00		011			0
R/W	R		R/W		R			R
Bit	7	6	5	4	3	2	1	0
Bit name	SELFREQ	PONSEQ	BSTSPT	TRC[1:0]		REFEN	MODE32	EXARBEN
Initial value	0	0	0	11		1	0	0
R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W

Bit	Bit name	Description
31-30	CASLATE	CAS latency selection This field sets the CAS latency of the SDRAM. 00 : Setting prohibited 01 : CAS latency = 2 10 : CAS latency = 3 11 : Setting prohibited
29-28	RASLATE	RAS latency selection This field sets the RAS latency of the SDRAM. 00 : Setting prohibited 01 : RAS latency = 2 10 : RAS latency = 3 11 : Setting prohibited

Bit	Bit name	Description
27-26	PREWAIT	Precharge command cycle This field sets the number of cycles needed for SDRAM precharge completion. 00 : Setting prohibited 01 : 2 cycles 10 : 3 cycles 11 : Setting prohibited
25-24	SETWAIT	Mode register setting command cycle This field sets the number of cycles needed for completion of the SDRAM mode register setting command. 00 : 1 cycle 01 : 2 cycles 10 : 3 cycles 11 : 4 cycles
23-22	BSTWAIT	Burst stop command cycle This field sets the number of cycles needed for completion of the burst stop command in the read cycle of SDRAM. 00 : Setting prohibited 01 : 2 cycles 10 : 3 cycles 11 : Setting prohibited
21-20	REFNUM	Refresh command number This field sets the number of refresh commands that are issued each refresh period. 00 : 1 01 : 2 10 : 3 11 : 4
19-18	TRASWAIT	Row address precharge command cycle number This field sets the minimum number of cycles until the precharge command is issued after the ROW address is issued to SDRAM. 00 : 4 cycles 01 : 5 cycles 10 : 6 cycles 11 : 7 cycles
17-16	SIZE	SDRAM size This field sets the size of the SDRAM to be used, and selects the range of addresses that is to be used for RAS address hit determination. 00 : Setting prohibited 01 : Using a 64Mbit SDRAM (x16) 10 : Using a 128Mbit SDRAM (x16) 11 : Using a 256Mbit SDRAM (x16)
15-14	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
13-12	WTPREWAIT	Numbers of precharge issuing cycles after data in. This field sets the minimum number of cycles until precharge command is issued after writing the final write data. 00 : 1 cycles 01 : 2 cycles 10 : 3 cycles 11 : 4 cycles
11-9	BL	Burst Length The burst size of the used SDRAM is set. This SDRAMIF is used

Bit	Bit name	Description
		under fixing at 8-word burst (read only). This setting is not directly related to write command because of using single write.
8	SELFON	Self-refresh mode on This bit indicates that the SDRAM is in self-refresh mode. 0 : Normal mode 1 : Self-refresh mode (power saving mode)
7	SELFREQ	Self-refresh mode request This bit controls the state shift to normal mode and self-refresh mode. 0 : Requests state shift to normal mode in the case that the access to SDRAM is reqested under normal or self-refresh mode. 1 : Requests state shift to self-refresh mode.
6	PONSEQ	Power on sequence Before accessing and refreshing SDRAM, it is necessary to carry out the power on sequence for SDRAM. The power on sequence for SDRAM is initiated by writing a "1" to this bit after reset (including soft reset and WDT reset). Once the power on sequence has been initiated, it can not be re-initiated until it is reset again.
5	BSTSPT	Burst stop command enable This bit enables the burst stop command in modes other than full page mode. 0 : Does not use burst stop command. 1 : Uses burst stop command.
4 – 3	TRC[1:0]	Refresh command delay time This field sets the minimum number of cycles between two consecutive refresh cycles. 00 : 7 cycles 01 : 8 cycles 10 : 9 cycles 11 : 10 cycles
2	REFEN	Refresh enable This bit enables the SDRAM refresh cycle. When the SDREFCNT register is at its default value (0x00000C30), the refresh cycle is performed 4096 times in 64ms at 100MHz. The refresh counter begins counting when this bit is set. 0 : Does not generate the refresh cycle. 1 : Generates the refresh cycle (normal mode).
1	MODE32	Always write a "0" to these bits in this SDRAMIF.
0	EXARBEN	Always write a "0" to these bits in this SDRAMIF.

8.3.2. Refresh period register

Register symbol: SDREFCNT
Address: 0xDA000004
Purpose: SDREFCNT sets the SDRAM refresh period.
SDREFCNT must be changed while the refresh operation is disabled. Operation is not guaranteed if the register is changed while SDRAM is being accessed.

Bit	31	30	29	28	27	26	25	24
Bit name	Reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	Reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	Reserved				PERI[13:8]			
Initial value	0				001100			
R/W	R				RW			
Bit	7	6	5	4	3	2	1	0
Bit name	PERI[7:0]							
Initial value	0x30							
R/W	RW							

Bit	Bit name	Description
31-14	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
13-0	PERI[13:0]	Refresh Period This field sets the SDRAM refresh period. The SDRAM clock is counted, and when the set value is reached refresh is performed the number of times set in the REFNUM field in the SDRAMBUS register. When using the initial settings of PERI[13:0]=0x0C30 and SDRAMBUS. REFNUM[1:0]=0x1 (twice), refresh is performed twice in 31.25μs, so these settings can satisfy the requirements of SDRAM that needs the refresh operation to be performed 4096 times in 64ms. How to set the numbers of the SDRAMIF refresh When executing n-time refresh command in the one refresh mode, the specification for SDRAM shows that y-time refresh must be carried out at x ms, and SDRAM of z MHz is operated. x/y × z × n × 1000 = Ans.Cycle Refresh period It moves to the refresh mode each Ans cycle. (e.g.) When refreshing SDRAM operating at 133MHz, the refresh period is calculated as executing 4096 refreshes at 64ms as shown below. 64/4096 × 133 × 2 × 1000 = 4156 (13h' 103c)

8.3.3. Base address register 0

Register symbol: SDBASE0
Address: 0xDA000008
Purpose: SDBASE0 specifies the memory space that is allocated to chip select XMCS0 for SDRAM.
SDBASE0 must be changed while the refresh operation is disabled. Operation is not guaranteed if the register is changed while SDRAM is being accessed.

Bit	31	30	29	28	27	26	25	24
Bit name	CBA[31:24]							
Initial value	0							
R/W	RW							
Bit	23	22	21	20	19	18	17	16
Bit name	CBA[23:20]				Reserved			
Initial value	0				0			
R/W	RW				R			
Bit	15	14	13	12	11	10	9	8
Bit name	CBAM[31:24]							
Initial value	11110010							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	CBAM[23:20]				Reserved			CE
Initial value	0				0			0
R/W	RW				R			RW

Bit	Bit name	Description
31-20	CBA[31:20]	Chip base address Set the upper 12 bits of the base address in CBA[31:20].
19-16	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
15-4	CBM[31:20]	Chip base address mask Set the mask for comparing the accessed address with the address specified by CBA[31:20]. If the bit logical product of the upper 12 bits of the address for which access was requested and CBM[31:20] matches that of CBA[31:20] and CBM[31:20] , XMCS0 is asserted.
3-1	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
0	CE	Chip select enable This bit enables/disables the generation of the chip select signals. 0 : Disabled 1 : Enabled

8.3.4. Base address register 1

Register symbol: SDBASE1
Address: 0xDA00000C
Purpose: SDBASE1 specifies the memory space that is allocated to chip select XMCS1 for SDRAM.
SDBASE1 must be changed while the refresh operation is disabled. Operation is not guaranteed if the register is changed while SDRAM

is being accessed.

Bit	31	30	29	28	27	26	25	24
Bit name	CBA[31:24]							
Initial value	00000010							
R/W	RW							
Bit	23	22	21	20	19	18	17	16
Bit name	CBA[23:20]				Reserved			
Initial value	0				0			
R/W	RW				R			
Bit	15	14	13	12	11	10	9	8
Bit name	CBAM[31:24]							
Initial value	11110010							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	CBAM[23:20]				Reserved			CE
Initial value	0				0			0
R/W	RW				R			RW

Bit	Bit name	Description
31-20	CBA[31:20]	Chip base address Set the upper 12 bits of the base address in CBA[31:20].
19-16	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
15-4	CAM[31:20]	Chip base address mask Set the mask for comparing the address that was accessed with the address specified by CBA[31:20]. If the bit logical product of the upper 12 bits of the address for which access was requested and CAM[31:20] matches that of CBA[31:20] and CAM[31:20] , XMCS1 is asserted.
3-1	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
0	CE	Chip select enable This bit enables/disables the generation of the chip select signals. 0 : Disabled 1 : Enabled

8.3.5. SD shadow register

Register symbol: SDSHDW
Address: 0xDA000010
Purpose: SDSHDW sets SDRAMIF operation mode.
SDSHDW must be changed while the refresh operation is disabled.
Operation is not guaranteed if the register is changed while SDRAM is being accessed.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	reserved							

Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	reserved			AUTOPRE	CASLTPLUS	CANEN	HDSK	RDSNG
Initial value	0			0	0	1	1	0
R/W	R			R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Description
31-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4	AUTOPRE	Semiauto-precharge mode SDRAMIF executes precharge command with the timing equal to auto-precharge. 0: normal mode 1: test mode
3	CASLTPLUS	Cas Latency Delay Mode One cycle is added to the value of CAS Latency only in the internal control. 0: normal mode 1: test mode
2	CANEN	BCU Cancel Request Enable Mode This makes a canceling request from BCU valid. 0: test mode 1: normal mode
1	HDSK	Handshake Mode Enable Access through handshake becomes possible. 0: test mode 1: normal mode
0	RDSING	Read Single Access Mode This executes the issue of read command by a single access. 0: normal mode 1: test

8.4. Description of Operation

8.4.1. Connection Example

The following diagram shows the address, data, and control signal connections between the LSI and SDRAM.

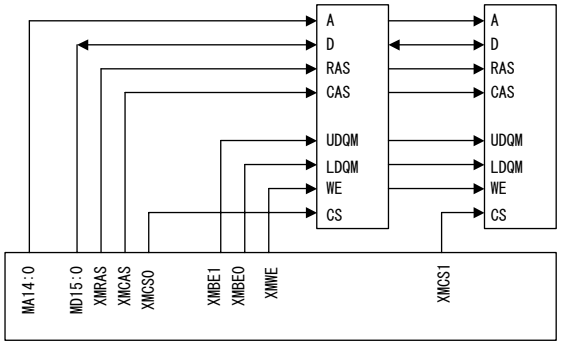


Figure 58 Connection of the addresses, data, and control signals to SDRAM

8.4.2. Clock Generation

This LSI supplies the operation clock to SDRAM from the SDCLK pin. The frequency of the clock that is supplied from the SDCLK pin is 1/2 or identical to the CPU clock. For details, refer to chapter 4 “Clock generator”. The clock that is input to the SDCKI pin is used for sampling the input data from SDRAM and the MDK input signals. When designing printed circuit boards, input the same clock to the SDCKI pin over an same wiring length with the case of inputing to SDRAM from the SDCLK pin.

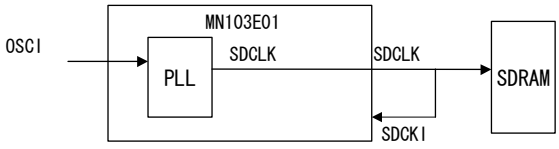


Figure 59 Clock generation

8.4.3. SDRAM Initialization

After a reset, initialize the memory controller.

(1) Bus controller (BCU) settings

Set the memory block,which is allocated to SDRAM, to the OpEX bus in the bus controller's BCCR register.

For example, when the 16MB from 0x98000000 to 0x98FFFFFF are allocated to OpEX bus, set "01" in B6AD[1:0] in the BCCR register as the setting for the corresponding block 6.

For details, refer to chapter 5 “Bus controller”.

(2) SDRAM chip base address settings

Set the appropriate base address and address mask in SDBASE0 and SDBASE1, and enable access to the appropriate chip by setting the CE bit to "1."

For example, when the 8MB from 0x98000000 to 0x987FFFFF are allocated to

XMCS0, set SDBASE0.CBA[31:20] = 0x980 and SDBASE0.CBM[31:20] = 0xFF8. Similarly, when the 8MB from 0x98800000 to 0x98FFFFFF are allocated to XMCS1, set SDBASE1.CBA[31:20] = 0x988 and SDBASE1.CBM[31:20] = 0xFF8.

(3) Refresh period settings and refresh initiation

Set the refresh period in the SDREFCNT register.

(4) SDRAM power on sequence and mode register settings

Set the SDRAM power on sequence and mode register by setting a suitable value in the SDRAMBUS register for the SDRAM to be used and writing a "1" to the PONSEQ bit. In this instance, set the access mode to burst read and single write mode. The burst length is set to 8 words.

8.4.4. Access Mode

In the SDRAM mode register settings, the burst read and single write mode is set with a burst length of 8 words.

In the case of a read access of less than 16 bytes (8 words), after the required number of words have been accessed, cancel the burst access by means of the burst stop command, a new read access, or the precharge command. In the case of a write access, perform single write accesses for the required number of words.

8.4.5. Access Data Alignment

This LSI supports Little Endian format for the byte data arrangement method, in which the least significant byte (LSB) is address 0.

Bit number	31 24	23 16	15 8	7 0
Memory address	(address 4n+3)	(address 4n+2)	(address 4n+1)	(address 4n)
Word data	Address : 4n			
Halfword data	Address : 4n+2		Address : 4n	
Byte data	Address : 4n+3	Address : 4n+2	Address : 4n+1	Address : 4n

Figure 60 Access data alignment

The bus width of the memory bus interface is 16 bits. The data alignment and the status of the byte access strobe (XMBE[1:0]) during write access are shown in the below table.

Table 61 Status of the byte access strobe (XMBE[1:0])

Access		Data bus MD[15:0]		Byte access strobe XMBE[1:0]	
		15-8	7-0	1	0
Address 0 8-bit access		-	7-0	-	○
Address 1 8-bit access		7-0	-	○	-
Address 2 8-bit access		-	7-0	-	○
Address 3 8-bit access		7-0	-	○	-
Address 0 16-bit access		15-8	7-0	○	○
ADDRESS 2 16-BIT ACCESS		15-8	7-0	○	○
Address 0 32-bit access	First time	15-8	7-0	○	○
	Second time	31-24	23-16	○	○

8.4.6. SDRAM Controller State Transitions

The following is the SDRAM state transition diagram for the SDRAM controller.

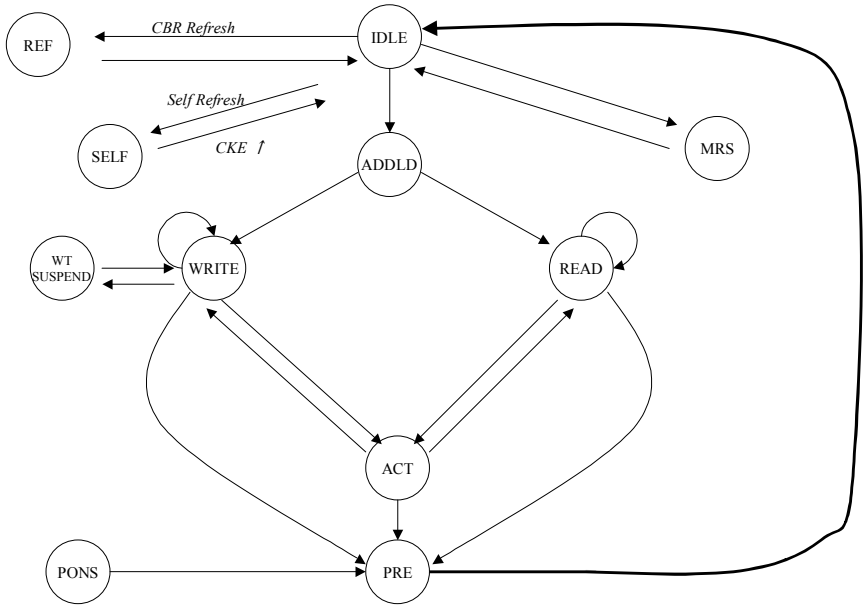


Figure 61 Transfer of the SDRAM status by this SDRAM controller

8.4.7. Addressing

This LSI supports 64Mbit, 128Mbit, and 256Mbit SDRAM. An example of the address line connections between this LSI and SDRAM is shown below.

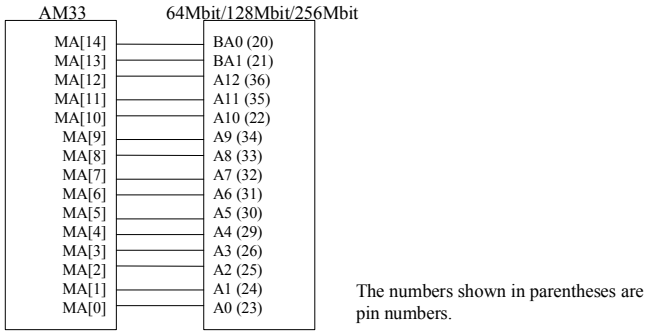


Figure 62 Example of the address connection between this LSI and SDRAM

MA13/MA14 are the bank select pins. In the cases of 64Mbit and 128Mbit SDRAM, MA12 are unused.
Because bits 21 and 22 of a memory address are the bank select bits, the maximum size of each bank is 1MB.

Table 62 Example of the address connection between this LSI and SDRAM

PA: Precharge All Bank, AP: Auto Precharge

MA output	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
SDRAM	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	BA	BA
SDRAM Pin	23	24	25	26	29	30	31	32	33	34	22	35	36	20	21
Row	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A24	A21	A22
Precharge	-	-	-	-	-	-	-	-	-	-	PA	-	-	A21	A22
Column	A1	A2	A3	A4	A5	A6	A7	A8	A23	X	AP	X	X	A21	A22

8.4.8. Timing Diagram

A typical timing chart is shown below. In this example, the burst length is 8 and the CAS latency is = 2.

8.4.8.1. Normal Operation Mode

8.4.8.1.1. Power up sequence setting

Initiate the power on sequence to SDRAM through writing “1” to the bit [6] (PONSEQ) of the SDRAMBUS register. Once the power on sequence has been initiated, it cannot be re-initiated until it is reset again.

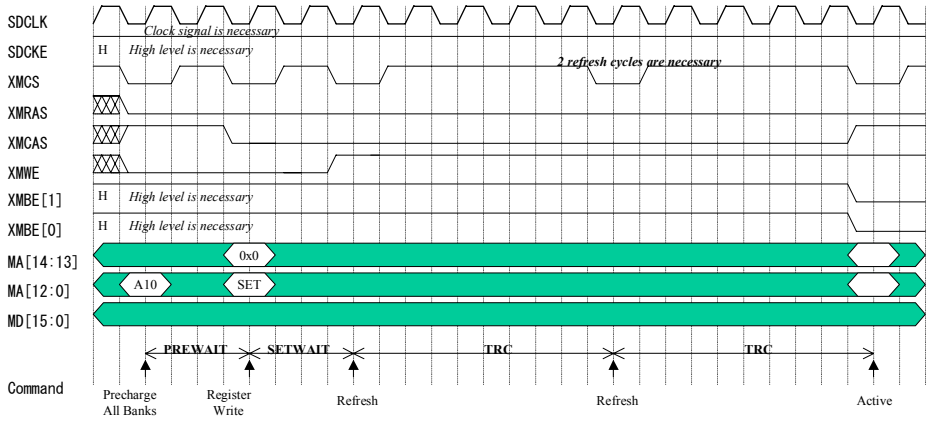


Figure 63 Setting of power up sequence

8.4.8.1.2. Mode Register Settings

The bits [31:30](CASLATE) and [17:16](SIZE) in the SDRAMBUS register have been rewritten, and then set the mode register for SDRAM.

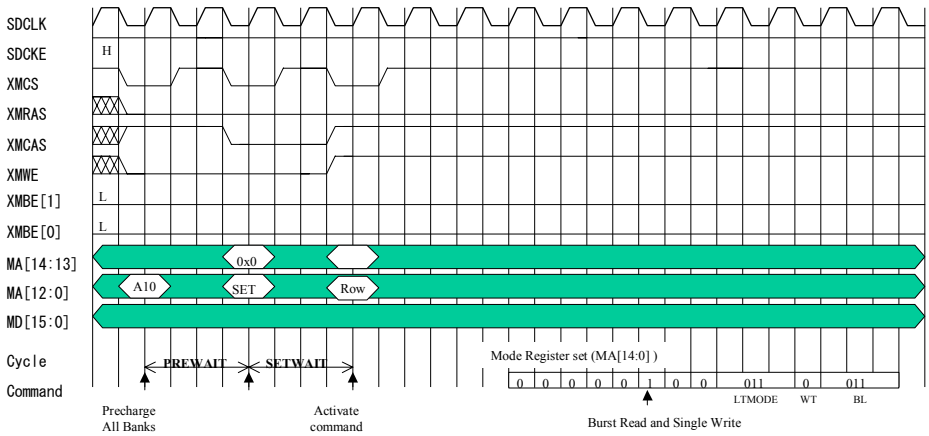


Figure 64 Setting of mode register

8.4.8.1.3. *Auto-refresh（CBR refresh）*

A count is started in SDCLK by writing “1” in the bit [2] (REFEN) of the SDRAMBUS register. When the count becomes the value of the bit in the SDREFCNT register, [13:0] (PERI), precharge all banks and issue the number of refresh commands which was set in the bit [21-20](REFNUM) of the SDRAMBUS register.

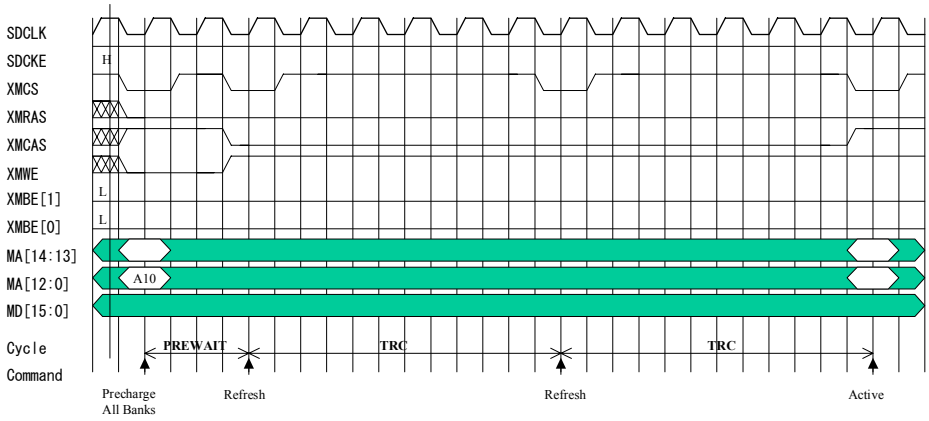


Figure 65 Auto-refresh (CBR refresh)

8.4.8.1.4. *Self-refresh*

Issue a self-refresh command for SDRAM by writing “1” to the bit [7] (SELFREQ) in the SDRAMBUS register. When “0” is written to the bit [7](SELFREQ) in the SDRAMBUS, or when access request occurs from BCU, deassert SDCKE and shift it from the self-refresh mode to the normal mode after TRC period.

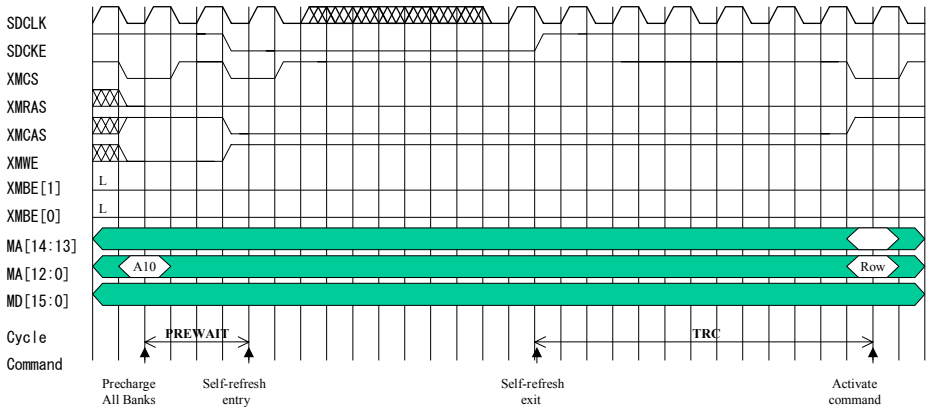


Figure 66 Self-refresh

8.4.8.1.5. Single write access (Page miss)

When a half-word access (16-bit) occurs, execute a single write access (16-bit) after issuing precharge and active command.

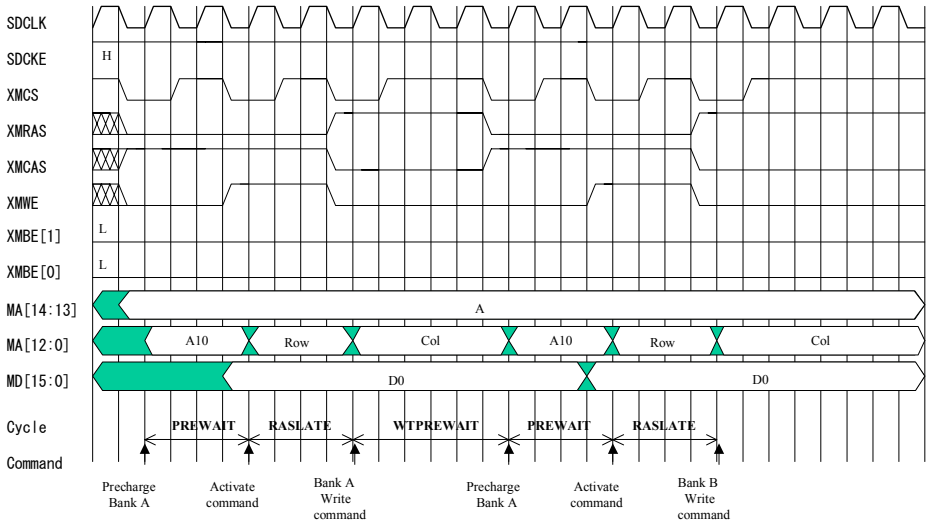


Figure 67 Single write access (Page miss)

8.4.8.1.6. 2-word write access (Page miss)

When a 32-bit write access occurs, issue 2 cycles of the write commands and execute a 2-word write access (32-bit) after issuing precharge and active command.

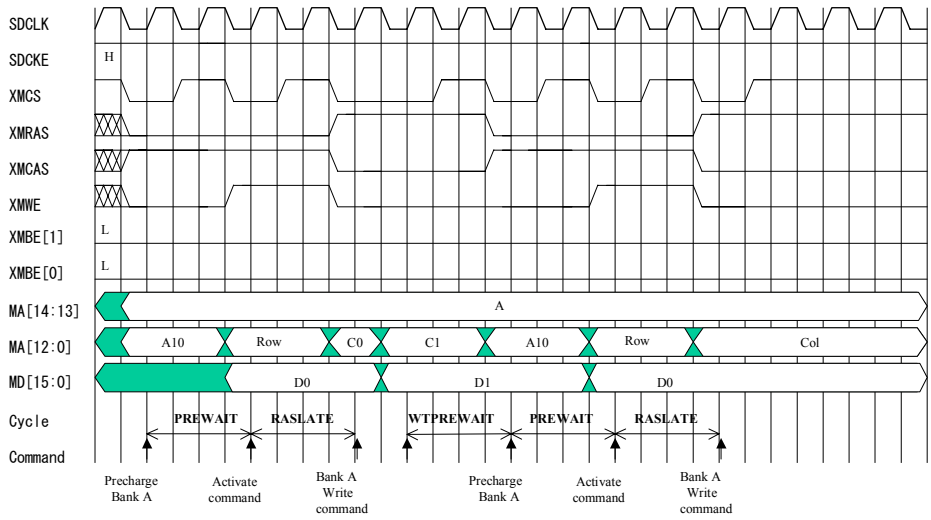


Figure 68 2-word write access (Page miss)

8.4.8.1.7. *8-word write access (Page miss)*

When 128-bit write access occurs like in the case of write back in CACHE and in the case of transferring DMA, issue 8 cycles of the write commands and execute a 8-word write access (128-bit) after issuing precharge and active command.

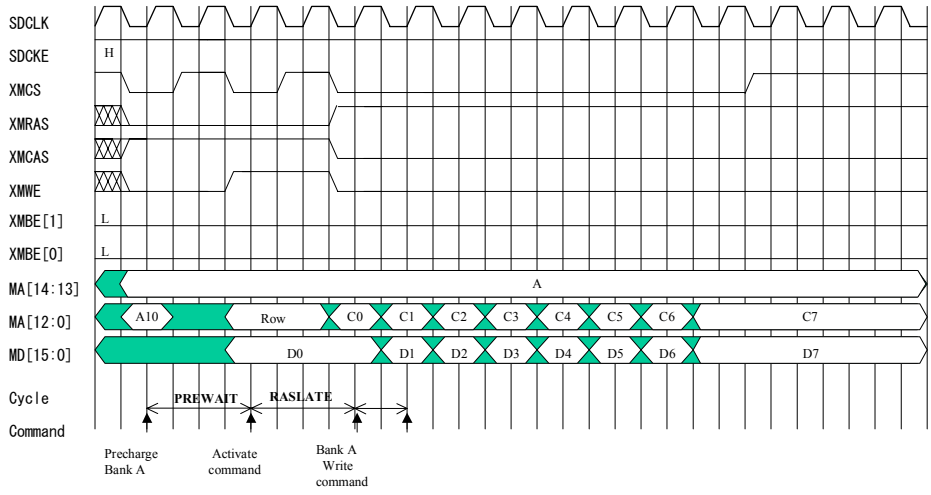


Figure 69 8-word write access (Page miss)

8.4.8.1.8. *1-word continuous write access (Page hit)*

When a 16-bit write access occurs for the same low address, issue a write command without issuing precharge and active command, and execute 1-word write access.

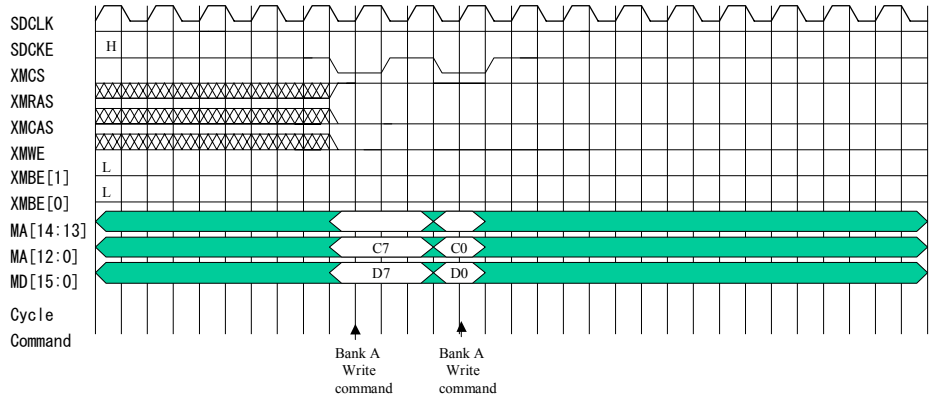


Figure 70 1-word continuous write access (Page hit)

8.4.8.1.9. 8-word continuous write access (Page hit)

When a 128-bit write access occurs for the same low address, issue 8 cycles of the write commands without precharge and active command and execute 8-word write access (128-bit).

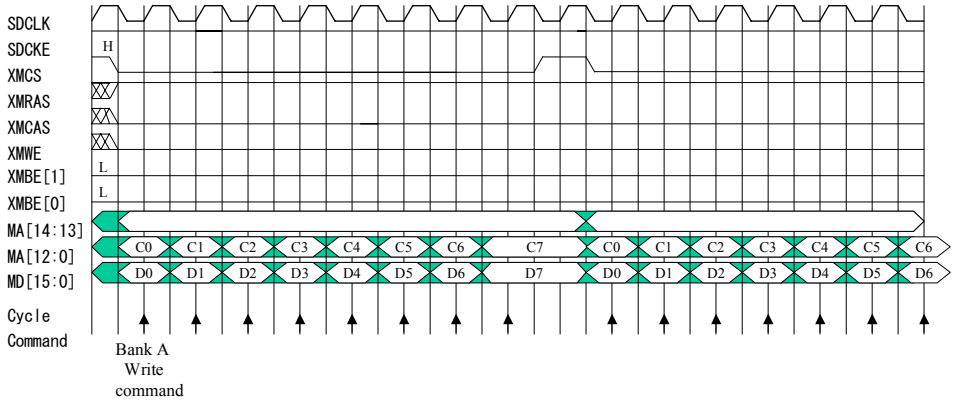


Figure 71 8-word continuous write access (Page hit)

8.4.8.1.10. Byte access

The lower 1 byte (upper 1 byte) of the word access is disabled through negating XMBE[0] (XMBE[1]) at issuing a write command.

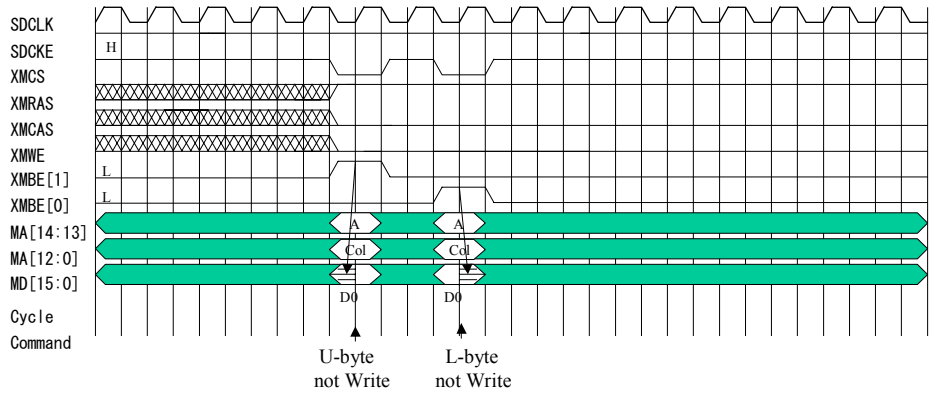


Figure 72 Byte access

8.4.8.1.11. 2-word burst read access (Precharge termination)

When a 32-bit read access occurs, issue a read command after issuing precharge and active command. Stop the burst read after the CASLATE cycle by issuing a precharge command after 2 cycles following the issue of the read command.

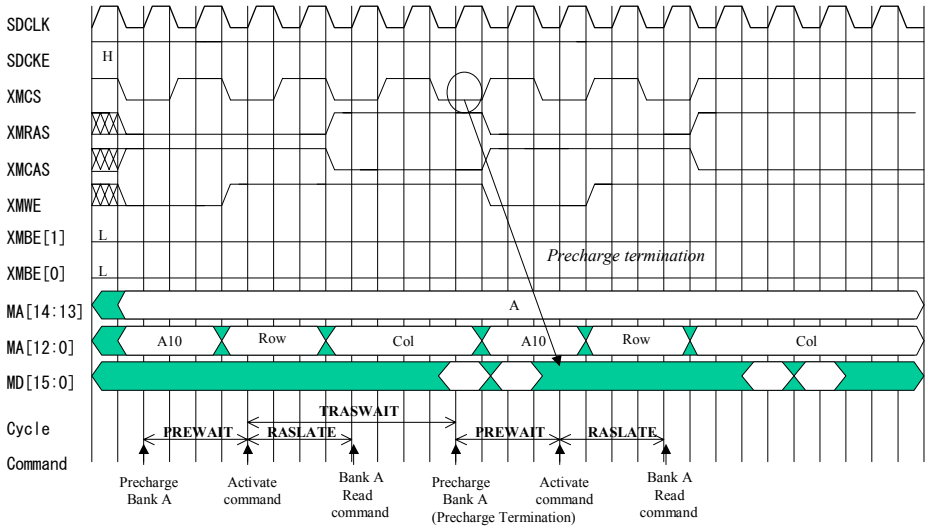


Figure 73 2-word burst read access (Precharge termination)

8.4.8.1.12. 2-word burst read access (Read termination)

When a 32-bit read access occurs, issue a read command after issuing precharge and active command. Stop the last burst read after the CASLATE cycle by issuing the next read command after 2 cycles following the issue of the read command.

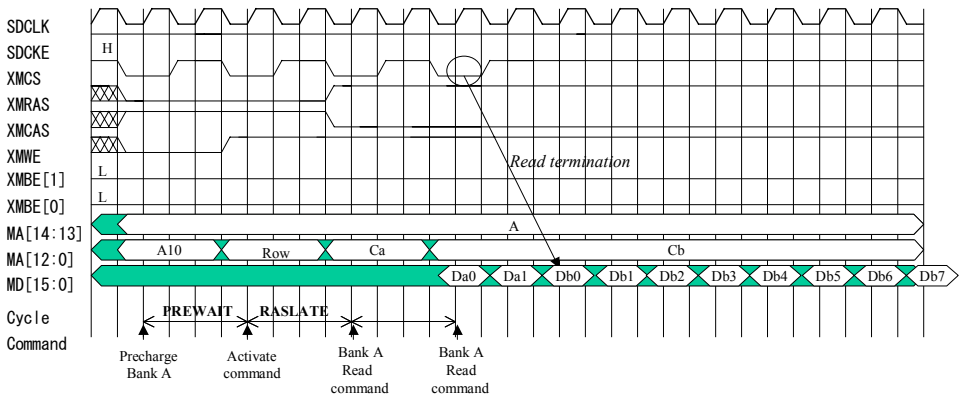


Figure 74 2-word burst read access (Read termination)

8.4.8.1.13. 2-word burst read access (Burst stop command termination)

When a 32-bit read access occurs, issue a read command after issuing precharge and active command. Stop the burst read after the BSTWAIT cycle by issuing a burst stop command after 2 cycles following the issue of the read command.

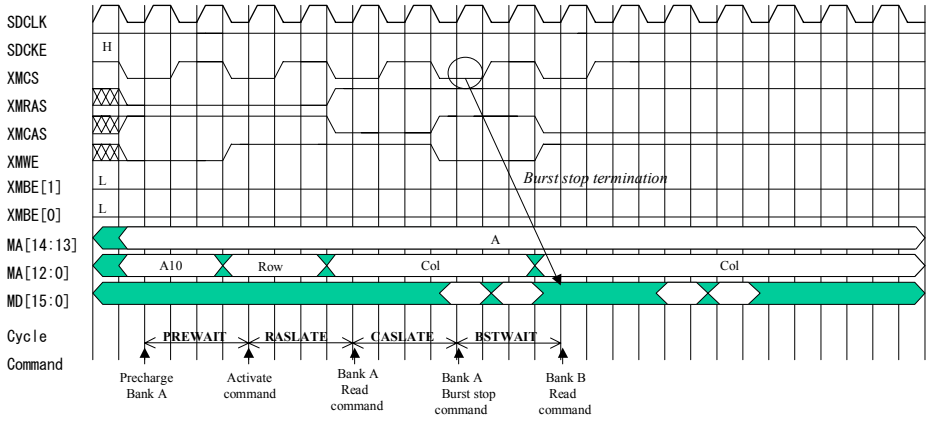


Figure 75 2-word burst read access (Burst stop command termination)

8.4.8.1.14. Continuous burst read access (Same bank & Page miss)

When executing the first and second burst read to the same bank, issue a precharge command before the PREWAIT cycle at the end of the first burst read, keep a low-active state after PREWAIT, and then issue the second read command after RASLATE. This includes the case that 128-bit write accesses occur continuously and to the same bank like at the refill of CACHE and at the transfer of DMA.

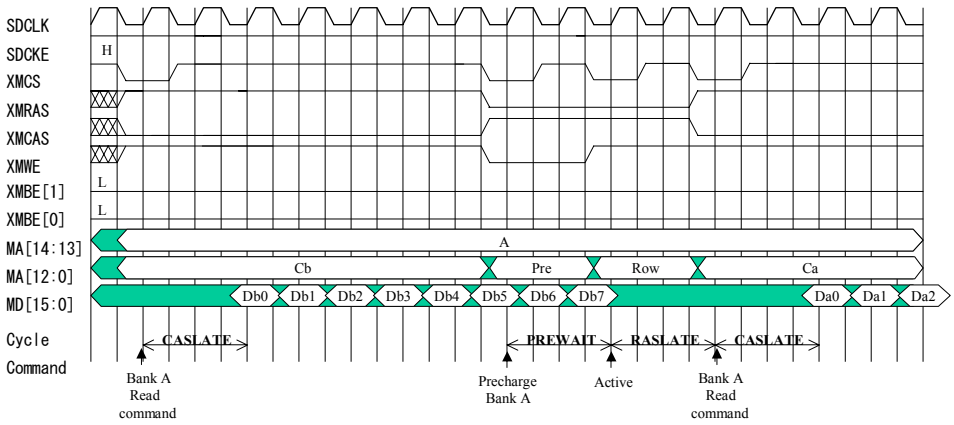


Figure 76 Continuous burst read access (Same bank & Page miss)

8.4.8.1.15. Continuous burst read access (Different Bank & Page miss)

When executing the first and second burst read to the different banks, issue a precharge command before the PREWAIT, RASLATE and CASLATE cycles at the end of the first burst read, keep a low-active state after PREWAIT, and then issue the second read command after RASLATE. This includes the case that 128-bit write accesses occur continuously and to the different banks like at the refill of CACHE and at the transfer of DMA.

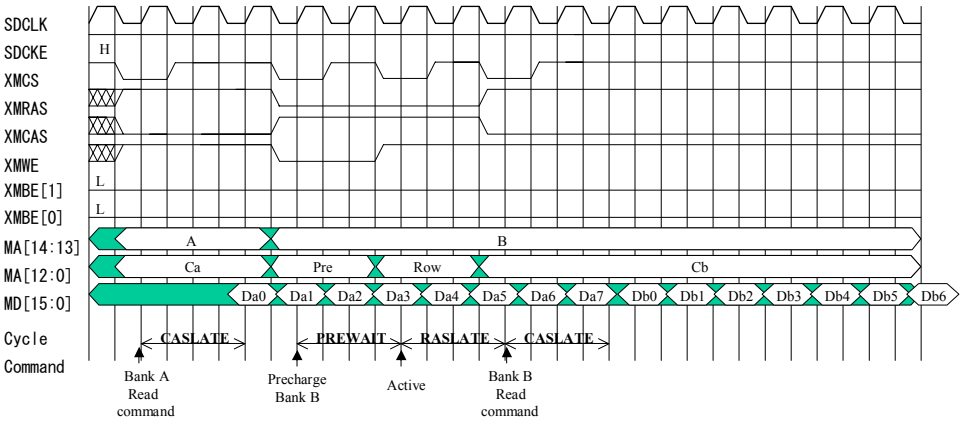


Figure 77 Continuous burst read access (Different Bank & Page miss)

8.4.8.2. Handshake Mode

The MDK input is provided for the purpose of interfacing with devices with an undefined CAS latency.
For read accesses, data transfer can be delayed by de-asserting MDK until the set number of CAS latency cycles have passed after the read command was input.
For write accesses, by de-asserting MDK until the set number of CAS latency cycles have passed after the write command was input, the next access will not start after the write command has ended until MDK is asserted again. Therefore, if a slave device has a write

buffer of at least burst size (16 bytes), then when there is data remaining in the write buffer, the start of the next read or write access can be prohibited by de-asserting MDK.
The MDK pin should be pulled up to high level on the board. In the case of normal SDRAM that does not require handshaking, the MDK signal is always asserted and access is conducted in normal mode.
The MDK signal is shared by multiple devices. Therefore, slave devices should always set this signal to the tristate condition. In addition, once this signal is de-asserted (i.e., is driven low from the tristate condition), it should be driven to the high-level condition once before being put back in the tristate condition.

8.4.8.2.1. Read access

When this SDRAM interface is used as an interface with a device with an undefined CAS latency, the data output in a single read can be delayed through asserting MDK until the number of the CAS latency cycles after inputting the read command.

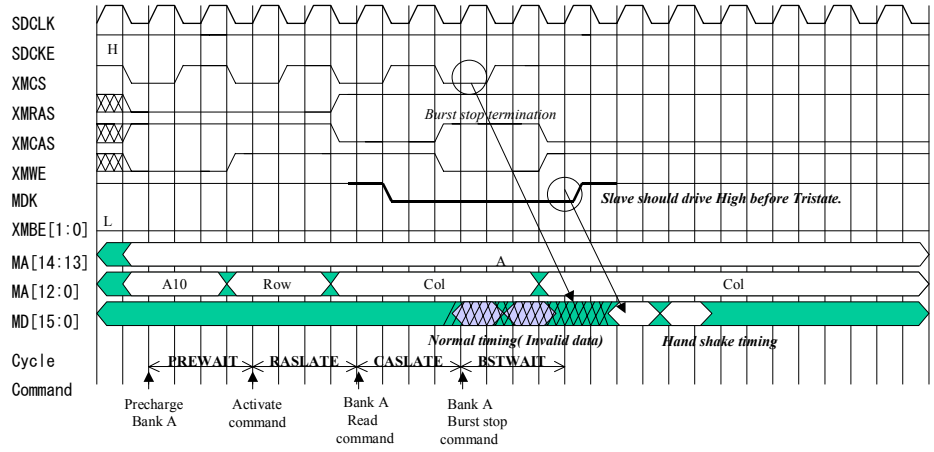


Figure 78 Read access

8.4.8.2.2. Burst Read Access

When this SDRAM interface is used as an interface with a device with an undefined CAS latency, the data output of a burst read can be delayed through asserting MDK until the number of the CAS Latency cycles after inputting a read command.

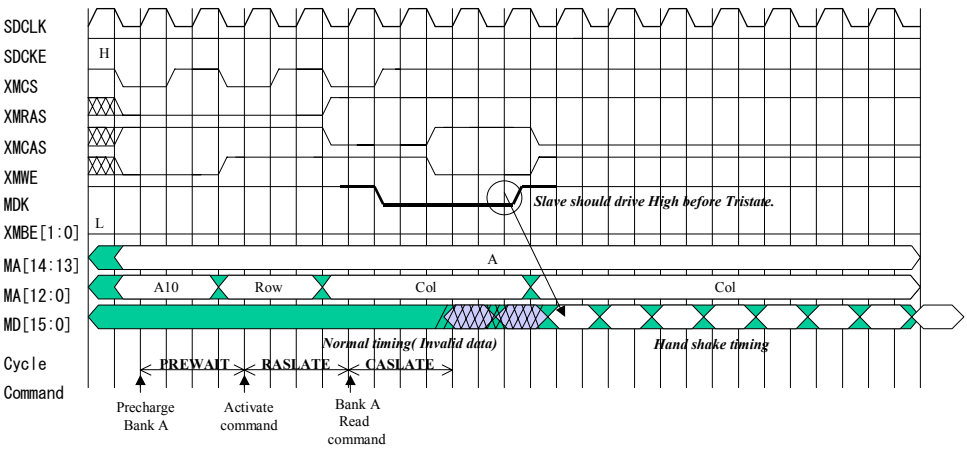


Figure 79 Burst read access

8.4.8.2.3. Write access

When this SDRAM interface is used as an interface with a device with an undefined CAS latency, the next access can be delayed until MDK has been reasserted through de-asserting MDK after inputting a write command.

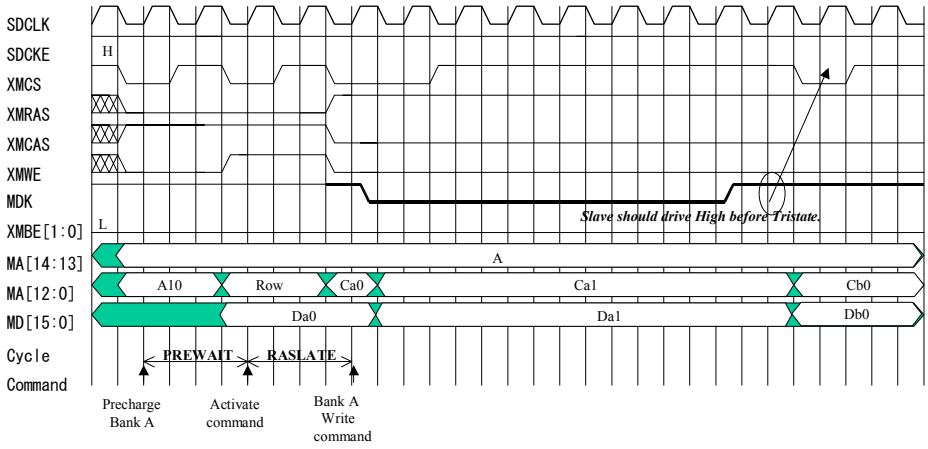


Figure 80 Write access

8.4.8.2.4. Burst Write Access

When this SDRAM interface is used as an interface with a device with an undefined CAS latency, the next command can be delayed until the write command has been completed and MDK has been reasserted through de-asserting MDK after inputting the write command.

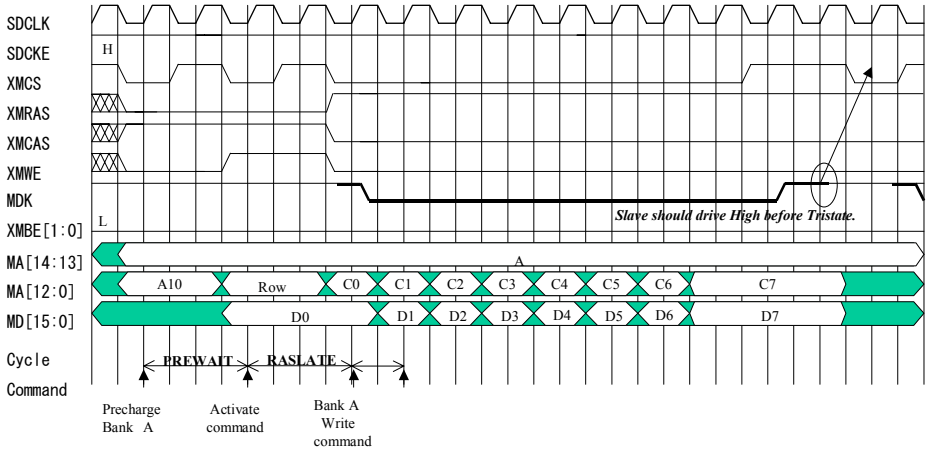


Figure 81 Burst write access

8.5. Cautions

Disable the refresh operation when changing the contents of the memory controller control registers. Operation is not guaranteed if the value in a register is changed while SDRAM is being accessed.

Chapter 9 DMA Controller (DMAC)

9

CHAPTER 9

DMA Controller (DMAC)

9.1. General

This LSI has an internal four-channel DMAC (Direct Memory Access Controller) that can perform data transfers between external memory, on-chip memory, and internal I/O.

9.2. Features

- Number of channels : 4 channels
- Transfer unit : 1/2/4/16 bytes
- Maximum number of transfer bytes : 1MB
- Initiation sources
  - External request : Request from the XDMR1/XDMR0 pin
  - External interrupt : External interrupt from the XIRQ1/XIRQ0 pin
  - Internal interrupt : Timer 0, 1, 2, 3 underflow
    Timer 6A compare/capture
    Serial 0, 1, 2 DMA initiation source
    Analog front end interrupt source
    A/D conversion end interrupt source
    IrDA interrupt source
    Real-time clock interrupt source
  - Software initiation (by writing a "1" to the TEN bit in DMnCTR)
- Transfer formats
  This LSI supports only 2 bus cycle transfer. 1 bus cycle transfer mode is not supported.
- Addressing mode
  "Fixed," "increment," and "decrement" can be specified for each source address and destination address. Incrementation and decrementation are performed automatically in accordance with the transfer unit.
- Transfer modes
  - Batch transfer
    the number of bytes specified in the DMnSIZ register are transferred in response to a single transfer request. When transfer is completed, a transfer end interrupt is generated.
  - Intermittent transfer
    After the number of transfers specified in the DMnCYC register have been performed in response to one transfer request, DMA transfer is interrupted and the DMAC waits for the next transfer request. When the number of bytes specified by the DMnSIZ register have all been transferred, a transfer end interrupt is generated.
- Priority ranking
  The priority ranking of the channels is DMA0 > DMA1 > DMA2 > DMA3. If there are transfer requests on multiple channels at the same time, the transfer on the channel with the highest priority is executed.
- Transfer requests from external devices
  External transfer requests can be issued by means of theXDMR1/XDMR0 signals or the XIRQ0/XIRQ1 signals.

9.3. Description of Registers

Address:	Symbol	Name	Number of bits	Initial value	Access size
0xD2000000	DM0CTR	DMA control register	32	0x 80000000	32
0xD2000004	DM0SRC	DMA source address register	32	0x 00000000	32
0xD2000008	DM0DST	DMA destination address register	32	0x 00000000	32
0xD200000C	DM0SIZ	DMA transfer word size register	32	0x 00000000	32
0xD2000010	DM0CYC	DMA intermittent transfer size register	32	0x 00000000	32
0xD2000100	DM1CTR	DMA control register	32	0x 80000000	32
0xD2000104	DM1SRC	DMA source address register	32	0x 00000000	32
0xD2000108	DM1DST	DMA destination address register	32	0x 00000000	32
0xD200010C	DM1 SIZ	DMA transfer word size register	32	0x 00000000	32
0xD2000110	DM1CYC	DMA intermittent transfer size register	32	0x 00000000	32
0xD2000200	DM2CTR	DMA control register	32	0x 80000000	32
0xD2000204	DM2SRC	DMA source address register	32	0x 00000000	32
0xD2000208	DM2DST	DMA destination address register	32	0x 00000000	32
0xD200020C	DM2SIZ	DMA transfer word size register	32	0x 00000000	32
0xD2000210	DM2CYC	DMA intermittent transfer size register	32	0x 00000000	32
0xD2000300	DM3CTR	DMA control register	32	0x 80000000	32
0xD2000304	DM3SRC	DMA source address register	32	0x 00000000	32
0xD2000308	DM3DST	DMA destination address register	32	0x 00000000	32
0xD200030C	DM3SIZ	DMA transfer word size register	32	0x 00000000	32
0xD2000310	DM3CYC	DMA intermittent transfer size register	32	0x 00000000	32

9.3.1. DMA Control Register

Register symbol: DMNCTR
Address: 0xD2000000 + (0x100 * n)
Purpose: This register controls DMA channel n, and indicates its status.

Bit	31	30	29	28	27	26	25	24
Bit name	XEND	Reserved						RQF
Initial value	1	0						0
R/W	RW	R						R
Bit	23	22	21	20	19	18	17	16
Bit name	RESERVED					RQM		TEN
Initial value	0					00		0
R/W	R					RW		RW
Bit	15	14	13	12	11	10	9	8
Bit name	Reserved	UT[1:0]		TM[1:0]		DAM[2:0]		
Initial value	0	00		00		000		
R/W	R	RW		RW		RW		
Bit	7	6	5	4	3	2	1	0
Bit name	SAM[2:0]			BG[4:0]				
Initial value	000			0000				
R/W	RW			RW				

Bit	Bit name	Description
31	XEND	DMA transfer end flag This bit indicates the status of the DMA transfer end interrupt. This flag can be cleared by writing a "1" to this bit. A "0" cannot be written to this bit. 0 : A DMA transfer end interrupt is being generated. 1 : A DMA transfer end interrupt is not being generated.
30-25	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
24	RQF	DMA transfer request flag This bit is "1" since from the point when a DMA transfer request is generated until the DMA transfer ends or until the next transfer interruption during intermittent transfer.
23-19	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
18-17	RQM[1:0]	External request input source mode This field sets the operation mode for transfer requests by external pins (XDMR0/XDMR1). 00 : Falling edge (Negedge) 01 : Rising edge (Posedge) 10 : Low level 11 : High level
16	TEN	DMA channel transfer enable This bit enables the DMA channel. 0 : DMA transfer disabled 1 : DMA transfer enabled A DMA transfer begins when the software source is set as the transfer request source (BG[4:0]=00000) and a "1" is written to the TEN bit. If a source other than the software source is set, the DMA transfer begins as soon as the transfer request is generated. No matter what the transfer source is, this bit is not cleared by an interruption of transfer in intermittent transfer mode.
15	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
14-13	UT[1:0]	DMA transfer unit This field sets the transfer unit. 00 : 1 byte 01 : 2 bytes 10 : 4 bytes 11 : 16 bytes
12-11	TM[1:0]	DMA transfer mode This field specifies the transfer mode. 00 : Batch transfer The number of bytes specified in the DMnSIZ register are transferred. 01 : Setting prohibited 10 : Intermittent transfer The number of bytes specified in the DMnCYC register are transferred in response to one transfer request. After the number of bytes specified by the DMnSIZ register have all been transferred, a transfer end interrupt is generated. 11 : Setting prohibited
10-8	DAM[2:0]	DMA transfer destination address mode This field specifies the address mode for the transfer destination address. 000 : Increment

Bit	Bit name	Description
		001 : Decrement 010 : Fixed 011 - 111 : Setting prohibited
7-5	SAM[2:0]	DMA transfer source address mode This field specifies the address mode for the transfer source address. 000 : Increment 001 : Decrement 010 : Fixed 011 - 111 : Setting prohibited
4-0	BG[4:0]	Transfer request source This field sets the transfer request source. 00000 : Software source 00001 : Setting prohibited 00010 : Serial 0 sending DMA initiation source 00011 : Serial 0 receiving DMA initiation source 00100 : Serial 1 sending DMA initiation source 00101 : Serial 1 receiving DMA initiation source 00110 : Serial 2 sending DMA initiation source 00111 : Serial 2 receiving DMA initiation source 01000 : Timer 0 underflow source 01001 : Timer 1 underflow source 01010 : Timer 2 underflow source 01011 : Timer 3 underflow source 01100 : Timer 6A compare/capture source 01101 : Analog front end interrupt source 01110 : A/D conversion end interrupt source 01111 : IrDA interrupt source 10000 : Real-time clock interrupt source 10001 : XIRQ0 pin input source (Level-edge condition is defined in the INTC block) 10010 : XIRQ1 pin input source (Level-edge condition is defined in the INTC block) 10011 : External request 0 (XDMR0 pin) input source 10100 : External request 1 (XDMR1 pin) input source 10101 – 11111 : Setting prohibited

9.3.2. DMA Source Address Register

Register symbol: DMNSRC
Address: 0xD2000004 + (0x100 * n)
Purpose: This register specifies the transfer source address for the DMA channel.

Bit	31	30	29	28	27	26	25	24
Bit name	SA[31:24]							
Initial value	0							
R/W	RW							
Bit	23	22	21	20	19	18	17	16
Bit name	SA[23:16]							
Initial value	0							
R/W	RW							
Bit	15	14	13	12	11	10	9	8
Bit name	SA[15:8]							

Initial value	0							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	SA[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
31-0	SA[32:0]	Transfer source address This field specifies the initial value for the transfer source address.

9.3.3. DMA Destination Address Register

Register symbol: DMNDST
Address: 0xD2000008 + (0x100 * n)
Purpose: This register specifies the transfer destination address for the DMA channel.

Bit	31	30	29	28	27	26	25	24
Bit name	DA[31:24]							
Initial value	0							
R/W	RW							
Bit	23	22	21	20	19	18	17	16
Bit name	DA[23:16]							
Initial value	0							
R/W	RW							
Bit	15	14	13	12	11	10	9	8
Bit name	DA[15:8]							
Initial value	0							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	DA[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
31-0	DA[31:0]	Transfer destination address This field specifies the initial value for the transfer destination address.

9.3.4. DMA Transfer Word Size Register

Register symbol: DMNSIZ
Address: 0xD200000c + (0x100 * n)
Purpose: This register specifies the number of bytes to be transferred for the DMA channel.

Bit	31	30	29	28	27	26	25	24
Bit name	Reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	Reserved				CT[19:16]			
Initial value	0				0			

R/W	R				RW			
Bit	15	14	13	12	11	10	9	8
Bit name	CT[15:8]							
Initial value	0							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	CT[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
31-20	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
19-0	CT[19:0]	Number of transfer bytes This field specifies the total number of bytes to be transferred. If "0" is specified, "0 bytes" is set as the number of transfer bytes. However, operation is not guaranteed if "0" is specified.

9.3.5. DMA Intermittent Transfer Count Register

Register symbol: DMNCYC
Address: 0xD200010 + (0x100 * n)
Purpose: This register specifies the number of intermittent transfers for the DMA channel.

Bit	31	30	29	28	27	26	25	24
Bit name	Reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	Reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	Reserved							
Initial value	0							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	CYC[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
31-8	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
7-0	CYC[7:0]	Number of intermittent transfers This field specifies the number of intermittent transfers. 0x00 : Interrupt transfer after one transfer 0xFF : Interrupt transfer after 256 transfers

9.4. Description of Operation

9.4.1. Types of Transfers

The DMA interface only supports 2-bus cycle DMA transfer which generates the read access from the transfer source, stores the read data in a buffer, and then generates a write access to the transfer destination.

9.4.2. Transfer modes

There are two transfer modes: batch transfer mode and intermittent transfer mode. The transfer mode is selected through the TM[1:0] bits in the DMnCTR register.

9.4.2.1. Batch Transfer Mode

The number of bytes specified in the DMnSIZ register are transferred.

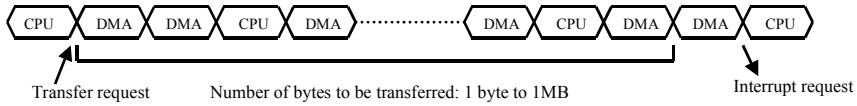


Figure 82 Batch Transfer Mode

9.4.2.2. Intermittent Transfer Mode

In an intermittent transfer mode, the number of transfers specified in the DMnCYC register are performed in response to one transfer request. After the number of bytes specified in the DMnSIZ register have all been transferred, a transfer end interrupt is generated.

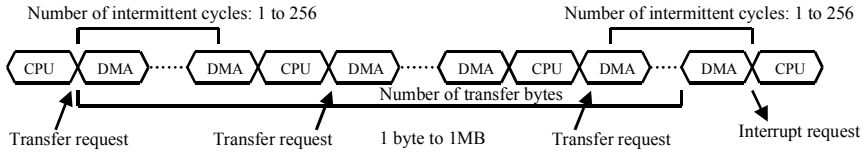


Figure 83 Intermittent Transfer Mode

9.4.3. Priority ranking

When there are multiple requests to use the bus, they are processed according to the following priority ranking:
DMA0 > DMA1 > DMA2 > DMA3

9.4.4. Bus Lock

The DMA controller occupies the bus controller's (BCU's) DMA bus until the end of the transfer (i.e. after reading from the source up to writing to the destination) of a transfer unit (set by DMnCTR.UT[1:0]); once the transfer of a transfer unit ends, the DMA controller temporarily releases the bus.
If there is a request from a DMA channel with a higher priority or a request from an external master device while a DMA transfer is in progress, the higher priority transfer is performed after the transfer of the current transfer unit ends.

9.4.5. Transfer unit:

The data transfer unit can be specified as either 1 byte, 2 bytes, 4 bytes or 16 bytes through the UT[1:0] field in the DMnCTR register.
As will be described later, only an address that is aligned with a 4-byte boundary can be specified for the transfer source and the transfer destination if the 16-byte transfer unit is specified. If the beginning or end of the transfer is not aligned with a 16-byte boundary, only the necessary number (4, 8, or 12) of bytes is performed as one transfer.

9.4.6. Number of intermittent transfers

If intermittent transfer is specified in the DMA control register as the DMA transfer mode, the DMA controller interrupts the transfer after the number of transfers specified in the intermittent transfer number register.
An address that is aligned with a 4-byte boundary can be specified for the transfer source and the transfer destination if the 16-byte transfer unit is specified. If the beginning or end of the transfer is not aligned with a 16-byte boundary, the transfer of the necessary number (4, 8, or 12) of bytes is counted as one transfer.

9.4.7. Transfer Addresses

9.4.7.1. Specification of Transfer Source and Transfer Destination Addresses

The starting addresses for the data transfer source and transfer destination are specified by the DMnSRC register and the DMnDST register, respectively. The value in the DMnSRC/DMnDST register indicates the next transfer address. Therefore, there is no need to reset these registers when executing another DMA in contiguous areas after a transfer ends.
There are three addressing modes that can be specified independently for both the transfer source and the transfer destination: increment, decrement, and fixed. If increment or decrement is specified as the addressing mode, the value in the DMnSRC/DMnDST register is incremented or decremented by the number of bytes in the transfer unit after the transfer of the data specified as the transfer unit is completed.

9.4.7.2. Relationship between the Transfer Unit and Address Alignment

If "1 byte," "2 bytes," "4 bytes," or "16 bytes" is specified as the transfer unit, only addresses that are aligned with 1-byte, 2-byte, 4-byte, or 16-byte address boundaries, respectively, are valid for the transfer source and the transfer destination. Operation is not guaranteed if an address that is aligned with a different byte from a 4-byte boundary is specified for the transfer source and the transfer destination.
If the addressing mode for the transfer source is different from the addressing mode for the transfer destination (for example, if one is specified as "increment" and the other is specified as "decrement"), specify addresses that are aligned with the number of bytes specified as the transfer unit. For example, if 2-byte transfer is specified, specify an address that is aligned with a 2-byte boundary.

9.4.7.3. Address Alignment Requirements for 16-byte Transfers

If the transfer unit is 16 bytes, an address that is aligned with a 4-byte boundary can normally be specified for the transfer source or the transfer destination. If the transfer unit is 16 bytes and the transfer destination addressing mode is set to "fixed," specify an address that is aligned with a 16-byte boundary for the transfer destination address. Similarly, if the

transfer unit is 16 bytes and the transfer source addressing mode is set to "fixed," specify an address that is aligned with a 16-byte boundary for the transfer source address.
If the transfer unit is specified as 16 bytes and an address is aligned with a 16-byte boundary for the transfer source or the transfer destination, the transfer of the two or more DMA channels is not guaranteed.

9.4.8. Transfer Size

The DMnSIZ register specifies the number of bytes of data that are to be transferred.
Operation is not guaranteed if DMA is initiated with a transfer size of "0."
The DMnSIZ register indicates the number of bytes remaining to be transferred. Therefore, because the value in this register becomes "0" at the end of the transfer, it needs to be reset if another DMA transfer is to be performed in a contiguous area.

9.4.9. Transfer Initiation

9.4.9.1. Transfer Initiation by an External Request

A transfer can be initiated by an edge or level input from the XDMR0 or XDMR1 pin. Set the source with the BG bit in the DMnCTR register, set the edge or level polarity with the RQM bit, and then set the TEN bit to "1." Even if edge input is set, maintain the system clock (SYSCLK) level for at least two cycles.
The XDMR0 and XDMR1 pins are dual-purpose pins. Before using the XDMR0/XDMR1 function of these pins, the appropriate general-purpose I/O port settings are needed.
Operation is not guaranteed if the general-purpose I/O port settings are changed to a different function while using the XDMR0/XDMR1 function of these pins.
If there is a new transfer request due to edge input during a DMA transfer on the DMA channel, execution of the DMA transfer in response to the new transfer request is not guaranteed.
Operation is not guaranteed if transfer initiation that is performed by the same external request source is specified for multiple channels

9.4.9.2. Transfer Initiation by an External Interrupt

A transfer can be initiated by an edge or level input from the XIRQ0 or XIRQ1 pin. The edge polarity setting in the interrupt controller (INTC) is valid. An edge mode must be set for the interrupt controller. Set the source with the BG bit in the DMnCTR register, and then set the TEN bit to "1." Writing to the BG bit and the TEN bit can be carried out in a single write operation.
If there is a new transfer request during a DMA transfer on the DMA channel, theexecution of the DMA transfer in response to the new transfer request is not guaranteed.
Operation is not guaranteed if transfer initiation that is performed by the same external interrupt source is specified for multiple channels.

9.4.9.3. Transfer Initiation by an Internal Interrupt

A transfer can be initiated by an interrupt source from an internal peripheral device. Set the source with the BG bit in the DMnCTR register, and then set the TEN bit to "1." Writing to the BG bit and the TEN bit can be carried out in a single write operation.
If there is a transfer request during a DMA transfer on the DMA channel, the execution of the DMA transfer in response to the new transfer request is not guaranteed.
Operation is not guaranteed if transfer initiation that is performed by the same internal interrupt source is specified for multiple channels.

9.4.9.4. Transfer Initiation by Software

A transfer can be initiated by setting the software source with the BG bit in the DMnCTR register and then setting the TEN bit to "1." Writing to the BG bit and the TEN bit can be carried out in a single write operation.
If there is a transfer request during a DMA transfer on the DMA channel, the execution of the DMA transfer in response to the new transfer request is not guaranteed. In the case of the intermittent transfer initiated by software, confirm that the previously requested transfer has been completed by clearing the RQF bit to "0" before executing the new transfer request.

9.4.10. Transfer Start/Interruption/End/Forced Termination

Transfer start
When the initiating source that is specified in the BG field of the DMnCTR register is generated, transfer begins if the TEN bit in the DMnCTR register is "1."
Transfer interruption
In intermittent transfer mode, once the specified number of intermittent transfers ends, the transfer process is interrupted and the controller waits for a new transfer request. In this case, although the RQF bit is "0," the TEN bit remains "1." Therefore, it is not necessary to enable the channel again.
Transfer end
Once the number of bytes set in the DMA transfer size register (DMnSIZ) have been transferred, the transfer ends.
Forced termination
Before a DMA transfer ends, it can be forcibly terminated by clearing the TEN bit in the DMnCTR register to "0." The contents of the register settings are not guaranteed in the event of a forced termination. No interrupt is generated in the event of a forced termination.

9.5. Cautions

9.5.1. Cautions about specifying transfer address

There are the cautions about specifying the addresses of the transfer destination and transfer source. Refer to 9.4.7 Transfer Addresses, page270.

9.5.2. Cautions about specifying the transfer size

There are the cautions about specifying the transfer size. Refer to 9.4.8 Transfer Size, page270.

9.5.3. Cautions about DMA transfer bus error

When a bus error occurs in writing by DMA, the transfer of a subsequent DMA may be stopped.

When a bus error occurs in DMA transfer, all valid DMA channel tranfers at the bus occurrence can not be guaranteed.

Chapter 10 8-bit Timer Module (TM8)

10

10.1. General

This LSI has four internal 8-bit down counters. These counters can be used as interval timers and event counter timers.

10.2. Features

- Clock sources
  Internal clocks: IOCLK, 1/8 IOCLK, 1/32 IOCLK, and timer 0 to 3 underflow
  External clock input: Counts the rising edge on the pin input.
  The clock sources that can be used differ according to the timer. Refer to the description of the TMnCK field in the TMnMD register.
- Cascaded connection
  Timers 0 through 3 can be cascaded together through the programming, allowing them to be used as a pure 16-, 24- or 32-bit timer.
- Interrupts: An interrupt is generated when a timer underflow occurs.
- General-purpose serial interface reference clock generation (timers 0, 1, 2, and 3)
- Start-stop synchronization-only serial interface reference clock generation (timers 2 and 3)
- DMA transfer can be initiated when an interrupt request is generated (timers 0, 1, 2, and 3)

Table 63 8-bit timer function chart

Timer	Timer 0	Timer 1	Timer 2	Timer 3
Up/down count	Down count	Down count	Down count	Down count
Interval timer	○	○	○	○
Event counter	○	○	○	○
Timer output	○	○	○	○
Interrupts	○	○	○	○
DMA initiation	○	○	○	○
UART0 clock source	○	×	○	×
UART1 clock source	×	○	×	○
UART2 clock source	×	×	○	○
Cascaded connection		○	○	○
Clock sources	Refer to the description of the TMnMD register.			

10.3. Description of Registers

Table 64 8-bit timer register

Address:	Symbol	Name	NUMBER OF BITS	Initial value	Access size
0xD4003000	TM0MD	Timer 0 mode register	8	0x 00	8,16,32
0xD4003001	TM1MD	Timer 1 mode register	8	0x 00	8
0xD4003002	TM2MD	Timer 2 mode register	8	0x 00	8,16
0xD4003003	TM3MD	Timer 3 mode register	8	0x 00	8

0xD4003010	TM0BR	Timer 0 base register	8	0x 00	8,16,32
0xD4003011	TM1BR	Timer 1 base register	8	0x 00	8
0xD4003012	TM2BR	Timer 2 base register	8	0x 00	8,16
0xD4003013	TM3BR	Timer 3 base register	8	0x 00	8
0xD4003020	TM0BC	Timer 0 binary counter	8	0x 00	8,16,32
0xD4003021	TM1BC	Timer 1 binary counter	8	0x 00	8
0xD4003022	TM2BC	Timer 2 binary counter	8	0x 00	8,16
0xD4003023	TM3BC	Timer 3 binary counter	8	0x 00	8
0xD4003071	TMPSCNT	Timer prescaler control register	8	0x 00	8

10.3.1. Timer Mode Register

Register symbol: TMNMD
Address: TM0MD: 0xD4003000
TM1MD: 0xD4003001
TM2MD: 0xD4003002
TM3MD: 0xD4003003
Purpose: This register sets the control conditions for timer n operation.

Bit	7	6	5	4	3	2	1	0
Bit name	TMnCNE	TMnLDE	Reserved			TMnCK[2:0]		
Initial value	0	0	0			0		
R/W	RW	RW	R			RW		

Bit	Bit name	Description
7	TMNCNE	Timer enable This bit controls the timer n count operation. 0 : Stops operation 1 : Enables operation
6	TMnLDE	Timer load enable This bit initializes timer n. 0 : Normal operation 1 : Loads the value in TMnBR into TMnBC. Resets the timer output n low.
5-3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2-0	TMnCK[2:0]	Timer clock source selection This field selects the clock source. Refer to the below table.

TMnCK[2:0]	Timer 0	Timer 1	Timer 2	Timer 3
000	IOCLK	IOCLK	IOCLK	IOCLK
001	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK
010	1/32 IOCLK	1/32 IOCLK	1/32 IOCLK	1/32 IOCLK
011	IOCLK	Cascaded with timer 0	Cascaded with timer 1	Cascaded with timer 2
100	SETTING PROHIBITED	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Setting prohibited	Timer 1 underflow	Timer 1 underflow
110	Timer 2 underflow	Timer 2 underflow	Setting prohibited	Timer 2 underflow
111	TM0IO	TM1IO	TM2IO	TM3IO

	pin input	pin input	pin input	pin input

When using the 1/8 IOCLK or 1/32 IOCLK setting, the prescaler settings must be enabled by TMPSCNT.
Set TMnCK while TMnCNE is "0."
Set TMnCNE to "1" while TMnLDE is "0."
Set TMnLDE to "1" while TMnCNE is "0."
Operation is not guaranteed if TMnCNE and TMnLDE are both "1" at the same time.

10.3.2. Timer Base Register

Register symbol: TMNBR
Address: TM0BR: 0xD4003010
TM1BR: 0xD4003011
TM2BR: 0xD4003012
TM3BR: 0xD4003013
Purpose: This register sets the timer n count period.

Bit	7	6	5	4	3	2	1	0
Bit name	TMnBR[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
7-0	TMnBR[7:0]	Timer n base register This field sets the initial value and the underflow cycle for the timer n binary counter (TMnBC). The underflow cycle is equal to the value set in TMnBR[7:0] plus one.

10.3.3. Timer Binary Counter

Register symbol: TMNBC
Address: TM0BC: 0xD4003020
TM1BC: 0xD4003021
TM2BC: 0xD4003022
TM3BC: 0xD4003023
Purpose: This register is the timer n binary counter. The value in this counter can be read.

Bit	7	6	5	4	3	2	1	0
Bit name	TMnBC[7:0]							
Initial value	0							
R/W	R							

Bit	Bit name	Description
7-0	TMnBC[7:0]	Timer n binary count The timer n binary count value can be read. This counter uses the value that is set in TMnBR as its initial value. An underflow occurs and an interrupt is generated when this counter counts the value set in TMnBR, plus one. The value in TMnBC is then initialized again to the value set in TMnBR.

10.3.4. Timer prescaler control register

Register symbol: TMPSCNT

Address: 0xD4003071
Purpose: This register enables the timer prescaler.
This prescaler is common for both 8-bit and 16-bit counters.

Bit	7	6	5	4	3	2	1	0
Bit name	TMPSCNE	Reserved						
Initial value	0	0						
RW	RW	R						

Bit	Bit name	Description
7	TMPSCNE	Timer prescaler enable This bit controls the prescaler operation. 0 : Stops prescaler operation 1 : Enables prescaler operation
6-0	Reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.

10.4. Description of Operation

When using an 8-bit timer as an interval timer, make the settings according to the below-described procedure. The timer will then operate as an interval timer that generates an interrupt on the set cycle.
When several 8-bit timers are cascaded together to form a 16-, 24-, or 32-bit timer, refer to 10.4.4"Cascaded connection" Page 278. This also applies when generating a serial interface reference clock.

10.4.1. Operation Start Procedure

(1) Timer frequency division ratio setting
Set the frequency division ratio in TMnBR. The interrupt cycle is equal to (TMnBR setting + 1) × clock source cycle.
(2) Clock source selection
The clock source is selected by TMnCK[2:0] in the TMnMD register.
Set the clock source while counting is stopped. Changing the clock source is prohibited while the counting operation is in progress.
When using 1/8 IOCLK or 1/32 IOCLK as the clock source, set TMPSCNE in the TMPSCNT register to "1," and enable the prescaler operation.
(3) Timer initialization
Initialize timer n by setting TMnLDE in the TMnMD register to "1." The value that is set in TMnBR is loaded into TMnBC as the initial value, and the timer output is reset.
After initialization, be sure to set TMnLDE to "0" in order to put the timer back in normal operation mode.
(4) I/O port setting
This setting is necessary only for using the timer output.
Select the timer in the port 0 mode register of the I/O port, and then set the output pin in the port 0 TM pin input/output control register.
(5) Enabling the counting operation
The counting operation starts when TMnCNE in the TMnMD register is set to "1."
Once the counting operation is enabled, an underflow interrupt request is generated on a constant cycle. Furthermore, the pin output inverts each time that the interrupt is generated, and the setting value in TMnBR is loaded into TMnBC.
If the value in the TMnBR register is changed while the counting operation is in progress, a new value is loaded as the initial value at the next occurrence of an underflow, and then the interrupt cycle changes.

10.4.2. Operation Stop Procedure

(1) Stop the timer counting operation.
The counting operation stops if TMnCNE in the TMnMD register is set to "0."
(2) Initialize the timer if necessary.
If TMnLDE is set to "1," the value that is set in TMnBR is loaded into TMnBC as the initial value, and the timer output is reset.

After the timer stops, the binary counter and the pin output are maintained in their previous states if TMnLDE is not set to "1." The counting operation can be resumed from its previous state by setting TMnCNE to "1."

10.4.3. Clock Source Selection

(1) Counting operation stop
The counting operation stops if TMnCNE in the TMnMD register is set to "0."
(2) Timer initialization
If TMnLDE in the TMnMD register is set to "1," the value that is set in TMnBR is loaded into TMnBC as the initial value, and the timer output is reset.
After the timer stops, the binary counter is maintained in its previous state if TMnLDE is not set to "1." The counting operation can be resumed from its previous state by setting TMnCNE to "1."

10.4.4. Cascaded connection

The 8-bit timers can be cascaded together in the settings described below as 8-bit timer cascade connection.

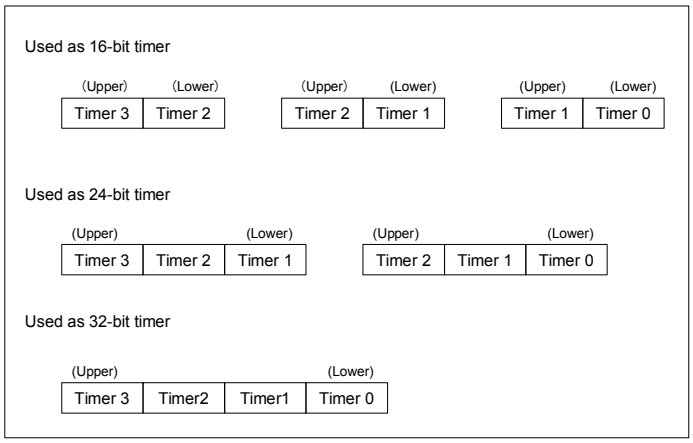


Figure 84 8-bit timer cascade connection

When the 8-bit timer is used through cascade connection, carry out the following settings.
(1) Setting of frequency division ratio in timer
Set the frequency division ratio in TMnBR.
(Example) In order to cascade timer 0 and timer 1 together for use as a 16-bit timer and to set the frequency division ratio to 0x1234, it is necessary to set 0x1234 - 1 = 0x1233 in TMnBR. Set 0x33 in the lower TM0BR, and 0x12 in the upper TM1BR.
Because TMnBR can be accessed by 16-bit or 32-bit access, it is possible to set multiple registers simultaneously. (When cascading timers 1 and 2 together, or when using three 8-bit timers as a 24-bit timer, it is not possible to simultaneously access only the registers for the cascaded timers.)

If the value set in TMnBR is to be changed while the counting operation is in progress, change TMnBR for all of the cascaded timers simultaneously.

(2) Clock source selection

Select any clock source for the lowest timer. Set "cascaded connection" as the clock source for all upper timers (i.e., all timers other than the lowest timer).

(Example 1) When using timer 0 and timer 1 as a 16-bit timer

Set any clock source for timer 0.

Set "cascaded connection" as the clock source for timer 1.

(Example 2) When using timer 0, 1, 2 and 3 as a 32-bit timer

Set any clock source for timer 0.

Set "cascaded connection" as the clock source for timer 1, 2 and 3.

(3) Timer initialization

Initialize all of the cascaded timers by setting the TMnLDE flag to "1." (It is not necessary to set all of the registers simultaneously.)

(4) Enabling the counting operation

Enable the counting operation by either of the following methods:

1) Enable the counting operation in each cascaded timer in sequence starting from the upper end.

2) Enable the counting operation in all of the cascaded timers simultaneously.

(5) Counting operation stop

Stop the counting operation by either of the following methods:

1) Stop the counting operation in each cascaded timer in sequence starting from the low end.

2) Stop the counting operation in all of the cascaded timers simultaneously.

(6) Timer outputs, interrupts

Of all the cascaded timers, only the timer output and interrupt request from the uppermost timer can be used. The operations of the timer outputs and interrupt requests from the lower timers are not guaranteed.

10.4.5. Example of Using the Prescaler and Cascaded Connection

(1) When the timer 1 clock source is set to "timer 0 underflow"

When TM0BC underflows, the setting value in TM0BR is loaded into TM0BC, and the value in TM1BC is decremented by one. When TM1BC underflows, the setting value in TM1BR is loaded into TM1BC.

(2) When timer 0 and timer 1 are set for cascaded connection

When TM1BC is not set to 0x00 and TM0BC underflows, TM0BC is set to 0xFF, and the value in TM1BC is decremented by one.

When TM1BC is set to 0x00 and TM0BC underflows, the setting values in TM0BR and TM1BR are loaded into TM0BC and TM1BC, respectively, and a timer 1 interrupt request is generated.

10.5. Cautions

Sampling the pin input is carried out through IOCLK. Input a signal with the pulse width of IOCLK×1.5 or over.

Moreover, event counting cannot be performed under the state that IOCLK is in stops. (HALT and STOP modes)

Chapter 11 16-bit Timer Module (TM16)

11

CHAPTER 11

16-bit Timer Module (TM16)

11.1. General

This LSI has eight internal 16-bit timer counters.
Seven of them are down counters that can be used as event counters, interval timers, and timer outputs.
The last one is equipped with two input capture registers, and can be used for an event counter, an interval timer, timer output (trigger output), PWM output, input capture, one-shot output (pulse, level), etc.

11.2. Features

Timers 4 and 5, and 7 to 11

- Down counter
- Clock sources
  Internal clocks: IOCLK
  1/8 IOCLK, 1/32 IOCLK, and timer 0 to 2 underflow
  External pin
  The clock sources that can be used differ according to the timer. Refer to the description of the TMnCK field in the TMnMD register.
- Timer output
  Output of 1/2 underflow frequency division
- Interrupts
  Interrupt is generated when a timer underflow occurs

Timer 6

- Up counter
- Clock sources
  Internal clocks: IOCLK
  1/8 IOCLK, 1/32 IOCLK, and timer 0 to 2 underflow
  External pins: TM6IOA, TM6IOB
- Timer output
  Toggled output
  Single-phase PWM with variable cycle and variable duty ratio
  Two-phase PWM with fixed cycle and variable duty ratio
  One-shot output (pulse, level)
  High-speed PWM (resolution: 10, 11, 12, 14 bits)
  Output polarity can be set
- Input capture
  TM6IOA: Rising edge/falling edge, both edges
  TM6IOB: Rising edge/falling edge, both edges
- Interrupts
  Binary counter overflow interrupt
  Compare/capture A interrupt (can initiate DMA)
  Compare/capture B interrupt

11.3. List of Functions

Table 65 16-bit timer function chart

TIMER	Timer 4	Timer 5	Timer 6	Timer 7	Timer 8	Timer 9	Timer 10	Timer 11
Up/down	Down count	Down count	Up count	Down count	Down count	Down count	Down count	Down count
Interval timer	○	○	○	○	○	○	○	○
Event counter	○	○	○	○	○	○	○	○
Timer output	○	○	○	○	○	○	○	○
Toggled output	×	×	○	×	×	×	×	×
PWM output	×	×	○	×	×	×	×	×
High-speed PWM output	×	×	○	×	×	×	×	×
One-shot output	×	×	○	×	×	×	×	×
Input capture (single edge)	×	×	○	×	×	×	×	×
Input capture (both edges)	×	×	○	×	×	×	×	×
Interrupts	Underflow	Underflow	Overflow Compare/capture A Compare/capture B	Underflow	Underflow	Underflow	Underflow	Underflow
DMA initiation	×	×	○	×	×	×	×	×
UART0 clock source	×	×	×	×	○	×	×	×
UART1 clock source	×	×	×	×	×	○	×	×
UART2 clock source	×	×	×	×	×	×	○	×

Cascaded connection		○	×	×	○	○	○	×	
Clock sources	Refer to the description of the TMnCK field in the TMnMD register.								

11.4. Description of Registers

Table 66 16-bit timer register

Address:	Symbol	Name	Number of bits	Initial value	Access size
0XD4003080	TM4MD	Timer 4 mode register	8	0x 00	8
0xD4003082	TM5MD	Timer 5 mode register	8	0x 00	8
0xD4003084	TM6MD	Timer 6 mode register	16	0x 0000	8,16
0xD4003086	TM7MD	Timer 7 mode register	8	0x 00	8
0xD4003088	TM8MD	Timer 8 mode register	8	0x 00	8
0xD400308A	TM9MD	Timer 9 mode register	8	0x 00	8
0xD400308C	TM10MD	Timer 10 mode register	8	0x 00	8
0xD400308E	TM11MD	Timer 11 mode register	8	0x 00	8
0xD4003090	TM4BR	Timer 4 base register	16	0x 0000	8,16,32
0xD4003092	TM5BR	Timer 5 base register	16	0x 0000	8,16
0xD4003096	TM7BR	Timer 7 base register	16	0x 0000	8,16
0xD4003098	TM8BR	Timer 8 base register	16	0x 0000	8,16,32
0xD400309A	TM9BR	Timer 9 base register	16	0x 0000	8,16
0xD400309C	TM10BR	Timer 10 base register	16	0x 0000	8,16,32
0xD400309E	TM11BR	Timer 11 base register	16	0x 0000	8,16
0xD40030A0	TM4BC	Timer 4 binary counter	16	0x 0000	8,16,32
0xD40030A2	TM5BC	Timer 5 binary counter	16	0x 0000	8,16
0xD40030A4	TM6BC	Timer 6 binary counter	16	0x 0000	8,16,32
0xD40030A6	TM7BC	Timer 7 binary counter	16	0x 0000	8,16
0xD40030A8	TM8BC	Timer 8 binary counter	16	0x 0000	8,16,32
0xD40030AA	TM9BC	Timer 9 binary counter	16	0x 0000	8,16
0xD40030AC	TM10BC	Timer 10 binary counter	16	0x 0000	8,16,32
0xD40030AE	TM11BC	Timer 11 binary counter	16	0x 0000	8,16
0xD40030B4	TM6MDA	Timer 6 compare/capture A mode register	8	0x 00	8,16
0xD40030B5	TM6MDB	Timer 6 compare/capture B mode register	8	0x 00	8
0xD40030C4	TM6CA	Timer 6 compare/capture register A	16	0x 0000	8,16
0xD40030D4	TM6CB	Timer 6 compare/capture register B	16	0x 0000	8,16
0xD4003071	TMPSCNT	Timer prescaler control register	8	0x00	8

11.4.1. Timer mode register

Register symbol: TMNMD
Address: TM4MD : 0xD4003080
TM5MD : 0xD4003082
TM7MD : 0xD4003086
TM8MD : 0xD4003088
TM9MD : 0xD400308A
TM10MD : 0xD400308C
TM11MD : 0XD400308E
Purpose: This register sets the control conditions for timer n operation.

Bit	7	6	5	4	3	2	1	0
Bit name	TMnCNE	TMnLDE	reserved			TMnCK[2:0]		
Initial value	0	0	0			0		
R/W	RW	RW	R			RW		

Bit	Bit name	Description
7	TMNCNE	Timer n count enable This bit controls the timer n count operation. 0 : Stops operation 1 : Enables operation
6	TMnLDE	Timer n load enable This bit initializes timer n. 0 : Normal operation 1 : Loads the value in TMnBR into TMnBC. Resets the timer output n low.
5-3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2-0	TMnCK[2:0]	Timer clock source This field selects the clock source. For the clock source of each timer, refer to Table 67 16-bit timer clock source.

Table 67 16-bit timer clock source

TMnCK[2:0]	Timer 4	Timer 5	Timer 6	Timer 7
000	IOCLK	IOCLK	IOCLK	IOCLK
001	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK
010	1/32 IOCLK	1/32 IOCLK	1/32 IOCLK	1/32 IOCLK
011	Setting prohibited	Cascaded with timer 4	Setting prohibited	Setting prohibited
100	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow
110	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow
111	TM4IO pin input	TM5IO pin input	TM6IOB pin input (single edge)	TM7IO pin input

TMnCK[2:0]	Timer 8	Timer 9	Timer 10	Timer 11
000	IOCLK	IOCLK	IOCLK	IOCLK
001	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK
010	1/32 IOCLK	1/32 IOCLK	1/32 IOCLK	1/32 ICOLK
011	Cascaded with timer 7	Cascaded with timer 8	Cascaded with timer 9	Setting prohibited
100	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow
110	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow
111	TM8IO pin input	TM9IO pin input	TM10IO pin input	TM11IO pin input

When using the 1/8 IOCLK or 1/32 IOCLK setting, the prescaler settings by TMPSCNT must be enabled.
Set TMnCK while TMnCNE is "0."
Set TM6CK while TM6CNE is "0."

Set TMnCNE to "1" while TMnLDE is "0."
Set TM6CNE to "1" while TM6LDE is "0."

Set TMnLDE to "1" while TMnCNE is "0."
Set TM6LDE to "1" while TM6CNE is "0."

Operation is not guaranteed if TMnCNE and TMnLDE are both "1" at the same time.
Operation is not guaranteed if TM6CNE and TMnLDE are both "1" at the same time.

11.4.2. Timer 6 mode register

Register symbol: TM6MD
Address: 0xD4003084
Purpose: This register sets the control conditions for the operation of the timer 6 .

Bit	15	14	13	12	11	10	9	8
Bit name	TM6CNE	TM6LDE	TM6PME	TM6PM[1:0]		reserved		
Initial value	0	0	0	00		0		
R/W	RW	RW	RW	RW		R		
Bit	7	6	5	4	3	2	1	0
Bit name	TM6TGE	TM6ONE	reserved	TM6CAE	reserved	TM6CK[2:0]		
Initial value	0	0	0	0	0	000		
R/W	RW	RW	R	RW	R	RW		

Bit	Bit name	Description
15	TM6CNE	Timer 6 count enable This bit controls the timer 6 count operation. 0 : Stops operation 1 : Enables operation
14	TM6LDE	Timer 6 load enable This bit initializes timer 6. 0 : Normal operation 1 : INITIALIZE. THIS SETTING CLEARS TM6BC. THE VALUES IN TM6CA AND TM6CB ARE UPDATED AS A COMPARE REGISTER BUFFER IF TM6CA AND TM6CB ARE SET AS A DOUBLE-BUFFER COMPARE REGISTER. THE TIMER 6 OUTPUT A AND B ARE RESET.
13	TM6PME	Timer 6 PWM enable This bit controls the PWM output to the TM6IOA/TM6IOB pins. 0 : Normal waveform 1 : PWM output. The resolution is set by TM6PM[1:0].
12-11	TM6PM[1:0]	Timer 6 PWM output resolution This field selects the PWM output resolution. 00 : 10 bits (basic output: 8 bits; additional output: 2 bits) 01 : 11 bits (basic output: 8 bits; additional output: 3 bits) 10 : 12 bits (basic output: 8 bits; additional output: 4 bits) 11 : 14 bits (basic output: 8 bits; additional output: 6 bits)
10-8	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
7	TM6TGE	Timer 6 trigger enable This bit enables the start of counting in response to the timer 6 external pin. 0 : Disabled 1 : Enables start of counting in response to TM6IOB pin input.
6	TM6ONE	Timer 6 one-shot enable This bit selects repeat/one-shot operation for timer 6. 0 : Repeat operation

Bit	Bit name	Description
		1 : One-shot operation (The TM6CNE flag is cleared when TM6BC and TM6CA match.)
5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4	TM6CAE	Timer 6 compare match A enable This bit selects whether to clear the counter in response to timer 6 compare match A. 0 : Do not clear 1 : Clear When TM6CA is a compare register: TM6BC is cleared when TM6BC and TM6CA match. When TM6CA is a capture register: TM6BC is cleared when input is captured in TM6CA.
3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2-0	TM6CK[2:0]	Selection of Timer 6 clock source This field selects the clock source. For the clock source of each timer, refer to Table 67 16-bit timer clock source.

11.4.3. Timer Base Register

Register symbol: TMNBR
Address: TM4BR : 0x D4003090
TM5BR : 0x D4003092
TM7BR : 0x D4003094
TM8BR : 0x D4003098
TM9BR : 0x D400309A
TM10BR : 0x D400309C
TM11BR : 0x D400309E
Purpose: This register sets the timer n count period.

Bit	15	14	13	12	11	10	9	8
Bit name	TMnBR[15:8]							
Initial value	0							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	TMnBR[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	DESCRIPTION
15-0	TMnBR[15:0]	Timer n base register This field sets the initial value and the underflow cycle for the timer n binary counter (TMnBC). The underflow cycle is equal to the value set in TMnBR[15:0], plus one.

11.4.4. Timer binary counter

Register symbol: TMNBC
Address: TM4BC : 0x D40030A0
TM5BC : 0x D40030A2
TM7BC : 0x D40030A6
TM8BC : 0x D40030A8
TM9BC : 0x D40030AA
TM10BC : 0x 40030AC
TM11BC : 0x 40030AE
Purpose: This register is the timer n binary counter. The value in this counter can be read.

Bit	15	14	13	12	11	10	9	8
Bit name	TMnBC[15:8]							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	TMnBC[7:0]							
Initial value	0							
R/W	R							

Bit	Bit name	Description
15-0	TMnBC[15:0]	Timer n binary count The timer n binary count value can be read. This counter uses the value that is set in TMnBR as its initial value. An underflow occurs and an interrupt is generated when this counter counts the value set in TMnBR, plus one. The value in TMnBC is then initialized again to the value set in TMnBR.

11.4.5. Timer 6 Compare/Capture A Mode Register

Register symbol: TM6MDA
Address: 0x D40030B4
Purpose: This register sets the control conditions for the operation of the timer 6 compare/capture register A.

Bit	7	6	5	4	3	2	1	0
Bit name	TM6AM[1:0]		TM6AEG	TM6ACE	reserved	TM6AO[2:0]		
Initial value	0		0	0	0	0		
R/W	RW		RW	RW	R	RW		

Bit	Bit name	DESCRIPTION
7-6	TM6AM[1:0]	Timer 6 compare/capture register A mode flag This field sets the operation mode for timer 6 compare register A. 00 : Compare register (single buffer) 01 : Compare register (double buffer) 10 : Capture register (single edge) 11 : Capture register (both edges)
5	TM6AEG	Timer 6 compare A edge select This field selects the valid edge for the TM6IOA pin input and the output polarity. 0 : Rising edge. Positive polarity output. 1 : Falling edge. Negative polarity output.
4	TM6ACE	Timer 6 capture A enable This bit enables/disables capture in TM6CA. 0 : Capture operation prohibited 1 : Capture operation enabled
3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2-0	TM6AO[2:0]	Timer 6 compare A output select This field selects the output waveform for the TM6IOA pin. 000 : Set when TM6BC and TM6CA match Reset when TM6BC and TM6CB match 001 : Set when TM6BC and TM6CA match Reset when TM6BC overflows 010 : Set when TM6BC and TM6CA match (Reset only when timer 6 is initialized) 011 : Reset when TM6BC and TM6CA match 100 : Invert (toggle) output when TM6BC and TM6CA match 101 : Setting prohibited 110 : Setting prohibited 111 : Setting prohibited

11.4.6. Timer 6 Compare/Capture B Mode Register

Register symbol: TM6MDB
Address: 0xD40030B5
Purpose: This register sets the control conditions for the operation of the timer 6 compare/capture register B.

Bit	7	6	5	4	3	2	1	0
Bit name	TM6BM[1:0]		TM6BEG	TM6BCE	reserved	TM6BO[2:0]		
Initial value	0		0	0	0	0		
R/W	RW		RW	RW	R	RW		

Bit	Bit name	Description
7-6	TM6BM[1:0]	Timer 6 compare/capture register B model flag This field sets the operation mode for timer 6 compare register B. 00 : Compare register (single buffer) 01 : Compare register (double buffer) 10 : Capture register (single edge) 11 : Capture register (both edges)
5	TM6BEG	Timer 6 compare B edge select This field selects the valid edge for the TM6IOB pin input and the output polarity. 0 : Rising edge. Positive polarity output. 1 : Falling edge. Negative polarity output.
4	TM6BCE	Timer 6 capture B enable This bit enables/disables capture in TM6CB. 0 : Capture operation prohibited 1 : Capture operation enabled
3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2-0	TM6BO[2:0]	Timer 6 compare B output select This field selects the output waveform for the TM6IOB pin. 000 : Set when TM6BC and TM6CB match Reset when TM6BC and TM6CA match 001 : Set when TM6BC and TM6CB match Reset when TM6BC overflows 010 : Set when TM6BC and TM6CB match (Reset only when timer 6 is initialized) 011 : Reset when TM6BC and TM6CB match 100 : Invert (toggle) output when TM6BC and TM6CB match 101 : Setting prohibited 110 : Setting prohibited 111 : Setting prohibited

11.4.7. Timer 6 Compare/Capture Register A

Register symbol TM6CA
Address 0xD40030C4
Purpose Used as timer 6 compare/capture register A.

Bit	15	14	13	12	11	10	9	8
Bit name	TM6CA[15:8]							
Initial value	0							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	TM6CA[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	DESCRIPTION
15-0	TM6CA[15:0]	Timer 6 compare When set as a compare register: An interrupt request is generated when TM6BC and TM6CA match. When TM6BC and TM6CA match, the timer 6 interval can be set by clearing TM6BC. The interval used will be the setting value plus one. When set as a double buffer compare register: Since the setting value for TM6CA will be stored once in the compare register buffer, the previously set value will sometimes be read after TM6CA has been written. The setting value is loaded from the compare register buffer to the compare register under the following conditions. In all cases, TM6BC is x0000. When timer 6 is initialized When there is an overflow (when TM6CAE = "0") When TM6BC and TM6CA match (TM6CAE = "1") When set as a capture register: When the edge selected through using TMnAEG is input to the TM6IOA pin, the value of TM6BC is captured in TM6CA and an interrupt request is generated. When the register is set to capture both edges, the edges are captured and an interrupt request is generated regardless of the edge type (rising or falling).

11.4.8. Timer 6 Compare/Capture Register B

Register symbol	TM6CB
Address	0xD40030D4
Purpose	Used as timer 6 compare/capture register B.

Bit	15	14	13	12	11	10	9	8
Bit name	TM6CB[15:8]							
Initial value	0							
R/W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	TM6CB[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
15-0	TM6CB[15:0]	Timer 6 compare When set as a compare register: An interrupt request is generated when TM6BC and TM6CB match. When TM6BC and TM6CB match, the timer 6 interval can be set by clearing TM6BC. The interval used will be the value set plus one. When set as a double buffer compare register: Since the setting value for TM6CB will be stored once in the compare register buffer, the previously set value will sometimes be read after TM6CB has been written. The setting value is loaded from the compare register buffer to the compare register under the following conditions. In all cases, TM6BC is 0x0000. When timer 6 is initialized When there is an overflow (when TM6CAE = "0") When TM6BC and TM6CB match (TM6CAE = "1") When set as a capture register: When the edge selected through using TM6AEG is input to the TM6IOB pin, the value of TM6BC is captured in TM6CB and an interrupt request is generated. When the register is set to capture both edges, the edges are captured and an interrupt request is generated regardless of the edge type (rising or falling).

11.4.9. Timer prescaler control register

Register symbol: TMPSCNT
Address: 0xD4003071
Purpose: This register enables the timer prescaler.
This prescaler is common for both 8-bit and 16-bit counters.

Bit	7	6	5	4	3	2	1	0
Bit name	TMPSCNE	reserved						
Initial value	0	0						
RW	RW	R						

Bit	Bit name	Description
7	TMPSCNE	Timer prescaler enable This bit controls the prescaler operation. 0 : Stops prescaler operation 1 : Enables prescaler operation
6-0	Reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.

11.5. Description of Operations of timers 4, 5, 7, 8, 9, 10 and 11

Timers 4, 5, 7-11 include a built-in down-counter and a register for initial settings, and can be used as interval timers or event counters.

11.5.1. Interval Timer and Timer Output

Set timers 4, 5, 7, 8, 9, 10 or 11 according to the procedure given below when using any of these timers as an interval timer. A timer set for this operation will function as an interval timer that generates an interrupt at set intervals. For details on using 32-bit timers under a cascaded connection, please see 11.5.3"Cascaded connection".

11.5.1.1. Operation Start Procedure

(1) Timer frequency division ratio setting
Set the division frequency ratio in TMnBR.
The interrupt cycle is equal to (TMnBR setting + 1) x clock source cycle.
(2) Clock source selection
The clock source is selected by TMnCK[2:0] in the TMnMD register.
When using 1/8 IOCLK or 1/32 IOCLK as the clock source, set TMPSCNE in the TMPSCNT register to "1," and enable the prescaler operation.
(3) Timer initialization
Initialize timer n by setting TMnLDE in the TMnMD register to "1."
The value that is set in TMnBR is loaded into TMnBC as the initial value, and the timer output is reset.
After initialization, make sure to set TMnLDE to "0" in order to put the timer back in the normal operation mode.
(4) I/O port setting (When using timer output)
Select the timer output in the I/O port output mode register, and then set the output pin in the input/output control register.
(5) Enabling the counting operation
The counting operation starts when TMnCNE in the TMnMD register is set to "1."

Once the counting operation is enabled, an underflow interrupt request is generated on a constant cycle. Furthermore, the pin output inverts each time that the interrupt is generated, and the setting value in TMnBR is loaded into TMnBC.
If the value in the TMnBR register is changed while the counting operation is in progress, the new value is loaded as the initial value at the next occurrence of underflow and then the interrupt cycle changes.

11.5.1.2. Operation Stop Procedure

(1) Counting operation stop
The counting operation stops if TMnCNE in the TMnMD register is set to "0."
(2) Initialize the timer as required.
If TMnLDE in the TMnMD register is set to "1," the value that is set in TMnBR is loaded into TMnBC as the initial value, and the timer output is reset.
After the timer stops, the binary counter and the pin output are maintained in their previous state if TMnLDE is not set to "1". The counting operation can be resumed from its previous state by setting TMnCNE to "1."

11.5.2. Event Count

Make settings according to the following procedure when using timers 4, 5, 7, 8, 9, 10 or 11 for event counts.

11.5.2.1. Operation Start Procedure

(1) Timer frequency division ratio setting
Set the frequency division ratio in TMnBR.
The interrupt cycle is equal to (TMnBR setting + 1) x clock source cycle.
(2) Count source selection
Set the TMnIO input pin as the count source by using TMnCK[2:0] of the TMnMD register.
(3) Timer initialization
Initialize timer n by setting TMnLDE in the TMnMD register to "1."
The setting value TMnBR is loaded into TMnBC as the initial value, and the timer output is reset.
After initialization, make sure to set TMnLDE to "0" in order to put the timer back in normal operation mode.
(4) I/O port setting
Set the input pin using the input/output control register for the I/O port.
(5) Enabling the counting operation
The counting operation starts when TMnCNE in the TMnMD register is set to "1."

The rising edge of the signal at the input pin is counted when count operations are enabled. An interrupt is generated and the value that is set in TMnBR is loaded into TMnBC when there is an underflow of the binary counter.
If the value in the TMnBR register is changed while the counting operation is in progress, that new value is loaded as the initial value the next time that an underflow occurs, and then the interrupt cycle changes.

11.5.3. Cascaded connection

Timers 4 and 5 can be cascaded together for use as a simple 32-bit timer. (Timer 5 represents the high order, while timer 4 represents the low order.) Timers 7 through 10 can also be cascaded together for use as a simple 32-bit timer (when timers 7-8, 8-9 or 9-10 are connected), 48-bit timer (when timers 7-8-9 or 8-9-10 are connected), or 64-bit timer (when 7-8-9-10 are connected).
16-bit timer can be cascaded together for use in combination shown in Figure 85　16-bit timer cascade connection.

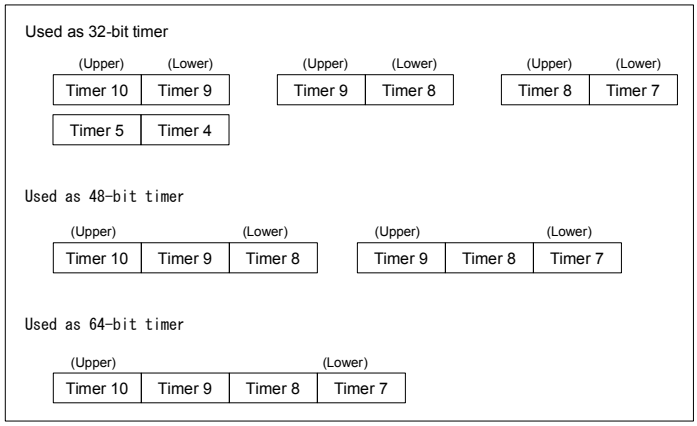


Figure 85　16-bit timer cascade connection

When cascading the 16-bit timers together for use, make the settings described below. The following describes an example in which timers 4 and 5 are used together as a 32-bit timer.
(1) Timer frequency division ratio setting
Set the frequency division ratio in TMnBR.
　(Example 1)
　To set 0x12345678 as the interrupt cycle while using timers 4 and 5 as a 32-bit timer, it is necessary to set 0x12345678 − 1 = 0x12345677 in TMnBR.　In other words, set 0x5677 in the lower TM4BR, and 0x1234 in the upper TM5BR.
Since TM4BR and TM5BR allow 32-bit access, they can be set simultaneously using a single instruction. Be sure to change TM4BR and TM5BR simultaneously using a single instruction when changing the setting value of TMnBR during counter operations.
(2) Clock source selection
　Select any clock source for the lower timer (timer 4).
　Set "cascaded connection" as the clock source for the upper timer (timer 5).
(3) Timer initialization
　Initialize by setting the TMnLDE flag for both timers 4 and 5 to "1". (It is not necessary to set both simultaneously.)
(4) Enabling the counting operation
Enable the counting operation by either of the following methods:
　1)　Enable count operations of the upper timer (timer 5) and then enable count

operations of the lower timer (timer 4).

2) Simultaneously enable count operations of both timers 4 and 5.

(5) Counting operation stop

Stop the counting operation by either of the following methods:

1) Stop count operations of the lower timer (timer 4) and then stop count operations of the upper timer (timer 5).
2) Simultaneously stop the count operations of both timer 4 and 5.

(6) Interrupts

An interrupt request can only be used with the upper timer (timer 5).
Interrupt request operations are not guaranteed in the case of the lower timer (timer 4).

11.6. Description of Operations of Timer 6

Timer 6 includes a built-in up-counter and two compare/capture registers. The compare/capture registers can be independently selected for use as compare or capture registers.

11.6.1. Binary Counter Settings

(1) Count source Setting
The count source is set using TM6CK[2:0] of the TM6MD register.
Be sure to enable prescaler operations using the timer prescaler control register (TMPSCNT) when 1/8 IOCLK or 1/32 IOCLK is selected.
(2) Timer initialization
Before enabling timer 6 count operations, initialize TM6BC by setting the TM6LDE flag of the TM6MD register to "1" and then reset this bit "0".
(3) Starting the timer
To start the timer from software, first initialize the timer and then set the TM6CNE bit of the TM6MD register to "1".
To start the timer using an external trigger, set the TM6CNE bit to "0".
(4) Clearing the binary counter
Set the TM6CAE bit of the TM6MD register to "1" when clearing of the TM6BC binary counter is to be controlled by the TM6CA register.
If TM6CA is to be used as a compare register, the binary counter is cleared when TM6BC and TM6CA match. If TM6CA is to be used as a capture register, the binary counter is cleared when data is captured into TM6CA.
(5) Starting the timer by using an external trigger
Set the TM6TGE bit of the TM6MD register to "1" when starting timer 6 based on trigger input to the TM6IO pin. The timer will be started upon the edge specified by TM6BEG of the TM6CB register and the opposite edge which inputs to the TM6IOB pin. .
(6) One-shot operation
Set the TM6ONE bit of the TM6MD register to "1" when stop control for the counter is to be handled by TM6CA. When TM6BC and TM6CA match, the TM6CNE flag is cleared and the counter is stopped. (Note: When using the MN103004, the TM6CNE bit is not cleared in this case.)
When starting the timer through using an external trigger, the timer will be re-started after it has been stopped whenever the start trigger is re-input to the pin. It is not necessary to reset the TM6TGE bit of the TM6MD register.

11.6.2. Compare/Capture Register Settings

When using compare registers A or B, be sure to make the following settings before initializing timer 6. Although the following description given is for compare register A, the settings for compare register B are handled in the same way.
(1) Compare/Capture Register Settings
The operational mode of TM6CA is set through using TM6AM[1:0] of the TM6MDB register.

TM6AM[1:0]	TM6CA operational mode
00	Compare register (single buffer)
01	Compare register (double buffer)
10	Capture register (single edge)
11	Capture register (both edges)

Single and double buffer modes are available when using compare register mode. In single buffer mode, the value written to the TM6CA register is activated immediately. In double buffer mode, the value written to the TM6CA register is first latched in a compare register buffer, and TM6CA register contents are not updated immediately after the value is written. Be sure to set double buffer mode when the value of the compare register is to be changed during count operations.
The compare register is updated under the following conditions.

1. When the TM6LDE bit of the TM6MD register is "1".
2. When the TM6CAE bit of the TM6MD register is "0" and TM6BC has overflowed.
3. When the TM6CAE bit is "1", TM6CA is set as a compare register, and TM6BC counts up while TM6CA and TM6BC match.

In addition to the above three conditions, the TM6CB register also updated under the following condition.

4. When the TM6CAE bit is "1", TM6CA is set as a capture register, and data has been captured into TM6CA.

Single edge mode and both edges mode are available in the capture mode.
(2) Selecting pin polarity
The polarity of the TM6IOA pin can be selected by using the TM6AEG bit of the TM6MDA register.

Function	When TM6AEG = 0	When TM6AEG = 1
Capture (single edge mode)	Rising edge	Falling edge
Timer start trigger	Falling edge	Rising edge
Pin output	When reset: "L" level When set: "H" level	When reset: "H" level When set: "L" level

The polarity of the TM6IOB pin can be selected by using the TM6BEG bit of the TM6MDB register.

Function	When TM6BEG = 0	When TM6BEG = 1
Capture (single edge mode)	Rising edge	Falling edge
Count source input (single edge mode)	Rising edge	Falling edge
Pin output	When reset: "L" level When set: "H" level	When reset: "H" level When set: "L" level

(3) Enabling capture operations
When TM6CA is set as a capture register, capture operations can be enabled or disabled by using the TM6ACE bit of TM6MDA.
(4) Setting the output pin
The mode of the output pin is set by using the TM6AO[2:0] (TM6BO[2:0]) field of the TM6MDA (TM6MDB) register. For details on the values that can be set, please see the description for each register.

11.6.3. High-Speed PWM Mode Settings

(1) Clock source setting
High-speed PWM mode can be set by setting the TM6PME bit in the TM6MD register to "1."

Set the resolution in TM6PM[1:0]. In this instance, set the TM6CAE bit to "0."
(2) Compare/capture mode setting
Set TM6AM[1:0] in TM6MDA (TM6MDB) to "compare register (double buffer)." In TM6AO[2:0], set "set when matches with TM6CA (TM6CB), reset when TM6BC overflows." (TM6AM[1:0] = 01, TM6AO[2:0] = 001)
(3) Compare value setting
In setting the data for TM6CA(TM6CB), shift and then write the data so that the MSB of the data for each bit width (10, 11, 12, or 14 bits) overlaps with the MSB of the register (16 bits). (For example, in the case of 10-bit data, shift the data 6 bits and write the 10 bits of data in bits 15 through 6.)
(4) Enabling the count operation
The counting operation is enabled, and a PWM waveform is output through setting the TM6CNE bit in the TM6MD register to "1"
The basic output is an 8-bit PWM waveform, and when a clock source is set as IOCLK, the frequency remains 130.20kHz regardless of the resolution. The duty ratio of the basic output is determined by the value of the upper 8 bits of the compare register.
The output cycle is determined by the resolution, and is the same as a free-run counter for each set bit. The frequency for one period is set as described below.

IOCLK frequency [MHz] / resolution = one-period frequency [kHz]
Table 11.6-1 Frequency

Resolution	Frequency for one period
10 bits	32.55kHz
11 bits	16.27kHz
12 bits	8.14kHz
14 bits	2.03kHz

11.7. Cautions

The pin input is sampled through IOCLK. Input the signals with the pulse width of IOCLK ×1.5 and more.
Event count can not be carried out under the state that IOCLK has stopped.(HALT and STOP modes)

Chapter 12 Serial interface (SIF)

12

12.1. General

This LSI has two types of internal serial interfaces.
One is a start-stop synchronous mode/clock synchronous mode/I2C mode specifiable serial interface, the other is a start-stop synchronous-only interface (with CTS).

12.2. Features

12.2.1. Serial Interface 0 (Serial Interface 1)

<Clock Synchronous Mode>

- Parity: None, 0 fixed, 1 fixed, even, or odd
- Character length: 7-bit or 8-bit
- Outgoing bit order: LSB or MSB
- Clock source: 1/8 or 1/32 of IOCLK
  1/8 of underflow of Timer 0 (Timer 1), Timer 2 (Timer 3), or Timer 8 (Timer 9)
  1/2 of underflow of Timer 2 (Timer 3)
  External clock
- Maximum transfer rate: 7.25 Mbps (IOCLK = 30MHz)
- Error detection during receive mode: Parity and overrun errors
- Buffers: Independent transmit and receive buffers and a dual transmit and receive buffer
- Interrupts
  - Transmit interrupts: Transmit end or transmit buffer empty (selectable)
  - Receive interrupts: Receive end or receive end on error (selectable)
- DMA requests
  - Transmit mode: Transmit end or transmit buffer empty (selectable)
  - Receive mode: Receive end

<Start-Stop Synchronous Mode>

- Parity: None, 0 fixed, 1 fixed, even, or odd
- Character length: 7-bit or 8-bit
- Outgoing bit order: LSB or MSB
- Clock source: 1/8 or 1/32 of IOCLK
  1/8 of underflow of Timer 0 (Timer 1), Timer 2 (Timer 3), or Timer 8 (Timer 9)
  1/8 of External Clock
- Maximum transfer rate: 38.8 kbps (IOCLK = 30MHz)
- Error detection during receive mode: Parity, overrun and framing errors
- Buffers: Independent transmit and receive buffers and a dual transmit and receive buffer
- Interrupts
  - Transmit interrupts: Transmit end or transmit buffer empty (selectable)
  - Receive interrupts: Receive end or receive end on error (selectable)
- DMA requests
  - Transmit mode: Transmit end or transmit buffer empty (selectable)
  - Receive mode: Receive end

<I2C Mode>

- Master transmit and master receive are possible but there is no collision detection for the start sequence.

12.2.2. Serial Interface 2

- Parity None, 0 fixed, 1 fixed, even, or odd
- Character length 7-bit or 8-bit
- Outgoing bit order LSB or MSB
- Clock source
  - Underflow of Timer 2, Timer 3 and Timer 10
  - External clock
- Maximum transfer rate 233.28 kbps (IOCLK = 30MHz)
- Error detection during receive mode Parity, overrun and framing errors
- Transmit break Transmission can be stopped through using the CTS pin.
- Buffers Independent transmit and receive buffers and a dual transmit and receive buffer
- Interrupts
  - Transmit interrupts Transmit end or transmit buffer empty (selectable)
  - Receive interrupts Receive end or receive end on error (selectable)
- DMA requests
  - Transmit mode Transmit end or transmit buffer empty (selectable)
  - Receive mode Receive end

12.3. Registers

Table 68 Serial controller register

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD4002000	SC0CTR	Serial 0 control register	16	0x 0000	8, 16
0xD4002004	SC0ICR	Serial 0 interrupt mode register	8	0x 00	8
0xD4002008	SC0TXB	Serial 0 transmit buffer	8	0x 00	8
0xD4002009	SC0RXB	Serial 0 receive buffer	8	0x 00	8
0xD400200C	SC0STR	Serial 0 status register	16	0x 0000	8,16
0xD4002010	SC1CTR	Serial 1 control register	16	0x 0000	8, 16
0xD4002014	SC1ICR	Serial 1 interrupt mode register	8	0x 00	8
0xD4002018	SC1TXB	Serial 1 transmit buffer	8	0x 00	8
0xD4002019	SC1RXB	Serial 1 receive buffer	8	0x 00	8
0xD400201C	SC1STR	Serial 1 status register	16	0x 0000	8,16
0xD4002020	SC2CTR	Serial 2 control register	16	0x 0000	8, 16
0xD4002024	SC2ICR	Serial 2 interrupt mode register	8	0x 00	8
0xD4002028	SC2TXB	Serial 2 transmit buffer	8	0x 00	8
0xD4002029	SC2RXB	Serial 2 receive buffer	8	0x 00	8
0xD400202C	SC2STR	Serial 2 status register	8	0x 00	8
0xD400202D	SC2TIM	Serial 2 timer register	8	0x 00	8

12.3.1. Serial Control Register

Register symbol SCNCTR
Address SC0CTR : 0x D4002000
SC1CTR : 0x D4002010
Purpose Sets the operational control conditions for serial interface n.

Bit	15	14	13	12	11	10	9	8
Bit name	SCnTXE	SCnRXE	SCnBKE	SCnIIC	SCnMD[1:0]		SCnOD	SCnTOE
Initial value	0	0	0	0	0		0	0
R/W	RW	RW	RW	RW	RW		RW	RW
Bit	7	6	5	4	3	2	1	0
Bit name	SCnCLN	SCnPB[2:0]			SCnSTB	SCnCK[2:0]		
Initial value	0	0			0	0		
R/W	RW	RW			RW	RW		

Bit	Bit name	Description
15	SCnTXE	Serial n Transmit Enable Enables transmission operations for serial interface n. 0 : Disabled 1 : Enabled
14	SCnRXE	Serial n receive enable Enables receive operations for serial interface n. 0 : Disabled 1 : Enabled
13	SCnBKE	Serial n Break Transmit Enable Enables the break signal from serial interface n 0 : Do not send break signal 1 : Send break signal (output of SBOn pins is always 0)
12	SCnIIC	Serial n I2C Mode Select Selects the I2C mode for serial interface n. 0 : The stop sequence is output when this bit changes from 1 to 0. 1 : The start sequence is output when this bit changes from 0 to 1.
11-10	SCnMD[1:0]	Serial n Mode Select Selects the mode of serial interface n. 00 : Start-stop synchronous mode 01 : Clock synchronous mode (1) SBOn pins are used for output and SBIn pins for input. 10 : I2C mode 11 : Clock synchronous mode (2) SBOn pins are used for output and input to SBIn pins is ignored.
9	SCnOD	Serial n Bit Order Selects the bit order for transmit/receive on serial interface n. 0 : LSB first 1 : MSB first
8	SCnTOE	Input/output Enable of Serial n SBTn pins Controls the output of SBTn pins for serial interface n. 0 : Data are output on SBTn pins only during transmitting or receiving when the internal clock is selected. Data are input during standby mode and when an external clock is

Bit	Bit name	Description
		selected. 1 : Data always output on SBTn pins when the internal clock is selected. Data is input when an external clock is selected.
7	SCnCLN	Serial n Character Length Sets the character length for serial interface n. 0 : 7-bit 1 : 8-bit
6-4	SCnPB[2:0]	Serial n Parity Bit Select Selects the parity bit for serial interface n. 000 : None 001, 010, 011 : Setting prohibited 100 : 0 fixed 101 : 1 fixed 110 : Even (number of 1 bits must be even) 111 : Odd (number of 1 bits must be odd)
3	SCnSTB	Serial n Stop Bit Select Selects the number of stop bits. (enabled only during start-stop synchronous mode) 0 : 1 bit 1 : 2 bits
2-0	SCnCK[2:0]	Serial n Clock Source Select Selects the clock source for serial interface n. 000 : Serial 0 :1/8 of underflow of Timer 8 Serial 1 : 1/8 of underflow of Timer 9 001 : 1/8 IOCLK 010 : 1/32 IOCLK 011 : Serial 0: 1/2 of underflow of Timer 2 (enabled only during clock synchronous mode) Serial 1: 1/2 of underflow of Timer 3 (enabled only during clock synchronous mode) 100 : Serial 0 :1/8 of underflow of Timer 0 Serial 1 : 1/8 of underflow of Timer 1 101 : Serial 0 : 1/8 of underflow of Timer 2 Serial 1 : 1/8 of underflow of Timer 3 110 : 1/8 of external clock (enabled only during start-stop synchronous mode) 111 : External clock (enabled only during clock synchronous mode)

SCnCK[2:0]	000	011	100	101
Serial10	Timer8	Timer2	Timer0	Timer2
Serial11	Timer9	Timer3	Timer1	Timer3

12.3.2. Serial Interrupt Mode Register

Register symbol	SCNICR
Address	SC0ICR : 0x D4002004 SC1ICR : 0x D4002014
Purpose	This set the transmit interrupt, receive interrupt and DMA trigger cause for serial interface n.

Bit	7	6	5	4	3	2	1	0
Bit name	SCnDMD	reserved	SCnTD	SCnTI	reserved	SCnRES	reserved	SCnRI
Initial value	0	0	0	0	0	0	0	0
R/W	RW	R	RW	RW	R	RW	R	RW

Bit	Bit name	Description
7	SCnDMD	Serial n Output Data Mode Sets the Data Output Maintain mode when transmitting through using an external clock. 0 : Data pin is "H" level during transmit mode. 1 : Data pin level is maintained during transmit mode.
6	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
5	SCNTD	Serial n Transmit DMA Trigger cause Selects the DMA trigger cause during transmit mode 0 : Transmit end 1 : Transmit buffer empty
4	SCnTI	Serial n Transmit Interrupt cause Selects the interrupt cause during transmit mode. 0 : TRANSMIT END 1 : Transmit buffer empty
3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2	SCnRES	Serial n Receive Error Select Selects the error interrupt cause during receive mode. 0 : Generates an interrupt request when an overrun, parity or framing error occurs. 1 : Generates an interrupt request when a parity error occurs.
1	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
0	SCNRI	Serial n Receive Interrupt cause Selects the interrupt cause during receive mode. 0 : Receive end 1 : Receive end on error

12.3.3. Serial Transmit Buffer

Register symbol	SCNTXB
Address	SC0TXB : 0x D4002008
	SC1TXB : 0x D4002018
Purpose	Writes transmit data for serial interface n.

Bit	7	6	5	4	3	2	1	0
Bit name	SCnTXB[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
7-0	SCnTXB[7:0]	Serial n Transmit Buffer Used as the transmit data buffer for serial interface n.

12.3.4. Serial Receive Buffer

Register symbol	SCNRXB
Address	SC0RXB : 0xD4002009
	SC1RXB : 0xD4002019
Purpose	Read receive data for serial interface n.

Bit	7	6	5	4	3	2	1	0
Bit name	SCnRXB[7:0]							
Initial value	0							
R/W	R							

Bit	Bit name	Description
7-0	SCnRXB[7:0]	Serial n Receive Buffer Used as the receive data buffer for serial interface n. Bit 7 is 0 when 7-bit transfer is used.

12.3.5. Serial Status Register

Register symbol	SCNSTR
Address	SC0STR : 0x D400200C
	SC1STR : 0x D400201C
Purpose	Indicates the status of serial interface n.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved						SCnSPF	SCnSTF
Initial value	0						0	0
R/W	R						R	R
Bit	7	6	5	4	3	2	1	0
Bit name	SCNTXF	SCnRXF	SCnTBF	SCnRBF	reserved	SCnFEF	SCnPEF	SCnOEF
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Bit name	Description

Bit	Bit name	Description
15-10	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
9	SCnSPF	Serial n Stop Sequence Detection Indicates I2C mode stop sequence detection. 0 : Not detected 1 : Detected Cleared when SCnRXB is read or SCnTXB is written.
8	SCnSTF	Serial n Start Sequence Detection Indicates I2C mode start sequence detection. 0 : Not detected 1 : Detected Cleared when SCnRXB is read or SCnTXB is written.
7	SCnTXF	Serial n Transmit Status Indicates the transmit status. 0 : Transmission possible 1 : Currently transmitting
6	SCnRXF	Serial n Receive Status Indicates the receive status. 0 : Receive possible 1 : Currently receiving
5	SCnTBF	Serial n Transmit Buffer Status Indicates the transmit buffer status. 0 : Transmit buffer empty 1 : Transmit buffer contains data
4	SCnRBF	Serial n Receive Buffer Status Indicates the receive buffer status. 0 : Receive buffer empty 1 : Receive buffer contains data
3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2	SCnFEF	Serial n Framing Error Detection Indicates the presence of a framing error. 0 : No error 1 : Framing error found
1	SCnPEF	Serial n Parity Error Detection Indicates the presence of a parity error. 0 : No error 1 : Parity error found
0	SCnOEF	Serial n Overrun Error Detection Indicates the presence of an overrun error. 0 : No error 1 : Overrun error found

12.3.6. Serial 2 Control Register

Register symbol	SC2CTR
Address	0xD4002020
Purpose	This sets the operational control conditions for serial interface 2.

Bit	15	14	13	12	11	10	9	8
Bit name	SC2TXE	SC2RXE	SC2BKE	SC2TWS	reserved		SC2OD	SC2TWE

Initial value	0	0	0	0	0		0	0
R/W	RW	RW	RW	RW	R		RW	RW
Bit	7	6	5	4	3	2	1	0
Bit name	SC2CLN	SC2PB[2:0]			SC2STB	reserved	SC2CK[1:0]	
Initial value	0	0			0	0	0	
R/W	RW	RW			RW	R	RW	

Bit	Bit name	Description
15	SC2TXE	Serial 2 Transmit Enable Enables transmission operations for serial interface 2. 0 : Disabled 1 : Enabled
14	SC2RXE	Serial 2 Receive Enable Enables receive operations for serial interface 2. 0 : Disabled 1 : Enabled
13	SC2BKE	Serial 2 Break Transmit Enable Enables the break signal from serial interface 2 0 : Do not send break signal 1 : Send break signal (output of SBO2 pins is always 0)
12	SC2TWS	Serial 2 Transmit Interrupt Select Selects the transmit interrupt code. 0 : Interrupt transmission when XCTS input is "H" level 1 : Interrupt transmission when XCTS input is "L" level
11-10	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
9	SC2OD	Serial 2 Bit Order Selects the bit order for transmit/receive on serial interface 2. 0 : LSB first 1 : MSB first
8	SC2TWE	Serial 2 Transmit Interrupt Enable Enables transmit interruption. 0 : Disabled 1 : Enabled Set to “0” when XCTS pin is not set.
7	SC2CLN	Serial 2 Character Length Sets the character length for serial interface 2. 0 : 7-bit 1 : 8-bit
6-4	SC2PB[2:0]	Serial 2 Parity Bit Select Selects the parity bit for serial interface 2. 000 : None 001, 010, 011 : Setting prohibited 100 : 0 fixed 101 : 1 fixed 110 : Even (number of 1 bits must be even) 111 : Odd (number of 1 bits must be odd)
3	SC2STB	Serial 2 Stop Bit Select Selects the number of stop bits. (enabled only during start-stop synchronous mode) 0 : 1 bit

Bit	Bit name	Description
		1 : 2 bits
2	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
1-0	SC2CK[1:0]	Serial 2 Clock Source Select Selects the clock source for serial interface 2. 00 : Underflow of Timer 10 01 : Underflow of Timer 2 10 : External clock 11 : Underflow of Timer 3

12.3.7. Serial 2 Interrupt Mode Register

Register symbol SC2ICR
Address 0xD4002024
Purpose Selects the transmit interrupt, receive interrupt, and DMA trigger cause for serial interface 2.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved		SC2TD	SC2TI	reserved	SC2RES	reserved	SC2RI
Initial value	0		0	0	0	0	0	0
R/W	R		RW	RW	R	RW	R	RW

Bit	Bit name	Description
7-6	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
5	SC2TD	Serial 2 Transmit DMA Trigger cause Selects the DMA trigger cause during transmit mode. 0 : Transmit end 1 : Transmit buffer empty
4	SC2TI	Serial 2 Transmit Interrupt cause Selects the interrupt cause during transmit mode. 0 : Transmit end 1 : Transmit buffer empty
3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2	SC2RES	Serial 2 Receive Error Select Selects the error interrupt cause during receive mode. 0 : Generates an interrupt request when an overrun, parity or framing error occurs. 1 : Generates an interrupt request when a parity error occurs.
1	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
0	SC2RI	Serial 2 Receive Interrupt cause Selects the interrupt cause during receive mode. 0 : Receive end 1 : Receive end on error

12.3.8. Serial 2 Transmit Buffer

Register symbol SC2TXB

Address 0xD4002028
Purpose This writes transmit data for serial interface 2.

Bit	7	6	5	4	3	2	1	0
Bit name	SC2TXB[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
7-0	SC2TXB[7:0]	Serial 2 Transmit Buffer Used as the transmit data buffer for serial interface 2.

12.3.9. Serial 2 Receive Buffer

Register symbol SC2RXB
Address 0xD4002029
Purpose This reads receive data for serial interface 2.

Bit	7	6	5	4	3	2	1	0
Bit name	SC2RXB[7:0]							
Initial value	0							
R/W	R							

Bit	Bit name	Description
7-0	SC2RXB[7:0]	Serial 2 Receive Buffer Used as the receive data buffer for serial interface 2. Bit 7 is 0 when 7-bit transfer is used.

12.3.10. Serial 2 Status Register

Register symbol SC2STR
Address 0xD400202C
Purpose Indicates the status of serial interface 2.

Bit	7	6	5	4	3	2	1	0
Bit name	SC2TXF	SC2RXF	SC2TBF	SC2RBF	SC2CTS	SC2FEF	SC2PEF	SC2OEF
Initial value	0	0	0	0	Note 1	0	0	0
R/W	R	R	R	R	R	R	R	

Note 1: SC2CTS reflects value of XCTS input. Refer SC2CTS description for more detail.

Bit	Bit name	Description
7	SC2TXF	Serial 2 Transmit Status Indicates the transmit status. 0 : Transmission possible 1 : Currently transmitting
6	SC2RXF	Serial 2 Receive Status Indicates the receive status. 0 : Receive possible 1 : Currently receiving
5	SC2TBF	Serial 2 Transmit Buffer Status Indicates the transmit buffer status.

Bit	Bit name	Description
		0 : Transmit buffer empty 1 : Transmit buffer contains data
4	SC2RBF	Serial 2 Receive Buffer Status Indicates the receive buffer status. 0 : Receive buffer empty 1 : Receive buffer contains data
3	SC2CTS	Serial 2 XCTS Input Pin Status 0 : XCTS input pin is "H" level 1 : XCTS input pin is "L" level
2	SC2FEF	Serial 2 Framing Error Detection Indicates the presence of a framing error. 0 : No error 1 : Framing error found
1	SC2PEF	Serial 2 Parity Error Detection Indicates the presence of a parity error. 0 : No error 1 : Parity error found
0	SC2OEF	Serial 2 Overrun Error Detection Indicates the presence of an overrun error. 0 : No error 1 : Overrun error found

12.3.11. Serial 2 Timer Register

Register symbol SC2TIM
Address 0xD400202D
Purpose This selects the internal timer for serial interface 2.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved	SC2TIM[6:0]						
Initial value	0	0						
R/W	R	RW						

Bit	Bit name	DESCRIPTION
7	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
6-0	SC2TIM[6:0]	Serial 2 internal Timer Selects the internal timer for serial interface 2. Cycle is performed at one more than the setting value of this register. An internal 7-bit only counter is built in for serial interface 2 to allow high-speed transfer rates to be maintained even when a relatively slow clock source is used. It is recommended that this register be set as given below when using a given transfer rate with using IOCLK. Divide the frequency for IOCLK with setting the value of Frequency Divisor 1 to the timer base register. Frequency Divisor 1 = INT (IOCLK frequency/Baud rate to be set/127) + 1

Bit	Bit name	DESCRIPTION
		Frequency Divisor 2 = INT (IOCLK frequency/Baud rate to be set/Division Factor 1 + 0.5) The value of Divisor 2, calculated using the above formula, minus 1 is written into SC2TIM . If the value of Divisor 1 is greater than or equal to 2, it is necessary to divide through using Timer 2 or Timer 3.

12.4. Description of Operation (Serial interface 0 or 1)

12.4.1. Connections

<Clock Synchronous Mode>

Connections for both uni-directional and bi-directional transfers are possible.
It is necessary to pull up the SBTn pins when output is on these pins only during transmission (SCnTOE = 0). It is necessary to also pull up the SBOn pins when using the SBOn pins for data I/O (SCnMD 1-0 = "11"). The pull up must be made externally.
The SBOn pins always output data and the SBIn pins always input data when the SBOn pin is used for data output and the SBIn pins for data input (SCnMD 1-0 = "01").
The SBOn pins are usually used for input and are used for output only during data transmission when using the SBOn pins for data I/O (SCnMD 1-0 = "11").
When SCnTOE = 0, the SBTn pins are usually used for input and are used for output only during transmissions using the internal clock. When SCnTOE = 1, the SBTn pins are always used for output when the internal clock is selected.

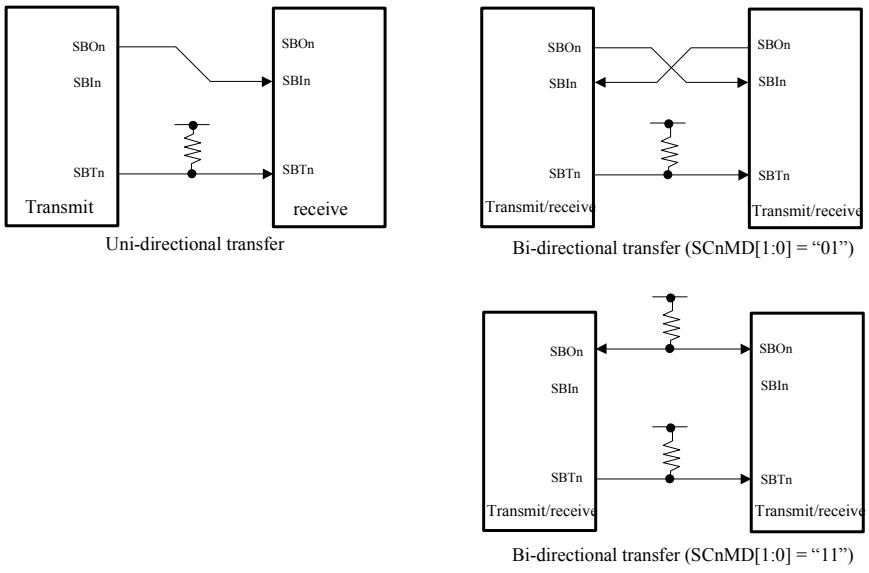


Figure 86 Clock synchronous mode connection

<Start-Stop Synchronous Mode>

Connections for both uni-directional and bi-directional transfers are possible. Here, the

SBOn pins are always used for output and the SBIn pins are always used for input.

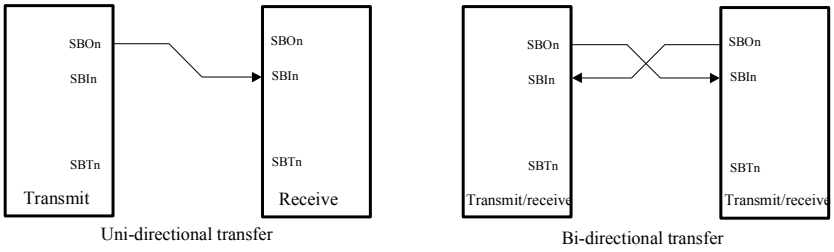


Figure 87 Start-Stop Synchronous Mode

<I2C Mode>

It is possible to connect a device that is capable of slave transmit and slave receive operations.

SDA and SCL must be pulled up to connect such a device. The pull up must be made externally.

The SBOn pins form an open-drain I/O port, while the SBTn pins form an open-drain output port.

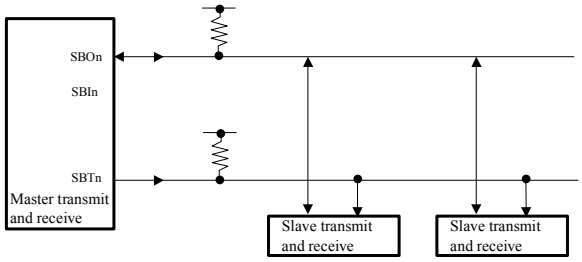


Figure 88 12C mode connection

12.4.2. Baud Rate

<Start-Stop Synchronous Mode>

In a start-stop synchronous mode, select the baud rate and the input clock to use for the serial interface. Set the input clock as given below.

Input clock = baud rate to be used × 8

When using IOCLK, divide the clock frequency through using Transfer Timer 0 or 2 (Timer 1 or 3). Set the frequency division ratio as follows.

Timer frequency division ratio = INT (IOCLK frequency/Baud rate to be used/8 + 0.5)

If the frequency division ratio is large, make Timer 0 a prescaler or make a cascade connection between Timer 1 and 2 (Timer 2 and 3). The baud rates that can actually be set when using IOCLK as an input clock are given as follows.

Baud rate = IOCLK frequency/Timer frequency division ratio/8

The baud rate error at this time can be found by using the following formula.

Baud Rate Error
= ABS (IOCLK frequency/Timer frequency division ratio/8/Baud rate to be set - 1)

Examples in which IOCLK is used are given below.

Table 69 Relationship Between the Timer Frequency Division ratio and the Baud Rate

When IOCLK = 20 MHz

BAUD RATE (BPS)	When making a cascade connection (IOCLK = 20 MHz)		When using a prescalar (IOCLK = 20 MHz)	
	Timer Frequency Division Ratio	Transfer rate error	Timer Frequency Division Ratio	Transfer rate error
56,000	45 *	0.79%	45 *	0.79%
38,400	65 *	0.16%	65 *	0.16%
19,200	130*	0.16%	130 *	0.16%
9,600	260	0.16%	260 (130 x 2)	0.16%
4,800	521	0.03%	520 (130 x 4)	0.16%
2,400	1042	0.03%	1040 (130 x 8)	0.16%
1,200	2083	0.02%	2080 (130 x 16)	0.16%

Note this can be implemented even without using a cascade connection or prescalar.

Table 70 Relationship Between the Timer Frequency Division ratio and the Baud Rate

When IOCLK = 25 MHz

BAUD RATE (BPS)	When making a cascade connection (IOCLK = 25 MHz)		When using a prescalar (IOCLK = 25 MHz)	
	Timer Frequency Division Ratio	Transfer rate error	Timer Frequency Division Ratio	Transfer rate error
56,000	56 *	0.35%	56 *	0.35%
38,400	81 *	0.47%	81 *	0.47%
19,200	163 *	0.15%	163 *	0.15%
9,600	326	0.15%	326 (163 × 2)	0.15%
4,800	651	0.01%	652 (163 × 4)	0.15%
2,400	1302	0.01%	1304 (163 × 8)	0.15%
1,200	2604	0.01%	2608 (163 × 16)	0.15%

Note this can be implemented even without using a cascade connection or pre-scalar.

When using a 1/8 external clock as the clock source, the "H" width and "L" width of the input clock must both be 2 cycles or more of IOCLK.

12.4.3. Using Clock Synchronous Mode

<Transmit> When using 8-bit length, no parity, 2-byte transfer

Transmission starts when data is written to SCnTXB while transmission is enabled. It is possible to serially transmit data when data is again written to SCnTXB during transmission. The MSB (bit 7) is ignored during 7-bit transfer.

The SCnTXF flag is "1" when data is written to SCnTXB and goes to "0" at transmit end. The SCnTBF flag is "1" when data is written to SCnTXB and goes to "0" at transmit start.

<Receive> When using 8-bit length, no parity, 2-byte transfer

After receive end (when the SCnRBF flag is "1"), SCnRXB is read and receive data is accepted. The MSB (bit 7) goes to 0 during 7-bit transfer.

The SCnRXF flag goes to "1" at receive start (at the falling edge of SBT) and goes to "0" at receive end.

The SCnRBF flag goes to "1" at receive end and goes to "0" when SCnRXB is read.

An overrun error occurs if after receive end the next data is received before SCnRXB is read. The first-received data is lost if this happens. The overrun error indicator flag (SCnOEF) is updated at the point that the last bit of data has been received.

A parity error occurs when "1" is received with the parity bit fixed to "0", when "0" is received with the parity bit fixed to "1", when odd data is received with the parity bit fixed to "even", or when even data is received with the parity bit fixed to "odd". The parity error indicator flag (SCnPEF) is updated at the point that the parity bit is received.

12.4.4. Using Start-Stop Synchronous Mode

<Transmit> When using 7-bit length, parity, stop bit 1 bit, 2-byte transfer

Transmission starts when data is written to SCnTXB while transmission is enabled.
It is possible to serially transmit data when data is again written to SCnTXB during transmission.
The MSB (bit 7) is ignored during 7-bit transfer.
The SCnTXF flag is "1" when data is written to SCnTXB and goes to "0" at transmit end.
The SCnTBF flag is "1" when data is written to SCnTXB and goes to "0" at transmit start.

<Receive> When using 7-bit length, parity, stop bit 1 bit, 2-byte transfer

After receive end (when the SCnRBF flag is "1"), SCnRXB is read and receive data is accepted. The MSB (bit 7) goes to 0 during 7-bit transfer.
The SCnRXF flag goes to "1" at receive start (when the start bit is detected) and goes to "0" at receive end.
The SCnRBF flag goes to "1" at receive end and goes to "0" when SCnRXB is read.
An overrun error occurs if after receive end the next data is received before SCnRXB is read. The first-received data is lost if this happens. The overrun error indicator flag (SCnOEF) is updated at the point that the last bit of data has been received.
A parity error occurs when "1" is received with the parity bit fixed to "0", when "0" is received with the parity bit fixed to "1", when odd data is received with the parity bit fixed to "even", or when even data is received with the parity bit fixed to "odd". The parity error indicator flag (SCnPEF) is updated at the point that the parity bit is received.
A framing error occurs when a "0" has been received as the stop bit. The framing error indicator flag (SCnFEF) is updated at the point that the stop bit is received.

12.4.5. Using I2C Mode

<Master Transmit>

Initial Settings

(1) Select the transmission clock. External clocks (SCnCK 2-0) may not be set.
(2) The parity bit is used to represent Ack. Be sure to fix the parity bit to "1" during transmission. (SCnPB 2-0)
Be sure to enable receive operations even during transmission when detecting the Ack signal output by the device used for slave receive. (This is detected as a parity error.)
(3) Set the character length and the outgoing bit order. (SCnCLN and SCnOD)
(4) Set the I2C mode select flag (SCnIIC) to "0".
(5) Set the protocol to I2C mode and set the SBT pin for output only during transmission. (SCnMD 1-0, SCnTOE)
The SBO and SBT pins form an open-drain output port.
(6) Enable transmit operations. (SCnTXE)
The SBO and SBT pins form an open-drain output port.

Enable receive when detecting an Ack signal or during master receive. (SCnRXE)

Start Sequence Detection

(1) "L" is output on the SBO pin as the start sequence when "1" is written into the I2C mode select flag (SCnIIC). The I2C start sequence detect bit (SCnSTF) goes to "1" when the start sequence is generated normally.
Even if a start sequence exists simultaneously at this time no detection is made regarding lost arbitration.

Data Transmit 1

(1) Data is transmitted when data is written into the serial transmit buffer (SCnTXB). SBOn pin output varies after the falling edge of the SBTn pin signal.
(2) SBOn pin output and SBTn pin output is maintained at "L" level after transmit end.

Data Transmit 2

(1) Data is written to the serial transmit buffer (SCnTXB) when more data is to be transmitted.

Stop Bit Detection

(1) "0" is written into the I2C mode select flag (SCnIIC) when data transmissions are to end. Writing must be performed not during transmitting.
(2) SBTn pin output goes "H" when data is being written. After 1 cycle, SBOn pin output goes "H" and the stop sequence is transmitted. The I2C stop sequence detect bit (SCnSPF) goes to "1" at this time.

<Master Receive>

It is always necessary for entering master receive mode to transmit the first byte in the master transmit. Accordingly, the settings after master transmit are described below.

Receive settings

(1) Enable receive operations. (SCnRXE)
(2) The parity bit is used to represent Ack. Be sure to fix the parity bit to "0" during reception. (SCnPB 2-0)

Data Receive

(1) The clock is output and receive operations are performed when the dummy

data "x'FF" is written to the serial transmit buffer (SCnTXB). The receive interrupt can be replaced with the transmit end interrupt.

(2) SBOn pin output and SBTn pin output is maintained at "L" level after receive end. Be sure to write the dummy data "x'FF" to the serial transmit buffer (SCnTXB) again when more data is to be received.

Stop Bit Detection

(1) "0" is written into the I2C mode select flag (SCnIIC) when data transmissions are to end. Writing must be performed not during transmitting.

(2) SBTn pin output goes "H" when data is being written. After 1 cycle, SBOn pin output goes "H" and the stop sequence is transmitted. The I2C stop sequence detect bit (SCnSPF) goes to "1" at this time.

(3) After the stop sequence is transmitted, be sure disable receive operations and initialize the receive mode.

The I2C start sequence detect bit (SCnSTF) and I2C stop sequence detect bit (SCnSPF) are cleared when the serial transmit buffer (SCnTXB) is written or when the serial receive buffer (SCnRXB) is read.

12.4.6. Receive Errors

When using clock synchronous mode, 8-bit length, parity

A DMA request is generated at receive end regardless of the presence of an error.
When the receive interrupt cause is set to receive end, an interrupt request is generated at receive end regardless of the presence of an error.
When the receive interrupt cause has been set to receive end on error, an interrupt request is generated when receive operations end with an error. (The interrupt request is not generated at the point the error occurs.)

12.5. Description of Operation (Serial interface 2)

12.5.1. Connections

<Start-Stop Synchronous Mode>

Connections for both uni-directional and bi-directional transfers are possible.
The SBO2 pin is always used for output and the SBI2 pin is always used for input.

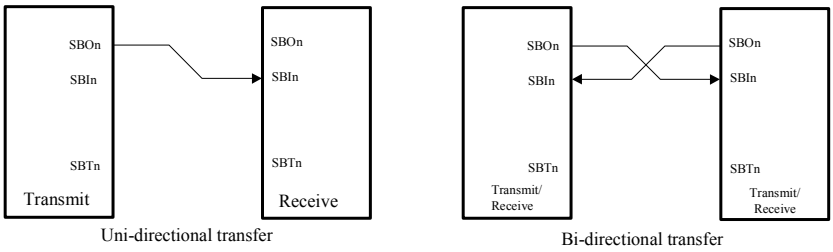


Figure 89 Start-Stop Synchronous Mode Connection

12.5.2. Baud Rate

An internal 7-bit only counter is built in the serial interface 2 to allow high-speed transfer rates to be maintained even when a relatively slow clock source is used.
It is recommended that this register be set as given below through selecting any of Timer 2, 3, or 10 in SC2CK[1:0] of SC2CTR register when setting a transfer rate while using IOCLK. The value of Frequency division ratio 1 is set to the timer base register the selected timer, and please carry out the frequency division of IOCLK. When the value of the frequency division 1 is 1, the value to the serial controller is the same frequency with that to IOCLK.

The value subtracted 1 from the value of the frequency division is written to the serial interface 2 timer register.

Frequency division ratio 1 = INT (IOCLK frequency/Baud rate to be set/127) + 1
Frequency division ratio 2 = INT (IOCLK frequency/Baud rate to be set/Frequency division ratio 1) + 0.5

The actual baud rates that can be set are as follows.
Baud rate[setting value] = IOCLK frequency/frequency division ratio 1/ frequency division ratio 2
The baud rate error at this time can be found by using the following formula.
Baud Rate Error
= ABS (Baud rate [setting value]/Baud rate to be set - 1)
Representative examples are described below.

Table 71 Relationship Between the Timer frequency division ratio and the Baud Rate

Bit rate (bps)	When IOCLK = 20 MHz		
	Timer frequency division ratio 1	Timer frequency division ratio 2	Transfer rate error
230,400	1	87	0.22%
115,200	2	87	0.22%
56,000	3	118	0.04%
38,400	5	104	0.16%
19,200	9	116	0.22%
9,600	17	123	0.37%
4,800	33	126	0.21%
2,400	66	126	0.21%
1,200	132	126	0.21%

Table 72 Relationship Between the Timer frequency division ratio and the Baud Rate

Bit rate (bps)	When IOCLK = 25 MHz		
	Timer frequency division ratio 1	Timer frequency division ratio 2	Transfer rate error
230,400	1	109	0.45%

115,200	2	109	0.45%
56,000	4	112	0.35%
38,400	6	109	0.45%
19,200	11	118	0.31%
9,600	21	124	0.01%
4,800	42	124	0.01%
2,400	83	126	0.4%
1,200	165	126	0.21%

12.6. Cautions

The transmission interrupt function generated by the XCTS pin does not disable actual transmitting operations. Therefore, when data remain in the transmission buffer and shift register for transmission while the XCTS pin is under transmission interruption, the transmission are interrupted after all remaining data have transmitted.
Write to SC2TXB must be performed after confirming that the transmission buffer is empty. The confirmation should be carried out by bit 5 of SC2STR, or by setting bit 4 of SC2ICR to 1 and during interrupt processing.
In using the serial interface, other registers must be set before bit 15 or 14 of SC0CTR, SC1CTR, SC2CTR is set to 1, transmitted, or allowed to receive. . Other registers must be set after the above-mentioned bits are set to 0, transmitted or allowed to receive. Do not change it during transmission or with the remaining data in the transmission buffer. If any change is found, operations are not guaranteed.

Chapter 13 Interrupt controller (INTC)

13

13.1. General

The interrupt controller processes nonmaskable interrupts and level interrupts (both internal and external). The interrupt controller possesses 8 external interrupt pins for external input and one nonmaskable interrupt pin. External level interrupts are processed by the external pin interrupt signal level that is maintained for 2 or more IOCLK cycles.

13.2. Features

- Interrupt groups
  42 interrupt groups
- Interrupt mask levels
  An interrupt mask level can be set for each interrupt group.
- External pin interrupt conditions
  Positive edge, negative edge, "H" level, or "L" level
  Recovery is possible from STOP, HALT and SLEEP modes by using an external pin interrupt.

13.3. Interrupt Signal Assignments

Table 73 Interrupt signal assingments

Interrupt source	Purpose/Point of connection	Internal/external to LSI
Group 0	External NMI pin (XNMI) Watchdog timer Asynchronous bus error	External Internal Internal
Group 1	System reserve	-
Group 2	Timer 0 underflow	Internal
Group 3	Timer 1 underflow	Internal
Group 4	Timer 2 underflow	Internal
Group 5	Timer 3 underflow	Internal
Group 6	Timer 4 underflow	Internal
Group 7	Timer 5 underflow	Internal
Group 8	Timer 6 overflow	Internal
Group 9	Timer 6A	Internal
Group 10	Timer 6B	Internal
Group 11	Timer 7 underflow	Internal
Group 12	Timer 8 underflow	Internal
Group 13	Timer 9 underflow	Internal
Group 14	Timer 10 underflow	Internal
Group 15	Timer 11 underflow	Internal
Group 16	DMA 0 transfer end	Internal
Group 17	DMA 1 transfer end	Internal

Group 18	DMA 2 transfer end	Internal
Group 19	DMA 3 transfer end	Internal
Group 20	Serial 0 receive	Internal
Group 21	Serial 0 transmit	Internal
Group 22	Serial 1 receive	Internal
Group 23	Serial 1 transmit	Internal
Group 24	Serial 2 receive	Internal
Group 25	Serial 2 transmit	Internal
Group 26	System reserve	-
Group 27	I2C Port 0	Internal
Group 28	I2C Port 1	Internal
Group 29	IrDA	Internal
Group 30	Analog front end	Internal
Group 31	AD conversion end	Internal
Group 32	Real-time clock	Internal
Group 33	System reserve	-
Group 34	External interrupt 0	External
Group 35	External interrupt 1	External
Group 36	External interrupt 2	External
Group 37	External interrupt 3	External
Group 38	External interrupt 4	External
Group 39	External interrupt 5	External
Group 40	External interrupt 6	External
Group 41	External interrupt 7	External

13.4. Description of Registers

Table 74 Interrupt controller register

Address	Symbol	NAME	Number of bits	Initial value	Access size
0xD4000000	G0ICR	Nonmaskable interrupt control Register	16	0x 0000	8, 16
0xD4000004	G1ICR	Group 1 interrupt control register	16	0x 0000	8, 16
0xD4000008	G2ICR	Group 2 interrupt control register	16	0x 0000	8, 16
0xD400000C	G3ICR	Group 3 interrupt control register	16	0x 0000	8, 16
0xD4000010	G4ICR	Group 4 interrupt control register	16	0x 0000	8, 16
0xD4000014	G5ICR	Group 5 interrupt control register	16	0x 0000	8, 16
0xD4000018	G6ICR	Group 6 interrupt control register	16	0x 0000	8, 16
0xD400001C	G7ICR	Group 7 interrupt control register	16	0x 0000	8, 16
0xD4000020	G8ICR	Group 8 interrupt control register	16	0x 0000	8, 16
0xD4000024	G9ICR	Group 9 interrupt control register	16	0x 0000	8, 16
0xD4000028	G10ICR	Group 10 interrupt control register	16	0x 0000	8, 16
0xD400002C	G11ICR	Group 11 interrupt control register	16	0x 0000	8, 16
0xD4000030	G12ICR	Group 12 interrupt control register	16	0x 0000	8, 16
0xD4000034	G13ICR	Group 13 interrupt control register	16	0x 0000	8, 16
0xD4000038	G14ICR	Group 14 interrupt control register	16	0x 0000	8, 16
0xD400003C	G15ICR	Group 15 interrupt control register	16	0x 0000	8, 16

0xD4000040	G16ICR	Group 16 interrupt control register	16	0x 0000	8, 16
0xD4000044	G17ICR	Group 17 interrupt control register	16	0x 0000	8, 16
0xD4000048	G18ICR	Group 18 interrupt control register	16	0x 0000	8, 16
0xD400004C	G19ICR	Group 19 interrupt control register	16	0x 0000	8, 16
0xD4000050	G20ICR	Group 20 interrupt control register	16	0x 0000	8, 16
0xD4000054	G21ICR	Group 21 interrupt control register	16	0x 0000	8, 16
0xD4000058	G22ICR	Group 22 interrupt control register	16	0x 0000	8, 16
0xD400005C	G23ICR	Group 23 interrupt control register	16	0x 0000	8, 16
0xD4000060	G24ICR	Group 24 interrupt control register	16	0x 0000	8, 16
0xD4000064	G25ICR	Group 25 interrupt control register	16	0x 0000	8, 16
0xD4000068	G26ICR	Group 26 interrupt control register	16	0x 0000	8, 16
0xD400006C	G27ICR	Group 27 interrupt control register	16	0x 0000	8, 16
0xD4000070	G28ICR	Group 28 interrupt control register	16	0x 0000	8, 16
0xD4000074	G29ICR	Group 29 interrupt control register	16	0x 0000	8, 16
0xD4000078	G30ICR	Group 30 interrupt control register	16	0x 0000	8, 16
0xD400007C	G31ICR	Group 31 interrupt control register	16	0x 0000	8, 16
0xD4000080	G32ICR	Group 32 interrupt control register	16	0x 0000	8, 16
0xD4000084	G33ICR	Group 33 interrupt control register	16	0x 0000	8, 16
0xD4000088	G34ICR	Group 34 interrupt control register	16	0x 0000	8, 16
0xD400008C	G35ICR	Group 35 interrupt control register	16	0x 0000	8, 16
0xD4000090	G36ICR	Group 36 interrupt control register	16	0x 0000	8, 16
0xD4000094	G37ICR	Group 37 interrupt control register	16	0x 0000	8, 16
0xD4000098	G38ICR	Group 38 interrupt control register	16	0x 0000	8, 16
0xD400009C	G39ICR	Group 39 interrupt control register	16	0x 0000	8, 16
0xD40000A0	G40ICR	Group 40 interrupt control register	16	0x 0000	8, 16
0xD40000A4	G41ICR	Group 41 interrupt control register	16	0x 0000	8, 16
0xD4000100	IAGR	Interrupt acceptance group register	16	0x 0000	8, 16
0xD4000200	EXTMD	External pin interrupt condition specification register	16	0x 0000	8, 16

13.4.1. Relationship Between the Timer frequency division ratio and the Baud Rate

For the details of the nonmaskable interrupt control register, refer to 2.3.4.7 “NMI control register”.

13.4.2. Group n Interrupt Control Register

Register symbol	GnICR
Address	0xD4000000 +(n * 4)
Purpose	These registers are for controlling the level interrupts for Group 2 through 41. They are used to enable, request and confirm detection of these interrupts.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved	LV[2:0]			reserved			IE

Initial value	0	0			0			0
R/W	R	RW			R			RW
Bit	7	6	5	4	3	2	1	0
Bit name	reserved			IR	reserved			ID
Initial value	0			0	0			0
R/W	R			RW	R			RW

Bit	Bit name	Description
15	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
14-12	LV[2:0]	Group n Interrupt Priority Level Sets the interrupt level. The interrupt for the corresponding interrupt group is enabled when the interrupt level set by LV[2:0] is smaller than IM[2:0] of PSW.
11-9	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
8	IE	Group n Interrupt Enable Flag Enables the interrupt. The interrupt is enabled when IE is "1". The interrupt is generated when IE is set while IR has already been set.
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4	IR	Group n Interrupt Request Flag Registers interrupt requests. This flag is cleared by software inside the interrupt processing program after the interrupt is accepted.
3-1	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
0	ID	Group n Interrupt Detect Flag Stores the logical product of IE and IR. When an interrupt enabled by IE is generated, the bit corresponding to the interrupt is set to "1".

The flag is changed depending on the data written to GnICR as given in the table below.
<Programming note>
Be sure to clear the interrupt request flag in the interrupt processing program in order to clear the generated interrupt request. Although IR=1 after receiving the interrupt, IR is cleared by writing IR=0 and ID=1.

WRITE DATA		Write result	
IR	ID	IR	ID
0	0	No change	No change
1	0	No change	No change
0	1	0	0
1	1	1	IE value

13.4.3. Interrupt Acceptance Group Register

Register symbol IAGR
Address 0xD4000100

Purpose Reads the group number for which an interrupt request was generated among those accepted interrupt levels.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	GN[5:0]						reserved	
Initial value	0						0	
R/W	R						R	

Bit	Bit name	Description
15-8	Reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
7-2	GN[5:0]	Group Number Register This stores the smallest group number of the groups registered as the maskable interrupts of the level shown by the value of PSW.IM.
1-0	Reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.

<Programming note>
If CPU accepts a maskable interrupt, IE becomes 0 because the accepted level is reflected to IM[2:0] of PSW. (Refer to 2.6.4.3 maskable interrupt) Normally, a group number can be specified by reading out IAGR up to PSW.IE=1 in the interrupt handler. When there is a interrupt request by the group with the smaller numbers before reading out IAGR, IAGR is updated to the group number.

13.4.4. External Pin Interrupt Condition Specification Register

Register symbol EXTMD
Address 0xD4000200
Purpose Sets the conditions for generating an external interrupt. Any level or edge is set for each pin.

Bit	15	14	13	12	11	10	9	8
Bit name	IR7[1:0]		IR6[1:0]		IR5[1:0]		IR4[1:0]	
Initial value	0		0		0		0	
R/W	RW		RW		RW		RW	
Bit	7	6	5	4	3	2	1	0
Bit name	IR3[1:0]		IR2[1:0]		IR1[1:0]		IR0[1:0]	
Initial value	0		0		0		0	
R/W	RW		RW		RW		RW	

Bit	Bit name	Description
15-14	IR7[1:0]	XIRQ7 Pin Trigger Condition Setting 00 : L LEVEL

Bit	Bit name	Description
		01 : H level 10 : Negative edge 11 : Positive edge The external interrupt signal should be maintained for 2 or more SYSCLK cycles.
13-12	IR6[1:0]	XIRQ6 Pin Trigger Condition Setting (same as IR7[1:0])
11-10	IR5[1:0]	XIRQ5 Pin Trigger Condition Setting (same as IR7[1:0])
9-8	IR4[1:0]	XIRQ4 Pin Trigger Condition Setting (same as IR7[1:0])
7-6	IR3[1:0]	XIRQ3 Pin Trigger Condition Setting (same as IR7[1:0])
5-4	IR2[1:0]	XIRQ2 Pin Trigger Condition Setting (same as IR7[1:0])
3-2	IR1[1:0]	XIRQ1 Pin Trigger Condition Setting (same as IR7[1:0])
1-0	IR0[1:0]	XIRQ0 Pin Trigger Condition Setting (same as IR7[1:0])

13.5. Description of Operation

The following interrupt processing is performed.

- Nonmaskable interrupts
  External NMI pin interrupt
  Watchdog timer overflow interrupt
  Asynchronous bus error interrupt
- Internal interrupts
  Timer, serial, DMA software modem I/F, A/D conversion, IrDA, I2C interrupts and real-time clock
- External interrupts
  8 external interrupt pins

When a level interrupt is generated, the interrupt group is determined and then a interrupt request us generated in CPU.
When an interrupt signal is accepted, a determination is made whether it is a nonmaskable interrupt or level interrupt. If it is a level interrupt and interrupts are enabled by the IE bit of the corresponding GnICR register, the interrupt group is determined by the group the accepted interrupt cause belongs to.
Once the interrupt group is determined, the interrupt control register (GnICR) for the group is controlled, and an interrupt request is sent by informing the CPU of the interrupt level of the interrupt group. The interrupt group number is also set in the interrupt acceptance group register (IAGR).
The interrupt level for a group can be found by reading the interrupt priority level LV [2:0] of the appropriate interrupt control register (GnICR).
When multiple interrupt signals have been accepted, the group for each is determined and the interrupt group having the highest priority level is selected. If the group level is the same, the group having the smallest group number is given priority.
An NMI interrupt request is sent directly to the CPU for nonmaskable interrupts.

13.6. Cautions

(1) Maintain the interrupt signal of external pins for 2 or more SYSCLK cycles.
(2) A write buffer for writing data to the bus controller (BCU) is implemented to achieve higher CPU processing speed. When returning from the interrupt program after it has cleared the IR and ID bits of the GnICR register, be sure to read the GnICR register between executing a write instruction and returning from the interrupt program in order to synchronize with the store buffer of the bus controller.
(3) Update GnICR after clearing the IE bit of the PSW.

Chapter 14 Watchdog timer (WDT)

14

14.1. General

Including an internal 25-bit binary counter, this microcontroller can be used as a 16 to 25-bit watchdog timer.
A nonmaskable interrupt or self-reset is generated if the watchdog timer overflows.
This feature can also be used as a timer for waiting for stabilization of PLL oscillation.

14.2. Features

- Number of bits of the binary counter can be selected
  Select from 16, 18, 20, 22 and 24 bits
  Overflow cycle
    3.276 ms to 838.860 ms (when the oscillation frequency is 20 MHz)
    1.986 ms to 508.400 ms (when the oscillation frequency is 33 MHz)
- A nonmaskable interrupt or self-reset can be generated when the watchdog timer overflows.
- PLL oscillation stabilization wait time
  When canceled by reset
    Number of bits: Fixed at 18 bits
      13.107 ms (when the oscillation frequency is 20 MHz)
      7.9438 ms (when the oscillation frequency is 33 MHz)
  When returning from STOP mode
    Number of bits: Setting value of WDCK[2:0]
      3.276 ms to 838.860 ms (when the oscillation frequency is 20 MHz)
      1.986 ms to 508.400 ms (when the oscillation frequency is 33 MHz)
- It is possible to perform a self-reset inside the chip by writing to the RSTCTR register.

14.3. Description of Registers

Table 75 Watchdog timer register

Address	Symbol	Name	Number of bits	Initial value	Access size
0x C0001000	WDBC	Watchdog binary counter	8	0x 00	8, 16
0x C0001002	WDCTR	Watchdog timer control register	8	Note	8, 16
0x C0001004	RSTCTR	Reset control register	8	0x 00	8, 16

Note: For the initial value, refer to 14.3.2 Watchdog Timer Control Register, page 339.

14.3.1. Watchdog Binary Counter

Register symbol WDBC
Address 0xC0001000
Purpose Reads the counter value of the upper 8 bits of the watchdog timer.

Bit	7	6	5	4	3	2	1	0
Bit name	WDBC[7:0]							
Initial value	0							
R/ W	R							

Bit	Bit name	Description
7-0	WDBC[7:0]	Counter Value of the Upper 8 Bits of the Watchdog Timer The read value is not guaranteed if the value of the upper 8 bits of the 25-bit watchdog timer changes while being read.

14.3.2. Watchdog Timer Control Register

Register symbol WDCTR
Address 0xC0001002
Purpose Sets the operational control conditions for the watchdog timer.

Bit	7	6	5	4	3	2	1	0
Bit name	WDCNE	WDRST	reserved			WDCK[2:0]		
Initial value	0	0	Note			001		
R/ W	RW	RW	RW			RW		

Note: For the initial values, refer to the following description.

Bit	Bit name	Description
7	WDCNE	Watchdog Timer Count Operation Control 0 : Stop count operation 1 : Enable count operation
6	WDRST	Binary Counter Reset 0 : Do not reset 1 : Reset

Bit	Bit name	Description
		The binary counter is reset by writing "1" here. The read value of the WDRST bit is always "0".
5-3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2-0	WDCK[2:0]	Clock Source Selection Selects the clock source for the upper 8 bits of the counter. 000 : $1/2^8$ (1/256) of OSCI input 001 : $1/2^{10}$ (1/1024) of OSCI input 010 : $1/2^{12}$ (1/4096) of OSCI input 011 : $1/2^{14}$ (1/16384) of OSCI input 100 : $1/2^{16}$ (1/65536) of OSCI input 101 : Setting prohibited 110 : Setting prohibited 111 : Setting prohibited

14.3.3. Reset Control Register

Register symbol RSTCTR
Address 0x C0001004
Purpose An internal reset can be generated by program.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved				WDREN	WDTRST	DBFRST	CHIPRST
Initial value	0				0	0	0	0
R/ W	R				RW	RW	RW	RW

Bit	Bit name	Description
7-4	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
3	WDREN	Watchdog Timer Reset Enable A self-reset (internal reset) is generated when a watchdog overflow has been caused by setting this flag to "1". The XRSTOUT pin is driven to "L" level during the self-reset cycle. 0 : Generates a nonmaskable interrupt 1 : Generates a self-reset (internal reset)
2	WDTRST	Watchdog Timer Reset Flag Indicates that a self-reset was generated through watchdog timer overflow. 0 : Not generated 1 : Generated "1" cannot be written to this bit. It is possible to clear the bits that are set to "1" by using software to write "0" . The WDTRST flag can be cleared by using an external reset signal or using a program to write "0" here.
1	DBFRST	Double Fault Reset Flag This flag is set to "1" when the CPU generates a self-reset caused by a double fault. The value is maintained even after a reset due to a double fault. "1" cannot be written to this bit. The DBFRST flag can be cleared by using an external reset signal or using a program to write "0" here.
0	CHIPRST	This flag is used to generate a self-reset (internal reset).

Bit	Bit name	Description
		A self-reset is generated by re-writing this flag from "0" to "1". No self-reset is generated if "1" is written to the flag while it is already set to "1". The value of this flag is maintained even after self-reset. The CHIPRST flag can be cleared using an external reset signal or using a program to write "0" here.

14.4. Description of Operation

14.4.1. Oscillation Stabilization Wait Operation

This microcontroller functions as an oscillation stabilization wait timer when operations are canceled by reset or when returning from STOP mode. This operation is enabled even when the WDCNE flag is "0".
This functions as a counter having the number of bits specified by WDCK[2:0] when returning from STOP mode. The oscillation stabilization wait time can be selected as a time calculated by:
overflow cycle = $2^{(n+WDCK\times2)}/(f \times 10^3)$[ms]
n = 16, WDCK = WDCK [2:0] where f is the oscillation input frequency given in units of MHz.
[Examples] 1.986ms when WDCK=0, f=33MHz
7.944ms when WDCK=1, f=33MHz
The oscillation cell for the external oscillator stops in STOP mode. Be sure to set a sufficient amount of time for the oscillation stabilization wait time by considering the actual time necessary for the oscillator to stabilize.
A nonmaskable interrupt is not generated if the WDCNE flag is "1" even when returning from STOP mode.

14.4.2. Watchdog Operation

It is possible to have the microcontroller function as a watchdog by setting the WDCNE bit of the WDCTR register to "1".
When the watchdog timer overflows, a nonmaskable interrupt is generated if the WDREN bit of the RSTCTR register is “0”, while a self-reset (internal reset) is generated if the WDREN bit is “1”. During the self-reset cycle, the XRSTOUT output signal is driven to "L" level.
Be sure to reset the counter by writing "1" to the WDRST bit before setting the WDCNE bit to "1".
Be sure to stop the watchdog timer by setting the WDCNE flag to "0" when transiting to HALT or SLEEP mode.

14.4.3. Self-Reset Operation

The chip is internally reset when "1" is written to the CHIPRST bit of the RSTCTR register. No oscillation stabilization wait operation is performed in this case.
Reset generated by writing to the CHIPRST flag is performed by a reset signal internal to the chip. The reset output pin (XRSTOUT) is driven to "L" level during the reset cycle.

14.5. Cautions

If the value of WDCK[2:0] is re-written, be sure to update after first stopping the watchdog timer and resetting the counter.

Chapter 15 AFE interface (AFE)

15

15.1. General

Possessing a function for interfacing with the external AFE (Analog Front End) device, it is possible to implement a modem function such as V.22 or V.34.

15.2. Features

- Serial communications with Analog front end devices
  Parallel-to-serial conversion of output data and serial-to-parallel conversion of input data
- Built-in transmit and receive FIFOs
  Each FIFO is 16 bits wide with 16 entries
- Interrupt generated based on status of transmit and receive FIFOs
  When transmit FIFO is FULL
  When transmit FIFO is EMPTY
  When the number of data words in the transmit FIFO is less than that set in the AFEFIFO register
  When the number of data words in the receive FIFO is greater than or equal to that set in the AFEFIFO register
  When the transmit FIFO or receive FIFO has overflowed
  When the transmit FIFO or receive FIFO has underflowed
- NCU control
  On-hook/off-hook control necessary when connecting to a public telephone and detection of various states can be performed using an external analog circuit. This control and detection is performed using the general-purpose I/O port.
- Eye pattern output
  This microcontroller supports an eye pattern output function for confirming the phase and signal level of the modem receive signal. After software processing is performed, the receive signal undergoes parallel-serial conversion and is output on the eye pattern output pin by writing to the eye pattern register.

15.3. Register

Table 76 Analog front end interface register

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8300000	AFESYS	AFE system control register	16	0x 0003	8, 16
0xD8300004	AFEINTM	AFE interrupt mask register	16	0x 00FF	8, 16
0xD8300008	AFESTAT	AFE status register	16	0x 0048	8, 16
0xD830000C	AFECTR	AFE control register	16	0x 0300	8, 16
0xD8300010	AFETBUF	AFE transmit buffer register	16	Undefined	16
0xD8300014	AFERBUF	AFE receive buffer register	16	Undefined	16
0xD8300018	AFEFIFO	AFE FIFO size register	16	0x1100	8, 16
0xD830001C	AFEEYE	AFE eye pattern register	16	0x0000	8, 16
0xD8300020	AFESEC	AFE second source register	16	0x000C	8, 16

15.3.1. Analog Front End System Control Register

Register symbol AFESYS
Address 0xD8300000
Purpose Sets the control mode of the AFE chip and resets the transmit and receive FIFOs.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/ W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	reserved		HXSSEL	reserved			TFRST	RFRST
Initial value	0		0	0			1	1
R/ W	R		RW	R			RW	RW

Bit	Bit name	Description
15-6	RESERVED	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
5	HXSSEL	Analog front End Control Mode Setting Sets the Analog front End control system. 0 : Software mode is used for AFE 1 : Hardware mode is used for AFE
4-2	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
1	TFRST	Transmit FIFO Reset Resets the data pointers of the transmit FIFO (read pointer and write pointer). If this bit is "1", the data pointer of the FIFO is reset to "0". The data pointers of the transmit FIFO can be changed by clearing this bit.
0	RFRST	Receive FIFO Reset Resets the data pointers of the receive FIFO. If this bit is "1", the data pointer of the FIFO is reset to "0". The data pointers of the receive FIFO can be changed by clearing this bit.

15.3.2. Analog Front End Interrupt Mask Register

Register symbol	AFEINTM
Address	0xD8300004
Purpose	Sets the interrupt mask for the AFE register.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/ W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	INTM[7:0]							
Initial value	FF							
R/ W	RW							

Bit	Bit name	Description
15-8	RESERVED	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
7-0	INTM[7:0]	Interrupt Mask Masks the corresponding interrupt cause of the AFESTAT register when each bit is "1".

15.3.3. Analog Front End Status Register

Register symbol	AFESTAT
Address	0xD8300008
Purpose	Sets the interrupt mask for the AFE register.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/ W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	TFFUL	TFHF	TFOVF	TFUDF	RFEMP	RFHF	RFOVF	RFUDF
Initial value	0	1	0	0	1	0	0	0
R/ W	R	R	R	R	R	R	R	R

Bit	Bit name	Description
15-8	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
7	TFFUL	Transmit FIFO Full When this bit is "1", it indicates that the transmit FIFO is full.
6	TFHF	Transmit FIFO Half-Full When this bit is "1" when transmitting data, it indicates that the amount of data in the transmit FIFO is less than the number of words set by TXN[3:0] of AFEFIFO.
5	TFOVF	Transmit FIFO Overflow A value of "1" indicates a transmit FIFO overflow. The value of this bit can be returned to "0" by resetting the transmit FIFO.
4	TFUDF	Transmit FIFO Underflow A value of "1" indicates a transmit FIFO underflow. The value of this bit

Bit	Bit name	Description
		can be returned to "0" by resetting the transmit FIFO.
3	RFEMP	Receive FIFO Empty When this bit is "1", it indicates that there is no data in the receive FIFO.
2	RFHF	Receive FIFO Half-Full When this bit is "1" while receiving data, it indicates that the amount of data in the receive FIFO is greater than or equal to the number of words set by TXN[3:0] of AFEFIFO.
1	RFOVF	Receive FIFO Overflow A value of "1" indicates a receive FIFO overflow. The value of this bit can be returned to "0" by resetting the receive FIFO.
0	RFUDF	Receive FIFO Underflow A value of "1" indicates a receive FIFO underflow. The value of this bit can be returned to "0" by resetting the receive FIFO.

15.3.4. Analog Front End Control Register

Register symbol	AFECTR
Address	0xD830000C
Purpose	Used to write the command to be transferred to the AFE chip.

Bit	15	14	13	12	11	10	9	8
Bit name	CTR[15:8]							
Initial value	0300							
R/ W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	CTR[7:0]							
Initial value	0							
R/ W	RW							

Bit	Bit name	Description
15-0	CTR[15:0]	Command Data The setting value in this field is passed to the Analog front End as a command the next time a register value is transferred to the Analog front End. It is necessary to set initial values into this field if you want to keep using the internal Analog front End register as initial values.

15.3.5. Analog Front End Transmit Buffer Register

Register symbol	AFETBUF
Address	0xD8300010
Purpose	Used to write transmit data to the Analog front End chip.

Bit	15	14	13	12	11	10	9	8
Bit name	TD[15:8]							
Initial value	Undefined							
R/ W	W							
Bit	7	6	5	4	3	2	1	0
Bit name	TD[7:0]							
Initial value	Undefined							
R/ W	W							

Bit	Bit name	Description
15-0	TD[15:0]	Transmit data Sets transmit data to the analog front end in this field.

15.3.6. Analog Front End Receive Buffer Register

Register symbol	AFERBUF
Address	0xD8300014
Purpose	Reads receive data from the analog front end chip.

Bit	15	14	13	12	11	10	9	8
Bit name	RD[15:8]							
Initial value	Undefined							
R/ W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	RD[7:0]							
Initial value	Undefined							
R/ W	R							

Bit	Bit name	Description
15-0	RD[15:0]	Receive data Stores data received from the analog front end into this field.

15.3.7. Analog Front End FIFO Size Register

Register symbol	AFEFIFO
Address	0xD8300018
Purpose	Sets the number of words in the FIFO for generating a transmit or receive FIFO interrupt.

Bit	15	14	13	12	11	10	9	8
Bit name	TXN[3:0]				RXN[3:0]			
Initial value	0001				0001			
R/ W	RW				RW			
Bit	7	6	5	4	3	2	1	0
Bit name	reserved							
Initial value	0							
R/ W	R							

Bit	Bit name	Description
15-12	TXN[3:0]	Transmit FIFO Half-Full If the data in the transmit FIFO is less than the number of words set by this field, the interrupt flag AFESTAT.TFHF for the CPU will be set. Note, however, that setting "0" and "1" is prohibited and operations are not guaranteed under these settings.
11-8	RXN[3:0]	Receive FIFO Half-Full If the amount of data in the receive FIFO is greater than or equal to the number of words set by this field, the interrupt flag AFESTAT.RFHF for the CPU will be set. Note, however, that setting "0" and "1" is prohibited and operations are not guaranteed under these settings.
7-0	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.

15.3.8. Analog Front End Eye Pattern Register

Register symbol	AFEEYE
Address	0xD830001C
Purpose	Sets the X and Y coordinates for the eye pattern.

Bit	15	14	13	12	11	10	9	8
Bit name	EYEX[7:0]							
Initial value	0							
R/ W	RW							
Bit	7	6	5	4	3	2	1	0
Bit name	EYEY[7:0]							
Initial value	0							
R/ W	RW							

Bit	Bit name	Description
15-8	EYEX[7:0]	Eye Pattern Data X Sets the X coordinate for the eye pattern.
7-0	EYEY[7:0]	Eye Pattern Data Y Sets the Y coordinate for the eye pattern.

15.3.9. Analog Front End Second Source Register

Register symbol AFESEC
Address 0xD8300020
Purpose Sets the specific control mode for each analog front end device.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							AFEREG
Initial value	0							0
R/W	R							R
Bit	7	6	5	4	3	2	1	0
Bit name	reserved				FS2SEL	FSIXO	FSPRD	FSINV
Initial value	0				1	1	0	0
R/W	R				RW	RW	RW	RW

Bit	Bit name	Description
15-9	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
8	AFEREG	AFE Transfer Register When this bit is "1", it indicates that the AFFS used for register value transfer is active and that a register value is currently being transferred.
7-4	Reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
3	FS2SEL	AFFS2 Enable/Disable Select Selects whether to use or not use AFFS2 for transferring register values. 1 : Do not use 0 : Use
2	FSIXO	AFFS Input/Output Select Sets the I/O mode for the AFFS signal. 0 : Creates AFFS internally and uses as an output signal 1 : Uses AFFS as an input signal from AFE
1	FSPRD	AFFS Active Period Sets the active period for AFFS. 0 : 1SCLK before data transfer starts 1 : Currently transferring data
0	FSINV	AFFS Inversion Sets the polarity for active AFFS. 0 : Positive logic 1 : Negative logic

15.4. Description of Operation

15.4.1. Data Transmit and Receive

Data transmission to analog front end is started by writing data to the transmit buffer register (AFETBUF). Data can be read from the receive buffer register (AFERBUF) for data received from the analog front end.
Transmit data written to the transmit buffer register accumulates in the 16-level, 16-bit transmit FIFO. Accumulated data is synchronized to the shift clock (AFSCLK) and undergoes parallel-serial conversion, and is output to AFTXD after frame synchronization signal (AFFS) output (input).
Data received from AFRXD is input in synchronization with AFSCLK after AFFS output (input). This data is input to the 16-level, 16-bit receive FIFO after serial-parallel conversion.
The following figure shows the configuration of the analog front end interface.

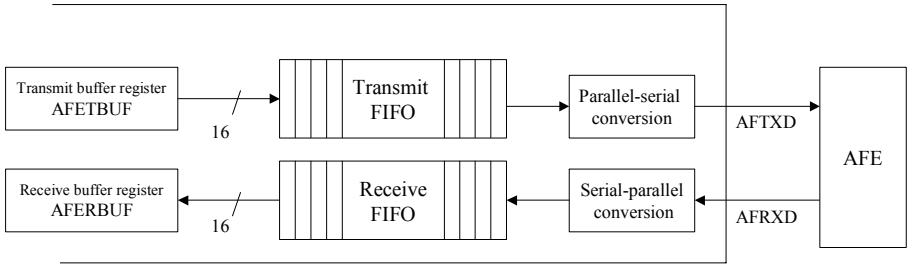


Figure 90 Analog front end E interface configuration

15.4.2. Interrupts

The AFE interface generates the following interrupts. The interrupt cause is indicated by the AFESTAT register. Interrupts enabled by the AFEINTM register can be generated.

- When the transmit FIFO is full of data
- When the receive FIFO is empty of data
- When the number of data words in the transmit FIFO is less than that set in the AFEFIFO register
- When the number of data words in the receive FIFO is greater than or equal to that set in the AFEFIFO register
- When the transmit FIFO or receive FIFO has overflowed
- When the transmit FIFO or receive FIFO has underflowed

15.4.3. NCU Control

NCU control is used for on-hook/off-hook control, on-hook/off-hook detection, circuit broken detection, and ring detection for analog telephone circuits. An external analog circuit is necessary for NCU control of these items. Control by the CPU is performed via the general-purpose I/O port.
The various control features are described below.

- Off-hook detection
  Detects that the connected telephone line is in use by another telephone on the same telephone line (off-hook status).
- Circuit broken detection
  Detects that the connected telephone line is broken.

- On-hook detection
  Detects that the connected telephone circuit is currently on-hook.
- Ring detection
  Detects that the telephone line is ringing. It is usually necessary to generate an interrupt upon detecting this signal.

15.4.4. Example Connections with Analog Front End Devices

The following figure shows an example of connecting to an AFE chip.

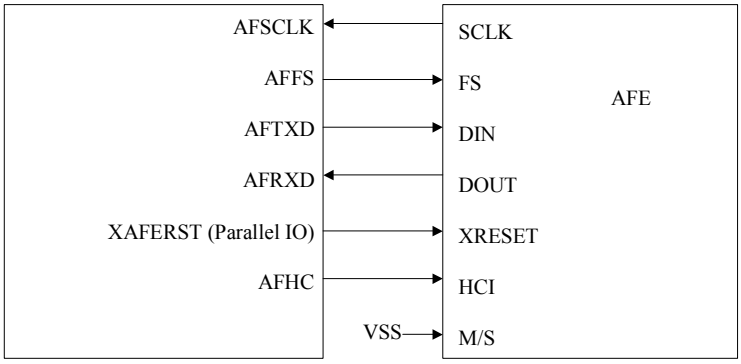


Figure 91 Example Connections with Analog Front End Devices

Chapter 16 A/D Converter (ADC)

16

16.1. General

This 10-bit load redistribution system A/D converter can process up to 8 analog signal channels.
1/4, 1/8, 1/16, or 1/32 of the IOCLK can be selected as the A/D conversion reference clock. When IOCLK = 20 MHz, A/D conversion is performed at a maximum conversion rate of 2.6 µs per channel (when 1/4 of IOCLK has been selected as the A/D conversion reference clock.)

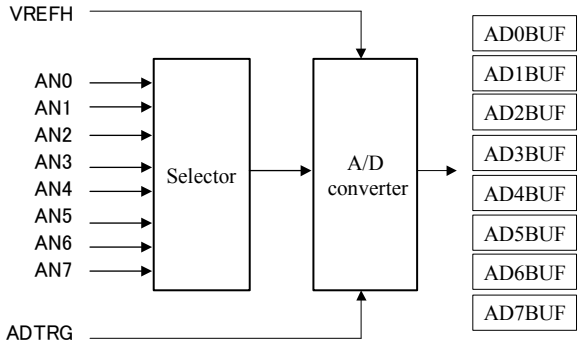


Figure 92 A/D converter

16.2. Features

- Conversion precision (relative precision) of 10 bits ±5 LSB. The value given by dividing VREFH (max. 3.3 V) by 1024 is stored in AD0BUF through AD7FBUF.
- Conversion reference clock
  1/4, 1/8, 1/16, or 1/32 can be selected as the IOCLK
  Be sure to set so that 1 cycle is 200 ns or more.
  (For example, if IOCLK = 25 MHz, be sure to set 1/8, 1/16, or 1/32.)
- Conversion rate
  2.6 µs/ch (IOCLK = 20 MHz, 1/4 division of IOCLK conversion reference clock)
  4.16 µs/ch (IOCLK = 25 MHz, 1/8 division of IOCLK conversion reference clock)
- 16 types of operation modes
  One-time conversion for channel 0, continuous conversion for channel 0
  One-time conversion for channel 1, continuous conversion for channel 1
  One-time conversion for channel 2, continuous conversion for channel 2
  One-time conversion for channel 3, continuous conversion for channel 3
  One-time conversion for channel 4, continuous conversion for channel 4
  One-time conversion for channel 5, continuous conversion for channel 5
  One-time conversion for channel 6, continuous conversion for channel 6
  One-time conversion for channel 7, continuous conversion for channel 7
- Conversion start
  Upon Timer 2 underflow
  Upon trigger input (falling edge) to external pin (ADTRG pin)
  Upon writing to register using software
- Interrupts
  An interrupt request is generated when a single channel ends.

16.3. Description of Registers

Table 77 A/D conversion register

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8500000	ADCTR	A/D conversion control register	16	0x0000	16
0xD8500010	AD0BUF	A/D 0 conversion data buffer	16	0x0000	16
0xD8500012	AD1BUF	A/D 1 conversion data buffer	16	0x0000	16
0xD8500014	AD2BUF	A/D 2 conversion data buffer	16	0x0000	16
0xD8500016	AD3BUF	A/D 3 conversion data buffer	16	0x0000	16
0xD8500018	AD4BUF	A/D 4 conversion data buffer	16	0x0000	16
0xD850001A	AD5BUF	A/D 5 conversion data buffer	16	0x0000	16
0xD850001C	AD6BUF	A/D 6 conversion data buffer	16	0x0000	16
0xD850001E	AD7BUF	A/D 7 conversion data buffer	16	0x0000	16

16.3.1. A/D Conversion Control Register

Register symbol ADCTR
Address 0xD8500000
Purpose Sets the A/D conversion mode.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved				STBY	ADSC[2:0]		
Initial value	0				0	0		
R/W	R				RW	RW		
Bit	7	6	5	4	3	2	1	0
Bit name	ADEN	ADST[1:0]		reserved	ADCK[1:0]		ADMD[1:0]	
Initial value	0	0		0	0		0	
R/W	RW	RW		R	RW		RW	

Bit	Bit name	Description
15-12	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
11	STBY	Standby Mode Switches the operational mode. 0 : Standby mode (conversion stop mode) 1 : Conversion mode
10-8	ADSC[2:0]	Selects the channel to be converted. 000 : AN0 001 : AN1 010 : AN2 011 : AN3 100 : AN4 101 : AN5 110 : AN6 111 : AN7
7	ADEN	Conversion Control Status Bit Indicates start, stop or conversion status for A/D conversion. 0 : Conversion stop 1 : Conversion start or conversion underway
6-5	ADST[1:0]	Conversion Start Trigger Select 00 : Uses software trigger 01 : Sets ADEN bit at falling edge of ADTRG external pin. 10 : Sets the ADEN bit upon a Timer 2 underflow interrupt. 11 : Setting prohibited
4	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
3-2	ADCK[1:0]	Selects the conversion reference clock. 00 : 1/4 IOCLK 01 : 1/8 IOCLK 10 : 1/16 IOCLK 11 : 1/32 IOCLK
1-0	ADMD[1:0]	Selects the conversion mode. 00 : One-time conversion for any single channel 10 : Continuous conversion for any single channel

16.3.2. A/D Conversion Data Buffer

Register symbol ADNBUF
Address 0xD8500010 +(0x2 * n)
Purpose Used to read the result of A/D conversion.

Bit	15	14	13	12	11	10	9	8
Bit name	ADnBUF[15:8]							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	ADnBUF[7:6]		reserved					
Initial value	0		0					
R/W	R		R					

Bit	Bit name	DESCRIPTION
15-6	ADnBUF[15:8]	The result of A/D conversion is stored in the 10 bits from 15 through 6. Bits 5 through 0 always read out as "0".
5 - 0	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.

16.4. Description of Operation

16.4.1. Selecting the Operational Mode

(1) One-time conversion for any single channel
Selecting one-time conversion for any single channel as the operational mode (ADMD[1:0]) converts one AN input one time only. Set the channel to be converted in the conversion channel select bit (ADSC[2:0]). (ADMC[2:0] is ignored.) An A/D interrupt request is generated when the conversion ends.
When starting conversion through using software, set conversion start trigger select bit (ADST[1:0]) to "00" and the conversion start execution flag (ADEN) to "1".
When the conversion start trigger select bit (ADST[1:0]) has been set to an external trigger, the conversion start execution flag (ADEN) is set to "1" when the falling edge is input into ADTRG, and the A/D conversion will be started.
In addition, when the conversion start trigger select bit (ADST[1:0]) has been set to the timer trigger, the conversion start execution flag (ADEN) is set to "1" when a Timer 2 underflow is generated, and the A/D conversion will be started.
The conversion start execution flag (ADEN) is "1" during conversion and "0" after conversion ends.

(2) Continuous conversion for any single channel
Selecting continuous conversion for any single channel as the operational mode (ADMD[1:0]) converts one AN input continuously. Set the channel to be converted in the conversion channel select bit (ADSC[2:0]). (ADMC[2:0] is ignored.) An A/D interrupt request is generated each time conversion ends.
When starting conversion through using software, set conversion start trigger select bit (ADST[1:0]) to "00" and the conversion start execution flag (ADEN) to "1".
When the conversion start trigger select bit (ADST[1:0]) has been set to an external trigger, the conversion start execution flag (ADEN) is set to "1" when the falling edge is input into ADTRG, and the A/D conversion will be started.
In addition, when the conversion start trigger select bit (ADST[1:0]) has been set to the timer trigger, the conversion start execution flag (ADEN) is set to "1" when a Timer 2 underflow is generated, and the A/D conversion will be started.
The conversion start execution flag (ADEN) is "1" during conversion. Since it is not cleared by hardware, be sure to write "0" to the conversion start execution flag (ADEN) when conversion is to be stopped.
When "0" is written to ADEN, data currently under conversion is not guaranteed because the A/D conversion circuit has stopped.

16.4.2. Selecting the Conversion Reference Clock

The A/D conversion rate per channel is 13 × IOCLK per clock select.
For example, if the conversion reference clock has been set to 1/8 of IOCLK, this will be IOCLK × 104 cycles per channel.
Be sure to set so that one cycle is 200 ns or more for the conversion reference clock.
When A/D conversion stop status (ADEN = "0") changes to A/D conversion start (ADEN = "1"), the standby mode for a maximum of one cycle of the conversion reference clock after ADEN = "1"until conversion actually starts.

16.5. Cautions

Select so that one cycle of the conversion reference clock is 200 ns or more.

Chapter 17 Real-time Clock (RTC)

17

CHAPTER 17

Real-time Clock (RTC)

17.1. General

The microcontroller includes a built-in real-time clock (RTC) and a 32.768-KHz oscillation pin for RTC.

17.2. Features

- Clock and calendar function representing the second, minute, hour, day, day of week, month, and year
  The last two digits of the Western year are used for the year display of the clock (00 represents the year 2000).
- Automatic compensation for leap years (Western calendar)
- Capable of using both BCD and binary formats for time and date
- 24-hour/12-hour system selectable
- Daylight savings time correction function
- Three interrupt functions:
  - Periodic interrupt (from 122 µs to 500 ms)
  - Alarm interrupt
  - Update ended interrupt

17.3. Registers

Table 78 Real time clock register

Address	Symbol	Name	Number of bits	Initial value as binary data x: Undefined	Access size
0xD8600000	RTSCR	Seconds count register	8	Undefined	8
0xD8600001	RTSAR	Seconds alarm register	8	Undefined	8
0xD8600002	RTMCR	Minutes count register	8	Undefined	8
0xD8600003	RTMAR	Minutes alarm register	8	Undefined	8
0xD8600004	RTHCR	Hours count register	8	Undefined	8
0xD8600005	RTHAR	Hours alarm register	8	Undefined	8
0xD8600006	RTDWCR	Day of the week count register	8	Undefined	8
0xD8600007	RTDMCR	Days count register	8	Undefined	8
0xD8600008	RTMTCR	Months count register	8	Undefined	8
0xD8600009	RTYCR	Years count register	8	Undefined	8
0xD860000A	RTCRA	Control register A	8	xx10xxxx	8
0xD860000B	RTCRB	Control register B	8	Undefined	8
0xD860000C	RTSRC	Status register C	8	xxxx0000	8

17.3.1. Seconds Count Register

Register symbol RTSCR
Address 0xD8600000
Purpose Allows the seconds counter value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	SCRD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	SCRD[7:0]	Seconds counter value Used to set and read the seconds counter value.

The seconds counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Value (decimal)	Setting range	
	Binary format	BCD format
0 - 59	00 - 3b	00 - 59

17.3.2. Seconds Alarm Register

Register symbol RTSAR
Address 0xD8600001
Purpose Allows the seconds alarm value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	SARD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	SARD[7:0]	Seconds alarm value Used to set and read the seconds alarm value.

The seconds counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Value (decimal)	Setting range	
	Binary format	BCD format
0 - 59	00 - 3b	00 - 59

17.3.3. Minutes Count Register

Register symbol RTMCR
Address 0xD8600002
Purpose Allows the minutes counter value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	MCRD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	MCRD[7:0]	Minutes counter value Used to set and read the minutes counter value.

The minutes counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Value (decimal)	Setting range	
	Binary format	BCD format
0 - 59	00 - 3b	00 - 59

17.3.4. Minutes Alarm Register

Register symbol RTMAR
Address 0xD8600003
Purpose Allows the minutes alarm value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	MARD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	MARD[7:0]	Minutes alarm value Used to set and read the minutes alarm value.

The minutes counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Value (decimal)	Setting range	
	Binary format	BCD format
0 - 59	00 - 3b	00 - 59

17.3.5. Hours Count Register

Register symbol RTHCR
Address 0xD8600004
Purpose Allows the hours counter value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	HCRD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	HCRD[7:0]	Hours counter value Used to set and read the hours counter value.

The hours counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Time format	Value (decimal)	Setting range			
		Binary format		BCD format	
12	1 - 12	AM	01 - 0C	AM	01 - 12
		PM	81 - 8C	PM	81 - 92
24	0 - 23	00 - 17		00 - 23	

17.3.6. Hours Alarm Register

Register symbol RTHAR
Address 0xD8600005
Purpose Allows the hours alarm value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	HARD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	HARD[7:0]	Hours alarm value Used to set and read the hours alarm value.

The hours counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Time format	Value (decimal)	Setting range			
		Binary format		BCD format	
12	1 - 12	AM	01 - 0C	AM	01 - 12
		PM	81 - 8C	PM	81 - 92
24	0 - 23	00 - 17		00 - 23	

17.3.7. Day of the Week Count Register

Register symbol RTDWCR
Address 0xD8600006
Purpose Allows the day of the week counter value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	DWCRD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	DWCRD[7:0]	Day of the week counter value Used to set and read the day of the week counter value.

The day of the week counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Value (decimal)	Setting range	
	Binary format	BCD format
1 - 7	01 - 07	01 - 07

17.3.8. Days Count Register

Register symbol	RTDMCR
Address	0xD8600007
Purpose	Allows the days counter value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	DMCRD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	DMCRD[7:0]	Days counter value Used to set and read the days counter value.

The days counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Value (decimal)	Setting range	
	Binary format	BCD format
1 - 31	01 - 0F	01 - 31

17.3.9. Months Count Register

Register symbol	RTMTCR
Address	0xD8600008
Purpose	Allows the months counter value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	MTCRD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	MTCRD[7:0]	Months counter value Used to set and read the months counter value.

The months counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Value (decimal)	Setting range	
	Binary format	BCD format
1 - 12	01 - 0C	01 - 12

17.3.10. Years Count Register

Register symbol RTYCR
Address 0xD8600009
Purpose Allows the years counter value to be set and read.

Bit	7	6	5	4	3	2	1	0
Bit name	YCRD[7:0]							
Initial value	Undefined							
R/W	RW							

Bit	Bit name	Description
7-0	YCRD[7:0]	Years counter value Used to set and read the years counter value. "00" is recognized as the Year 2000.

The years counter value can be set in the following range. Operations are not guaranteed if an out-of-bounds value is written here.

Value (decimal)	Setting range	
	Binary format	BCD format
0 - 99	00 - 63	01 - 99

17.3.11. RTC Control Register A

Register symbol RTCRA
Address 0xD860000A
Purpose Used to read the updated status of the clock, initialize the frequency divider, and set the periodic interrupt rate.

Bit	7	6	5	4	3	2	1	0
Bit name	UIP	DVR	reserved	reserved	RS[3:0]			
Initial value	Undefined	Undefined	1	0	Undefined			
R/W	RW	RW	R	R	RW			

Bit	Bit name	Description
7	UIP	Clock update flag Indicates the clock is being updated. 0 : Clock not being updated 1 : Clock currently being updated This bit is set to "1" from 244 µs before clock update until 244 µs after.
6	DVR	Frequency divider reset The frequency divider is reset by writing "1" here. Reset can be canceled by writing "1" and then "0".
5	reserved	These are reserved bits. "1" is always returned when these bits are read. Always write a "1" to these bits.
4	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.

Bit	Bit name	Description
3-0	RS[3:0]	Sets the periodic interrupt cycle The periodic interrupt cycle is set as given in the table below.

RS[3:0]	Periodic interrupt cycle
0000	Do not generate interrupt
0001	3.90625 ms
0010	7.8125 ms
0011	122.070μs
0100	244.141μs
0101	488.281μs
0110	976.5625μs
0111	1.953125 ms
1000	3.90624 ms
1001	7.8125 ms
1010	15.625 ms
1011	31.25 ms
1100	62.5 ms
1101	125 ms
1110	250 ms
1111	500 ms

17.3.12. RTC Control Register B

Register symbol RTCRB
Address 0xD860000B
Purpose Sets the operational control for the real-time clock.

Bit	7	6	5	4	3	2	1	0
Bit name	SET	PIE	AIE	UIE	reserved	DM	TM	DSE
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit name	Description
7	SET	Clock Update Disable Enables updating of the clock. 1 : Enable clock update 0 : Disable clock update
6	PIE	Periodic Interrupt Enable 0 : Periodic interrupt enabled 1 : Periodic interrupt disabled
5	AIE	Alarm Interrupt Enable 0 : Alarm interrupt enabled 1 : Alarm interrupt disabled
4	UIE	Update Interrupt Enable 0 : Update interrupt enabled 1 : Update interrupt disabled
3	reserved	These are reserved bits.
2	DM	Sets the format of numeric values.

Bit	Bit name	Description
		Sets the numeric value format to use when reading or writing values from the clock counter and alarm counter. 0 : BCD format 1 : Binary format This bit is not affected by reset. Therefore, be sure to always set this bit before writing a value to a clock counter or alarm counter.
1	TM	Time Format Setting Sets the time format to use when reading or writing values from a clock counter or alarm counter. 0 : 12-hour system 1 : 24-hour system This bit is not affected by reset. Therefore, be sure to always set this bit before writing a value to a clock counter or alarm counter.
0	DSE	Daylight Savings Setting Enables daylight savings time. 0 : Disable daylight savings time 1 : Enable daylight savings time The time is advanced as given in the table below when daylight savings time is enabled. It is necessary to reset the value of the clock counter when the value of this bit is changed. It is impossible to correctly judge the end of daylight savings time if the clock is set to a time falling between 1:00:00 AM and 2:00:00 AM on the last Sunday of October. Avoid settings the clock during this time. This bit is not affected by reset. Therefore, be sure to always set this bit before writing a value to a clock counter or alarm counter.

Day and hour	Clock time before update	Clock time after update	Remarks
Last Sunday of April	1:59:59 AM	3:00:00 AM	Start of daylight savings time
Last Sunday of October	1:59:59 AM	1:00:00 AM	End of daylight savings time

17.3.13. RTC Status Register C

Register symbol RTSRC
Address 0xD860000C
Purpose Used to read the operational status of the real-time clock.

Bit	7	6	5	4	3	2	1	0
Bit name	IRQF	PF	AF	UF	reserved			
Initial value	Undefined	Undefined	Undefined	Undefined	0			
R/W	R	R	R	R	R			

Bit	Bit name	Description
7	IRQF	Interrupt Flag Indicates if an interrupt is being generated. 0 : No interrupt 1 : Interrupt being generated If any of the flags PF, AF or UF are set to "1" and the corresponding enable bit of the RTCRB register (PIE, AIE, UIE) is "1", the IRQF bit will be "1". This bit can be cleared to "0" by reading the RTSRC register.
6	PF	Periodic Interrupt Flag Indicates whether or not there is a periodic interrupt request. 0 : No interrupt request 1 : Interrupt request present An interrupt is generated if the PIE bit of the RTCRB register is "1". This bit can be cleared to "0" by reading the RTSRC register.
5	AF	Alarm Interrupt Flag Indicates whether or not there is an alarm interrupt request. 0 : No interrupt request 1 : Interrupt request present An interrupt is generated if the AIE bit of the RTCRB register is "1". This bit can be cleared to "0" by reading the RTSRC register.
4	UF	Update End Interrupt Flag Indicates whether or not there is an update end interrupt request. 0 : No interrupt request 1 : Interrupt request present An interrupt is generated if the UIE bit of the RTCRB register is "1". This bit can be cleared to "0" by reading the RTSRC register.
3-0	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.

17.4. Description of Operation

17.4.1. Initial Settings

Make the initial settings described below after RTC power is turned on.

(1) Initialize by writing "0" to the RTCRB register. RTCRB register values are undefined after RTC power is turned on.
(2) Read the RTSRC register. RTSRC register values are undefined after RTC power is turned on. Each interrupt flag is cleared when the RTSRC register is read.
(3) Write "1" to the SET bit of the RTCRB register to stop the clock counter.
(4) Set the RTCRB register. Write "1" for the SET bit at this time.
(5) Write "1" to the DVR bit of the RTCRA register to reset the frequency divider.
(6) Set the clock.
(7) Write "0" to the DVR bit of the RTCRA register to restart the frequency divider.
(8) Write "0" to the SET bit of the RTCRB register to start the clock counter.

Note the following points when writing values to a clock counter register or alarm set register.

- Always set the DM, TM and DSE bits of the RTCRB register before writing a value. (The initial value of these bits are undefined because they are not affected by reset.)
- Set a value from the range defined for each value to the corresponding register. Operations are not guaranteed if an out-of-bounds value is specified.
- When setting daylight savings time, avoid setting the clock between 1:00 AM to 2:00 AM on the last Sunday of October as it is impossible to correctly determine the end of daylight savings time when this hour has been set.
- Reset the value of the clock counter when the value of the DSE bit has been changed between 1:00 AM on the last Sunday of April and 3:00 AM on the last Sunday of October.

17.4.2. Updating the Time

The time is updated once per second when the SET bit of the RTCRB register is set to "0". The clock is not updated when the SET bit is set to "1". The UIP bit of the RTCRA register is set to "1" from 244 µs before updating until 244 µs after updating.

17.4.3. Alarm Function

After the clock is updated, it is compared to the alarm setting. If it matches the alarm setting, "1" is written to the AF bit of the RTSRC register. If the AF bit is "1" and the AIE bit of the RTCRB register is "1", an interrupt is generated and the IRQF bit of the RTSRC register is set. The AF and IRQF bits are cleared to "0" when the RTSRC register is read.
The alarm is set using the seconds, minutes and hours alarm set registers (RTSAR, RTMAR, RTHAR). If both bits 7 and 6 of the hours alarm set register (RTHAR) are set to "1", RTHAR is not compared to the clock and the alarm goes off every hour at the specified minute and second.

17.4.4. Re-setting the Clock

Temporarily stop the clock and set it according to the procedure shown for making initial settings when re-setting it.
Re-setting the clock while it is running will cause errors in set data in the case that the clock is updated while registers are being written. The UIP bit of the RTCRA register is set to "1" from 244 µs before updating until 244 µs after updating. The clock will therefore not be updated for at least

244 µs when the UIP bit is "0". Use the following procedure to check and reset the UIP bit.

(1) Check the value of the UIP bit of the RTCRA register. The clock is being updated if it is "1". Keep re-reading the UIP bit until it goes to "0".
(2) Check that the UIP bit is "0" and reset the clock. Be sure to finish writing data within 244 µs.
(3) After re-set, check again that the UIP bit is "0". Note that data may not have been written correctly if the UIP bit is "1". Return to step (1) if this is the case.

17.4.5. Reading the Clock

It is impossible to read the correct value when reading the clock counter value while the clock is being updated. It is therefore necessary to read the correct value with avoiding the period when the clock is being updated by using any of the following methods.

(1) Using the UIP bit value
The clock is updated at 244 µs later after the UIP bit of the RTCRA register goes to "1". Check that the UIP bit is "0" before and after reading data and read the correct value by using the same procedure as for re-setting the clock,.
(2) Using the update ended interrupt
An update ended interrupt is generated after the clock is updated. After the update ended interrupt has been generated, the value of the clock counter will be fixed without being updated for about 999 ms.
(3) Using the periodic interrupt
A periodic interrupt longer than the clock update cycle (488 µs) cannot be generated while the clock is being updated. The value of the clock counter is therefore fixed without being updated for half the cycle of the periodic interrupt that has been set.

17.4.6. Interrupts

The following three types of interrupts are generated. It is possible to enable interrupts to be generated based on each of the PIE, AIE and UIE bits of the RTCRB register.
Read the value of RTSRC and check the cause of an interrupt after one has been generated. All bits of RTSRC are cleared when RTSRC is read.

- Periodic interrupts
  13 settings are available from every 500 ms to every 122 µs.
- Alarm interrupt
  4 settings are available from once per day to once per second.
- Update ended interrupt
  Generated once per second.

Chapter 18 IrDA Controller (IRC)

18

CHAPTER 18

IrDA Controller (IRC)

18.1. General

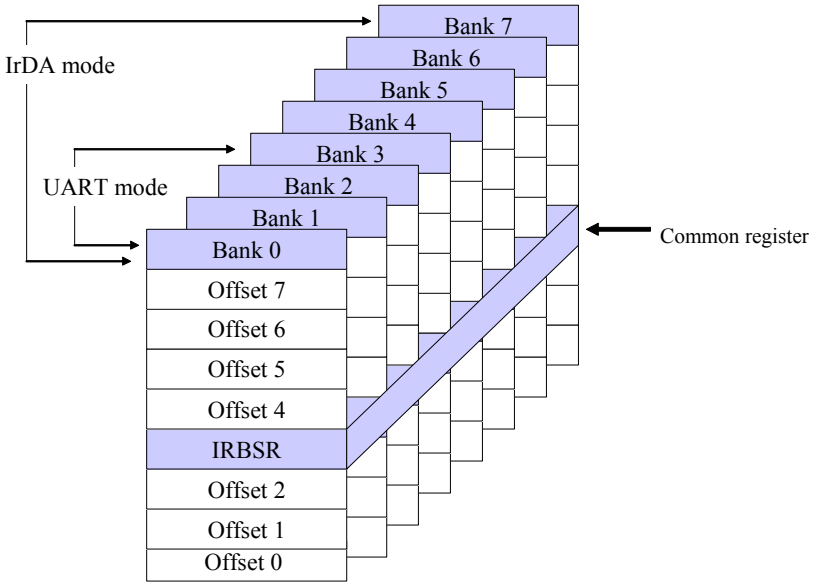
IrDA controller (Infrared Data Association Controller) is an infrared transmission controller.

18.2. Features

- IrDA 1.0SIR (- 115.2Kbps, Half-Duplex)
- IrDA 1.1 MIR (0.579Kbps, 115.2Mbps, Half-Duplex)
- IrDA 1.1 FIR (4.0Mbps, Half-Duplex)
- UART (- 1.5Mbps, Full-Duplex)
- 48MHz clock input (built-in baud rate generator function)

18.3. Register

The registers of the IrDA controller consists of eight banks. Each bank is composed of 8 bytes, and these banks are mapped at the same address. The third byte of each bank is mapped at the bank select register, and only any one bank can be accessed by setting this registers.



The registers from banks 0 to 3 are used for both UART and IrDA modes, and the registers from 4 to 7 are used for IrDA mode. The list of each bank is shown below.

Table 79 Bank list

Bank	USRT mode	IR mode	Overview
0	○	○	Control status register
1	○	○	16550 compatible register
2	○	○	Baud generator divisor
3	○	○	ID/shadow register
4		○	Timer counter
5		○	IR control, Status FIFO
6		○	IR physical layer configuration
7		○	CEIR/optical communication configuration

CHAPTER 18

IrDA Controller (IRC)

Table 80 IrDA controller register

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080	-	Offset 0	8	Note	8
0xD8700081	-	Offset 1	8	Note	8
0xD8700082	-	Offset 2	8	Note	8
0xD8700083	-	Offset 3	8	Note	8
0xD8700084	-	Offset 4	8	Note	8
0xD8700085	-	Offset 5	8	Note	8
0xD8700086	-	Offset 6	8	Note	8
0xD8700087	-	Offset 7	8	Note	8

Note: The register differs depending on each bank. Refer to the register list.

Table 81 Register configuration

	Bank 0	Bank 1	Bank 2	Bank 3	Bank 4	Bank 5	Bank 6	Bank 7
Offset 0	IRTDR IRRDR	IRDLLR	IRDLLR	-	IRTMRL	-	IRCR3	-
Offset 1	IRIER IREIER	IRDLUR	IRDLUR	IRSHLCR	IRTMRH	-	IRMIRPW	-
Offset 2	IRIIR IREIIR IRFCR	IRIIR IREIIR IRFCR	IREXCR1	IRSHFCR	IRCR1	IRPMDR	IRSIRPW	-
Offset 3	IRLCR IRBSR	IRLCR IRBSR	IRLCR IRBSR	IRLCR IRBSR	IRLCR IRBSR	IRLCR IRBSR	IRLCR IRBSR	IRLCR IRBSR
Offset 4	IRMCR IRMDR	IRMCR IRMDR	IREXCR2	-	IRTFLL	IRCR2	IRBFPL	IRCFG1
Offset 5	IRLSR IRELSR	IRLSR IRELSR	-	-	IRTFLU	IRFRST	-	IRCFG2
Offset 6	IRMSR	IRMSR	IRTFLL	-	IRRMLL	IRFRLL	-	-
Offset 7	IRSCR IRASCR	IRSCR IRASCR	IRRFLV	-	IRRMLU	IRFRLU	IRFIRPW	IRCFG4

Table 82 Bank 0 Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080	IRTDR IRRDR	IrDA transmit data register IrDA receive data register	8	0x00	8
0xD8700081	IRIER IREIER	IrDA interrupt enable register IrDA extended interrupt enable register	8	0x00 0x00	8
0xD8700082	IRIIR IREIIR IRFCR	IrDA interrupt identification register IrDA extended interrupt identification register IrDA FIFO control register	8	0x01 0x01 0x00	8
0xD8700083	IRLCR IRBSR	IrDA link control register IrDA bank select register	8	0x00 0x00	8

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700084	IRMCR IRMDR	IrDA modem control register IrDA mode control register	8	0x00 0x00	8
0xD8700085	IRLSR IRELSR	IrDA link status register IrDA extended link status register	8	0x60 0x60	8
0xD8700086	IRMSR	IrDA modem status register	8	Note 1	8
0xD8700087	IRSCR IRASCR	IrDA scratch IrDA extended status/control register	8	0xFF 0x00	8

Note 1: For the initial values, refer to page 396, 18.3.1.14 IrDA modem status register.

Table 83 Bank 1 Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080	IRDLLR	IrDA divisor latch lower register	8	0xFF	8
0xD8700081	IRDLUR	IrDA divisor latch upper register	8	0xFF	8
0xD8700082	-	Shared with bank 0 offset 2	8	0x01	8
0xD8700083	-	Shared with bank 0 offset 3	8	Note 2	8
0xD8700084	-	Shared with bank 0 offset 4	8	0x00	8
0xD8700085	-	Shared with bank 0 offset 5	8	0x60	8
0xD8700086	-	Shared with bank 0 offset 6	8	0x00	8
0xD8700087	-	Shared with bank 0 offset 7	8	0xFF	8

Note 2: For the initial values, refer to page 391, 18.3.1.9 IrDA bank control register.

Table 84 Bank 2 Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080	-	Shared with Bank 1 Offset 0	8	0xFF	8
0xD8700081	-	Shared with Bank 1 Offset 1	8	0xFF	8
0xD8700082	IREXCR1	IrDA extended control register 1	8	0x00	8
0xD8700083	-	Shared with Bank 0 Offset 3	8	0xE0	8
0xD8700084	IREXCR2	IrDA extended control register 2	8	0x00	8
0xD8700085	-	reserved	8	0x00	8
0xD8700086	IRTFLV	IrDA transmit FIFO data level register	8	0x00	8
0xD8700087	IRRFLV	IrDA receive FIFO data level register	8	0x00	8

Table 85 Bank 3 Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080		reserved	8	0x00	8
0xD8700081	IRSHLCR	IrDA link control shadow register	8	Note 3	8
0xD8700082	IRSHFCR	IrDA FIFO control shadow register	8	0x00	8
0xD8700083		Shared with Bank 0 Offset 3	8	0xE4	8

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700084		reserved	8	0x00	8
0xD8700085		reserved	8	0x00	8
0xD8700086		reserved	8	0x00	8
0xD8700087		reserved	8	0x00	8

Note 3: For the initial values, refer to page 402, 18.3.4.1 IrDA link control shadow register.

Table 86 Bank 4 registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080	IRTMRL	IrDA timer initial register lower register	8	0x00	8
0xD8700081	IRTMRH	IrDA timer initial register upper register	8	0x00	8
0xD8700082	IRCR1	IrDA infrared control register 1	8	0x00	8
0xD8700083		Shared with Bank 0 offset 3	8	0xE8	8
0xD8700084	IRTFLL	IrDA transmitter frame-length lower count register	8	0x00	8
0xD8700085	IRTFLU	IrDA transmitter frame-length upper count register	8	0x08	8
0xD8700086	IRRMLL	IrDA receiver frame maximum-length lower count register	8	0x00	8
0xD8700087	IRRMLU	IrDA receiver frame maximum-length upper count register	8	0x08	8

Table 87 Bank 5 registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080		reserved	8	0x00	8
0xD8700081		reserved	8	0x00	8
0xD8700082	IRPMDR	IrDA pipelined mode register	8	0x00	8
0xD8700083		Shared with Bank 0 offset 3	8	0xEC	8
0xD8700084	IRCR2	IrDA Infrared control register 2	8	0x00	8
0xD8700085	IRFRST	IrDA status FIFO frame-status register	8	0x00	8
0xD8700086	IRFRLL	IrDA status FIFO frame-length lower count register	8	0x00	8
0xD8700087	IRFRLU	IrDA status FIFO frame-length upper count register	8	0x00	8

Table 88 Bank 6 registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8700080	IRCR3	IrDA infrared control register 3	8	0x00	8
0xD8700081	IRMIRPW	IrDA MIR pulse-width register	8	0x0A	8
0xD8700082	IRSIRPW	IrDA SIR pulse-width register	8	0x00	8
0xD8700083		Shared with Bank 0 offset 3	8	0xF0	8
0xD8700084	IRBFPL	IrDA start flags/preamble length	8	0x2A	8

Address	Symbol	Name	Number of bits	Initial value	Access size
		register			
0xD8700085		reserved	8	0x00	8
0xD8700086		reserved	8	0x00	8
0xD8700087	IRFIRPW	IrDA FIR pulse setting register	8	0x01	8

Table 89 Bank 7 registers

Address	Symbol	Name	Number of bits	Initial value	Access
0xD8700080		reserved	8	0x00	8
0xD8700081		reserved	8	0x00	8
0xD8700082		reserved	8	0x00	8
0xD8700083		Shared bank 0 offset 3	8	0xF4	8
0xD8700084	IRCFG1	IrDA infrared interface control register 1	8	0x00	8
0xD8700085	IRCFG2	IrDA infrared interface control register 2	8	0x00	8
0xD8700086		reserved	8	0x00	8
0xD8700087	IRCFG4	IrDA extended control register 4	8	0x00	8

18.3.1. Bank 0

18.3.1.1. IrDA transmitter data register

Symbol IRTDR
Address 0xD8700080
Purpose This writes transmit data

Bit	7	6	5	4	3	2	1	0
Bit name	TD [7:0]							
Initial value	0							
R/W	W							

Bit	Bit name	Description
7-0	TD[7:0]	Transmit data Data is written from the transmitter FIFO in the mode using FIFO and from the transmitter register in the mode not using FIFO.

18.3.1.2. IrDA receiver data register

Symbol IRRDR
Address 0xD8700080
Purpose This can read out the receive data.

Bit	7	6	5	4	3	2	1	0

Bit name	RD[7:0]
Initial value	0
R/W	R

Bit	Bit name	Description
7-0	RD[7:0]	Receive data Data is read out from the receive FIFO in the mode using FIFO and from the receive register in the mode not using FIFO.

<Programming note>
IrDA receive data register is valid when FEN (Bit 0) IrDA FIFO control register is “1”.

18.3.1.3. IrDA interrupt enable register

Symbol IRIER
Address 0xD8700081
Purpose This permits the occurrence of interrupts. Each interrupt can enable a corresponding interrupt depending on its setting. Moreover, this can prohibit all interrupts.
This is usable at the UART mode/SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved				MDS	RLS	TDE	RDA
Initial value	0				0	0	0	0
R/W	R				R/W	R/W	R/W	R/W

Bit	Bit name	Description
7-4	reserved	These bits are reserved. A “0” is always returned when these bits are read. When writing this register, always write a “0” to these bits.
3	MDS	Modem status interrupt This permits modem status interrupts. 0: disable 1: enable
2	RLS	Receive link status interrupt This permits receive link status interrupt. 0: disable 1: enable
1	TDE	Transmit data register empty interrupt This permits transmit data register empty interrupt. 0 : disable 1 : enable
0	RDA	Receive data available interrupt This permits receive data available interrupt and timeout interrupt. 0 : disable 1 : enable

<Programming note>
When this register switches the interrupt to the enable side during operation, the interrupt cause which occurred in the past occurs an interrupt if it is valid at the time.

<Programming note>
When writing to this register, the same setting value is written to IrDA extended interrupt enable register. However, in the case of writing to IrDA extended interrupt enable register, this register is unchanged.

18.3.1.4. IrDA extended interrupt enable register

Symbol	IREIER
Address	0xD8700081
Purpose	This permits interrupt occurrence. Each interrupt can enable corresponding interrupt depending on the setting. This can prohibit all interrupts. This can be used at the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	TMRIE	SFIE	TEIE	reserved	MDS	LSIE	TLIE	RLIE
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit name	Description
7	TMRIE	Timer interrupt When the timer count value is “0”, this generates an interrupt. 0: disable 1: enable
6	SFIF	Status FIFO interrupt This permits status FIFO interrupt. 0 : disable 1 : enable
5	TEIE	Transmit empty/pipeline load interrupt This permits transmit empty/pipeline load interrupt. 0: disable 1: enable
4	reserved	This bit is reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
3	MDS	Modem status interrupt This permits the modem status interrupt. 0 : disable 1 : enable
2	LSIE	Receive link status interrupt This permits receive link status interrupt. 0: disable 1: enable
1	TLIE	Transmit low level interrupt This permits transmit low level interrupt. 0 : disable 1 : enable
0	RLIE	Receive high level interrupt This permits receive high level interrupt. 0: disable 1: enable

<Programming note>
IrDA extended interrupt enable register is valid, when EXTSL (Bit 0) of IrDA extended control register is “1”.

18.3.1.5. IrDA interrupt identification register

Symbol IRIIR
Address 0xD8700082
Purpose This records the occurred interrupts.
The interrupts are divided into four priority level and recorded in the UART/SIR mode.
This can be used at the UART/SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	FIE[1:0]		reserved		TMI	ILV[1:0]		IPD
Initial value	0		0		0	0		1
R/W	R		R		R	R		R

Bit	Bit name	Description
7-6	FIE[1:0]	FIFO enable 0 : FIFO not using mode 1 : FIFO using mode
5-4	reserved	These are reserved. A “0” is always returned when the bit is read.
3	TMI	Timeout interrupt flag 0: No timeout interrupt 1: Timeout interrupt
2-1	ILV[1:0]	Interrupt level flag There are priorities in the following order for the interrupts, and the highest-priority interrupt is shown at reading. 00: Modem status interrupt 01: IrDA transmit data register empty interrupt 10: Receive data available interrupt, Timeout interrupt 11: Receive line status interrupt
0	IPD	Interrupt pending flag 0: Interrupt 1: No interrupt

18.3.1.6. IrDA extended interrupt identification register

Symbol IREIIR
Address 0xD8700082
Purpose This records the events which occurred.
When a corresponding interrupt is approved, all other bits are cleared.
There is no priority order.
This is usable at the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	TMREV	SFEV	TEPL	reserved		LSEV	TLEV	RLEV
Initial value	0	0	0	0		0	0	1
R/W	R	R	R	R		R	R	R

Bit	Bit name	Description
7	TMREV	TMR event This is set to “1” when TMR timer count value is “0”. When “1” is written to PLCT (bit 7) of IrDA extended status/control register, this is set to “0”.
6	SFEV	Status FIFO event When the number of the status FIFO data is more than the threshold or when timeout occurs, this is set “1”. If writing is executed to transmit FIFO, this is negated.
5	TEPL	Transmit empty/pipeline load event This is set to “1” when data in the transmit section is empty. When the pipeline function is enabling and the pipeline writes to transmit FIFO on loading, this is negated.
4-3	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
2	LSEV	Link status event This is set to “1” under the following conditions. 1. EOF reaches the bottom of the receive FIFO. Negated by reading out LSR. 2.Overrun at receiving Negated by reading out LSR. 3.Underrun at transmitting Negated by only soft reset. 4.Transmit halt at finishing frame Negated when “1” is written to THFE (bit 3) of IrDA extended status/control register and restart is executed,
1	TLEV	Transmit low data level event This is set to “1” when there are spaces for the trigger level in transmit FIFO. Negated if over the level.
0	RLEV	Receive high-level event This is set to “1” under the following conditions. When transmit FIFO data reaches the trigger level. (negated if under the level) When timeout occurs in receive FIFO (Negated by reading out transmit FIFO)

<Programming note>
IrDA extended interrupt identification register is valid when EXRSL (bit 0) of IrDA extended control register 1 is “1”.

18.3.1.7. IrDA FIFO control register

Symbol IRFCR
Address 0xD8700082
Purpose This sets FIFO mode switch, transmit and receive FIFO reset, and the trigger level of the interrupt occurrence in the receive FIFO (number of the FIFO levels).

Bit	7	6	5	4	3	2	1	0
Bit name	RTL[1:0]		TTL[1:0]		reserved	TFR	RFR	FEN
Initial value	0		0		0	0	0	0
R/W	W		W		W	W	W	W

Bit	Bit name	Description
7-6	RTL[1:0]	Receive FIFO trigger level This sets the trigger level by the number of data stored in 16-level receive FIFO in the FIFO using mode. 00:1/1 01:4/8 10:8/16 11:14/30
5-4	TTL[1:0]	Transmit FIFO trigger level This sets the trigger level by the number of empty levels in 16-level transmit FIFO in the FIFO using mode. 00:1/1 01:4/8 10:8/16 11:16/32
3	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
2	TFR	Transmit FIFO reset When “1” is set for this bit, transmit FIFO pointer is reset. When “1” written for this bit, it is automatically cleared to “0” after FIFO resetting.
1	RFR	Receive FIFO reset When “1” is set for this bit, Receive FIFO pointer is reset. When “1” written for this bit, it is automatically cleared to “0” after FIFO resetting.
0	FEN	FIFO enable This bit setting sets the FIFO using mode. 0: FIFO not using mode 1: FIFO using mode

<Programming note>
Bit 0 is valid in only the UART mode/SIR mode. It cannot be used in the MIR/FIR mode.

18.3.1.8. IrDA link control register

Symbol	IRLCR
Address	0xD8700083
Purpose	This sets the specifications of the transmit/receive data format and the divisor latch access bit (DLAB). For the access method, refer to Table 90 Readable setting and Table 91 Setting at writable time. This is usable at the UART mode/SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	DLAB	BRC	STP	EPS	PE	STL	CHL[1:0]	
Initial value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit name	Description
7	DLAB	Divisor latch access bit This is an address bit to access to IrDA divisor latch lower register/IrDA divisor latch upper register.
6	BRC	Brake control This controls the space state (fixed at “0”) of SOUT pin in the UART mode. In the cases except the UART mode, there is no effect. TE (bit 6) of IrDA link status register becomes “1” and a break occurs, and then transmit data are written and this bit is set to “0”. Afterward, it becomes “1” again and can transmit a break of the time for 1 character.
5	STP	Fixed parity bit This is used for setting the parity of a fixed value. The parity is set to be enabled (PE=1), and when this bit is “1”, the fixed value is determined by EPS (bit 4). 0: When EPS=0, set the parity depending on EPS (bit 4). 1: When EPS=0, fix at “0”, when EPS=1, fix at “1”.
4	EPS	Parity selection bit 0:Odd parity 1:Even parity
3	PE	Parity enable 0:Parity disable 1:Parity enable
2	STL	Stop-bit length This sets the stop-bit length at transmit. At receive, this checks only 1 bit as stop bit regardless of its setting. 0:1 bit 1:1.5 bit in the case of CHL[1:0] = 00, 2 bits in other cases.
1-0	CHL[1:0]	Character-bit length This sets transmit/receive character-bit length 00:5 bit 01:6 bit 10:7 bit 11:8 bit

18.3.1.9. IrDA bank control register

Symbol IRBSR
Address 0xD8700083
Purpose This switches a bank register.

Bit	7	6	5	4	3	2	1	0
Bit name	BCR	BSR [6:0]						
Initial value	0	0						
R/W	R/W	R/W						

Bit	Bit name	Description
7	BCR	Bank selection control bit 0:can set bank 0. 1:Bank selection can be executed by the setting values of BSR[6-0](bit 0-bit6).
6-0	BSR[6:0]	Bank selection bit When BCR (bit 7) is “1”, bank selection can be carried out by the setting indicated below.

Setting bit								Setting bank
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
0	X	X	X	X	X	X	X	0
1	0	X	X	X	X	X	X	1
1	1	X	X	X	X	1	X	1
1	1	X	X	X	X	X	1	1
1	1	1	0	0	0	0	0	2
1	1	1	0	0	1	0	0	3
1	1	1	0	1	0	0	0	4
1	1	1	0	1	1	0	0	5
1	1	1	1	0	0	0	0	6
1	1	1	1	0	1	0	0	7

X: Both “0” and “1” can be bit setting values.

Reading as follows is executed and the writing register is set through this register’s switching the BANK

Table 90 Readable setting

Setting bank	Reading
0	IRBSR
1	IRBSR
2	IRBSR
3	IRLCR
4	IRBSR
5	IRBSR
6	IRBSR
7	IRBSR

Table 91 Setting at writable time

Bit 7	Writing
0	IRBSR/IRLCR
1	IRBSR

18.3.1.10. IrDA modem control register

Symbol	IRMCR
Address	0xD8700084
Purpose	This sets the values of the modem control output signals, and sets the loop back test mode and auto follow enable. This LSI cannot set the operations corresponding to the purposes because this register is not implemented in the LSI. This is usable in the UART mode/ SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved							
Initial value	0							
R/W	R/W							

Bit	Bit name	Description
7-0	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.

18.3.1.11. IrDA mode control register

Symbol	IRMDR
Address	0xD8700084
Purpose	This can carry out mode selection and change the mode. This also sets the transmission of the interaction pulse and transmission delay function. This register is valid only in the extended mode. For the extended mode setting, refer to 400, 18.3.3.1 IrDA extended control register 1.

Bit	7	6	5	4	3	2	1	0
Bit name	MDSL			SIP	TXDF	reserved		
Initial value	0			0	0	0		
R/W	R/W			R/W	R/W	R/W		

Bit	Bit name	Description
7-5	MDSL	Mode selection 000:UART Mode 001:reserved 010:reserved 011:SIR Mode 100:MIR Mode 101:FIR Mode 110:reserved 111:reserved
4	SIP	Interaction pulse (valid only for MIR mode /FIR mode) This transmits an infrared interaction pulse.
3	TXDF	Transmit delay function (valid only for MIR mode/FIR mode) This validates a transmit delay function when using transmit FIFO. *<Programming note>* *The number of the set FIFO levels must execute writing data of the 2*

Bit	Bit name	Description
		and more levels.
2-0	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.

<Programming note>
Bits 7-2 must be initialized after changing from the extended mode to unextended mode.

<Programming note>
All values are cleared by reset.

<Programming note>
Bits 4-3 are valid only in the MIR mode/FIR mode. They are not used in the UART/SIR mode.

18.3.1.12. IrDA link status register

Symbol IRLSR
Address 0xD8700085
Purpose This shows the information about the receive data and transmit register.
Bits 4-1 shows the interrupt status of the link status.
This is usable in the UART mode/SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	FE	TE	TDRE	BRI	FME	PTE	OVE	RDR
Initial value	0	1	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Bit name	Description
7	FE	FIFO error 0:FIFO without error 1:FIFO with error When any of the following errors is detected during receiving the FIFO in the FIFO using mode, this is set to “1”. ・Parity error ・Framing error ・Break interrupt When the data including errors is only at the bottom of the FIFO and read is carried out, this is set to “0”. This is always “0” at reset and in FIFO not using mode.
6	TE	Transmit empty 0: Transmit data 1: No transmit data When the data are empty in all blocks of the FIFO, register, P/S, this is sets to “1”. This is “1” at reset and “0” at writing IrDA transmit data register.
5	TDRE	Transmit data register empty 0: Transmit data 1: Not transmit data The transmit-side FIFO or register is empty, this is set to “1”. This is “1” at reset and “0” at writing the IrDA transmit data register.

Bit	Bit name	Description
4	BRI	Break event detection 0:No break interrupt 1:Break interrupt This is set to “1” when the data with the break interrupt bit of “1" reach the bottom of the receive FIFO. (Immediately after the error detection when not using FIFO) This is “0” at reset and reading LSR.
3	FME	Framing error 0: No framing error 1: Framing error This is set to “1” when the framing error occurs. This is set to “1” when the data with the framing error bit of “1" reach the bottom of the receive FIFO. (Immediately after the error detection when not using FIFO) This is “0” at reset or reading LSR.
2	PTE	Parity error 0: No parity error 1: Parity error This is set to “1” when a parity error occurs. This is set to “1” when the data with the parity error bit of “1" reach the bottom of the receive FIFO. (Immediately after the error detection when not using FIFO) This is “0” at reset and reading LSR.
1	OVE	Overrun error 0: No overrun error 1: Overrun error This is set to “1” when an overrun error occurs. This is set to “1” when the overrun error bit reaches the bottom of the receive FIFO. (Immediately after the error detection when not using FIFO) This is “0” at reset and reading this register.
0	RDR	Receive data decision This shows that the receive data can read from the host. This is “0” at reset and reading this register. 0: No receive data 1: Receive data

<Programming note>
The occurrence of errors may not be detected if errors occur during reading because of the higher priority of negation by reading.

18.3.1.13. IrDA extended link status register

Symbol	IRELSR
Address	0xD8700085
Purpose	This shows the information about the receive data and transmit register. The bits 4 to 1 and bit 7 indicate the link status event. This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	FE	TE	TFE	ML	PLE	CRE	OVE	RDR
Initial value	0	1	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Bit name	Description
7	FE	Frame end This is set to “1” when it reaches the bottom of the frame final data FIFO after receiving. This is “0” at reset.
6	TE	Transmit data empty This is set to “1” when the data in the all blocks of FIFO, register and transmit processes are empty. This is set to”0” through writing the IrDA transmit data register.
5	TFE	Transmit FIFO empty This is set when the transmit FIFO is empty. This is “0” when writes are executed to IrDA transmit data register
4	ML	Maximum receive frame length In the case of receiving the longer frame than the maximum length of the receive frame set by IrDA receive frame maximum-length lower count register/IrDA receive frame maximum-length upper count register, this is set to “1” when the frame final data reach the bottom of the receive FIFO.
3	PLE	Physical layer error [MIR mode] In the case that abortion is detected in the received bit stream, this is set to “1” when the frame final data reach the bottom of the FIFO. [FIR mode] In the case that an encoding error is detected in the received bit stream, this is set to “1” when the frame final data reach the bottom of the FIFO. (1) A stream which cannot be the PPM modulation is detected in the data area. (2) Although the STO flag has not been detected, the next STA flag is detected. This is “0” at reset or by reading this register.
2	CRE	CRC error At receiving, 1 is set when the result of CRC is an error and the final data of the receive frame reaches the bottom of FIFO.
1	OVE	Overrun error If the data are to be loaded to the receive FIFO and status FIFO at receiving and the data are abandoned because the receive FIFO and status FIFO are full, this is set to “1” when the EOF reaches the bottom of the receive FIFO. This is “0” at reset or by reading this register.
0	RDR	Receive data determination This indicates that the receive data can be read from the host. This is “0” at reset and by reading this register. 0: No receive data 1: Receive data

18.3.1.14. IrDA modem status register

Symbol IRMSR
Address 0xD8700086
Purpose This shows the level and transition status of four modem control signal input pins from the modem or peripheral devices. This shows the status of the values set by bits 3 to 0 of the IrDA modem control register.

Bit	7	6	5	4	3	2	1	0
Bit name	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
Initial value	Note				0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note : The values are changeable depending on the status of the input pins.

Bit	Bit name	Description
7	DCD	Data detection This shows the status of the input pin NDCD.
6	RI	Link indication This shows the status of the input pin NRI.
5	DSR	Data setting preparation This shows the status of the input pin NDSR.
4	CTS	Transmission deletion This shows the status of the input pin NCTS.
3	DDCD	Delta data detection This is set to “1” if DCD (Bit 7) changes as compared with the last read.
2	TERI	Edge after link indication This is set to “1” if RI (bit 6) changes Low to High as compared with the last read.
1	DDSR	Delta data setting preparation This is set to “1” if DSR (bit 5) changes as compared with the last read.
0	DCTS	Delta transmission deletion This is set to “1” if CTS (bit 4) changes as compared with the last read.

18.3.1.15. IrDA scratch register

Symbol IRSCR
Address 0xD8700087
Purpose This stores the data temporarily.
This is usable in the UART mode/SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	SCR[7:0]							
Initial value	FF							
R/W	R/W							

Bit	Bit name	Description
7-0	SCR[7:0]	Scratch register

Bit	Bit name	Description
		This stores the data temporarily.

18.3.1.16. IrDA extended status/control register

Symbol IRASCR
Address 0xD8700087
Purpose This register is accessed when selecting the extended operation mode.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	PLCT	TXUD	RBSY	LFRF	THFE	SEOT	EOF	RFTO
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R	R

Bit	Bit name	Description
7	PLCT	Pipeline load status This is “1” when a pipeline load operation is generated, and this is cleared by reading this register. TMREV (bit 7) of IrDA extended identification register is cleared to “0” by writing “1” to this bit. *<Programming note>* *When MBPE (bit 0) of IrDA pipeline mode register is set and the transmission data is empty, this bit is set to “1”.* *If the transmission data is empty and the pipeline operation starts during reading this register, the pipeline load operation may not be detected by reading this bit.* *When carry out the pipeline operation, MBPE must be set under the condition that the transmission data is empty (TE (bit 6) of IrDA link control register is set to “1”.).*
6	TXUD	Transmission underrun This is “1” when a transmission underrun occurs. This is cleared by writing “1” to this register.
5	RBSY	Receive operation This is set to “1” during the frame receive operation.
4	LFRF	Frame abandon flag When the receive frame is abandoned, this is set to “1” and stores the contents of IRFRST (bit 6). This is “0” by resetting software.
3	THFE	Transmission stop at the final frame This is valid only in the mode (frame-end stop mode) which finishes the transition of the frame by the internal counter. For the frame-end stop mode setting, refer to 409, 18.3.6.2 IrDA infrared control register 2. Transmission stops and the bit is set to “1” when the transition of the frame length set by IrDA transmission frame-length lower count register/IrDA transmission frame-length upper count register. Transmission is restarted by writing “1” to this bit.
2	SEOT	Final transmission data set If this is set to “1” before writing transmit FIFO transmission data, it is indicated that the data is the final frame. This is “0” by writing the transmission data to after setting.

Bit	Bit name	Description
1	EOF	EOF byte (Receive FIFO) This is set to “1” when there is one frame final data at least in the receive FIFO. This is “0” when the final data is gone by reading. This is “0” by resetting software.
0	RFTO	When receive FIFO timeout occurs, this is set to “1”. This is cleared by reading the data from the receive FIFO.

18.3.2. Bank 1

18.3.2.1. Divisor latch register

18.3.2.1.1. IrDA devisor latch lower register

Symbol	IRDLLR
Address	0xD8700080
Purpose	This stores the lower 8 bits of the baud-generator division. This is usable in the USRT mode/SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	DLLR[7:0]							
Initial value	FF							
R/W	R/W							

Bit	Bit name	Description
7-0	DLLR[7:0]	Baud generator Baud generator must be set by the following setting. Divisor latch lower register → Divisor latch upper register

<Programming note>
Do not access this register in the MIR mode/FIR mode.

18.3.2.1.2. IrDA divisor latch upper register

Symbol	IRDLUR
Address	0xD8700081
Purpose	This stores the upper 8 bits of the baud generator division. This is usable in the UART mode/SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	DLUR[7:0]							
Initial value	FF							
R/W	R/W							

Bit	Bit name	Description
7-0	DLUR	Baud generator Baud generator must be set by the following setting. Divisor latch lower register → Divisor latch upper register

<Programming note>
Do not access this register in the MIR/FIR mode.

Table 92 Baud generator setting list

Baud rate	Frequency division value	Baud rate	Frequency division value
50	2304	3600	32
75	1536	4800	24
110	1047	7200	16
134.5	857	9600	12
150	768	14400	8
300	384	19200	6
600	192	28800	4
1200	96	38400	3
1800	64	57600	2
2000	58	115200	1
2400	48	…	…

18.3.3. Bank 2

18.3.3.1. IrDA extended control register 1

Symbol IREXCR1
Address 0xD8700082
Purpose This selects the extended mode or unextended mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved							EXTSL
Initial value	0							0
R/W	R							R/W

Bit	Bit name	Description
7-1	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
0	EXTSL	Expansion selection 0: Unextended mode 1:Extended mode Mode selection is set by the IrDA control register.

18.3.3.2. IrDA extended control register 2

Symbol IREXCR2
Address 0xD8700084
Purpose This is used for selecting the sizes of the transmit FIFO and receive FIFO.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0

Bit name	reserved	RFSZ	TFSZ
Initial value	0	0	0
R/W	R	R/W	R/W

Bit	Bit name	Description
7-4	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
3-2	RFSZ	Selection of the receive FIFO size 00:16 bit 01:32 bit 10:reserved 11:reserved
1-0	TFSZ	Selection of the transmit FIFO size 00:16 bit 01:32 bit 10:reserved 11:reserved

18.3.3.3. IrDA receive FIFO data level register

Symbol IRTFLV
Address 0xD8700086
Purpose This returns the number of the receive FIFO data. This is used for debugging software.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved		RFL					
Initial value	0		0					
R/W	R		R					

Bit	Bit name	Description
7-6	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
5-0	RFL	Receive FIFO level This shows the number of the receive FIFO data.

18.3.3.4. IrDA transmit FIFO data level register

Symbol IRRFLV
Address 0xD8700087
Purpose This returns the number of the receive FIFO data. This is used for debugging software.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved		TFL					
Initial value	0		0					

R/W	R	R

Bit	Bit name	Description
7-6	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
5-0	TFL	Transmit FIFO level This shows the number of the transmit FIFO data.

18.3.4. Bank 3

18.3.4.1. IrDA link control shadow register

Symbol IRSHLCR
Address 0xD8700081
Purpose This returns the IrDA link control register value through reading this register.

Bit	7	6	5	4	3	2	1	0
Bit name	SHLCR[7:0]							
Initial value	Note							
R/W	R							

Note: This reflects the result of the IrDA link control register.

Bit	Bit name	Description
7-0	SHLCR[7:0]	The value of this register reflects only the result of writing to IrDA link control register.

<Programming note>
This register is reading only. The operations during writing are not guaranteed.

18.3.4.2. IrDA FIFO control shadow register

Symbol IRSHFCR
Address 0xD8700082
Purpose The value of the IrDA FIFO control register returns when this register is read out.

Bit	7	6	5	4	3	2	1	0
Bit name	RTL		TTL		reserved	TFR	RFR	FEN
Initial value	0		0		0	0	0	0
R/W	R		R		R	R	R	R

Bit	Bit name	Description
7-6	RTL	Receive FIFO trigger level This indicates the receive FIFO trigger level. 00:1/1 01:4/8 10:8/16 11:14/30
5-4	TTL	Transmit FIFO trigger level

Bit	Bit name	Description
		This indicates the transmit FIFO trigger level. 00:1/1 01:4/8 10:8/16 11:16/32
3	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
2	TFR	Transmit FIFO soft reset “0” is always read out.
1	RFR	Transmit FIFO soft reset “0” is always read out.
0	FEN	Enable FIFO 0: FIFO not using mode 1: FIFO using mode *<Programming note>* *Valid only in the UART mode/SIR mode* *Always “0” in the MIR mode/FIR mode*

<Programming note>
This register is read only. The operations at writing are not guaranteed.

18.3.5. Bank 4

18.3.5.1. Timer initial value register

This register sets the timer of the general-purpose 12-bit down counter. The resolution of the timer is 1ms and the maximum count time is approximately 4 seconds. Countdown starts and stops at the count value “0” if “1” is set to the TMRST (bit 2) of the IrDA infrared control register after the count value is written to the IrDA timer initial value lower register/IrDA timer initial value upper register under the condition that the TMREN (bit 0) =1 IrDA infrared control register 1. Continuous operations need to rewrite “1” to the TMRST (Bit 2) of the IrDA infrared control register.
TMRST (bit 2) of the IrDA infrared control register 1 is automatically reset when the count starts. TMRIE (bit 7) =1 of the IrDA extended interrupt enable register must be set for generating interrupts at the end of the timer count.

18.3.5.1.1. IrDA timer initial value lower register

Symbol	IRTMRL
Address	0xD8700080
Purpose	This decides the counter load value of the lower bytes to the general-purpose timer. The setting and count values are returned through reading. Selection of the reading values is carried out at CTBT (bit 1) of the IrDA infrared control register 1. This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	TMRL[7:0]							

Initial value	0
R/W	R/W

Bit	Bit name	Description
7-0	TMRL[7:0]	This indicates the counter load value of the lower byte to the general-purpose timer. The setting and count values are returned through reading.

18.3.5.1.2. IrDA timer initial value upper register

Symbol IRTMRH
Address 0xD8700081
Purpose This indicates the counter load value of the upper 4 bits to the general-purpose timer. The setting and count values are returned through reading. A value to read is selected by CTST (bit 1) of the IrDA infrared control register 1.
This is usable in the MIR/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved				TMRH[3:0]			
Initial value	0				0			
R/W	R				R/W			

Bit	Bit name	Description
7-4	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
3-0	TMRH[3:0]	This indicates the counter load value of the upper 4 bit to the general-purpose timer. The setting and count values are returned through reading.

18.3.5.2. IrDA infrared control register 1

Symbol IRCR1
Address 0xD8700082
Purpose This is used for controlling the timer and counter.
All bits are to “0” through reset.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved					TMRST	CTST	TMREN
Initial value	0					0	0	0
R/W	R					R/W	R/W	R/W

Bit	Bit name	Description
7-3	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
2	TMRST	Timer start The counter of the general-purpose timer starts up through writing “1”. This is automatically reset when the counter recognizes the assertion

Bit	Bit name	Description
		of this bit.
1	CTST	Counter test 0: Return the executing count value during reading out the IrDA timer initial value lower register/IrDA timer initial value upper register. 1: Return the initial value during reading out the IrDA timer initial value lower register/IrDA timer initial value upper register.
0	TMREN	Timer enable 1: General-purpose timer counter enable

18.3.5.3. Transmit frame length counter

The counter is included to control the frame length at transmission. The initial values are set or the count is read out by the following register. When the count is decremented from the initial values and reaches “0”, the frame finishing process. And then this transmits CRC and STO and waits for the next frame transmit.

The internal counter is 13 bit. Accordingly, writes to bits 7 to5 of the IrDA transmission frame length upper count register are invalid because the initial values set by the IrDA transmit frame-length lower count register/IrDA transmit frame-length upper count register are also 13 bit. “0” is always returned in reading these bits. “1” is the minimum value that can be set as initial value. Do not set “0”. The data to be actually transmitted adds CRC to this setting value. If setting “N”, the following data are treated as transmit data: (N+2) bytes in the MIR mode, (N+4) bytes in the FIR mode.

18.3.5.3.1. IrDA transmit frame length lower count register

Symbol	IRTFLL
Address	0xD8700084
Purpose	This is accessed as a set of the transfer frame length 8LSB at transmitting in writing and as the number of data bytes, 8LSB, during frame transmission in reading. This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	TFLL[7:0]							
Initial value	0							
R/W	R/W							

Bit	Bit name	Description
7-0	TFLL[7:0]	Transmit frame length lower count Writes: 8LSB of transmission frame length Reads: 8LSB of the number of the data bytes during transmitting the frame.

18.3.5.3.2. IrDA transmit frame length upper count register

Symbol	IRTRLU
Address	0xD87000085
purpose	This is accessed as a set of the transmission frame length 5MSB at transmitting in writing and as the number of data bytes, 5MSB during frame transmission in reading. This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			TFLU				
Initial value	0			01000				
R/W	R			R/W				

Bit	Bit name	Description
7-5	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
4-0	TFLU	Transmit frame length upper count Writes: 5MSB of transmission frame length Reads: 5MSB of the number of the data bytes during transmitting the frame.

18.3.5.4. Receiver frame maximum-length counter

This can set the most allowable maximum-frame length at receiving by the following register. An internal counter is included to count the receiver frame length, and when the value is over the maximum length, the later writing of FIFO is not carried out.

The internal counter is 13 bit. Accordingly, the maximum length set by the above-mentioned registers is also 13 bit. Writes to bits 7 to 5 of the IrDA receiver frame maximum-length upper count register are invalid. “0” is always returned in reading these bits. “4” is the minimum value that can be set as maximum length. The maximum length to be set must include CRC.

If setting “N”, the allowable numbers of data are (N-2) bytes in the MIR mode, (N-4) bytes in the FIR mode. When being over the maximum length, FIFO writes of CRC cannot be carried out.

18.3.5.4.1. IrDA receiver frame maximum-length lower count register

Symbol IRRMLL
Address 0xD8700086
Purpose This is accessed as the maximum frame length 8LSB at receiving in writing and as the number of data bytes, 8LSB, during frame receiving in reading.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	IRRMLL[7:0]							
Initial value	0							
R/W	R/W							

Bit	Bit name	Description
7-0	RMLL	Receiver frame maximum-length lower count Writes: 8LSB of receiver maximum-frame length Reads: 8LSB of the number of the data bytes during transmitting the frame.

18.3.5.4.2. IrDA receiver frame maximum-length upper count register

Symbol IRRMLU
Address 0xD8700087
Purpose This is accessed as a set of the transmission frame length 5MSB at receiving in writing and as the number of data bytes, 5MSB during frame receiving in reading.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			RMLU[4:0]				
Initial value	0			01000				
R/W	R			R/W				

Bit	Bit name	Description
7-5	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
4-0	RMLU[4:0]	Receiver frame maximum-length upper count Writes: 5MSB of receiver maximum-frame length Reads: 5MSB of the number of the data bytes during transmitting the frame.

18.3.6. Bank 5

18.3.6.1. IrDA pipeline mode register

Symbol IRPMDR
Address 0xD8700082
Purpose This controls the pipeline operation and sets the mode.

Bit	7	6	5	4	3	2	1	0
Bit name	PLMD			reserved		reserved		MBPE
Initial value	0			0		0		0
R/W	R/W			R		R/W		R/W

Bit	Bit name	Description
7-5	PLMD	Pipeline mode selection The mode is decided when MBPE is “1” and the transmission data is empty. 000:UART 001:reserved 010:reserved 011:SIR 100:MIR 101:FIR 110:reserved 111:reserved
4-1	reserved	These are reserved. A “0” is always returned when the bit is read.

Bit	Bit name	Description
		When writing this register, always write a “0” to these bits.
0	MBPE	Mode bit pipeline starting enable When this bit is “1” and the transmit data is empty, the pipeline operation starts.

<Programming note>
This register is valid only when the transfer from SIR mode to MIR mode/FIR mode is to be carried out.

18.3.6.2. IrDA infrared control register 2

Symbol	IRCR2
Address	0xD8700084
Purpose	This performs the controls for the infrared transmission. This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved	SFTH	FREC	AIRS	TXMD	MRSL	reserved	
Initial value	0	0	0	0	0	0	0	
R/W	R	R/W	R/W	R/W	R/W	R/W	R	

Bit	Bit name	Description
7	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
6	SFTH	Status FIFO trigger level A status FIFO interrupt occurs because of reaching the trigger level or timeout. This bit decides the trigger level. 0: 2 Byte 1: 4 Byte
5	FREC	Frame end control This validates the frame final data generation counter in the PIO mode. 0: PIO mode
4	AIRS	Infrared input auxiliary selection This is used for selecting the IrDA input pin. For the details, refer to 419, 18.3.8.3 IrDA extended control register 4.
3	TXMD	Transmit mode selection This recognizes the end of the frame by using the counter. 1: frame-end stop mode *<Programming note>* *This is used only in the MIR mode/FIR mode.* *<Programming note>* *The transmission can be restart through clearing TXHFE (bit 3) of the IrDA extended status/control register.*
2	MRSL	MIR rate selection This decides the transfer speed in the MIR mode. 0: 1.152 Mbps 1: 0.576 Mbps
1	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
0	FDPX	Infrared full duplex mode 0: Half duplex transmission 1: Full duplex transmission *<Programming note>* *The Infrared receive is interrupted when the transmission is being executed during the half duplex transmission.* *The operations are not guaranteed when the host performs the writes of the transmit FIFO and the transmit starts.*

<Programming note>
Bit 4 is also valid in SIR.

18.3.6.3. IrDA status-FIFO frame status register

Symbol IRFRST
Address 0xD8700085
Purpose This stores the status FIFO.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	VLD	LTFR	reserved	MFLE	PHLE	CRCE	OERF	OETF
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Bit name	Description
7	VLD	Status FIFO valid entry This is cleared to “1” when all data of the 24 bits stored at the bottom of the status FIFO are valid. In the following cases, this is “0”. 1）There is no data in the status FIFO. 2）Setting of 24 bits has not completed yet during updating the bottom of the status FIFO.
6	LTFR	Frame-abandoning indication flag This is set to “1” when the frame is abandoned because of overrun. The contents stored in IrDA status FIFO frame-length lower count register/IrDA status FIFO frame-length upper count register change depending on this value.
5	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
4	MFLE	Maximum-frame length overrun This is set to “1” in the case of receiving the data over the frame maximum allowable length which is set in the IrDA receive frame maximum length lower count register/IrDA receive frame maximum length upper count register.
3	PHLE	Physical layer error This is set to “1” when the abortion and encoding error are detected in respectively the MIR mode and FIR mode.
2	CRCE	CRC error This is set to “1” when the received CRC data and the CRC data calculated in the receiver are matched to each other.
1	OERF	Receiver FIFO overrun error This is set to “1” when the frame data is abandoned because the receiver FIFO is full.
0	OETF	Status FIFO overrun error This is set to “1” when the frame data is abandoned because the status FIFO is full.

18.3.6.4. IrDA status FIFO frame-length count register

18.3.6.4.1. IrDA status FIFO frame-length lower count register

Symbol	IRFRLL
Address	0xD8700086
Purpose	When 8LSB, the status FIFO bottom frame length, or the frame has a loss because the FIFO is full, a soft frame account is read out. This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	FRLL[7:0]							
Initial value	0							
R/W	R							

Bit	Bit name	Description
7-0	FRLL[7:0]	Status FIFO frame-length lower count 8LSB, status FIFO bottom frame-length or the frame has a loss because the FIFO is full, the count of the lost frame is read out. When the LTFR (bit 6) of the IrDA status FIFO frame status register is “0”, the frame length, 8 LSB, is stored. When the LTFR (bit 6) of the IrDA status FIFO frame status register is “1”, the number of the lost frames is stored.

18.3.6.4.2. IrDA status FIFO frame length upper count register

Symbol IRFRLU
Address 0xD8700087
Purpose 5 MSB, the status FIFO bottom frame length is read out.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	FRLU[7:0]							
Initial value	0							
R/W	R							

Bit	Bit name	Description
7-0	FRLU[7:0]	IrDA status FIFO frame length upper count 5 MSB, status FIFO bottom frame length is read out. When the LTFR (bit 6) of IrDA status FIFO frame status register is “0”, the frame length, 5 MSB is stored. When the LTFR (bit 6) of IrDA status FIFO frame status register is “1”, all bits are “0”.

18.3.7. Bank 6

18.3.7.1. IrDA infrared control register 3

Symbol IRCR3
Address 0xD8700080
Purpose This is used for selecting the operating mode of the infrared interface.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved					ICRC	DCRC	reserved
Initial value	0					0	0	0
R/W	R					R/W	R/W	R

Bit	Bit name	Description
7-3	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
2	ICRC	CRC invert transmission 1: CRC is reversed and transmitted.
1	DCRC	CRC transmission disable 1: CRC is transmission
0	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.

18.3.7.2. IrDA MIR pulse setting register

Symbol	IRMIRPW
Address	0xD8700081
Purpose	This sets the pulse width at modulation-demodulation in the MIR mode. This is usable in the MIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved				MPW[3:0]			
Initial value	0				1010			
R/W	R				R/W			

Bit	Bit name	Description
7-4	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
3-0	MPW[3:0]	MIR signal pulse width

Encode pulse width

	MRSL=0(1.152Mbps)	MRSL=1(0.576Mbps)
00xx	reserved	reserved
0100	83.33 ns	166.66 ns
0101	104.16 ns	208.33 ns
0110	125 ns	250 ns
0111	145.83 ns	291.56 ns
2000	166.66 ns	333.33 ns
1001	187.50 ns	374.99 ns
1010	208.33 ns	416.66 ns
1011	229.16 ns	458.33 ns
1100	250 ns	500 ns
1101	270.83 ns	541.66 ns
1110	291.66 ns	583.32 ns
1111	312.5 ns	625 ns

<Programming note>
When a value except the above-mentioned values is written, the operation is not guaranteed.

18.3.7.3. IrDA SIR pulse width register

Symbol	IRSIRPW
Address	0xD8700082
Purpose	This sets the pulse width at the modulation-demodulation in the SIR mode. This is usable in the SIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved				SPW[3:0]			
Initial value	0				0			
R/W	R				R/W			

Bit	Bit name	Description
7-4	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
3-0	SPW[3:0]	SIR signal pulse width 0000: 3/16 bit time 1101:1.6 μs fixed

<Programming note>
When a value except the above-mentioned values is written, the operation is not guaranteed.

18.3.7.4. IrDA beginning/preamble length register

Symbol IRBFPL
Address 0xD8700084
Purpose This sets the number of the beginning flag in the MIR mode and the number of the preamble flag in the FIR mode.
This is usable in the MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	MBF[3:0]				FPL[3:0]			
Initial value	0				1010			
R/W	R/W				R/W			

Bit	Bit name	Description
7-4	MBF[3:0]	FIR preamble length This shows the number of the FIR preamble. 0000: reserved 1000: 10 0001: 1 1001: 12 0010: 2 1010: 16 0011: 3 1011: 20 0100: 4 1100: 24 0101: 5 1101: 28 0110: 6 1110: 32 0111: 8 1111: reserved
3-0	FPL[3:0]	MIR beginning flag This specifies the number of the beginning flag of the MIR frame. 0000: reserved 1000:10 0000: 1 1001:12 0010: 2 1010:16 0011: 3 1011:20 0100: 4 1100:24 0101: 5 1101:28 0110: 6 1110:32 0111: 8 1111:reserved

18.3.7.5. IrDA FIR pulse width register

Symbol	IRFIRPW
Address	0xD8700087
Purpose	This sets the pulse width in the FIR mode. This is usable in the FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved					FPWS	FPWD	
Initialvalue	0					0	01	
R/W	R					R/W	R/W	

Bit	Bit name	Description
7-3	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
2	FPWS	FIR receive minimum single pulse width This sets the minimum width for pulse identification in the FIR mode receive. 0: XIN positive edge x2 (This needs the pulse width of 42ns and more regardless of the receive pulse and the phase of the system clock for the IrDA block (XIN).) 1: XIN positive edge x 3 (This needs the pulse width of 65ns and more regardless of the receive pulse and the phase of the system clock for the IrDA block (XIN).)
1-0	FPWD	FIR receive maximum single-pulse width This sets the maximum width identified as single pulse. (the boundary value between single and double) 00: XIN positive edge x 9 (This is identified as single pulse if it is under 167ns regardless of the phase of the receive pulse and the system clock for the IrDA block system (XIN).) 01: XIN positive edge x 10 (This is identified as single pulse if it is under 187ns regardless of the phase of the receive pulse and the system clock for the IrDA block system (XIN).) 10: XIN positive edge x 11 (This is identified as single pulse if it is under 208ns regardless of the difference between the receive pulse and the system clock (XIN).)

<Programming note>
This can normally receive anything with a pulse width in conformance with the specifications regardless of the above register setting.

18.3.8. Bank 7

18.3.8.1. IrDA infrared interface control register 1

Symbol	IRCFG1
Address	0xD8700084
Purpose	This sets a transmit alignment in the SIR mode.

The automatic alignment controls the directory in the transmit operational mode when not enabled.
The lower 4 bits are also used for reading the identical data of the plug-and-play of the infrared adapter.
This can be usable in the SIR mode/MIR mode/FIR mode.
This cannot use the IRCFG1 register because the IRSL control is not implemented in this LSI.
ID can always read out “0”.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved							
Initial value	0							
R/W	R							

Bit	Bit name	Description
7-0	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.

18.3.8.2. IrDA infrared interface control register 2

Symbol	IRCFG2
Address	0xD8700085
Purpose	This sets the transmit alignment. This is usable in the MIR mode/FIR mode. This cannot use the IRCFG2 register because the IRSL control is not implemented in this LSI.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved							
Initial value	0							
R/W	R							

Bit	Bit name	Description
7-0	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.

18.3.8.3. IrDA extended control register 4

Symbol IRCFG4
Address 0xD8700087
Purpose This uses the pin assignment of the receive data path and enables the shape-pin automatic selection. This sets to “0” after reset.
This is usable in the SIR mode/MIR mode/FIR mode.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved	RXMD	reserved	RXIV	reserved			
Initial value	0	0	0	0	0			
R/W	R	R/W	R	R/W	R			

Bit	Bit name	Description
7	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
6	RXMD	This selects an input pin depending on the high-speed or low-speed IrDA mode. 0: Input from a single pin to the SIR mode/MIR mode/FIR mode. 1: Separate input depending on the SIR mode/MIR mode/FIR mode. IrDA assignment register (Extended index 04h when the RXIV is “0”.)

RXMD	AIRS	IrDA mode	Input pin
0	0	All modes	IRRXDS
0	1	All modes	IRRXDF
1	x	SIR mode	IRRXDS
1	x	MIR mode/FIR mode	IRRXDF

IrDA assignment register
(Extended index 04h when the RXIV is “1”.)

RXMD	AIRS	IrDA mode	Input pin
0	0	All modes	IRRXDF
0	1	All modes	IRRXDS
1	x	SIR mode	IRRXDF
1	x	MIR mode/FIR mode	IRRXDS

5	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.
4	RXIV	This can control the internal capture polarity of the SI-pin input signal in the IrDA mode. 1: Capture in reverse.
3-0	reserved	These are reserved. A “0” is always returned when the bit is read. When writing this register, always write a “0” to these bits.

18.4. Operational description

18.4.1. Transmit data FIFO

The configuration of the FIFO is 9 bits x 32 steps, and the used area is different depending on the mode.

O IrDA Version 1.0 mode (UART mode/SIR mode)
- 8 bits x 32 steps (Fixed step number)

O IrDA Version 1.1 mode (MIR mode/FIR mode)
- 9 bits x 16/32 steps (The step number is changeable.)
  The 9 bits consists of the 8 bit data and frame final data flag.
  The switch of the step numbers is selected through the TFSZ (bit 1-0) of the IrDA extended control register 2.

18.4.2. Receive data FIFO

The configuration of the FIFO is 11 bits x 32 steps, and the used area is different depending on the mode.

O IrDA Version 1.0 mode (UART mode/SIR mode)
- 11 bits x 32 steps (Fixed step number)
  8-bit data + Framing error bit + Break interrupt bit + Parity error bit

O IrDA Version 1.1 mode (MIR mode/FIR mode)
- 9 bits x 16/32 steps (The step number is changeable.)
  The 9 bits consists of the 8 bit data and frame final data flag.
  The switch of the step numbers is selected through the IrDA extended control register 2 (Bits 3-2).

18.4.3. Status FIFO

This is used only in the IrDA Version 1.1 mode. The frame status information is stored in the internal FIFO per end of the frame receive. The FIFO consists of 24 bits x 8 steps.

The contents of 24 bits are shown below.

- Each frame status information (8 bit)
  (1) Status FIFO overrun
  (2) Receive FIFO overrun
  (3) CRC error
  (4) Physical layer error
  (5) Frame maximum-length error
  (6) Frame lost
  (7) Status FIFO active
- The frame length of each frame or the count of the lost frame (16 bit)

When there is no problem in the above-mentioned status information, 8 bit, the frame length, 16 bit, is stored. When there are errors and the frame is lost, the lost frame count is stored in the 8 bits of them.

The status FIFO, which is necessary for that the CPU gets the information such as the frame boundary in the case of transmitting multiple frames to the memory at the receives in the MIR mode/FIR mode, is included in the IrDA block. The configuration of the status FIFO is as follows.

- Frame status information 8 bits x 8 steps
  Error flags and etc. are stored.
- Frame length information 16 bits x 8 steps
  The frame length is stored. However, when the status FIFO is full and abandoned, the count of the lost frame is stored in 8 bits of them.

The status FIFO loads all of 24 bits when the frame receive process finishes through the detection of the stop flag and CRT inspections.

When the receive FIFO overflow or the status FIFO overflow occurs in the process of receiving, the status loading is not carried out. Instead, the counter of the internal lost frame is counted up. Subsequently, not only the status data of the frame with overflow, but also those of the following frame are lost. This is caused because the overflow status and lost frame count are loaded to the status FIFO when the first frame without overflow finishes.

18.4.4. IrDA Version 1.0 mode

IrDA Version 1.0 mode includes the following modes.

O UART mode
The internal UART block uses the interface to communicate with the outside.

O SIR mode
This communicates through performing the 3/16 modulation and demodulation to the input and output of the internal UART block.

This consists of the following functions.

- UART
  This carries out the asynchronous serial transfer and transmit processing. This block also performs the interface with the host and FIFO.

- SIR modulation
  At transmitting in the SIR mode, this modulates the pulse width of the serial NRZ data to the 3/16 of the bit time or 1.6μs fix, and outputs to the outside.

- SIR demodulation
  At receiving in the SIR mode, this demodulates the bit stream input from the outside to the NRZ data, and outputs the UART.

O IRRX selection
When receiving, this selects the serial data input to the UART according to modes.
UART mode: External input SIN
SIR mode: Serial data performed the SIR demodulation for.

18.4.4.1. UART mode

This carries out an asynchronous serial data transmission. After reset, this is set in the UART mode. When transmitting, this performs the PS conversion for the data written from CPU and adds the control bits (start, parity, stop) to the data, and then it outputs the data from the SOUT pin. When receiving, this eliminates the control bits from the data which is input from the SIN pin and performs the PS conversion for the data, and then the data is stored in the inside and can be read out from the CPU and DMAC. The transfer rate is -1.5Mbps and fill duplex.

The character format for transmission and reception is shown below.

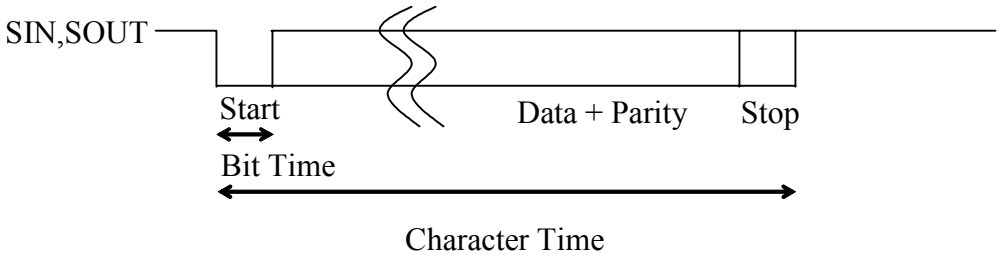


Figure 93 UART mode pulse

18.4.4.2. SIR mode

When transmitting, this modulates the data with the asynchronous parallel/serial conversion and outputs it from the IRTXD pin. When receiving, this demodulates the input pulse from the IRRXDS pin and outputs it through the asynchronous serial/parallel conversion. The transfer rate is -115.2KHz and half duplex. The pulse width at modulating and demodulating is 3/16 of the bit time or 1.6 μs fix.

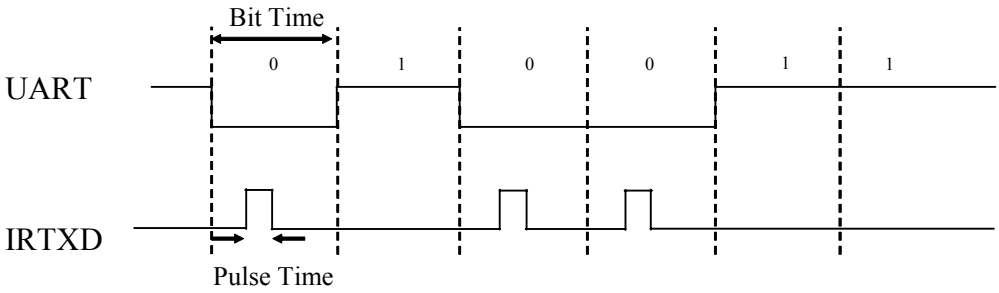


Figure 94 SIR mode pulse

18.4.5. IrDA Version 1.1 mode

IrDA version 1.1 includes the following modes.

O MIR mode
This performs the serial transmit in the HDLC format frame unit. The transfer rate is 576 Mbps/1.152Mbps.

O FIR mode
This performs the serial transmission in the 4PPM modulation/demodulation format frame unit. The transfer rate is 4 Mbps.

18.4.5.1. MIR mode

This performs the serial data transmission in the frame unit.

O Transmission
- Converses the parallel data to the serial data, and arranges the bits in the order from the LSB.
- Performs the 16-bit CRC calculation and adds the bits in the order from the LSB behind the data when the frame finishes.
- Inserts “0” to these bit stream
- Outputs them from the SOUT pin after adding the flags at the front and back and performing the 1/4 modulation.

O Reception
- Performs the 1/4 demodulation through inputting from the SIN.
- Starts the decode processing when the start flag is detected.
- Eliminates “0”.
- Performs the parallel conversion and the CRC check when the stop flag is detected, and outputs the data to the FIFO in the case of no problem or sets up the flag in the case of NG.

The frame format is shown below.

STA	STA	DATA	CRC16	STO

STA: 8 bit 01111110
DATA: Address & Control & Information (-2048 byte)
CRC16: 16bit CRC operational result
STO: 8 bit 01111110

The transmission is carried out in the half duplex fashion, and the transfer rate is 0.576 Mbps or 1.152 Mbps. The pulse width at modulation/demodulation is variable.

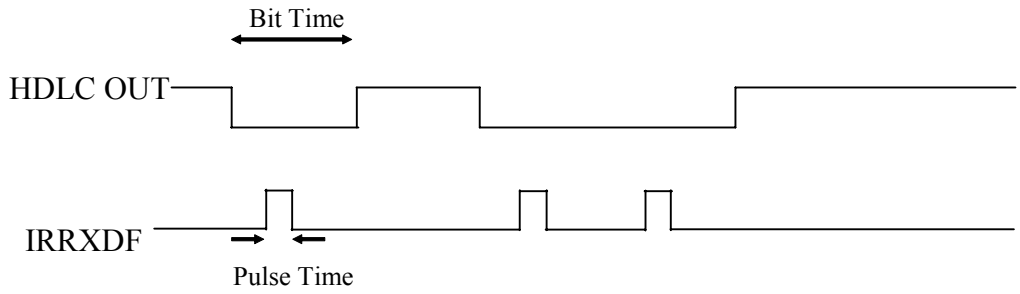


Figure 95 MIR mode pulse

18.4.5.2. FIR mode

This performs the serial transmission through the 4 PPM modulation/demodulation. The transfer rate is 4 Mbps and half duplex. The frame format is shown below.

PA	STA	DATA	CRC32	STO

PA: 16bit 1000 0000 1010 1000
STA: 32bit0000 1100 0000 1100 0110 0000 01110 0000
DATA: Address 1 byte, Control 1 byte, Information 64-2048 byte
CRC32: CRC operational result
STO: 32bit0000 1100 0000 1100 0000 0110 0000 0110

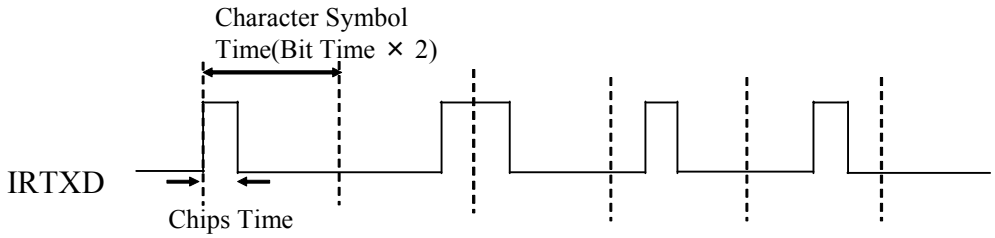


Figure 96 FIR mode pulse

18.4.6. Interrupt

Interrupts are set to enable by the IrDA interrupt enable register, and the IrDA controller interrupt occurs with the occurrence of the cause. The types of the interrupts can be recognized through reading the followings: IrDA interrupt identification register, IrDA extended interrupt identification register, IrDA link status register, IrDA extended link status register, IrDA modem status register. These are different depending on the types of the interrupt sources or the modes of the read formats.

18.4.6.1. Interrupt priority

18.4.6.1.1. UART mode/SIR mode

The priorities to the interrupt causes in the UART mode/SIR mode are as follows.

IRIIR[3:0]	Priority	Interrupt types
0001	-	No interrupts
0110	1	Receive line status
0100	2	Receive data ready
1100	2	Character timeout
0010	3	Transmit data buffer empty

1) Receive line status interrupt

The receive line status interrupt is enabled when the RLS (bit 2) of the IrDA interrupt enable register is “1”.
The interrupt cause can be recognized by reading the IrDA link status register.
The interrupt causes are as follows.

- The OVE (bit 1) of the IrDA link status register is 1: overrun error
  When the receive data is overwritten before reading the data from the host.
- The PTE (bit 2) is 1: parity error
  When the parity error is detected in the receive data.
- The FME (bit 3) of the IrDA link status register is 1: framing error
  When the stop bit of the receive data is not detected.
- The BRI (bit 4) of the IrDA link status register is 1: break interrupt
  When all receive data are “0”.

2) Receive data acquisition interrupt

The receive data acquisition interrupt is enabled when the RDA (bit 0) of the IrDA interrupt enable register is “1”.

The cause of interrupts in the FIFO using mode is that the data above the trigger level is stored in the receive FIFO. If the data is under the trigger level at reading the IrDA receive data register, it is negated.

The Cause of interrupts in the FIFO using mode is that the data is stored in the IrDA receive data register. The data is negated by reading out the IrDA receive data register.

3) Timeout interrupt

The timeout interrupt is enabled in the FIFO using mode when the RDA (bit 0) of the IrDA

interrupt enable register is “1”.
The cause of interrupts is that there is no access to the receive data FIFO in the case of over 4-character time.

This is negated through reading the IrDA receiver data register.

4) IrDA transmit data register empty interrupt
The transmit data buffer empty interrupt is enabled when the TDE (bit 1) of the IrDA interrupt enable register is “1”.

The cause of interrupts in the FIFO using mode is that the transmit data FIFO is empty.

The cause of interrupts in the FIFO non-using mode is that the IrDA transmit data register is empty.

This is negated by reading the IrDA interrupt identification register if the followings are not carried out; the writes to the IrDA transmit register or the interrupt processing of the higher priority ranking.

18.4.6.1.2. MIR/FIR mode

There is no priority to the interrupt cause in the MIR/FIR mode.

The interrupt source in the IrDA extended interrupt identification register and the interrupt enable in the IrDA extended interrupt enable register correspond to each other at the ratio of 1:1.

1) Receive data high level interrupt
when the RLIE (bit 0) of the IrDA extended interrupt enable register is “1”, the receive data high-level interrupt is enabled.

The interrupt causes are the following two points.

<Receive data FIFO over-trigger level>
- Causes
  The data of the receive data FIFO is over the trigger level.
- Negate
  When the data of the receive data FIFO is under the trigger level, the data is negated at the read finishing edge of the IrDA receive data register.

Table 93 Trigger level of the receive data FIFO

RFCR [7:6] RTL	Trigger level (FIFO size 16/32)
00	1/1
01	4/8
10	8/16
11	14/30

<Receive data FIFO timeout>
This occurs when the following three conditions happen.

1) The receive data FIFO includes the over-1 byte data.
2) The receive data to the 64μs FIFO is not sent.
3) No reads of the 64μs FIFO.

- Negate
  This reads the FIFO data. (The RDR read finishing edge)

2) Transmit low data level interrupt

When the TLIE (bit 1) of the IrDA extended interrupt enable register is “1”, the transmit low data level interrupt is enabled.

- Causes
  When the space for the trigger level is in the transmit data FIFO.
- Negate
  When the space in the transmit data FIFO is under the trigger level, this is negated at the timing of the IrDA transmit data register write finishing edge.

Table 94 Trigger level of the transmit data FIFO

TTL (FCR Bit 5 to Bit 4)	Trigger Level(FIFO Size 16/32)
00	1/1
01	4/8
10	8/16
11	16/32

3) Line status interrupt

When the LSIE (bit 2) of the IrDA extended interrupt enable register is “1”, the line status interrupt is enabled.

The line status interrupt occurs because of the following causes.

- When the frame final data reaches the bottom of the receive FIFO,
  the FE (bit 7) of the IrDA link status register is set to “1”.
  This is negated at reading the IrDA link status register.

- When the overrun occurs at receiving,
  the OVE (bit 1) of the IrDA link status register is set to “1”.
  This is negated at reading the IrDA link status register.

- When the underrun occurs at transmitting,
  the TXUD (bit 6) of the IrDA extended status/control register is set to “1”.
  This is negated only by soft reset.

- When the transmit stops at the end of the frame,
  the THFE (bit 3) of the IrDA extended status/control register is set to “1”.
  After stopping the transmit, this is negated when the THFE of the IrDA extended status/control register is written “1” and restarted.

  <Programming note>
  This has a function to stop the transmit after finishing the frame. (described later)

4) Transmit empty/pipeline load interrupt

When the TEIE (bit 5) of the IrDA extended interrupt enable register is “1”, this

interrupt is enabled.
When no data are in the overall transmit part, this is asserted. When the pipeline is enabled, the next mode is loaded.

5) Status FIFO event interrupt
The SFIF (bit 6) of the IrDA extended interrupt enable register is “1”, this interrupt is enabled.

The interrupt causes are the following two points.

<Status FIFO over-trigger level>

- Causes
  When the status FIFO is over the trigger level.
- Negate
  When the Status FIFO is under the trigger level.
  (the read finishing edge in the order of FRST, FRLL, and FRLU.)

The status FIFO trigger level is certified at the SFTH (bit 6) of the IrDA infrared control register.

Table 95 Status FIFO trigger level

IRCR2(bit 6) SFTH	Trigger level
0	2 byte
1	4 byte

<Status FIFO timeout>
When the following conditions happen, this occurs.

1) The receive data FIFO includes the over-1 byte data.
2) The receive data to the FIFO is not sent in the case of over 1 ms.
3) No reads of the FIFO in the case of over 1 ms.

- Negate
  This reads the FIFO data. (the read finishing edge in the order of FRST, FRLL, and FRLU.)

6) Timer interrupt
When the TMRIE (bit 7) of the IrDA extended interrupt enable register is “1”, this interrupt is enabled.

18.4.7. FIFO timeout

The IrDA block becomes time-out in the case of no access during a period even if the there is the data in the FIFO at receiving. The conditions are different depending on the modes.

18.4.7.1. UART mode/SIR mode

<Timeout condition>

This occurs when the following conditions are included.

1) FIFO mode
2) There is over-1-character data in the FIFO.
3) The data is not received to the FIFO during over-4-character time.
4) There is no read of the FIFO during over-4-character time.

<Operations after occurrence>

- The interrupts occur if enabled.
- The time-out status can be read out at the reads of the IrDA interrupt identification register.

18.4.7.2. MIR/FIR mode

These modes respectively have timeouts because they use the receive data FIFO and status FIFO at receiving.

<Timeout conditions>

O Receive data FIFO

This occurs when the following conditions are included.

1) There is over-1-byte data in the receive data FIFO.
2) There is no write to the receive data FIFO during over-64 μs.
3) There is no read of the receive data FIFO during over-64 μs.

O Status FIFO

1) There is over-1-byte data in the status FIFO.
2) There is no write to the status FIFO during over-1ms.
3) There is read of the status FIFO during over-1ms.

<Operations after occurrence>

- The interrupts occur if enabled.
- The time-out status can be read out at the reads of the IrDA extended status/control register.

18.4.8. FIFO underrun prevent function (Transmit deferral)

There is a risk that the MIR/FIR generates the FIFO underrun at transmit because the IR port side has high speed. Immediately after the underrun occurs, the transmit part reverses and transmits the CRC operation result and then returns to the initial value. The receive part purposely generates an error to the frame and shows the problem. In order to prevent this problem, this IrDA locks the transmit (only in the PIO mode) until 14 bytes in the case of the 16-step FIFO and 30 bytes in the 32-step FIFO are stored and then starts the transmit.
This function is enabled by setting the TXDF (bit 3) of the IrDA mode control register. The TXDF (bit 3) is internally cleared at releasing the lock.

18.4.9. Transmit stop function

The MIR mode/FIR mode has a function to stop the transmission once after finishing the frame transmit. This function can reset the frame length while stopping. In other words, different sizes can be frame-transmitted without the re-initialization of the DMAC.
To be more specific, this option is set through setting the TXMD (bit 3) of the IrDA infrared control

register 2. Subsequently, the transmit stops when the frame-finishing-flag is identified and the final data is transmitted. Restart is carried out through writing “1” to the THFE (bit 3) of the IrDA extended status/control register.

18.4.10. IR-UNIT interface

There are the following specifications for carrying out the IR-UNUT interface.
1) ID3-0: used for ID reads. (“0000” internally fixed)
2) IRSL2-0 output: used for selecting the IR-UNIT mode. (allocated to GPIO)

18.4.11. Interaction pulse (SIP) transmission

This function is valid only in the MIR mode/FIR mode and transmits at the end of the interaction pulse frame. The pulse width is 1.6μs. The pulse transmit is enabled through setting “1” to the SIP (bit 4) of the IrDA mode control register and is disabled immediately after transmission. The transmit waveshape of the pulse in the MIR mode is shown below.

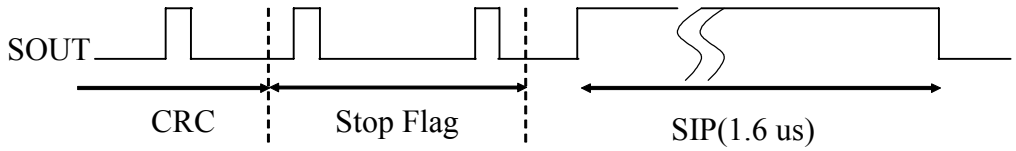


Figure 97 SIP pulse generation

Chapter 19 I2C Controller (I2C)

19

CHAPTER 19
I2C Controller (I2C)

19.1. General

This microcontroller includes two independent I2C controllers that support multiple masters.

19.2. Features

- Hardware-based bus arbitration and bus monitoring function
  Software can set any timing for communications start regardless of the status of the I2C bus.
- Programmable slave address
  Matches the slave address that has been set
- Supports four modes
  Master transmit, master receive, slave transmit and slave receive
- Interrupt generator function
  Communications end, bus acquisition and failure, stop condition detection
- Programmable serial clock frequency
- Open drain output (SDA/SCL) and 3.3 V signal level
  A pull-up resistor (3.3 V) is required on the board.

19.3. Registers

Address	Symbol	Name	Number of bits	Initial value	Access size
0xD8400000	IIC0DTRM	I2C transmit data register 0	32	0x00000000	32
0xD8400004	IIC0DREC	I2C receive data register 0	32	0x000009FF	32
0xD8400008	IIC0MYAD	I2C slave address register 0	32	0x00000000	32
0xD840000C	IIC0CLK	I2C clock register 0	32	0x00000000	32
0xD8400010	IIC0BRST	I2C bus reset register 0	32	0x00000001	32
0xD8400014	IIC0BSTS	I2C bus status register 0	32	Undefined	32
0xD8401000	IIC1DTRM	I2C transmit data register 1	32	0x00000000	32
0xD8401004	IIC1DREC	I2C receive data register 1	32	0x000009FF	32
0xD8401008	IIC1MYAD	I2C slave address register 1	32	0x00000000	32
0xD840100C	IIC1CLK	I2C clock register 1	32	0x00000000	32
0xD8401010	IIC1BRST	I2C bus reset register 1	32	0x00000001	32
0xD8401014	IIC1BSTS	I2C bus status register 1	32	Undefined	32

Note: Please write to all registers through 32 bits.

19.3.1. I2C Transmit Data Register

Symbol	IICnDTRM
Address	IIC0DTRM : 0xD8400000 IIC1DTRM : 0xD8401000
Purpose	Used to write transmit data and to control.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	reserved					STA	STO	ACK
Initial value	0					0	0	0
R/W	R					RW	RW	RW
Bit	7	6	5	4	3	2	1	0
Bit name	DATA[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
31-11	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
10	STA	Start Control Bit Controls I2C bus transfer start. (See table below.)
9	STO	Stop Control Bit Controls I2C bus transfer stop. (See table below.)
8	ACK	Acknowledge Control Controls the acknowledge output of each byte. 0 : Outputs an acknowledge signal 1 : Does not output an acknowledge signal An acknowledge is output regardless of the value of the ACK bit if the device address matches or if the address is a global call address (address 0).
7-0	DATA[7:0]	Transmit data Writes transmit data.

A command for 12C controller is determined by the value written in STA.STO and the current mode of the 12C controller. The determined commands and operations as the result of executing the commands are shown below.

Mode	STA	STO	Command	Operation
All	0	0	NOP	No mode change
Slave receive	1	0	START	Data is sent by using IICnDTRAM.DATA[7:1] as the target slave address. A transit is made to master transmit mode when IICnDTRM.DATA[0] (R/W) = 0. A transit is made to master receive mode when IICnDTRM.DATA[0] (R/W) = 1.
Slave transmit	1	0	REPEAT START	Same as above.
Master receive	0	1	STOP READ	Stop conditions are transmitted and a transit is made to slave receive mode.
Master transmit	0	1	STOP WRITE	Same as above.
All	1	1	None	No mode change

Do not use any combination of the current mode and the STA.STO value that is not described above.

19.3.2. I2C Receive Data Register

Symbol IICNDREC
Address IIC0DREC : 0xD8400004
IIC1DREC : 0xD8401004
Purpose Used to read receive data and status.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	reserved	MODE[1:0]		STS	LRB	AAS	LAB	BB
Initial value	0	00		0	1	0	0	1
R/W	R	R		R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Bit name	DATA[7:0]							
Initial value	FF							
R/W	R							

Bit	Bit name	Description
31-15	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
14-13	MODE[1:0]	Device Mode Indicates the current mode. 00 : Slave receive (SLV/REC) 01 : Slave transmit (SLV/TRM) 10 : Master receive (MST/REC)

Bit	Bit name	Description
		11 : Master transmit (MST/TRM)
12	STS	Stop Condition Flag 0 : No stop condition 1 : Stop condition present
11	LRB	Last Receive Data Bit This bit maintains the last serial receive data. It is usually used to confirm acknowledge cycle data. 0 : Acknowledge present 1 : No acknowledge
10	AAS	Slave Bit "1" is set when the slave address matches the address register or when the general call address (address 0) is used. This bit is cleared when the receive register is read.
9	LAB	Lost Arbitration Flag This flag is set when arbitration fails. This bit is cleared when the start condition is set to IICnDTRM.
8	BB	Bus Busy Flag Indicates whether or not the I2C controller is busy. 0 : Busy 1 : Idle This flag is cleared to "0" in the start conditions and it is set to "1" in the stop conditions.
7-0	DATA[7:0]	Receive data

19.3.3. I2C Slave Address Register

Symbol	IICNMYAD
Address	IIC0MYAD : 0xD8400008 IIC1MYAD : 0xD8401008
Purpose	Used to set the slave address of the I2C controller.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	reserved	ADD[6:0]						
Initial value	0	0						
R/W	R	RW						

Bit	Bit name	Description
31-7	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
6-0	ADD[6:0]	Slave Address These set its own address in the slave mode.

19.3.4. I2C Clock Register

Symbol	IICNCLK
Address	IIC0CLK : 0xD840000C IIC1CLK : 0xD840100C
Purpose	Used to set the output clock frequency at the master operations.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	CLK[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
31-8	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
7-0	CLK[7:0]	Clock Frequency This sets the output clock during master operations and can calculate the output clock frequency by using the below formula. Output Clock Frequency = IOCLK Frequency [KHz]/2(CLK[7:0]+1)[KHz] This is used for frequency division ratio, and the operations can be guaranteed in the design up tp 500KHz.

19.3.5. I2C Bus Reset Register

Symbol	IICNBRST
Address	IIC0BRST : 0xD8400010 IIC1BRST : 0xD8401010
Purpose	Resets the I2C bus.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	reserved							BRST
Initial value	0							1
R/W	R							RW

Bit	Bit name	Description
31-1	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
0	BRST	I2C Bus Reset Performs control reset for the I2C bus. 0 : Reset 1 : Reset released The bus can be reset only when it is in the idle mode (IICnDREC.BB = 1) during the slave receive mode (IICnDREC.MODE[1:0] = 00). The I2C controller registers are not reset by a bus reset.

19.3.6. I2C Bus Status Register

Symbol	IICNBSTS
Address	IIC0BSTS : 0xD8400014 IIC1BSTS : 0xD8401014
Purpose	Used to read the I2C bus status.

Bit	31	30	29	28	27	26	25	24
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	23	22	21	20	19	18	17	16
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	reserved						SDA	SCL
Initial value	0						Undefined	Undefined
R/W	R						R	R

Bit	Bit name	Description
31-2	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
1	SDA	Data Pin Status reads the status of the SDA (data) pin.
0	SCL	Clock Pin Status reads the status of the SCL (clock) pin.

19.4. Description of Operation

19.4.1. Master Transmit

The procedure for performing master transmit is given below.

(1) Write the followings into the IICnDTRM register; the target slave address and the command data to enter master transmit mode
    STA = 1, STO = 0, ACK = 1 (START command)
    DATA[7:1] = slave address (7 bits), DATA[0] (R/W) = 0 (master transmit)
    This will output the slave address on the I2C bus.

(2) A communication end interrupt is generated when an acknowledgement is returned from the slave device. Read the IICnDREC register and check that LRB = 0 and LAB = 0 (that there was an acknowledgement from the slave).

(3) Write the transmit data to the IICnDTRM register.
    STA = 0, STO = 0, ACK = 1 (NOP command)
    DATA[7:0] = transmit data
    Subsequently, the necessary amount of data will be transmitted by repeating steps (2) and (3).

(4) After all data has been transmitted, write the stop command (STA = 0, STO = 1, ACK = 1) to the IICnDTRM register.
    STA = 0, STO = 1, ACK = 1 (STOP command)
    DATA[7:0] = any data
    The I2C controller will output a STOP condition and enter a slave receive mode.

19.4.2. Master Receive

The procedure for performing master receive is given below.

(1) Write the followings into the IICnDTRM register; the target slave address and the command data for entering master transmit mode.
    STA = 1, STO = 0, ACK = 1 (START command)
    DATA[7:1] = slave address (7 bits), DATA[0] (R/W) = 1 (master receive)
    This will output the slave address on the I2C bus.

(2) A communication end interrupt is generated when an acknowledgement is returned from the slave device. Read the IICnDREC register and check that LRB = 0 and LAB = 0 (that there was an acknowledgement from the slave).

(3) Write the acknowledgment setting to the IICnDTRM register.
    STA = 0, STO = 0, ACK = 0 (NOP command, enable ACK generation)
    DATA[7:0] = any data

(4) An interrupt will be generated with the output of an acknowledgement that data from the slave device is to be received. Read the IICnDREC register and check that LAB = 0 (that data has been received from the slave). At the same time, read the receive data DATA[7:0] of the IICnDREC register.
    Subsequently, the necessary amount of data is received by repeating steps (3) and (4).
    Execute step (3) when software is ready to receive the next data. (SCL (clock) output to the I2C bus will start and the next data from the slave will be transmitted.)

(5) , For finishing the data receive operation, write the stop command (STA = 0, STO =1, ACK =1) to the IICnDTRM register after the last data has been received.
    STA = 0, STO = 1, ACK = 1 (STOP command)
    DATA[7:0] = any data
    The I2C controller will output a STOP condition and enter a slave receive mode.

19.4.3. Slave Transmit

The procedure for performing slave transmit is given below.

(1) The I2C controller returns an acknowledgement to the master and sends an interrupt to the CPU when the address sent from the master matches the value of the IICnMYAD register or when the general call address (address 0) has been used.
(2) Read the IICnDREC register and check that AAS = 1 and that the LSB (DATA[0]) of the receive data is "1" (slave transmit).
(3) Write the transmit data to the IICnDTRM register.
DATA[7:0] = transmit data
(4) An interrupt is generated when the transmission is complete. Repeat steps (3) and (4) when the IICnDREC register is read and LRB = 0 (when there is an acknowledgement from the master). The transmission has ended if LRB =1.

19.4.4. Slave Receive

The procedure for performing slave receive is given below.

(1) The I2C controller returns an acknowledgement to the master and sends an interrupt to the CPU when the address sent from the master matches the value of the IICnMYAD register or when the general call address (address 0) has been used.
(2) Read the IICnDREC register and check that AAS=1 and that the LSB (DATA[0]) of the receive data is "0" (slave receive).
(3) Set the ACK bit of the IICnDTRM register to "0" and make settings to return an acknowledgement to the master.
(4) The I2C controller returns an acknowledgement and sends an interrupt to the CPU each time data is received. Repeat steps (3) and (4) when STS=0 in the IICnDREC register. Execute step (3) when software is ready to receive the next data. (Data transmit start from the master is prompted by canceling the low output of SCL (clock) on the I2C bus.) Set the ACK bit of the IICnDTRM register to "1" and end the reception when STS = 1 (STOP condition present).

19.4.5. Interrupt Causes

An interrupt is sent to the CPU for any of the following three reasons.

- Communications end interrupt
  A communications end interrupt is generated when the data receive and transmit in the master/slave mode has ended.
- Bus acquisition failure interrupt
  A bus acquisition failure interrupt is generated when the bus cannot be secured in the master mode.
  The LAB bit of the IICnDREC register is set to "1" when this interrupt has been generated.
- Stop condition detected interrupt (slave receive mode)
  A stop condition detected interrupt is generated in the slave receive mode when the stop conditions have been detected.
  The STS bit of the IICnDREC register is set to "1" when this interrupt has been generated.
- Slave address match interrupt
  A slave address match interrupt is generated in the slave mode when the slave address matches (or when the general call address is included). When this interrupt is generated, the AAS bit of the IICnDREC register is set to "1".

The relationship between the IICnDREC register bits and interrupts types is shown in the table below.

Table 96 Relationship between the IICnDREC register bits and interrupts types

Interrupts	MODE[1:0]	STS	LAB	AAS
Master communications end interrupt	1x	x	0	x
Bus acquisition failure interrupt	1x	x	1	x
Stop condition detected interrupt	00	1	x	0
Slave address match interrupt	0x	x	x	1

Chapter 20 I/O Ports (PIO)

20

CHAPTER 20

I/O Ports (PIO)

20.1. General

This LSI incorporates six I/O ports from Port 0 to Port 5. These ports can all be accessed by a program as an internal I/O memory space.

20.2. Pin Configuration

20.2.1. I/O Port 0

Port 0 is an 8-bit I/O port. Port 0 is also used for timer clock I/O (TM0IO-TM6IOB) and test signal output (EYECLK, EYED) for the analog front end interface.
The mode of bit n is switched by setting P0MDn [1:0] in the P0MD register.

Table 97 Configuration of port 0

P0MDn[1:0]	00	01	10	11	Initial status
PIO0[0]	P0IN[0]	P0OUT[0]	TM0IO	EYECLK	P0IN[0]
PIO0[1]	P0IN[1]	P0OUT[1]	TM1IO	EYED	P0IN[1]
PIO0[2]	P0IN[2]	P0OUT[2]	TM2IO	-	P0IN[2]
PIO0[3]	P0IN[3]	P0OUT[3]	TM3IO	-	P0IN[3]
PIO0[4]	P0IN[4]	P0OUT[4]	TM4IO	XCTS	P0IN[4]
PIO0[5]	P0IN[5]	P0OUT[5]	TM5IO	-	P0IN[5]
PIO0[6]	P0IN[6]	P0OUT[6]	TM6IOA	-	P0IN[6]
PIO0[7]	P0IN[7]	P0OUT[7]	TM6IOB	-	P0IN[7]

20.2.2. I/O Port 1

Port 1 is an 5-bit I/O port. Port 1 is also used for timer clock I/O, A/D conversion trigger input (ADTRG), DMA request input (XDMR[1:0]), and CPU clock multiplier setting input (FRQS[1:0]).
The mode of bit n is switched by setting P1MDn [1:0] in the P1MD register.

Table 98 Configuration of port 1

P1MDn[1:0]	00	01	10	11	Initial status
PIO1[0]	P1IN[0]	P1OUT[0]	TM7IO	ADTRG	P1IN[0]
PIO1[1]	P1IN[1]	P1OUT[1]	TM8IO	XDMR[0]	P1IN[1]
PIO1[2]	P1IN[2]	P1OUT[2]	TM9IO	XDMR[1]	P1IN[2]
PIO1[3]	P1IN[3]	P1OUT[3]	TM10IO	FRQS[0]	FRQS[0]
PIO1[4]	P1IN[4]	P1OUT[4]	TM11IO	FRQS[1]	FRQS[1]

20.2.3. I/O Port 2

Port 2 is a 5-bit I/O port. Port 2 is also used for timer clock I/O, DRAM CAS signal (XSCAS[3:0]) and for initial settings pins at time of reset (BOOTBW, BOOTSEL, CMOD, CKIO).
The mode of bit n is switched by setting P2MDn [1:0] in the P2MD register.

Table 99 Configuration of port 2

P2MDn[1:0]	00	01	10	11	Initial status
PIO2[0]	P2IN[0]	P2OUT[0]	-	BOOTBW	BOOTBW
PIO2[1]	P2IN[1]	P2OUT[1]	-	BOOTSEL	BOOTSEL
PIO2[2]	P2IN[2]	P2OUT[2]	-	-	P2IN[2]
PIO2[3]	P2IN[3]	P2OUT[3]	-	CKIO	CKIO
PIO2[4]	P2IN[4]	P2OUT[4]	-	CMOD	CMOD

20.2.4. I/O Port 3

Port 3 is a 5-bit I/O port. Port 3 is also used for analog front end interface pins (AFRXD, AFTXD, AFSCLK, AFFS, and AFEHC).
The mode of bit n is switched by setting P3MDn [1:0] in the P3MD register.

Table 100 Configuration of port 3

P3MDn[1:0]	00	01	10	11	Initial status
PIO3[0]	P3IN[0]	P3OUT[0]	AFRXD	-	P3IN[0]
PIO3[1]	P3IN[1]	P3OUT[1]	AFTXD	-	P3IN[1]
PIO3[2]	P3IN[2]	P3OUT[2]	AFSCLK	-	P3IN[2]
PIO3[3]	P3IN[3]	P3OUT[3]	AFFS	-	P3IN[3]
PIO3[4]	P3IN[4]	P3OUT[4]	AFEHC	-	P3IN[4]

20.2.5. I/O Port 4

Port 4 is a 8-bit I/O port. PIO4 [3:0] is also used for I2C controller pins (SCL[1:0], SDA[1:0]). PIO4 [3:0] uses open drain output pins. PIO4 [7:4] is also used for a serial port.
The mode of bit n is switched by setting P4MDn [1:0] in the P4MD register.

Table 101 Configuration of port 4

P4MDn[1:0]	00	01	10	11	Initial status
PIO4[0]	P4IN[0]	P4OUT[0]	SCL[0]	-	P4IN[0]
PIO4[1]	P4IN[1]	P4OUT[1]	SDA[0]	-	P4IN[1]
PIO4[2]	P4IN[2]	P4OUT[2]	SCL[0]	-	P4IN[2]
PIO4[3]	P4IN[3]	P4OUT[3]	SDA[1]	-	P4IN[3]
PIO4[4]	P4IN[4]	P4OUT[4]	SBO[0]	-	SBO[0]
PIO4[5]	P4IN[5]	P4OUT[5]	SBO[1]	-	SBO[1]
PIO4[6]	P4IN[6]	P4OUT[6]	SBT[0]	-	SBT[0]
PIO4[7]	P4IN[7]	P4OUT[7]	SBT[1]	-	SBT[1]

20.2.6. I/O Port 5

Port 5 is a 3-bit I/O port. Port 5 is also used for IrDA controller pins (IRTXD, IRRXDS, and IRRXDF).
The mode of bit n is switched by setting P5MDn [1:0] in the P5MD register.

Table 102 Configuration of port 5

P5MDn[1:0]	00	01	10	11	Initial status
PIO5[0]	P5IN[0]	P5OUT[0]	IRTXD	SOUT	P5IN[0]
PIO5[1]	P5IN[1]	P5OUT[1]	IRRXDS	SIN	P5IN[1]
PIO5[2]	P5IN[2]	P5OUT[2]	IRRXDF	-	P5IN[2]

20.3. Registers

Table 103 I/O port register

Address	Symbol	Name	Number of bits	Initial value	Access size
0xDB000000	P0MD	Port 0 mode register	16	0x0000	8, 16
0xDB000004	P0IN	Port 0 pin register	8	Undefined	8
0xDB000008	P0OUT	Port 0 output register	8	0x00	8
0xDB00000C	P0TMIO	Port 0TM pin I/O control register	8	0x00	8
0xDB000100	P1MD	Port 1 mode register	16	0x03C0	8, 16
0xDB000104	P1IN	Port 1 pin register	8	Undefined	8
0xDB000108	P1OUT	Port 1 output register	8	0x00	8
0xDB00010C	P1TMIO	Port 1TM pin I/O control register	8	0x00	8
0xDB000200	P2MD	Port 2 mode register	16	0x00FF	8, 16
0xDB000204	P2IN	Port 2 pin register	8	Undefined	8
0xDB000208	P2OUT	Port 2 output register	8	0x00	8
0xDB000300	P3MD	Port 3 mode register	16	0x0000	8, 16
0xDB000304	P3IN	Port 3 pin register	8	Undefined	8
0xDB000308	P3OUT	Port 3 output register	8	0x00	8
0xDB000400	P4MD	Port 4 mode register	16	0x AA00	8, 16
0xDB000404	P4IN	Port 4 pin register	8	Undefined	8
0xDB000408	P4OUT	Port 4 output register	8	0x00	8
0xDB000500	P5MD	Port 5 mode register	16	0x0000	8, 16
0xDB000504	P5IN	Port 5 pin register	8	Undefined	8
0xDB000508	P5OUT	Port 5 output register	8	0x00	8

20.3.1. Port 0 Mode Register

Symbol	P0MD
Address	0xDB000000
Purpose	Sets the mode of each pin of Port 0.

Bit	15	14	13	12	11	10	9	8
Bit name	P0MD7[1:0]		P0MD6[1:0]		P0MD5[1:0]		P0MD4[1:0]	
Initial value	00		00		00		00	
R/W	RW		RW		RW		RW	
Bit	7	6	5	4	3	2	1	0
Bit name	P0MD3[1:0]		P0MD2[1:0]		P0MD1[1:0]		P0MD0[1:0]	
Initial value	00		00		00		00	
R/W	RW		RW		RW		RW	

Bit	Bit name	Description
15-14	P0MD7[1:0]	PIO0[7] pin mode setting Sets the mode of the PIO0[7] pin. 00 : Functions as an input I/O port (values can be read from P0IN[7]) 01 : Functions as an output I/O port (values can be written to P0OUT[7]) 10 : Functions as a TM6B pin 11 : Setting prohibited
13-12	P0MD6[1:0]	PIO0[6] pin mode setting Sets the mode of the PIO0[6] pin. 00 : Functions as an input I/O port (values can be read from P0IN[6]) 01 : Functions as an output I/O port (values can be written to P0OUT[6]) 10 : Functions as a TM6A pin 11 : Setting prohibited
11-10	P0MD5[1:0]	PIO0[5] pin mode setting Sets the mode of the PIO0[5] pin. 00 : Functions as an input I/O port (values can be read from P0IN[5]) 01 : Functions as an output I/O port (values can be written to P0OUT[5]) 10 : Functions as a TM5 pin 11 : Setting prohibited
9-8	P0MD4[1:0]	PIO0[4] pin mode setting Sets the mode of the PIO0[4] pin. 00 : Functions as an input I/O port (values can be read from P0IN[4]) 01 : Functions as an output I/O port (values can be written to P0OUT[4]) 10 : Functions as a TM4 pin 11 : Functions as an XCTS pin
7-6	P0MD3[1:0]	PIO0[3] pin mode setting Sets the mode of the PIO0[3] pin.

Bit	Bit name	Description
		00 : Functions as an input I/O port (values can be read from P0IN[3]) 01 : Functions as an output I/O port (values can be written to P0OUT[3]) 10 : Functions as a TM3 pin 11 : Setting prohibited
5-4	P0MD2[1:0]	PIO0[2] pin mode setting Sets the mode of the PIO0[2] pin. 00 : Functions as an input I/O port (values can be read from P0IN[2]) 01 : Functions as an output I/O port (valuescan be written to P0OUT[2]) 10 : Functions as a TM2 pin 11 : Setting prohibited
3-2	P0MD1[1:0]	PIO0[1] pin mode setting Sets the mode of the PIO0[1] pin. 00 : Functions as an input I/O port (values can be read from P0IN[1]) 01 : Functions as an output I/O port (values can be written to P0OUT[1]) 10 : Functions as a TM1 pin 11 : Functions as an EYED pin
1-0	P0MD0[1:0]	PIO0[0] pin mode setting Sets the mode of the PIO0[0] pin. 00 : Functions as an input I/O port (values can be read from P0IN[0]) 01 : Functions as an output I/O port (values can be written to P0OUT[0]) 10 : Functions as a TM0 pin 11 : Functions as an EYECLK pin

20.3.2. Port 0 Pin Register

Symbol	P0IN
Address	0xDB000004
Purpose	reads the value of each pin of Port 0.

Bit	7	6	5	4	3	2	1	0
Bit name	P0IN[7:0]							
Initial value	Note 1							
R/W	R							

Note 1: This reflects the pin status.

Bit	Bit name	Description
7-0	P0IN[7:0]	PIO0 input data Read the value of the PIO0[7:0] pin.

20.3.3. Port 0 Output Register

Symbol	P0OUT
Address	0xDB000008
Purpose	sets the value to be output to each pin of Port 0.

Bit	7	6	5	4	3	2	1	0
Bit name	P0OUT[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
7-0	P0OUT[7:0]	PIO0 output data Writes the value to be output into the PIO0[7:0] pin. (This register reads the written values to the register and may not reflects a value of PIO0[7:0] pin.)

20.3.4. Port 0 Timer Pin Input/Output Control Register

Symbol	P0TMIO
Address	0xDB00000C
Purpose	Sets the clock input/output direction when each pin of port 0 is defined as a timer input/output function.

Bit	7	6	5	4	3	2	1	0
Bit name	P0TMIO7	P0TMIO6	P0TMIO5	P0TMIO4	P0TMIO3	P0TMIO2	P0TMIO1	P0TMIO0
Initial value	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit name	Description
7	P0TMIO7	TM6IOB Pin Input/Output Setting Sets the timer clock input/output status when the P0MD7[1:0] bits of P0MD register is set to “10” and functions as TM6IOB pin. 0: Input 1: Output
6	P0TMIO6	TM6IOA Pin Input/Output Setting Sets the timer clock input/output status when the P0MD6[1:0] bits of P0MD register is set to “10” and functions as TM6IOA pin. 0: Input 1: Output
5	P0TMIO5	TM5IO Pin Input/Output Setting Sets the timer clock input/output status when the P0MD5[1:0] bits of P0MD register is set to “10” and functions as TM5IO pin. 0: Input 1: Output
4	P0TMIO4	TM4IO Pin Input/Output Setting Sets the timer clock input/output status when the P0MD4[1:0] bits of P0MD register is set to “10” and functions as TM4IO pin. 0: Input 1: Output
3	P0TMIO3	TM3IO Pin Input/Output Setting Sets the timer clock input/output status when the P0MD3[1:0] bits of P0MD register is set to “10” and functions as TM3IO pin. 0: Input 1: Output
2	P0TMIO2	TM2IO Pin Input/Output Setting Sets the timer clock input/output status when the P0MD2[1:0] bits of P0MD register is set to “10”, the pin functions as TM2IO pin. 0: Input 1: Output
1	P0TMIO1	TM1IO Pin Input/Output Setting Sets the timer clock input/output status when the P0MD1[1:0] bits of P0MD register is set to “10” and functions as TM1IO pin. 0: Input 1: Output
0	P0TMIO0	TM0IO Pin Input/Output Setting Sets the timer clock input/output status when the P0MD0[1:0] bits of

Bit	Bit name	Description
		P0MD register is set to “10” and functions as TM0IO pin. 0: Input 1: Output

20.3.5. Port 1 Mode Register

Symbol	P1MD
Address	0xDB000100
Purpose	Sets the mode of each pin of Port 1.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved						P1MD4[1:0]	
Initial value	0						11	
R/W	R						RW	
Bit	7	6	5	4	3	2	1	0
Bit name	P1MD3[1:0]		P1MD2[1:0]		P1MD1[1:0]		P1MD0[1:0]	
Initial value	11		00		00		00	
R/W	RW		RW		RW		RW	

Bit	Bit name	Description
15-10	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
9-8	P1MD4[1:0]	PIO1[4] pin mode setting Sets the mode of the PIO1[4] pin. 00 : Functions as an input I/O port (values can be read from P1IN[4]) 01 : Functions as an output I/O port (values can be written to P1OUT[4]) 10 : Functions as a TM11 pin 11 : Functions as an FRQS[1] pin
7-6	P1MD3[1:0]	PIO1[3] pin mode setting Sets the mode of the PIO1[3] pin. 00 : Functions as an input I/O port (values can be read from P1IN[3]) 01 : Functions as an output I/O port (values can be written to P1OUT[3]) 10 : Functions as a TM10 pin 11 : Functions as an FRQS[0] pin
5-4	P1MD2[1:0]	PIO1[2] pin mode setting Sets the mode of the PIO1[2] pin. 00 : Functions as an input I/O port (values can be read from P1IN[2]) 01 : Functions as an output I/O port (values can be written to P1OUT[2]) 10 : Functions as a TM9 pin 11 : Functions as an XDMR[1] pin
3-2	P1MD1[1:0]	PIO1[1] pin mode setting Sets the mode of the PIO10[1] pin. 00 : Functions as an input I/O port (values can be read from P1IN[1])

Bit	Bit name	Description
		01 : Functions as an output I/O port (values can be written to P1OUT[1])
		10 : Functions as a TM8 pin
		11 : Functions as an XDMR[0] pin
1-0	P1MD0[1:0]	PIO1[0] pin mode setting Sets the mode of the PIO1[0] pin. 00 : Functions as an input I/O port (values can be read from P1IN[0]) 01 : Functions as an output I/O port (values can be written to P1OUT[0]) 10 : Functions as a TM7 pin 11 : Functions as an ADTRG pin

20.3.6. Port 1 Pin Register

Symbol	P1IN
Address	0xDB000104
Purpose	Reads the value of each pin of Port 1.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			P1IN[4:0]				
Initial value	0			x				
R/W	R			R				

Bit	Bit name	Description
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4-0	P1IN[4:0]	PIO1 input data Reads the value of the PIO1[4:0] pin.

20.3.7. Port 1 Output Register

Symbol	P1OUT
Address	0xDB000108
Purpose	Sets the value to be output to each pin of Port 1.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			P1OUT[4:0]				
Initial value	0			0				
R/W	R			RW				

Bit	Bit name	Description
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4-0	P1OUT[4:0]	PIO0 output data Writes the value to be output to the PIO1[7:0] pin.

20.3.8. Port 1 Timer Input/Output Control Register

Symbol	P1TMIO
Address	0xDB00010C
Purpose	Sets the input/output when each pin of Port 1 is defined for the timer input/output functions.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			P1TMIO4	P1TMIO3	P1TMIO2	P1TMIO1	P1TMIO0
Initial value	0			0	0	0	0	0
R/W	R			RW	RW	RW	RW	RW

Bit	Bit name	Description
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4	P1TMIO4	TM7IO Pin Input/Output Setting This sets the timer clock input/output status when the P1MD4[1:0] bits of P1MD register is set to “10” and functions as TM7IO pin. 0: Input 1: Output
3	P1TMIO3	TM8IO Pin Input/Output Setting This sets the timer clock input/output status when the P1MD3[1:0] bits of P1MD register is set to “10” and functions as TM8IO pin. 0: Input 1: Output
2	P1TMIO2	TM9IO Pin Input/Output Setting This sets the timer clock input/output status when the P1MD2[1:0] bits of P1MD register is set to “10” and functions as TM9IO pin. 0: Input 1: Output
1	P1TMIO1	TM10IO Pin Input/Output Setting This sets the timer clock input/output status when the P1MD1[1:0] bits of P1MD register is set to “10” and functions as TM10IO pin. 0: Input 1: Output
0	P1TMIO0	TM11IO Pin Input/Output Setting This sets the timer clock input/output status when the P1MD0[1:0] bits of P1MD register is set to “10” and functions as TM11IO pin. 0: Input 1: Output

20.3.9. Port 2 Mode Register

Symbol	P2MD
Address	0xDB000200
Purpose	Sets the mode of each pin of Port 2.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved						P2MD4[1:0]	
Initial value	0						11	
R/W	R						RW	
Bit	7	6	5	4	3	2	1	0
Bit name	P2MD3[1:0]		P2MD2[1:0]		P2MD1[1:0]		P2MD0[s1:0]	
Initial value	11		00		11		11	
R/W	RW		RW		RW		RW	

Bit	Bit name	Description
15-10	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
9-8	P2MD4[1:0]	PIO2[4] pin mode setting Sets the mode of the PIO2[4] pin. 00 : Functions as an input I/O port (values can be read from P2IN[4]) 01 : Functions as an output I/O port (values can be written to P2OUT[4]) 10 : Setting prohibited 11 : Functions as a CMOD pin
7-6	P2MD3[1:0]	PIO2[3] pin mode setting Sets the mode of the PIO2[3] pin. 00 : Functions as an input I/O port (values can be read from P2IN[3]) 01 : Functions as an output I/O port (values can be written to P2OUT[3]) 10 : Setting prohibited 11 : Functions as a CKIO pin
5-4	P2MD2[1:0]	PIO2[2] pin mode setting Sets the mode of the PIO2[2] pin. 00 : Functions as an input I/O port (values can be read from P2IN[2]) 01 : Functions as an output I/O port (values can be written to P2OUT[2]) 10 : Setting prohibited 11 : Setting prohibited
3-2	P2MD1[1:0]	PIO2[1] pin mode setting Sets the mode of the PIO2[1] pin. 00 : Functions as an input I/O port (values can be read from P2IN[1]) 01 : Functions as an output I/O port (values can be written to P2OUT[1]) 10 : Setting prohibited 11 : Functions as a BOOTSEL pin

Bit	Bit name	Description
1-0	P2MD0[1:0]	PIO2[0] pin mode setting Sets the mode of the PIO2[0] pin. 00 : Functions as an input I/O port (values can be read from P2IN[0]) 01 : Functions as an output I/O port (values can be written to P2OUT[0]) 10 : Setting prohibited 11 : Functions as a BOOTBW

20.3.10. Port 2 Pin Register

Symbol	P2IN
Address	0xDB000204
Purpose	Reads the value of each pin of Port 2.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			P2IN[4:0]				
Initial value	0			x				
R/W	R			R				

Bit	Bit name	Description
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4-0	P2IN[4:0]	PIO2 input data Reads the value of the PIO2[4:0] pin.

20.3.11. Port 2 Output Register

Symbol	P2OUT
Address	0xDB000208
Purpose	Sets the value to be output to each pin of Port 2.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			P2OUT[4:0]				
Initial value	0			0				
R/W	R			RW				

Bit	Bit name	Description
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4-0	P2OUT[4:0]	PIO2 output data Writes the value to be output to the PIO2[4:0] pin.

20.3.12. Port 3 Mode Register

Symbol	P3MD
Address	0xDB000300
Purpose	Sets the mode of each pin of Port 3.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved						P3MD4[1:0]	
Initial value	0						00	
R/W	R						RW	
Bit	7	6	5	4	3	2	1	0
Bit name	P3MD3[1:0]		P3MD2[1:0]		P3MD1[1:0]		P3MD0[1:0]	
Initial value	00		00		00		00	
R/W	RW		RW		RW		RW	

Bit	Bit name	Description
15-10	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
9-8	P3MD4[1:0]	PIO3[4] pin mode setting Sets the mode of the PIO3[4] pin. 00 : Functions as an input I/O port (values can be read from P3IN[4]) 01 : Functions as an output I/O port (values can be written to P3OUT[4]) 10 : Functions as an AFEHC pin 11 : Setting prohibited
7-6	P3MD3[1:0]	PIO3[3] pin mode setting Sets the mode of the PIO3[3] pin. 00 : Functions as an input I/O port (values can be read from P3IN[3]) 01 : Functions as an output I/O port (values can be written to P3OUT[3]) 10 : Functions as an AFFS pin 11 : Setting prohibited
5-4	P3MD2[1:0]	PIO2[2] pin mode setting Sets the mode of the PIO2[2] pin. 00 : Functions as an input I/O port (values can be read from P2IN[2]) 01 : Functions as an output I/O port (values can be written to P2OUT[2]) 10 : Functions as an AFSCLK pin 11 : Setting prohibited
3-2	P3MD1[1:0]	PIO3[1] pin mode setting Sets the mode of the PIO3[1] pin. 00 : Functions as an input I/O port (values can be read from P3IN[1]) 01 : Functions as an output I/O port (values can be written to P3OUT[1]) 10 : Functions as an AFTXD pin 11 : Setting prohibited

Bit	Bit name	Description
1-0	P3MD0[1:0]	PIO3[0] pin mode setting Sets the mode of the PIO3[0] pin. 00 : Functions as an input I/O port (values can be read from P3IN[0]) 01 : Functions as an output I/O port (values can be written to P3OUT[0]) 10 : Functions as an AFRXD pin 11 : Setting prohibited

20.3.13. Port 3 Pin Register

Symbol P3IN
Address 0xDB000304
Purpose Used to read the value of each pin of Port 3.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			P3IN[4:0]				
Initial value	0			x				
R/W	R			R				

Note 1: Refer to the description below

Bit	Bit name	Description
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4-0	P3IN[4:0]	PIO3 input data Reads the value of the PIO3[4:0] pin.

20.3.14. Port 3 Output Register

Symbol P3OUT
Address 0xDB000308
Purpose Used to set the value to be output to each pin of Port 3.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved			P3OUT[4:0]				
Initial value	0			0				
R/W	R			RW				

Bit	Bit name	Description
7-5	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
4-0	P3OUT[4:0]	PIO3 output data Writes the value to be output to the PIO3[4:0] pin.

20.3.15. Port 4 Mode Register

Symbol	P4MD
Address	0xDB000400
Purpose	Sets the mode of each pin of Port 4.

Bit	15	14	13	12	11	10	9	8
Bit name	P4MD7[1:0]		P4MD6[1:0]		P4MD5[1:0]		P4MD4[1:0]	
Initial value	10		10		10		10	
R/W	RW		RW		RW		RW	
Bit	7	6	5	4	3	2	1	0
Bit name	P4MD3[1:0]		P4MD2[1:0]		P4MD1[1:0]		P4MD0[1:0]	
Initial value	00		00		00		00	
R/W	RW		RW		RW		RW	

Bit	Bit name	Description
15-14	P4MD7[1:0]	PIO4[7] pin mode setting Sets the mode of the PIO4[7] pin. 00 : Functions as an input I/O port (values can be read from P4IN[7]) 01 : Functions as an output I/O port (values can be written to P4OUT[7]) 10 : Functions as an SBT[1] pin 11 : Setting prohibited
13-12	P4MD6[1:0]	PIO4[6] pin mode setting Sets the mode of the PIO4[6] pin. 00 : Functions as an input I/O port (values can be read from P4IN[6]) 01 : Functions as an output I/O port (values can be written to P4OUT[6]) 10 : Functions as an SBT[0] pin 11 : Setting prohibited
11-10	P4MD5[1:0]	PIO4[5] pin mode setting Sets the mode of the PIO4[5] pin. 00 : Functions as an input I/O port (values can be read from P4IN[5]) 01 : Functions as an output I/O port (values can be written to P4OUT[5]) 10 : Functions as an SBO[1] pin 11 : Setting prohibited
9-8	P4MD4[1:0]	PIO4[4] pin mode setting Sets the mode of the PIO4[4] pin. 00 : Functions as an input I/O port (values can be read from P4IN[4]) 01 : Functions as an output I/O port (values can be written to P4OUT[4]) 10 : Functions as an SBO[0] pin 11 : Setting prohibited
7-6	P4MD3[1:0]	PIO4[3] pin mode setting Sets the mode of the PIO4[3] pin.

Bit	Bit name	Description
		00 : Functions as an input I/O port (values can be read from P4IN[3]) 01 : Functions as an output I/O port (values can be written to P4OUT[3]) 10 : Functions as an SDA[1] pin 11 : Setting prohibited
5-4	P4MD2[1:0]	PIO4[2] pin mode setting Sets the mode of the PIO4[2] pin. 00 : Functions as an input I/O port (values can be read from P4IN[2]) 01 : Functions as an output I/O port (values can be written to P4OUT[2]) 10 : Functions as an SCL[1] pin 11 : Setting prohibited
3-2	P4MD1[1:0]	PIO4[1] pin mode setting Sets the mode of the PIO4[1] pin. 00 : Functions as an input I/O port (values can be read from P4IN[1]) 01 : Functions as an output I/O port (values can be written to P4OUT[1]) 10 : Functions as an SDA[0] pin 11 : Setting prohibited
1-0	P4MD0[1:0]	PIO4[0] pin mode setting Sets the mode of the PIO4[0] pin. 00 : Functions as an input I/O port (values can be read from P4IN[0]) 01 : Functions as an output I/O port (values can be written to P4OUT[0]) 10 : Functions as an SCL[0] pin 11 : Setting prohibited

20.3.16. Port 4 Pin Register

Symbol	P4IN
Address	0xDB000404
Purpose	Used to read the value of each pin of Port 4.

Bit	7	6	5	4	3	2	1	0
Bit name	P4IN[7:0]							
Initial value	-							
R/W	R							

Bit	Bit name	Description
7-0	P4IN[3:0]	PIO4 input data Reads the value of the PIO4[7:0] pin.

20.3.17. Port 4 Output Register

Symbol	P4OUT
Address	0XDB000408
Purpose	Used to set the value to be output to each pin of Port 4.

Bit	7	6	5	4	3	2	1	0
Bit name	P4OUT[7:0]							
Initial value	0							
R/W	RW							

Bit	Bit name	Description
7-0	P4OUT[7:0]	PIO4 output data Writes the value to be output to the PIO4[7:0] pin.

20.3.18. Port 5 Mode Register

Symbol P5MD
Address 0xDB000500
Purpose Sets the mode of each pin of Port 5.

Bit	15	14	13	12	11	10	9	8
Bit name	reserved							
Initial value	0							
R/W	R							
Bit	7	6	5	4	3	2	1	0
Bit name	reserved		P5MD2[1:0]		P5MD1[1:0]		P5MD0[1:0]	
Initial value	0		00		00		00	
R/W	R		RW		RW		RW	

Bit	Bit name	Description
15-6	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
5-4	P5MD2[1:0]	PIO5[2] pin mode setting Sets the mode of the PIO5[2] pin. 00 : Functions as an input I/O port (values can be read from P5IN[2]) 01 : Functions as an output I/O port (values can be written to P5OUT[2]) 10 : Functions as an XRRXDF pin 11 : Setting prohibited
3-2	P5MD1[1:0]	PIO5[1] pin mode setting Sets the mode of the PIO5[1] pin. 00 : Functions as an input I/O port (values can be read from P5IN[1]) 01 : Functions as an output I/O port (values can be written to P5OUT[1]) 10 : Functions as an IRRXDS pin 11 : Setting prohibited
1-0	P5MD0[1:0]	PIO5[0] pin mode setting Sets the mode of the PIO5[0] pin. 00 : Functions as an input I/O port (values can be read from P5IN[0]) 01 : Functions as an output I/O port (values can be written to P5OUT[0]) 10 : Functions as an IRTXD pin 11 : Functions as an SOUT pin

20.3.19. Port 5 Pin Register

Symbol	P5IN
Address	0xDB000504
Purpose	Used to read the value of each pin of Port 5.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved					P5IN[2:0]		
Initial value	0					x		
R/W	R					R		

Bit	Bit name	Description
7-3	reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2-0	P5IN[4:0]	PIO5 input data Reads the value of the PIO5[2:0] pin.

20.3.20. Port 5 Output Register

Symbol	P5OUT
Address	0XDB000508
Purpose	Used to set the value to be output to each pin of Port 5.

Bit	7	6	5	4	3	2	1	0
Bit name	reserved					P5OUT[2:0]		
Initial value	0					000		
R/W	R					RW		

Bit	Bit name	Description
7-3	Reserved	These are reserved bits. "0" is always returned when these bits are read. Always write a "0" to these bits.
2-0	P5OUT[2:0]	PIO5 output data Writes the value to be output to the PIO5[2:0] pin.

Chapter 21 Electrical Specifications

21

CHAPTER 21

Electrical Specifications

21.1. Absolute maximum ratings

The following table shows absolute maximum ratings.

Table 104 Absolute maximum ratings

Items		Symbols	Rating	Unit
A1	Power-supply voltage 1	V_{DD33}	-0.3-4.6	V
A2	Power-supply voltage 2	AV_{DD}	-0.3-4.6	V
A3	Power-supply voltage 3	PV_{DD}	-0.3-4.6	V
A4	Power-supply voltage 4	RV_{DD}	-0.3-4.6	V
A5	Power-supply voltage 5	V_{DD18}	-0.3-2.5	V
A6	Input voltage 1	V_{I1}	-0.3-VDD33+0.3(up to 4.6)	V
A7	Input voltage 2	V_{I2}	-0.3-RVDD+0.3(up to 4.6)	V
A8	Output voltage	V_O	-0.3-VDD33+0.3(up to 4.6)	V
A9	Average output current 1	I_{O1}	±12	mA
A10	Average output current 2	I_{O2}	±24	mA
A11	Acceptable loss	P_D	1.7	W
A12	Operation peripheral temperature	Topr	-20-70	℃
A13	Storage temperature	Tstg	-50-150	℃

(Note)

- The absolute maximum rating shows a limit value to cause no destruction regardless of the application to a chip, and it is not for guaranteeing the operations.
- All VDD33, AVDD, PVDD, RVDD, and VDD18 pins must be externally and directly connected to the power.
- All VSS, AVSS, PVSS pins must be externally and directly connected to the ground.
- One bypass condenser at least of over 0.1 μF must be inserted around the this chip between VDD33 and VSS pins, AVDD and AVSS pins, PVDD and PVSS pins, RVDD and VSS pins, and VDD18 and VSS pins.
- In this LSI, 3.3V must be provided for PAD, ADC, PLL and RTC, and 1.8V must be provided for an internal (digital) circuit. For the corresponding pins, refer to the configuration.
- When any power is not provided, the internal state of LSI becomes unstable. Although the order for turning on the powers is not defined, they mast be turn on as simultaneously as possible.

21.2. Operational requirements

The following table shows the operational requirements.

Table 105 Operational requirements

VSS,PVSS,AVSS = 0.0 V, Ta = -20 ℃～+70 ℃

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
B1	Power-supply voltage 1	VDD33	-	3.135	3.3	3.465	V
B2	Power-supply voltage 2	AVDD	-	3.135	3.3	3.465	V
B3	Power-supply voltage 3	PVDD	-	3.135	3.3	3.465	V
B4	Power-supply voltage 4	RVDD	-	3.175	3.3	3.465	V
B5	Power-supply voltage 5	VDD18	-	1.71	1.8	1.89	V
B6	Operation peripheral temperature	Topr	-	-20	-	70	℃
Crystal oscillation							
B7	Oscillating frequency (OSCI)	fosc	FRQS[1:0] = 00	25	-	33.33	MHz
			FRQS[1:0] = 01	28.75	-	40	
B8	Oscillating frequency (RCLKI)	fRCLK	-	-	32.768	-	kHz

(Note)
Equal voltage levels must be supplied to VDD33, AVDD, and PVDD.

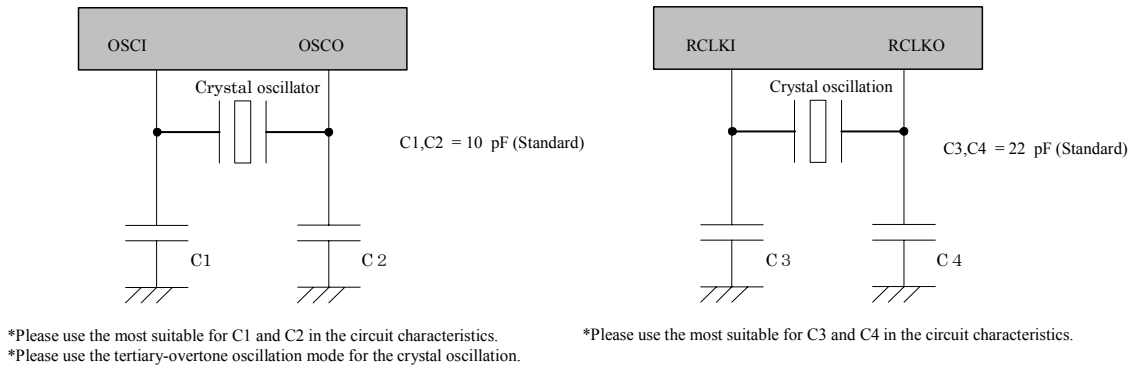


Figure 98 Self-excited oscillation recommended circuit

21.3. DC characteristics

The following tables show the DC characteristics.

Table 106 DC characteristics

VSS,PVSS,AVSS = 0.0 V, Ta = -20 ℃～+70 ℃

Item		Symbol	Conditions	Acceptable value			Unit
				Minimum	Standard	Maximum	
C1	Operating power-supply current (VDD18)	I_{DD18A}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = 33.33 MHz FRQS[1:0] = 0,0、Output release	—	—	460	mA
C2	SLEEP mode Power-supply current (VDD18)	I_{DD18B}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = 33.33 MHz FRQS[1:0] = 0,0、Output release	—	—	160	mA
C3	HALT mode Power-supply current (VDD18)	I_{DD18C}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = 33.33 MHz FRQS[1:0] = 0,0、Output release	—	—	60	mA
C4	Static power-supply current (VDD18)	I_{DD18D}	VDD18 = 1.89 V VDD33, PVDD, AVDD, RVDD = 3.465 V fOSC = halt FRQS[1:0] = 0,0、Output release Tj = 70 ℃	—	—	50	mA
C5	Operating power-supply current (VDD33)	I_{DD33A}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = 33.33 MHz FRQS[1:0] = 0,0、Output release	—	—	180	mA
C6	SLEEP mode power-supply current (VDD33)	I_{DD33B}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = 33.33 MHz FRQS[1:0] = 0,0、Output release	—	—	130	mA

Item		Symbol	Conditions	Acceptable value			Unit
				Minimum	Standard	Maximum	
C7	HALT mode power-supply current (VDD33)	I_{DD33C}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = 33.33 MHz FRQS[1:0] = 0,0、Output release	—	—	10	mA
C8	Static power-supply current (VDD33)	I_{DD33D}	VDD18 = 1.89 V VDD33, PVDD, AVDD, RVDD = 3.465 V fOSC = halt FRQS[1:0] = 0,0、output release Tj = 70 ℃	-	-	120	μA
C9	Power-supply current when ADC and PLL are operating. (AVDD, PVDD)	I_{APDD1}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = 33.33 MHz FRQS[1:0] = 0,0、output release	-	-	5	mA
C10	Power-supply current When ADC and PLL are halted (AVDD, PVDD)	I_{APDD2}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = halt FRQS[1:0] = 0,0、output release	-	-	40	μA
C11	Power-supply current when RTC are operating (RVDD)	I_{RDD1}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = 33.33 MHz FRQS[1:0] = 0,0、output release fRCLK=32.768 kHz	-	-	120	μA
C12	Power-supply current when RTC is halted (RVDD)	I_{RDD2}	VDD18 = 1.8 V VDD33, PVDD, AVDD, RVDD = 3.3 V fOSC = halt FRQS[1:0] = 0,0、Output release	-	-	40	μA

Table 107 DC characteristics

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Input/output pin 1 <Output:Push-pull/Input:LVTTL level> PIO0[0]～PIO0[7], PIO1[0]～PIO1[4], PIO2[0]～PIO2[4], PIO3[0]～PIO3[4] PIO5[0]～PIO5[2], SBO0, SBO1							
C13	Input voltage High level	V_{IH}	-	2.2	-	VDD33	V
C14	Input voltage Low level	V_{IL}	-	0	-	0.6	V
C15	Output voltage High level	V_{OH}	IO = -4 mA	2.4	-	-	V
C16	Output voltage Low level	V_{OL}	IO = 4 mA	-	-	0.4	V
C17	Output leak current	I_{OZ}	Condition of VO = Hi-Z	-5	-	5	µA
Input/Output pin 2 <Output: Nch open drain/Input:LVTTL level> PIO4[0]～PIO4[3]							
C18	Input voltage High level	V_{IH}	-	2.2	-	VDD33	V
C19	Input voltage Low level	V_{IL}	-	0	-	0.6	V
C20	Output voltage Low level	V_{OL}	IO = 4 mA	-	-	0.4	V
C21	Output leak current	I_{OZ}	Condition of VO = Hi-Z	-5	-	5	µA

Table 108 DC characteristics

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Input/output pin 3 <Output:Push-pull/Input:LVTTL level> SBT0, SBT1							
C22	Input voltage High level	V_{IH}	-	2.2	-	VDD33	V
C23	Input voltage Low level	V_{IL}	-	0	-	0.6	V
C24	Output voltage High level	V_{OH}	IO = -4 mA	2.4	-	-	V
C25	Output voltage Low level	V_{OL}	IO = 4 mA	-	-	0.4	V
C26	Output leak current	I_{OZ}	Condition of VO = Hi-Z (Pull-up condition)	-10	-	10	μA
C27	Pull-up resistance	R_{PU}	VI = 0 V	8.67	26	78	kΩ
Input/output pin 4 <Output: Push-pull/Input: LVTTL level> SRXW, XSAS, XSDK							
C28	Input voltage High level	V_{IH}	-	2.2	-	VDD33	V
C29	Input voltage Low level	V_{IL}	-	0	-	0.6	V

C30	Output voltage High level	V_{OH}	IO = -8 mA	2.4	-	-	V
C31	Output voltage Low level	V_{OL}	IO = 8 mA	-	-	0.4	V
C32	Output leak current	I_{OZ}	Condition of VO = Hi-Z (Pull-up condition)	-10	-	10	μA
C33	Pull-up resistance	R_{PU}	VI = 0 V	8.67	26	78	kΩ

Table 109 DC characteristics

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Input/output pin 5 <Output: Push-pull/Input: LVTTL level> MD0～MD15, SA0～SA31							
C34	Input voltage High level	VIH	-	2.2	-	VDD33	V
C35	Input voltage Low level	VIL	-	0	-	0.6	V
C36	Output voltage High level	VOH	IO = -8 mA	2.4	-	-	V
C37	Output voltage Low level	VOL	IO = 8 mA	-	-	0.4	V
C38	Output leak current	IOZ	Condition of VO = Hi-Z	-5	-	5	μA
Input/output pin 6 <Output: Push-pull/Input: LVTTL level> EXTRG, SD0～SD31, SSZ0, SSZ1							
C39	Input voltage High level	VIH	-	2.2	-	VDD33	V
C40	Input voltage Low level	VIL	-	0	-	0.6	V
C41	Output voltage High level	VOH	IO = -8 mA	2.4	-	-	V
C42	Output voltage Low level	VOL	IO = 8 mA	-	-	0.4	V
C43	Output leak current	IOZ	Condition of VO = Hi-Z (Pull-down condition)	-10	-	10	μA
C44	Pull-down resistance	RPD	VI = VDD33	8	24	72	kΩ

Table 110DC Characteristics

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0V, Ta = -20 ℃～+70 ℃

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Input pin 1 <Input: LVTTL level> TCK, XRESET, XIRQ0～XIRQ7, XNMI, CLK48							
C45	Input voltage High level	V_{IH}	-	2.2	-	VDD33	V
C46	Input voltage Low level	V_{IL}	-	0	-	0.6	V
C47	Input leak current	I_{OZ}	VI = VDD33 or 0 V	-5	-	5	μA
Input pin 2 <Input: LVTTL level> PWROK							
C48	Input voltage High level	V_{IH}	RVDD = 3.3 V ± 0.165 V	2.2	-	RVDD	V
C49	Input voltage Low level	V_{IL}	RVDD = 3.3 V ± 0.165 V	0	-	0.6	V
C50	Input leak current	I_{OZ}	VI = RVDD or 0 V	-5	-	5	μA

(Note)
PWROK is connected to RV_{DD} or the ground through resistance when RTC is not used.

Table 111 DC characteristics

V_{DD33}, PV_{DD}, AV_{DD}, RV_{DD} = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
V_{SS}, PV_{SS}, AV_{SS} = 0.0 V, Ta = -20 ℃～+70 ℃

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Input pin 3 <Input: LVTTL level> SDCKI, XSBR, TMS, MDK, SBI0～SBI2, SBT2, TDI							
C51	Input voltage High level	V_{IH}	-	2.2	-	VDD33	V
C52	Input voltage Low level	V_{IL}	-	0	-	0.6	V
C53	Input leak current	I_{OZ}	VI = VDD33	-10	-	10	μA
C54	Pull-up resistance	R_{PU}	VI = 0 V	8.67	26	78	kΩ
Input pin 4 <Input: LVTTL level> TRSTMOD							
C55	Input voltage High level	V_{IH}	-	2.2	-	VDD33	V
C56	Input voltage Low level	V_{IL}	-	0	-	0.6	V
C57	Input leak current	I_{OZ}	VI = 0 V	-10	-	10	μA
C58	Pull-down resistance	R_{PD}	VI = VDD33	8	24	72	kΩ

Table 112 DC characteristics

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Output pin 1 <Output: Push-pull> SBO2							
C59	Output voltage High level	VOH	IO = -4 mA	2.4	-	-	V
C60	Output voltage Low level	VOL	IO = 4 mA	-	-	0.4	V
C61	Output leak current	IOZ	Condition of VO = Hi-Z	-5	-	5	μA
Output pin 2 <Output: Push-pull> MA0～MA14, SDCKE, SDCLK, SYSCLK, TRCCLK, TRCD0～TRCD7, TRCST XMBE0, XMBE1, XMCAS, XMRAS, XMCS0, XMCS1, XMWE, XRSTOUT, XSBG XSCS0～XSCS7, XSRE, XSWE0～XSWE3, TDO							
C62	Output voltage High level	VOH	IO = -8 mA	2.4	-	-	V
C63	Output voltage Low level	VOL	IO = 8 mA	-	-	0.4	V
C64	Output leak current	IOZ	Condition of VO = Hi-Z	-5	-	5	μA
OSCI pin							
C65	Internal feedback resistance	RFB	VI = VDD33 or 0 V VDD33 = 3.3 V	1.0	3.0	9.0	kΩ
RCLKI pin							
C66	Internal feedback resistance	RFB	VI = RVDD33 or 0 V RVDD = 3.3 V	1.7	5.0	15.0	MΩ

Table 113 DC characteristics

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Input/output capacitance							
C67	Input pin	C_{IN}	VDD33 = VI = 0 V f = 1 MHz	-	-	16	pF
C68	Output pin	C_{OUT}		-	-	16	pF
C69	Input/output pin	C_{IO}		-	-	16	pF

21.4. A/D converter characteristics

The following table shows the A/D converter characteristics.

Table 114 A/D converter characteristics

V_{DD33}, PV_{DD}, AV_{DD}, RV_{DD} = 3.3 V±0.165 V, $V_{DD}18$ = 1.8 V±0.09 V
V_{SS}, PV_{SS}, AV_{SS} = 0.0 V, Ta = -20 ℃～+70 ℃

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
D1	Resolution	-	-	-	-	10	Bit
D2	A/D conversion relative accuracy	-	V_{REFH}= 3.3 V Conversion reference clock = 4.166 MHz	-	-	±4	LSB
D3	A/D conversion differential nonlinear error	-		-	-	±4	LSB
D4	A/D conversion period	-	-	2.6	-	-	μs
D5	Reference input voltage	V_{REF+}	-	AV_{DD}-0.3	-	AVDD	V
D6	Analog input voltage	V_{IA}	-	AV_{SS}	-	AVDD	V

21.5. AC characteristics

The following tables show AC characteristics

21.5.1. Reset signal timing

.

Table 115 AC characteristics (1)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Specification			Unit
				Minimum	Standard	Maximum	
Reset input timing							
E1	Reset signal pulse width (XRESET)	t_{RSTW}	-	8 t_{SCYC}	-	-	ns
E2	Reset hold period on power-up	t_{RSTN}	-	1	-	-	ms
E3	Mode input setup period	t_{MDS}	-	0.1	-	-	ms
E4	Mode input hold time	t_{MDH}	-	0.1	-	-	ms

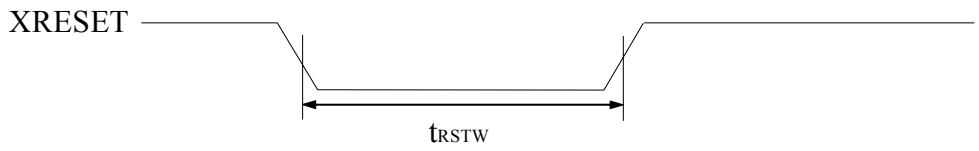


Figure 99 Reset input timing (1)

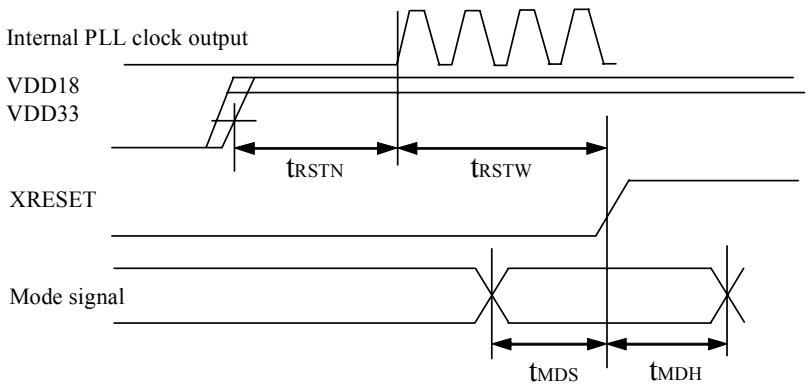


Figure 100 Reset input timing (2)

21.5.2. Clock timing

Table 116 AC characteristics (2)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Specification			Unit
				Minimum	Standard	Maximum	
Clock timing							
E5	Clock input frequency (OCSI)	t_{EXF}	FRQS=00	25	-	33.33	MHz
			FRQS=01	28.75	-	40	
E6	Clock input cycle time (OCSI)	t_{EXCCYC}	FRQS=00	30	-	40	ns
			FRQS=01	25	-	34.8	
E7	CPU operating frequency (MCLK)	t_{MF}	FRQS=00	100	-	133.3	MHz
			FRQS=01	57.5	-	80	
E8	CPU operating cycle time (MCLK)	t_{MCYC}	FRQS=00	7.5	-	10	ns
			FRQS=01	12.5	-	17.4	
E9	System clock output frequency (SYSCLK)	t_{SF}	FRQS=00	25	-	33.33	MHz
			FRQS=01	28.75	-	40	
E10	System clock output cycle time (SYSCLK)	t_{SCYC}	FRQS=00	30	-	40	ns
			FRQS=01	25	-	34.8	
E11	System clock output duty（SYSCLK）	t_{SCYCH}	-	45	-	55	%
E12	System clock output rise time（SYSCLK）	t_{SCYCR}	C_L=50 pF	-	-	4	ns
E13	System clock output fall time（SYSCLK）	t_{SCYCCF}	C_L=50 pF	-	-	4	ns
E14	SDRAM clock output frequency（SDCLK）	t_{SDF}	FRQS=00	100	-	133.3	MHz
			FRQS=01	57.5	-	80	
E15	SDRAM clock output cycle time（SDCLK）	t_{SDCYC}	FRQS=00	7.5	-	10	ns
			FRQS=01	12.5	-	17.4	

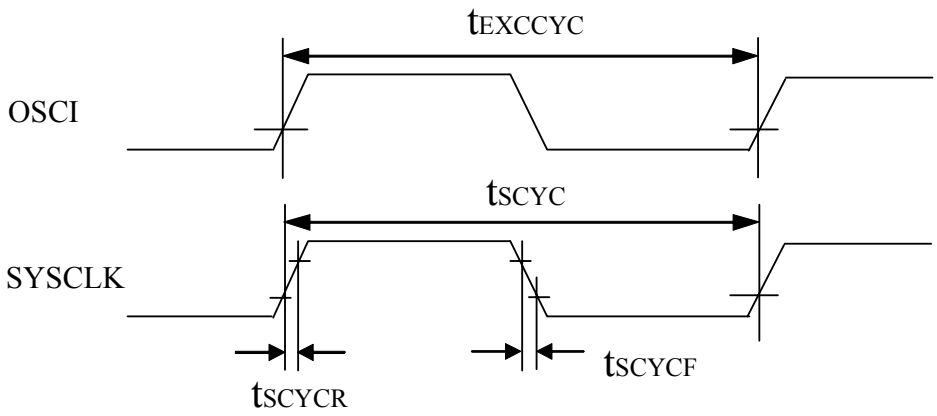


Figure 101 Clock timing

21.5.3. System bus signal timing

Table 117 AC characteristics (3)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Specification			Unit
				Minimum	Standard	Maximum	
System bus signal output timing							
E16	Address strobe signal output delay time (XSAS)	t_{SASOD}	-	0	-	7	ns
E17	Address strobe signal output hold time (XSAS)	t_{SASOH}	-	0	-	-	ns
E18	System bus address output delay time (SA)	t_{SAOD}	-	0	-	7	ns
E19	System bus address output hold time (SA)	t_{SAOH}	-	0	-	-	ns
E20	Data transfer size signal output delay time (SSZ)	t_{SSZOD}	-	0	-	7	ns
E21	Data transfer size signal output hold time (SSZ)	t_{SSZOH}	-	0	-	-	ns
E22	Chip select signal output delay time (XSCS)	t_{SCSD}	-	0	-	7	ns
E23	Chip select signal output hold time (XSCS)	t_{SCSH}	-	0	-	-	ns
E24	Read enable signal output delay time (XSRE)	t_{SRED}	-	0	-	5	ns
E25	Read enable signal output hold time (XSRE)	t_{SREH}	-	0	-	-	ns
E26	Write enable signal output delay time (XSWE)	t_{SWED}	-	0	-	5	ns
E27	Write enable signal output hold time (XSWE)	t_{SWEH}	-	0	-	-	ns
E28	Read/write condition signal output delay time	t_{SRWOD}	-	0	-	7	ns

Item		Symbol	Condition	Specification			Unit
				Minimum	Standard	Maximum	
	(SRXW)						
E29	Read/write condition signal output hold time (SRXW)	t_{SRWOH}	-	0	-	-	ns
E30	System data output delay time (SD)	t_{SDOD}	-	2	-	8	ns
E31	System data output hold time (SD)	t_{SDOH}	-	0	-	-	ns
E32	Output tri-state output delay time	t_{OBOFF}	-	0	-	25	ns
E33	Output buffer on delay time	t_{OBON}	-	0	-	25	ns
System bus signal input timing							
E34	System data input setup time (SD)	t_{SDIS}	-	7	-	-	ns
E35	System data input hold time (SD)	t_{SDIH}	-	0	-	-	ns
E36	Data acknowledge signal input setup time (XSDK)	t_{SDKIS}	-	11	-	-	ns
E37	Data acknowledge signal input hold time (XSDK)	t_{SDKIH}	-	0	-	-	ns

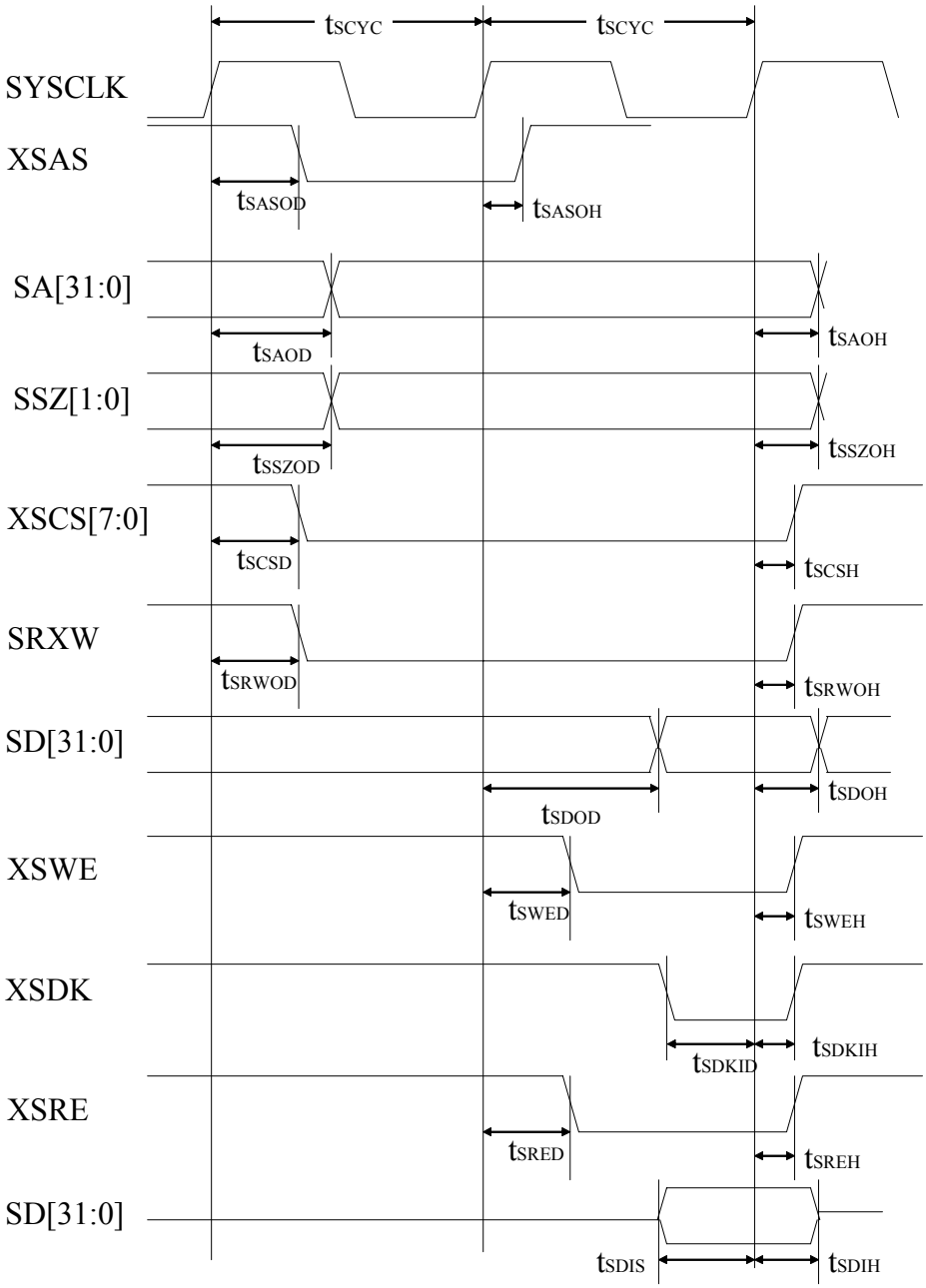


Figure 102 System bus signal input/output timing (1)

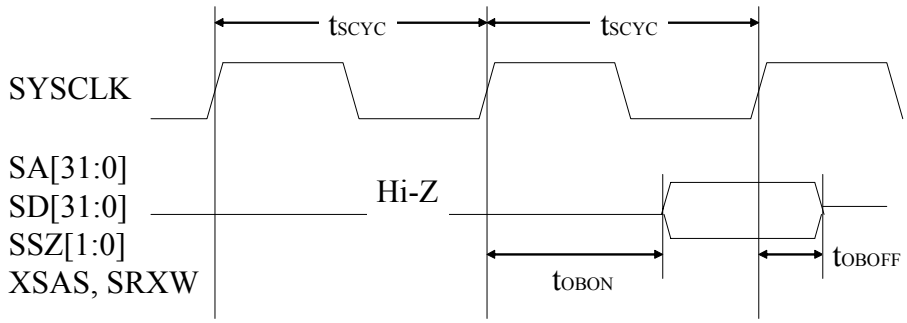


Figure 103 System bus signal input/output timing (2)

Table 118 AC characteristics (4)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
System bus signal timing (in the external master cycle)							
E38	Bus request signal input setup time (XSBR)	t_{SBRIS}	-	5	-	-	ns
E39	Bus request signal input hold time (XSBR)	t_{SBRIH}	-	2	-	-	ns
E40	Bus grant signal output delay time (XSBG)	t_{SBGOD}	-	0	-	10	ns
E41	Address strobe signal input setup time (XSAS)	t_{SASIS}	-	10	-	-	ns
E42	Address strobe signal input hold time (XSAS)	t_{SASIH}	-	2	-	-	ns
E43	System bus address input setup time (SA)	t_{SAIS}	-	0	-	-	ns
E44	System bus address input hold time (SA)	t_{SAIH}	-	0	-	-	ns
E45	Data transfer size signal input setup time (SSZ)	t_{SSZIS}	-	0	-	-	ns
E46	Data transfer size signal input hold time (SSZ)	t_{SSZIH}	-	0	-	-	ns
E47	Read/write condition signal input setup time (SRXW)	t_{SRWIS}	-	6	-	-	ns

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
E48	Read/write condition signal input hold time (SRXW)	t_{SRWIH}	-	2	-	-	ns
E49	Data acknowledge signal output delay time (XSDK)	t_{SDKOD}	-	0	-	10	ns
E50	Data acknowledge signal output hold time (XSDK)	t_{SDKOH}	-	0	-	-	ns

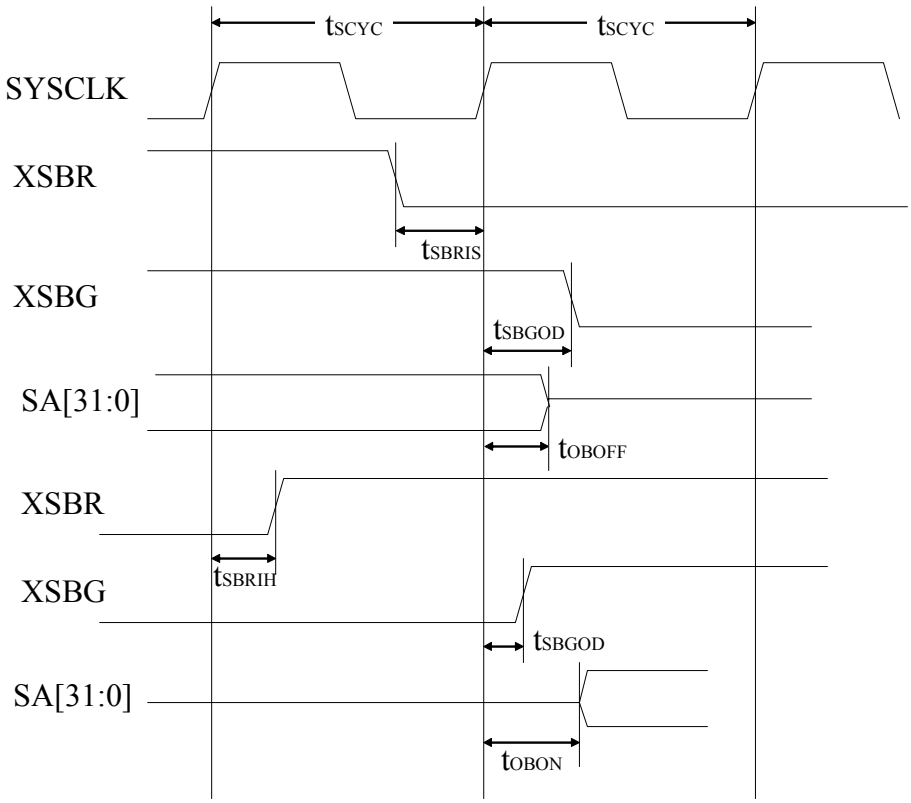


Figure 104 System bus signal input/output timing (in the external master cycle)(1)

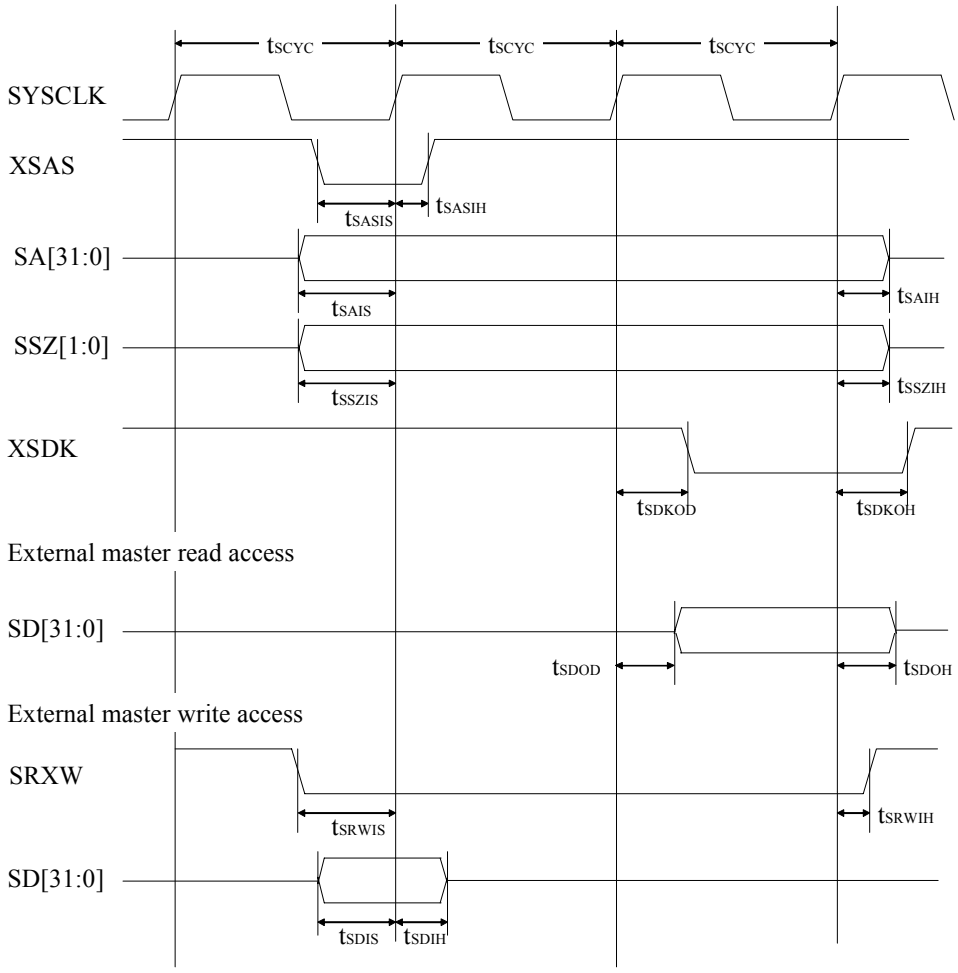


Figure 105 System bus signal input/output timing (in the external master cycle)(2)

21.5.4. Memory bus signal timing

Table 119 AC characteristics (5)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Memory bus signal output timing							
E51	SDRAM clock frequency (SDCLK)	t_{SDCF}	-	-	-	133.33	MHz
E52	SDRAM clock cycle time (SDCLK)	t_{SDCCYC}	-	7.5	-	-	ns
E53	SDRAM clock high-level time (SDCLK)	t_{SDCHP}	t_{SDCCYC}=7.5	3	-	-	ns
E54	SDRAM clock low-level time (SDCLK)	t_{SDCLP}	t_{SDCCYC}=7.5	3	-	-	ns
E55	SDRAM clock enable signal output delay time (SDCKE)	t_{CKEOD}	-	0	-	t_{SDCYC}/2+1.75	ns
E56	SDRAM clock enable signal output hold time (SDCKE)	t_{CKEOH}	-	2.5	-	-	ns
E57	Chip-select signal output delay time (XMCS)	t_{MCSOD}	-	0	-	t_{SDCYC}/2+1.75	ns
E58	Chip-select signal output hold time (XMCS)	t_{MCSOH}	-	2.5	-	-	ns
E59	SDRAM RAS signal output delay time (XMRAS)	t_{MROD}	-	0	-	t_{SDCYC}/2+1.75	ns
E60	SDRAM RAS signal output hold time (XMRAS)	t_{MROH}	-	2.5	-	-	ns
E61	SDRAM CAS signal output delay time (XMCAS)	t_{MCOD}	-	0	-	t_{SDCYC}/2+1.75	ns
E62	SDRAM CAS signal output hold time (XMCAS)	t_{MCOH}	-	2.5	-	-	ns
E63	SDRAM write	t_{MWOD}	-	0	-	t_{SDCYC}/2+1.	ns

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
	enable signal output delay time					75	
E64	SDRAM write enable signal output hold time (XMWE)	t_{MWOH}	-	2.5	-	-	ns
E67	Data byte enable signal output delay time (XMBE)	t_{MBOD}	-	0	-	t_{SDCYC}/2+1.75	
E68	Data byte enable signal output hold time (XMBE)	t_{MBOH}	-	2.5	-	-	
E69	Memory bus address output delay time (MA)	t_{MAOD}	-	0	-	t_{SDCYC}/2+2.25	ns
E70	Memory bus address output hold time (MA)	t_{MAOH}	-	2.5	-	-	ns
E71	Memory bus data output delay time (MD)	t_{MDOD}	-	0	-	tSDCYC/2+2.25	ns
E72	Memory bus data output hold time (MD)	t_{MDOH}	-	2.5	-	-	ns
E73	Memory bus data output tri-state delay time (MD)	t_{MDOFF}	-	0	-	t_{SDCYC}/2+3.25	ns
E74	Memory bus data output buffer on delay time (MD)	t_{MDON}	-	0	-	-	ns
Memory bus signal input timing							
E75	Skew time between SDCLK-clock output and SDCKI-clock input	t_{CSKEW}	-	0	-	-	ns
E76	Memory bus data input setup time (MD)	t_{MDIS}	-	2	-	-	ns
E77	Memory bus data input hold time (MD)	t_{MDIH}	-	2	-	-	ns

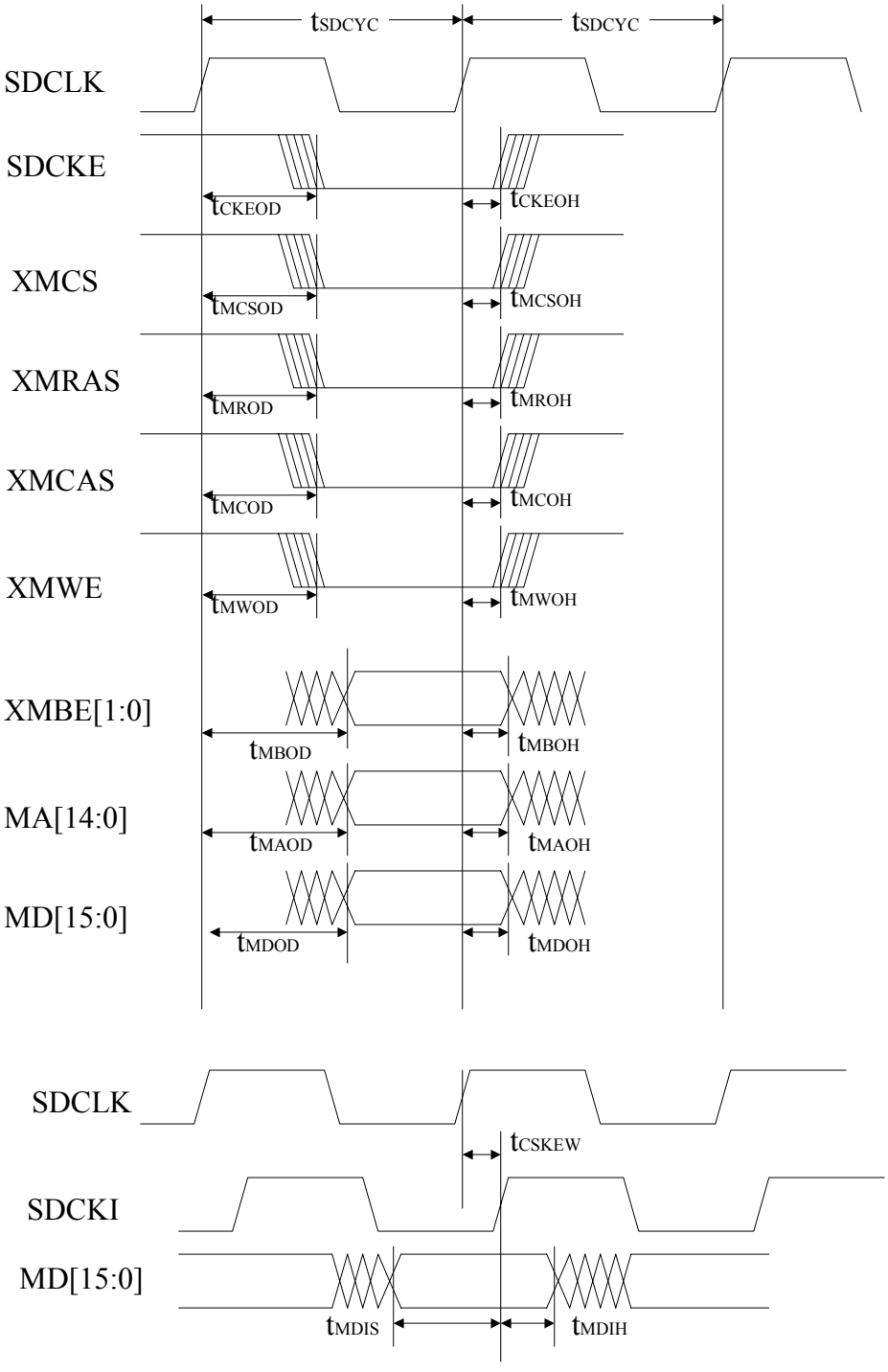


Figure 106 Memory bus signal input/output timing

21.5.5. DMA signal timing

Table 120 AC characteristics (6)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
DMA transfer request signal input timing							
E78	DMA transfer request signal input pulse width (XDMR)	t_{DMRW}	-	t_{SCYC} × 1.5	-	-	ns

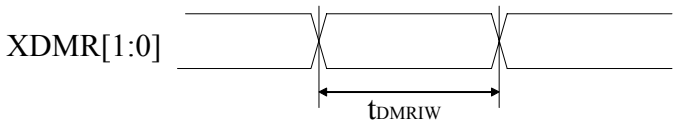


Figure 107 DMA transfer request signal input timing

21.5.6. Timer counter signal timing

Table 121 AC characteristics (7)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL=50 pF

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Timer counter signal input timing							
E79	Timer counter input signal pulse width (TM0IO-TM3IO, TM4IO-TM5IO, TM6IOA,TM6IOB, TM7IO-TM11IO)	t_{TCIW}	-	t_{SCYC} × 1.5	-	-	ns
Timer counter signal output timing							
E80	Timer counter output signal pulse width (TM0IO-TM3IO,TM4IO-TM5IO, TM6IOA, TM6IOB, TM7IO-TM11IO)	t_{TCOW}	-	t_{SCYC} × 0.8	-	-	ns

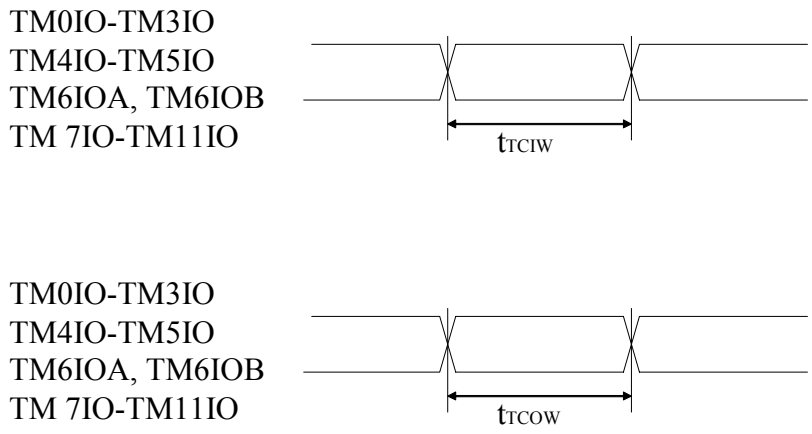


Figure 108 Timer counter signal input/output timing

21.5.7. External interrupt signal timing

Table 122 AC characteristics (8)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
External interrupt signal input timing							
E81	External interrupt input signal pulse width (XIRQ)	t_{SRQW}	-	t_{SCYC} × 2	-	-	ns
E82	External NMI input signal pulse width (XNMI)	t_{NMIW}	-	t_{SCYC} × 2	-	-	ns

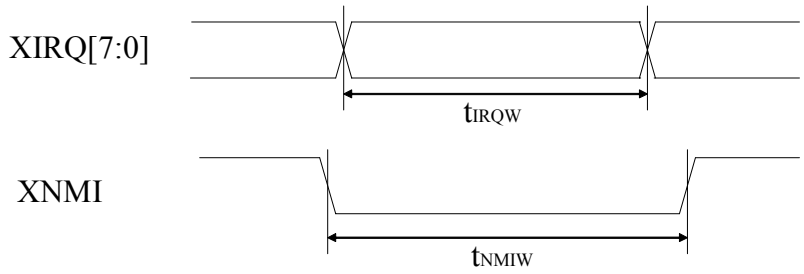


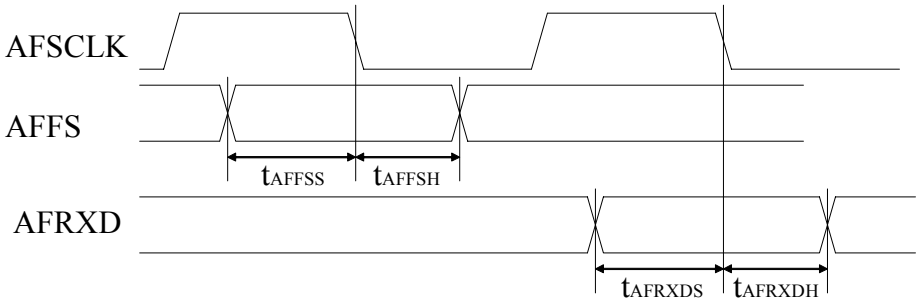
Figure 109 External interrupt signal input timing

21.5.8. Analog Front End signal timing

Table 123 AC characteristics (9)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
Analog Front End interface signal input timing							
E83	AFE frame synchronization signal input setup time (AFFS)	t_{AFFSS}	―	60	―	―	ns
E84	AFE frame synchronization signal input hold time (AFFS)	t_{AFFSH}	―	60	―	―	ns
E85	AFE data input setup time (AFRXD)	t_{AFRXDS}	―	60	―	―	ns
E86	AFE data input hold time (AFRXD)	t_{AFRXDH}	―	60	―	―	ns
E87	AFE data clock input cycle time (AFSCLK)	t_{AFSCYC}	―	250	―	―	ns
Analog Front End interface signal output timing							
E88	AFE frame synchronization signal output delay time (AFFS)	t_{AFFSOD}	―	―	―	70	ns
E89	AFE data output delay time (AFTXD)	$t_{AFTXDOD}$	―	―	―	70	ns



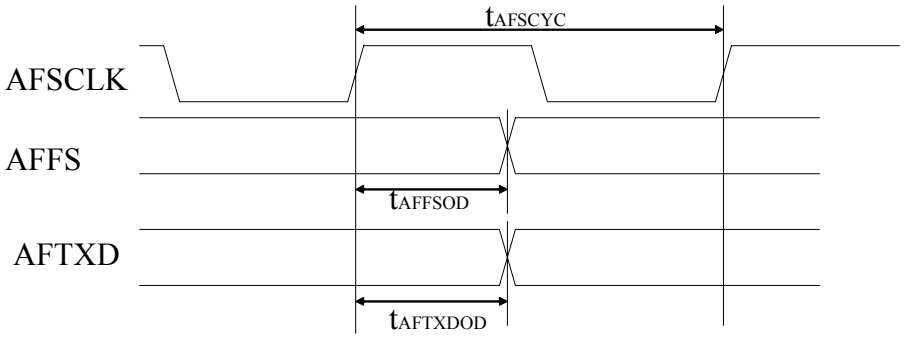


Figure 110 AFE interface signal input/output timing

21.5.9. A/D conversion signal timing

Table 124 AC characteristics (10)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
A/D conversion signal input timing							
E90	ADTRG input signal pulse width	t_{ADTRGW}	-	t_{SCYC} × 1.5	-	-	ns

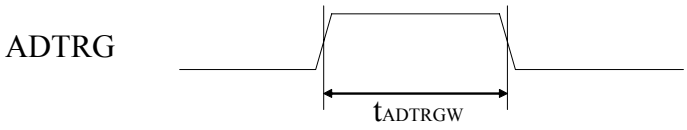


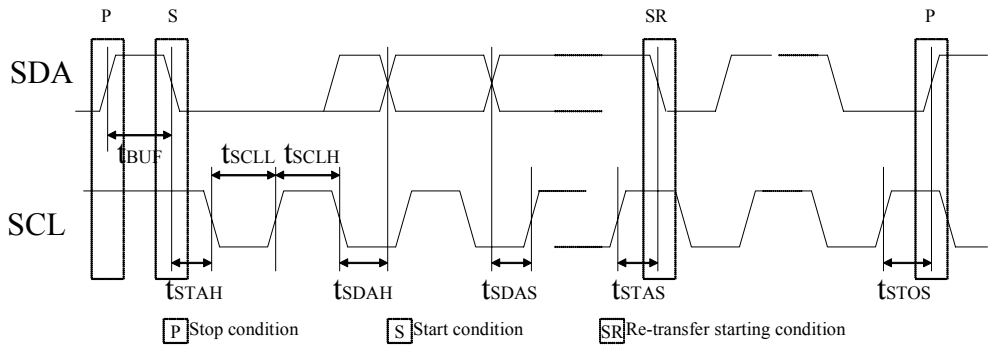
Figure 111 A/D conversion signal input timing

21.5.10. I2C Interface signal timing

Table 125 AC characteristics (11)

VDD33, PVDD, AVDD, RVDD = 3.3 V±0.165 V, VDD18 = 1.8 V±0.09 V
VSS, PVSS, AVSS = 0.0 V, Ta = -20 ℃～+70 ℃, CL = 50 pF

Item		Symbol	Condition	Acceptable value			Unit
				Minimum	Standard	Maximum	
12C controller signal timing							
E91	SCL operating frequency	t_{SCLF}	-	-	-	400	kHz
E92	SCL clock high time	t_{SCLH}	-	0.6	-	-	μs
E93	SCL clock low time	t_{SCLL}	-	1.3	-	-	μs
E94	Bus-free period (from the stop to start conditions)	t_{BUF}	-	1.3	-	-	μs
E95	Start condition hold time	t_{STAH}	-	0.6	-	-	μs
E96	Stop condition setup time	t_{STOS}	-	0.6	-	-	μs
E97	Start condition setup time	t_{STAS}	-	0.6	-	-	μs
E98	SDA input setup time	t_{SDAS}	-	100	-	-	ns
E99	SDA input hold time	t_{SDAH}	-	1	-	-	μs
E100	SDA output low determination time	t_{SDALD}	-	-	-	0.9	μs
E101	SDA output off determination time	t_{SDAOFF}	-	-	-	0.9	μs
E102	SCL/SDA input rise time	t_{I2CR}	-	-	-	300	ns
E103	SCL/SDA input fall time	t_{I2CF}	-	-	-	300	ns
E104	Spike width which can be removed through input filter	t_{I2CSP}	-	-	-	50	ns



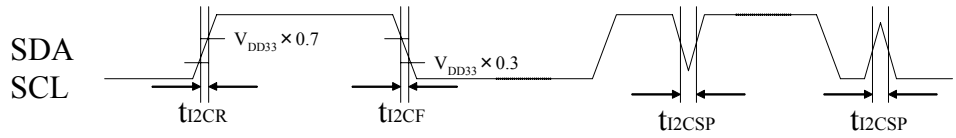


Figure 112 12C controller signal timing

21.5.11. AC characteristics measuring conditions

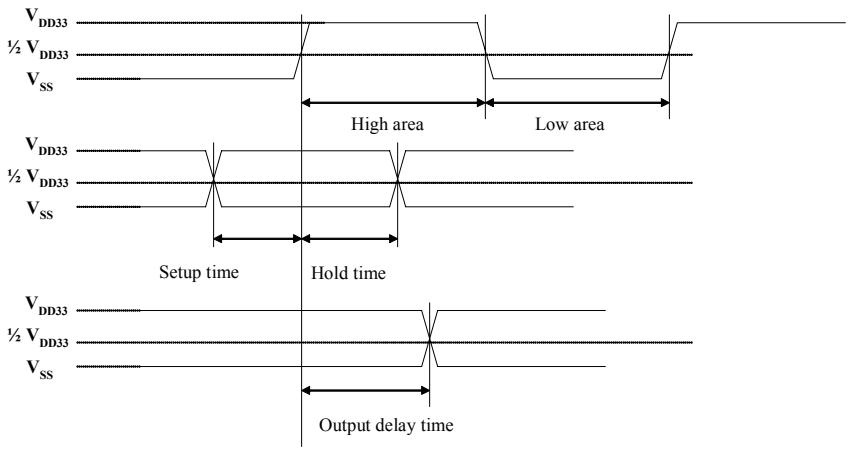


Figure 113 AC characteristics measuring conditions

Chapter 22 Appendix

22

CHAPTER 22

APPENDIX

22.1. Pin list

Pin name	I/O	Schmitt	Pull	Pin reset		Soft reset		SLEEP mode		STOP/HALT mode		Unassigned
				I/O	Value	I/O	Value	I/O	Value	I/O	Value	
VDD33	Power supply	-	-	-	-	-	-	-	-	-	-	-
VDD18	Power supply	-	-	-	-	-	-	-	-	-	-	-
VSS	Power supply	-	-	-	-	-	-	-	-	-	-	-
VREFH	Power supply	-	-	-	-	-	-	-	-	-	-	-
AVDD	Power supply	-	-	-	-	-	-	-	-	-	-	-
AVSS	Power supply	-	-	-	-	-	-	-	-	-	-	-
PVDD	Power supply	-	-	-	-	-	-	-	-	-	-	-
PVSS	Power supply	-	-	-	-	-	-	-	-	-	-	-
RVDD	Power supply	-	-	-	-	-	-	-	-	-	-	-
PWROK	I	-	-	I	-	I	-	-	-	-	-	-
OSCI	I	-	-	I	-	I	-	-	-	-	-	-
OSCO	O	-	-	O	-	O	-	O	-	O	-	Open
TCPOUT (Analog)	O	-	-	O	Undefined	O	Undefined	O	Undefined	O	Undefined	Open
SYSCLK	O	-	-	O	-	O	-	O	-	O	Low	Open
XRESET	I	Schmitt	-	I	-	I	-	I	-	I	-	-
XRSTOUT	O	-	-	O	Low	O	Low	O	High	O	High	Open
SA[31:16]	I/O	-	-	O		O	Undefined	O	Undefined	O	Undefined	Open
SA[15:0]	I/O	-	-	O	Undefined	O	Undefined	O	Undefined	O	Undefined	Open
SD[31:16]	I/O	-	down	I	-	I	-	I	-	I	-	Open
SD[15:0]	I/O	-	down	I	-	I	-	I	-	I	-	Open
XSAS	I/O	-	up	O	High	O	High	O	-	O	Note 3	Open
XSCS[7:4]	O	-	-	O	High	O	High	O	-	O	Note 3	Open
XSCS[3:0]	O	-	-	O	High	O	High	O	-	O	Note 3	Open
SSZ[1:0]	I/O	-	down	O	Low	O	Low	O	-	O	Note 3	Open
XSRE	O	-	-	O	High	O	High	O	-	O	Note 3	Open
XSWE[3:0]	O	-	-	O	High	O	High	O	-	O	Note 3	Open
SRXW	I/O	-	up	O	High	O	High	O	-	O	Note 3	Open

Pin name	I/O	Schmitt	Pull	Pin reset		Soft reset		SLEEP mode		STOP/HALT mode		Unassigned
XSDK	I/O	-	up	I	-	I	-	I	-	I	-	Open
XSBR	I	-	up	I	-	I	-	I	-	I	-	Open
XSBG	O	-	-	O	High	O	High	O	-	O	Note 3	Open
MA[14:0]	O	-	-	O	Low	O	Low	O	Undefined	O	Undefined	Open
MD[15:0]	I/O	-	-	O	Low	O	Low	I	-	I	-	Open
XMCS[1:0]	O	-	-	O	High	O	High	O	-	O	Note 3	Open
XMBE[1:0]	O	-	-	O	High	O	High	O	-	O	Note 3	Open
XMRAS	O	-	-	O	High	O	High	O	-	O	Note 3	Open
XMCAS	O	-	-	O	High	O	High	O	-	O	Note 3	Open
XMWE	O	-	-	O	High	O	High	O	-	O	Note 3	Open
MDK	I	-	up	I	-	I	-	I	-	I	-	Open
SDCLK	O	-	-	O	-	O	-	O	-	O	Low	Open
SDCKE	O	-	-	O	High	O	High	O	-	O	Note 3	Open
SDCKI	I	-	up	I	-	I	-	I	-	I	-	Open
XIRQ[7:0]	I	Schmitt	-	I	-	I	-	I	-	I	-	Pull-up
XNMI	I	Schmitt	-	I	-	I	-	I	-	I	-	Pull-up
SBI[2:0]	I	-	up	I	-	I	-	I	-	I	-	Open
SBO2	O	-	-	O	Undefined	O	Undefined	O	-	O		Open
SBT2	I	-	up	I	-	I	-	I	-	I	-	Open
RCLKI	I	-	-	I	-	I	-	I	-	I	-	Power supply (0V)
RCLKO	O	-	-	O	-	O	-	O	-	O	-	Open
AN[7:0]	I	-	-	I	-	I	-	I	-	I	-	Power supply (0V)
CLK48	I	Schmitt	-	I	-	I	-	I	-	I	-	Power supply (0V)
TDI	I	-	up	I	-	I	-	I	-	I	-	Open
TDO	O	-	-	O	Undefined	O	Undefined	O	Undefined	O	Undefined	Open
TCK	I	Schmitt	-	I	-	I	-	I	-	I	-	Note 1
TMS	I	-	up	I	-	I	-	I	-	I	-	Open
TRSTMOD	I	-	down	I	-	I	-	I	-	I	-	Open
TRCD[7:0]	O	-	-	O	Undefined	O	Undefined	O	Undefined	O	Undefined	Open
TRCST	O	-	-	O	Undefined	O	Undefined	O	Undefined	O	Undefined	Open
EXTRG	I/O	Schmitt	down	I	-	I	-	I	-	I	-	Open
TRCCLK	O	-	-	O	Undefined	O	Undefined	O	Undefined	O	Undefined	Open
PIO0[0]/TM0IO/EYECLK	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO0[1]/TM1IO/EYED	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note4	Note 1
PIO0[2]/TM2IO	I/O	Schmitt	-	I	-	I		-	-	Note 3	Note 4	Note 1
PIO0[3]/TM3IO	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1

Pin name	I/O	Schmitt	Pull	Pin reset		Soft reset		SLEEP mode		STOP/HALT mode		Unassigned
PIO0[4]/TM4IO/XCTS	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO0[5]/TM5IO	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO0[6]/TM6IOA	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO0[7]/TM6IOB	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO1[0]/TM7IO/ADTRG	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO1[1]/TM8IO/XDMR[0]	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO1[2]/TM9IO/XDMR[1]	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO1[3]/TM10IO/FRQS[0]	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO1[4]/TM11IO/FRQS[1]	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO2[0]/BOOTBW	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 2
PIO2[1]/BOOTSEL	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 2
PIO2[2]	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 2
PIO2[3]/CKIO	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 2
PIO2[4]/CMOD	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 2
PIO3[0]/AFRXD	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO3[1]/AFTXD	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO3[2]/AFSCLK	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO3[3]/AFFS	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO3[4]/AFEHC	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO4[0]/SCL[0]	I/O	Schmitt	OD	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO4[1]/SDA[0]	I/O	Schmitt	OD	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO4[2]/SCL[1]	I/O	Schmitt	OD	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO4[3]/S	I/O	Schmitt	OD	I	-	I	-	-	-	Note 3	Note 4	Note 1

Pin name	I/O	Schmitt	Pull	Pin reset		Soft reset		SLEEP mode		STOP/HALT mode		Unassigned
DA[1]												
PIO4[4]/ SBO0	I/O	Schmitt	-	O	Undefined	O	Undefined	O	Undefined	Note 3	Note 4	Open
PIO4[5]/ SBO1	I/O	Schmitt	-	O	Undefined	O	Undefined	O	Undefined	Note 3	Note 4	Open
PIO4[6]/ SBT0	I/O	Schmitt	up	I	-	I	-	I	-	Note 3	Note 4	Open
PIO4[7]/ SBT1	I/O	Schmitt	up	I	-	I	-	I	-	Note 3	Note 4	Open
PIO5[0]/I RTXD /SOUT	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO5[1]/I RRXDS /SIN	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1
PIO5[2]/I RRXDF	I/O	Schmitt	-	I	-	I	-	-	-	Note 3	Note 4	Note 1

Note 1 Input is executed by resetting. In the case of not assigning the pin, and the case of the design that its defined level is not driven by the external device, pill-up or pull-down must be necessary.
Note 2 Input is executed by resetting. This pin carries out the mode setting in the chip when releasing the reset (the rising edge of the XRESET pin).
Note 3 This maintains the state right before the clock stops.
Note 4 This maintains the state right before the clock stops in the case of outputting.

22.2. Address Map

		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Interrupt	0xC000000X			IVAR3				IVAR2				IVAR1				IVAR0	
control	0xC000001X							IVAR6				IVAR5				IVAR4	
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
CPU control	0xC000002X					System reserve				TBR						CPUP	
/Interrupt	0xC000003X					DEAR				System reserve				System reserve			
control	0xC000004X									SISR						CPUM	
	0xC000005X													CPUREV			
	0xC000006X													System reserve			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Cache	0xC000007X															CHCTR	
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
MMU	0xC000009X	MMUFCR				PTBR					PIDR			MMUCTR			
	0xC00000AX					IPTEL2				IPTEU				IPTEL			
	0xC00000BX					DPTEL2				DPTEU				DPTEL			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
System reserve	0xC000010X					System reserve				System reserve				System reserve			
	0xC000012X	System reserve				System reserve				System reserve				System reserve			
	0xC000014X	System reserve				System reserve				System reserve				System reserve			
	0xC000015X													System reserve			
	0xC000016X	System reserve				System reserve				System reserve				System reserve			
	0xC000017X													System reserve			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
WDT	0xC000100X												RSTCTR		WDCTR		WDBC
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BCU	0xC000200X													BCCR			
	0xC000201X													BCBERR			
	0xC000202X													BCBEAR			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Instruction	0xC800000X	Way0 entry0 offset3				Way0 entry0 offset2				Way0 entry0 offset1				Way0 entry0 offset0			
cache	0xC800001X	Way0 entry1 offset3				Way0 entry1 offset2				Way0 entry1 offset1				Way0 entry1 offset0			
data address	:	:				:				:				:			
	0xC8000FEX	Way0 entry254 offset3				Way0 entry254 offset2				Way0 entry254 offset1				Way0 entry254 offset0			
	0xC8000FFX	Way0 entry255 offset3				Way0 entry255 offset2				Way0 entry255 offset1				Way0 entry255 offset0			
	0xC800100X	Way1 entry0 offset3				Way1 entry0 offset2				Way1 entry0 offset1				Way1 entry0 offset0			
	0xC800101X	Way1 entry1 offset3				Way1 entry1 offset2				Way1 entry1 offset1				Way1 entry1 offset0			
	:	:				:				:				:			
	0xC8001FEX	Way1 entry254 offset3				Way1 entry254 offset2				Way1 entry254 offset1				Way1 entry254 offset0			
	0xC8001FFX	Way1 entry255 offset3				Way1 entry255 offset2				Way1 entry255 offset1				Way1 entry255 offset0			
	0xC800200X	Way2 entry0 offset3				Way2 entry0 offset2				Way2 entry0 offset1				Way2 entry0 offset0			
	0xC800201X	Way2 entry1 offset3				Way2 entry1 offset2				Way2 entry1 offset1				Way2 entry1 offset0			
	:	:				:				:				:			
	0xC8002FEX	Way2 entry254 offset3				Way2 entry254 offset2				Way2 entry254 offset1				Way2 entry254 offset0			
	0xC8002FFX	Way2 entry255 offset3				Way2 entry255 offset2				Way2 entry255 offset1				Way2 entry255 offset0			
	0xC800300X	Way3 entry0 offset3				Way3 entry0 offset2				Way3 entry0 offset1				Way3 entry0 offset0			
	0xC800301X	Way3 entry1 offset3				Way3 entry1 offset2				Way3 entry1 offset1				Way3 entry1 offset0			
	:	:				:				:				:			
	0xC8003FEX	Way3 entry254 offset3				Way3 entry254 offset2				Way3 entry254 offset1				Way3 entry254 offset0			
	0xC8003FFX	Way3 entry255 offset3				Way3 entry255 offset2				Way3 entry255 offset1				Way3 entry255 offset0			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Instruction	0xC810000X													Way0 entry0			
cache	0xC810001X													Way0 entry1			
tag address	:													:			
	0xC8100FEX													Way0 entry254			
	0xC8100FFX													Way0 entry255			
	0xC810100X													Way1 entry0			
	0xC810101X													Way1 entry1			
	:													:			
	0xC8101FEX													Way1 entry254			
	0xC8101FFX													Way1 entry255			
	0xC810200X													Way2 entry0			
	0xC810201X													Way2 entry1			
	:													:			
	0xC8102FEX													Way2 entry254			
	0xC8102FFX													Way2 entry255			
	0xC810300X													Way3 entry0			
	0xC810301X													Way3 entry1			
	:													:			
	0xC8103FEX													Way3 entry254			
	0xC8103FFX													Way3 entry255			

		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data cache	0xC820000X	Way0 entry0 offset3				Way0 entry0 offset2				Way0 entry0 offset1				Way0 entry0 offset0			
data address	0xC820001X	Way0 entry1 offset3				Way0 entry1 offset2				Way0 entry1 offset1				Way0 entry1 offset0			
	:	:				:				:				:			
	0xC8200FEX	Way0 entry254 offset3				Way0 entry254 offset2				Way0 entry254 offset1				Way0 entry254 offset0			
	0xC8200FFX	Way0 entry255 offset3				Way0 entry255 offset2				Way0 entry255 offset1				Way0 entry255 offset0			
	0xC820100X	Way1 entry0 offset3				Way1 entry0 offset2				Way1 entry0 offset1				Way1 entry0 offset0			
	0xC820101X	Way1 entry1 offset3				Way1 entry1 offset2				Way1 entry1 offset1				Way1 entry1 offset0			
	:	:				:				:				:			
	0xC8201FEX	Way1 entry254 offset3				Way1 entry254 offset2				Way1 entry254 offset1				Way1 entry254 offset0			
	0xC8201FFX	Way1 entry255 offset3				Way1 entry255 offset2				Way1 entry255 offset1				Way1 entry255 offset0			
	0xC820200X	Way2 entry0 offset3				Way2 entry0 offset2				Way2 entry0 offset1				Way2 entry0 offset0			
	0xC820201X	Way2 entry1 offset3				Way2 entry1 offset2				Way2 entry1 offset1				Way2 entry1 offset0			
	:	:				:				:				:			
	0xC8202FEX	Way2 entry254 offset3				Way2 entry254 offset2				Way2 entry254 offset1				Way2 entry254 offset0			
	0xC8202FFX	Way2 entry255 offset3				Way2 entry255 offset2				Way2 entry255 offset1				Way2 entry255 offset0			
	0xC820300X	Way3 entry0 offset3				Way3 entry0 offset2				Way3 entry0 offset1				Way3 entry0 offset0			
	0xC820301X	Way3 entry1 offset3				Way3 entry1 offset2				Way3 entry1 offset1				Way3 entry1 offset0			
	:	:				:				:				:			
	0xC8203FEX	Way3 entry254 offset3				Way3 entry254 offset2				Way3 entry254 offset1				Way3 entry254 offset0			
	0xC8203FFX	Way3 entry255 offset3				Way3 entry255 offset2				Way3 entry255 offset1				Way3 entry255 offset0			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data cache	0xC830000X													Way0 entry0			
tag address	0xC830001X													Way0 entry1			
	:													:			
	0xC8300FEX													Way0 entry254			
	0xC8300FFX													Way0 entry255			
	0xC830100X													Way1 entry0			
	0xC830101X													Way1 entry1			
	:													:			
	0xC8301FEX													Way1 entry254			
	0xC8301FFX													Way1 entry255			
	0xC830200X													Way2 entry0			
	0xC830201X													Way2 entry1			
	:													:			
	0xC8302FEX													Way2 entry254			
	0xC8302FFX													Way2 entry255			
	0xC830300X													Way3 entry0			
	0xC830301X													Way3 entry1			
	:													:			
	0xC8303FEX													Way3 entry254			
	0xC8303FFX													Way3 entry255			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Cache purge	0xC840000X													Way0 entry0			
address	0xC840001X													Way0 entry1			
	:													:			
	0xC8400FEX													Way0 entry254			
	0xC8400FFX													Way0 entry255			
	0xC840100X													Way1 entry0			
	0xC840101X													Way1 entry1			
	:													:			
	0xC8401FEX													Way1 entry254			
	0xC8401FFX													Way1 entry255			
	0xC840200X													Way2 entry0			
	0xC840201X													Way2 entry1			
	:													:			
	0xC8402FEX													Way2 entry254			
	0xC8402FFX													Way2 entry255			
	0xC840300X													Way3 entry0			
	0xC840301X													Way3 entry1			
	:													:			
	0xC8403FEX													Way3 entry254			
	0xC8403FFX													Way3 entry255			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
DMA	0xD200000X	DM0SIZ				DM0DST				DM0SRC				DM0CTR			
	0xD200001X													DM0CYC			
	0xD200010X	DM1SIZ				DM1DST				DM1SRC				DM1CTR			
	0xD200011X													DM1CYC			
	0xD200020X	DM2SIZ				DM2DST				DM2SRC				DM2CTR			
	0xD200021X													DM2CYC			
	0xD200030X	DM3SIZ				DM3DST				DM3SRC				DM3CTR			
	0xD200031X													DM3CYC			

		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
INTC	0xD400000X			G3ICR				G2ICR				G1ICR				NMICR G0ICR	
	0xD400001X			G7ICR				G6ICR				G5ICR				G4ICR	
	0xD400002X			G11ICR				G10ICR				G9ICR				G8ICR	
	0xD400003X			G15ICR				G14ICR				G13ICR				G12ICR	
	0xD400004X			G19ICR				G18ICR				G17ICR				G16ICR	
	0xD400005X			G23ICR				G22ICR				G21ICR				G20ICR	
	0xD400006X			G27ICR				G26ICR				G25ICR				G24ICR	
	0xD400007X			G31ICR				G30ICR				G29ICR				G28ICR	
	0xD400008X			G35ICR				G34ICR				G33ICR				G32ICR	
	0xD400009X			G39ICR				G38ICR				G37ICR				G36ICR	
	0xD40000AX											G4ICR				G40ICR	
	0xD400010X															IAGR	
	0xD400020X															EXTDM	
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
8-bit timer	0xD400300X													TM3MD	TM2MD	TM1MD	TM0MD
	0xD400301X													TM3BR	TM2BR	TM1BR	TM0BR
	0xD400302X													TM3BC	TM2BC	TM1BC	TM0BC
	0xD400307X															TMPSCNT	
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
16-bit timer	0xD400308X		TM11MD		TM10MD		TM9MD		TM8MD		TM7MD	TM6MD			TM5MD		TM4MD
	0xD400309X	TM11BR		TM10BR		TM9BR		TM8BR		TM7BR				TM5BR		TM4BR	
	0xD40030AX	TM11BC		TM10BC		TM9BC		TM8BC		TM7BC		TM6BC		TM5BC		TM4BC	
	0xD40030BX											TM6MDB	TM6MDA				
	0xD40030CX											TM6CA					
	0xD40030DX											TM6CB					
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Serial control	0xD400200X				SC0STR			SC0RXB	SC0TXB				SC0ICR			SC0CTR	
	0xD400201X				SC1STR			SC1RXB	SC1TXB				SC1ICR			SC1CTR	
	0xD400202X			SC2TIM	SC2STR			SC2RXB	SC2TXB				SC2ICR			SC2CTR	
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
AFE	0xD830000X			AFECTR				AFESTAT				AFEINTM				AFESYS	
	0xD830001X			AFEEYE				AFEFIFO				AFERBUF				AFETBUF	
	0xD830002X															AFESEC	
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
I2C	0xD840000X	IIC0CLK				IIC0MYAD				IIC0DREC				IIC0DTRM			
	0xD840001X									IIC0BSTS				IIC0BRST			
	0xD840100X	IIC1CLK				IIC1MYAD				IIC1DREC				IIC1DTRM			
	0xD840101X									IIC1BSTS				IIC1BRST			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
A/D	0xD850000X															ADCTR	
	0xD850001X	AD7BUF		AD6BUF		AD5BUF		AD4BUF		AD3BUF		AD2BUF		AD1BUF		AD0BUF	
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
RTC	0xD860000X				RTSRC	RTCRB	RTCRA	RTYCR	RTMTCR	RTDMCR	RTDWCR	RTHAR	RTHCR	TRMAR	RTMCR	RTSAR	RTSCR
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
IrDA	0xD870008X									IRSCR IRASCR IRRFLV IRRMLU IRFRLU IRFIRPW IRCFG4	IRMSR IRTFLV IRRMLL IRFRLL	IRLSR IRELSR IRTFLU IRFRST IRCFG2	IRMCR IRMDR IREXCR2 IRTFLL IRCR2 IRBFPL IRCFG1	IRLCR IRBSR	IRIIR IREIIR IRFCR IREXCR1 IRSHFCR IRCR1 IRPMDR	IRIER IREIER IRDLUR IRSHLCR IRTMRH IRMIRPW	IRTDR IRRDR IRDLLR IRTMRL IRCR3
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
SBC	0xD8C0010X													SBBASE0			
	0xD8C0011X													SBBASE1			
	0xD8C0012X													SBBASE2			
	0xD8C0013X													SBBASE3			
	0xD8C0014X													SBBASE4			
	0xD8C0015X													SBBASE5			
	0xD8C0016X													SBBASE6			
	0xD8C0017X													SBBASE7			
	0xD8C0020X					SBCNTRL02				SBCNTRL01				SBCNTRL00			
	0xD8C0021X					SBCNTRL12				SBCNTRL11				SBCNTRL10			
	0xD8C0022X					SBCNTRL22				SBCNTRL21				SBCNTRL20			
	0xD8C0023X					SBCNTRL32				SBCNTRL31				SBCNTRL30			
	0xD8C0024X					SBCNTRL42				SBCNTRL41				SBCNTRL40			
	0xD8C0025X					SBCNTRL52				SBCNTRL51				SBCNTRL50			
	0xD8C0026X					SBCNTRL62				SBCNTRL61				SBCNTRL60			
	0xD8C0027X					SBCNTRL72				SBCNTRL71				SBCNTRL70			
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
MBC	0xDA00000X	SDBASE1				SDBASE0						SDREFCNT		SDRAMBUS			
	0xDA00001X																SDSHDW
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
I/O	0xDB00000X				P0TMIO				P0OUT				P0IN			P0MD	
	0xDB00010X				P1TMIO				P1OUT				P1IN			P1MD	
	0xDB00020X								P2OUT				P2IN			P2MD	
	0xDB00030X								P3OUT				P3IN			P3MD	
	0xDB00040X								P4OUT				P4IN			P4MD	
	0xDB00050X								P5OUT				P5IN			P5MD	

Memory allocation of the peripheral-circuit register is aligned by a word (32 bits).
For example, all accesses from the 0th to third addresses become the access to the 0th address.
Similarly, all accesses from the fourth to seventh addresses, the eighth to B addresses, and the C to F addresses become the accesses to respectively the fourth, the eighth, and C addresses.
The access width is 8-bit (a byte9, 16-bit (half-word9, 32-bit (a word) accessible.

22.3. Instruction Set List

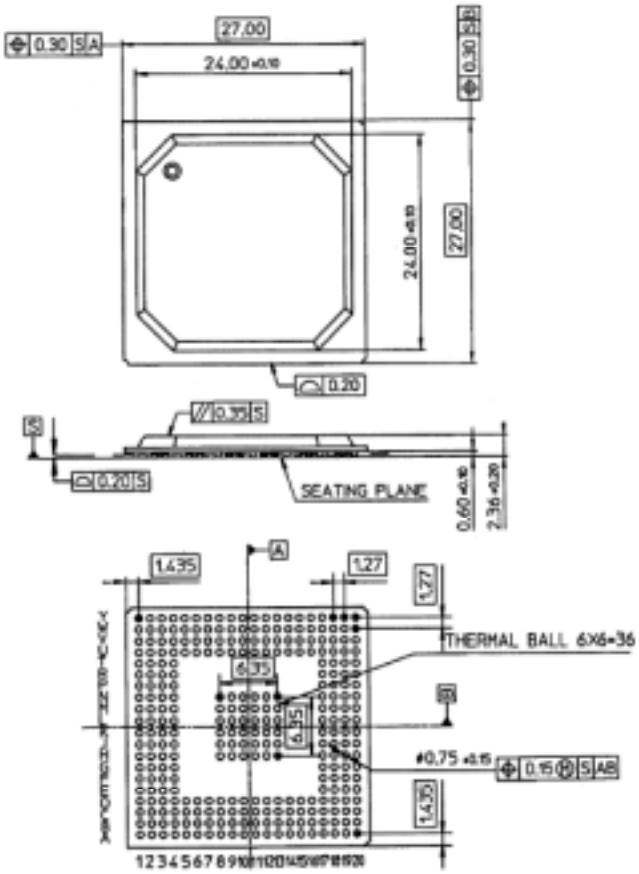
Refer to the instruction of AM33-2 core in the instruction manual of the MN103E series.

CHAPTER 22

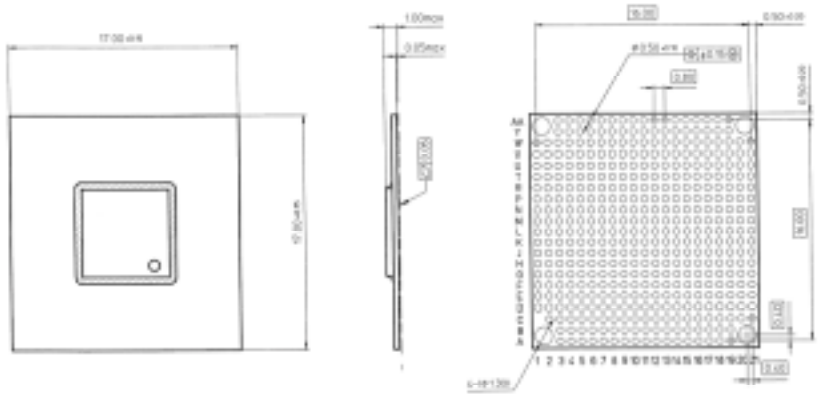
APPENDIX

22.4. Outer dimensions

MN103E010HRA



MN103E040HYB



TOP VIEW　　BOTTOM VIEW

Revision Record

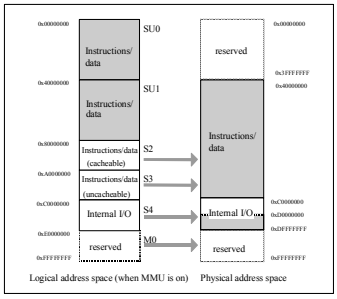
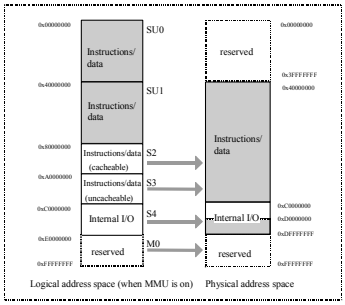
The revised parts of MN103E010H LSI User's manual from version 1.0 to version 2.0 are shown below.

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
Cover Page				
-	MN103E010H LSI User's manual	-	MN103E010H/040H LSI User's manual	Changed
All chapters				
-	MN103E010HYB	-	MN103E040HYB	Changed
1.2 Features				
31-32	○ Interrupts 42 sources (39 groups) Internal interrupts: …1; double fault timer: 14 …)	41	○ Interrupts 41 sources (42 groups: 3 of them are system-reserved.) Internal interrupts: …1; timer: 14 …)	Changed/ Deleted
35	○ IrDA interface IrDa 1.0 SIR (−115.2 Kb/s, half-duplex) UART (−1.5 Mbp/s, full-duplex)	43	○ IrDA controller IrDa 1.0 SIR (-115.2 Kb/s, half-duplex) UART (-1.5 Mbp/s, full-duplex)	Changed
35	○ I2C interface	43	○ I2C controller	Changed
1.4.1 Pin assignments				
37	Figure 2 MN103E010HYB pin assignments VSS covered with gray : A14, AA19, AA18	45	Figure 2 MN103E040HYB pin assignments VSS not covered with gray : A14, AA19, AA18	Changed / Deleted
37	Description for Figure 2 MN103E010HYB pin assignments VSS	45	Description for Figure 2 MN103E040HYB pin assignments VSS (including AVSS, PVSS)	Addition
1.4.2 Pin functions				
49	Category: 8-bit timer Pin name: and Pin functions: TM0IO – TM3IO Event count input/toggle output	57	Category: 8-bit timer Pin name: and Pin functions: TM0IO Timer 0 input/output TM1IO Timer 1 input/output TM2IO Timer 2 input/output TM3IO Timer 3 input/output	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
49	Category: 16-bit timer Pin name: and Pin functions: TM04O – TM11IO Event count input/toggle output	57-58	Category: 16-bit timer Pin name: and Pin functions: TM4IO Timer 4 input/output TM5IO Timer 5 input/output TM6IOA Timer 6 input/output A TM6IOB Timer 6 input/output B TM7IO Timer 7 input/output TM8IO Timer 8 input/output TM9IO Timer 9 input/output TM10IO Timer 10 input/output TM11IO Timer 11 input/output	Changed
51	Analog front end	59	Analog front end interface	Addition
51	I2C interface	59	I2C controller	Changed
51	IrDA interface	59	IrDA controller	Changed
1.5 Register set				
56	Table 10 Memory bus controller registers Initial value: 0x00000100 0x00000000 0x00000000	64	Table 10 Memory bus controller registers Initial value: 0xAA96061C 0x0000F200 0x0000F200	Changed
56	Table 11 DMA controller registers Name: DMA #0 DMA #1 DMA #2 DMA #3	64	Table 11 DMA controller registers Name: DMA DMA DMA DMA	Deleted
57	Table 13 16-bit timer registers Address: 0xD4003094 Name: TM6BR Symbol: Timer 6 base register Number of bits: 16 Initial value: 0x0000 Access size: 8, 16, 32			Deleted

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
58	Table 13 16-bit timer registers	66	Table 13 16-bit timer registers Address: 0xD4003071 Symbol: TMPCNT Name: Timer prescaler control register Number of bits : 8 Initial value: 0x00 Access size: 8	Addition
60	Table 17 Analog Front end registers Symbols: Initial value: AFESYS 0x0000 AFEINTM 0x0003 AFESTAT 0x00FF AFECTR 0x0048 AFESEC 0x00C0	68	Table 17 Analog Front end interface registers Symbols: Initial value: AFESYS 0x0003 AFEINTM 0x00FF AFESTAT 0x0048 AFECTR 0x0300 AFESEC 0x000C	Changed/ Addition
2.3.2.1	EPSW/PSW: Processor Status Word			
74	Bit description (Bit 20): <Programming note> if the floating-point unit has been implemented in the hardware.	82	Bit description (Bit 20): <Programming note> if the floating-point unit hardware has been implemented.	Changed
2.3.2.1	Multiply/Divide Register			
77	Extended Multiply/Divide Register	85	Multiply/Divide Register	Changed
77	This register is used by the fast multiply instruction that is executed by the extended operation unit. The register stores the upper 32 bits of the 64-bit multiplication result. For details, refer to the description of operation of individual instructions	85	This register is used in the multiply/divide instructions. In the case of multiplication, this stores the upper 32 bits of 64-bit multiplication result. In the case of division, this stores the 32-bit surplus.	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
2.3.4.2	CPU mode register			
85	Bit description (bit 4): The CPU changes back to NORMAL mode by an interrupt after the waiting time for the oscillation stabilization that the watchdog timer set.	93	Bit description (bit 4): The CPU changes back to NORMAL mode by an interrupt after confirming the operational stabilization of the clock generator.	Changed
2.3.4.4	CPU Revision Register			
87	Bit description (bits 27-24): Each core type is defined. The following table shows AM33-2 core release type. This LSI belongs to 0001. CRTYP E[3:0] / Core release type name 0000 / - 0001 / AM33-2-18LM-R2 0010 / AM33-2-18LM-R3 Others / Reserved	95	Bit description (bit 27-24): This LSI reads out 0001.	Changed
88	Bit description (bits 3-0): This field indicates the name of the CPU core type.	95	Bit description (bits 3-0): This field indicates the name of the CPU core release type.	Addition
2.3.4.6	Supervisor Interrupt Status Register			
90	Bit description (bit 31):	97	Bit description (bit 31): Multi-synchronous exception	Addition
2.3.4.9	Trap Base Register			
95	Bit table (Bits 23-0): R/W: R/W	102	Bit table (Bits 23-0): R/W: R	Changed
2.3.4.11	Process Identifier Register			
98	Bit description (Bits 5-11)	105	Bit description (Bits 5-8)	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
2.6.1 Overview of Interrupts				
119	More interrupts can be handled by grouping multiple interrupts into one level through an interrupt controller that is external to the core.	126	Each interrupt can be grouped into these levels by the LSI interrupt controller.	Changed
2.6.2.2 Nonmaskable Interrupts				
121	… the NMI entry (TBA[31:0] + 0x240 or 0x248) regardless …	128	… the NMI entry (TBA[31:0] + 0x008) regardless …	Changed
2.6.2.4 MMU Exception				
124	<Programming Note> EPSW.T is saved, not cleared when a debug mode is 0 or 1.	131		Deleted
2.7.1 Address space				
148	Table of Address space when using an MMU 	155	Table of Address space when using an MMU 	Changed
2.9.2.1 Floating-point format				
172	Quiet NaN (qNaN) 0xFFF000000000001-	181	Quiet NaN (qNaN) 0xFFF0000000000001-	Addition
4.2 Features				
186	● Supplies a programmable twofold or fourfold input frequency as the CPU clock (MCLK).	192	● Supplies a twofold or fourfold input frequency (FRQS-pin setting) as the CPU clock (MCLK).	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
5.6 Memory space				
196	0xBC000000-0xBFFFFFF	203	0xBC000000-0xBFFFFFFF	Addition
8.3 Description of Registers				
233	Memory bus controller register Initial value: 0x00000100 0x00000000 0x00000000	239	Memory of bus controller register Initial value: 0xAA96061C 0x0000F200 0x0000F200	Changed
8.4.1 Connection example				
240	Connection of the addresses, data, and control signals to SDRAM: MA 11:0	246	Connection of the addresses, data, and control signals to SDRAM: MA 14:0	Changed
8.4.5 Access Data Alignment				
241	The data alignment and the status of the byte access strobe (XMBE[3:0]) during …	247	The data alignment and the status of the byte access strobe (XMBE[1:0]) during …	Changed
241	Status of the byte access strobe (XMBE[3:0])	247	Status of the byte access strobe (XMBE[1:0])	Changed
8.4.8 Timing diagram				
243-250	All figures of 8.4.8.1.1 Power up sequence setting to 8.4.8.1.15: DQMU DQML MBE[1:0]	249-256	All figures of 8.4.8.1.1 Power up sequence setting to 8.4.8.1.15: XMBE[1] XMBE[0] MA[14:13]	Changed
244	Mode register setting (MBE[1:0], MA[12:0])	250	Mode register setting (MA[14:0])	Changed
247	Byte Access … is masked through asserting DQML (DQMU) at issuing a write command.	253	Byte Access … is disabled through negating XMBE [0] (XMBE[1]) at issuing a write command.	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
251-252	All figures of 8.4.8.2.1 Power up sequence setting to 8.4.8.2.4: DOMU/L MBE[1:0]	257-258	All figures of 8.4.8.2.1 Power up sequence setting to 8.4.8.2.4: XMBE[1:0] MA[14:13]	Changed
9.3 Description of registers				
257	Name: DMA #0 DMA #1 DMA #2 DMA #3	263	Name: DMA DMA DMA DMA	Deleted
10.3 Description of registers				
268	8-bit timer register Symbol: TMMD	274	8-bit timer register Symbol: TM0MD	Addition
10.3.1 Timer mode register				
269	Address: 0xD4003000+(0x1*n)	274	Address: TM0MD:0xD4003000 TM1MD:0xD4003001 TM2MD:0xD4003002 TM3MD:0xD4003003	Addition
269	Bit description (Bits 2-0) TMnCK[2:0]: Timer 3 010: 1/32 IOLK	275	Bit description (Bits 2-0) TMnCK[2:0]: Timer 3 010: 1/32 IOCLK	Addition
10.3.2 Timer Base Register				
270	Address: 0xD4003010+(0x1*n)	276	Address: TM0BR:0xD4003010 TM1BR:0xD4003011 TM2BR:0xD4003012 TM3BR:0xD4003013	Addition
10.3.3 Timer binary counter				
270	Address: 0xD4003020+(0x1*n)	276	Address: TM0BC:0xD4003020 TM1BC:0xD4003021 TM2BC:0xD4003022 TM3BC:0xD4003023	Addition

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
11.2 Features				
276	Timers 4, 5 and 7 to 11 ● Clock sources Internal clocks: IOCLK (25MHz)	282	Timers 4, 5 and 7 to 11 ● Clock sources Internal clocks: IOCLK	Deleted
276	Timers 6 ● Clock sources Internal clocks: IOCLK (25MHz)	282	Timers 6 ● Clock sources Internal clocks: IOCLK	Deleted
11.4 Description of Registers				
278	Table 47 16-bit timer register Address: 0xD4003094 Symbol: TM6BR Name: Timer 6 base register Number of bits: 16 Initial value: 0x0000 Access size: 8, 16, 32	284		Deleted
278	16-bit timer register	284	16-bit timer register Address: 0xD4003071 Symbol: TMPSCNT Name: Timer prescaler control register Number of bits: 8 Initial value: 0x00 Access size: 8	Addition
11.4.1 Timer mode register				
279-280	16-bit timer clock source 010: Timer 7 : 1/32 IOLK 010: Timer 11: 1/32 IOLK	285-256	16-bit timer clock source 010: Timer 7 : 1/32 IOCLK 010: Timer 11: 1/32 IOCLK	Addition
11.4.5 Timer 6 compare capture A mode register				
285	Bit description (Bits 7-6) Timer 6 compare A register mode flag 10: Register (single edge) 11: Register (both edge)	291	Bit description (Bits 7-6) Timer 6 compare capture register A mode flag 10: Capture register (single edge) 11: Capture register (both edge)	Changed/ Addition

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
11.4.6 Timer 6 compare capture B mode register				
286	Bit description (Bits 7-6) Timer 6 compare B register mode flag 10: Register (single edge) 11: Register (both edge)	292	Bit description (Bits 7-6) Timer 6 compare capture register B mode flag 10: Capture register (single edge) 11: Capture register (both edge)	Changed/ Addition
12.2.1 Serial Interface 0 (Serial interface 1)				
298	<Clock Synchronous Mode> Clock source 1/8 or 1/32 of IOCLK (30.375 MHz) Maximum transfer rate 15.2 Mbps (IOCLK = 30.375MHz)	304	<Clock Synchronous Mode> Clock source 1/8 or 1/32 of IOCLK Maximum transfer rate 7.25 Mbps (IOCLK = 30MHz)	Deleted/ Changed
298	<Start-Stop Synchronous Mode> Maximum transfer rate 38.8 kbps (IOCLK = 30.375MHz)	304	<Start-Stop Synchronous Mode> Maximum transfer rate 38.8 kbps (IOCLK = 30MHz)	Changed
12.2.2 Serial Interface 2				
299	Maximum transfer rate 233.28 kbps (IOCLK = 30.375 MHz)	305	Maximum transfer rate 233.28 kbps (IOCLK = 30MHz)	Changed
13.3 Interrupt Signal Assignments				
320	Interrupt source: Group 8 Purpose/point of connection: Timer 6 underflow	328	Interrupt source: Group 8 Purpose/point of connection: Timer 6 overflow	Changed
15.2 Features				
335	● Serial communications with AFE devices used as software modems. Parallel-serial conversion of output data and serial-parallel conversion of input data	345	● Serial communications with analog front end devices. Parallel-to-serial conversion of output data and serial-to-parallel conversion of input data	Deleted/ Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
15.3 Register				
336	Table 61 Analog Front end registers Symbols: Initial value: AFESYS 0x0000 AFEINTM 0x0003 AFESTAT 0x00FF AFECTR 0x0048 AFESEC 0x00C0	340	Table Analog Front end interface registers Symbols: Initial value: AFESYS 0x0003 AFEINTM 0x00FF AFESTAT 0x0048 AFECTR 0x0300 AFESEC 0x000C	Addition/ Changed
15.3.3 Analog front end status register				
338	Bit table (Bits 7-0): R/W: R/W	348	Bit table (Bits 7-0): R/W: R	Deleted
15.3.4 analog front end control register				
339	Bit table (Bits 15-0): Initial value: 0	349	Bit table (Bits 15-0): Initial value: 0300	Changed
15.4.1 Data transmit and receive				
343	Figure 85 AFE interface configuration : MN103E010	353	Figure: AFE interface configuration :	Deleted
15.4.4 Example Connections with AFE Devices				
344	Example Connections with AFE Devices MN103E010	354	Example Connections with AFE Devices AFE	Deleted/Addition
18.3 Registers				
375-410	UART/SIR mode MIR/FIR mode SIR/MIR/FIR mode	385-420	UART mode/SIR mode MIR mode/FIR mode SIR mode/MIR mode/FIR mode	Addition
18.3.1.5 IrDA interrupt identification register				
377	Bit description (Bit 3) 0: Timeout interrupt 1: No timeout interrupt	387	Bit description (Bit 3) 0: No timeout interrupt 1: Timeout interrupt	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
18.3.1.7 IrDA FIFO control register				
379	Bit description (bits 5-4) 11:16/30	389	Bit description (bits 5-4) 11:16/32	Changed
18.3.1.13 IrDA extended link status register				
385	Bit description (bit 2) In receiving, when the CRC result is an error and the final data of the receive frame reaches the bottom of the FIFO, "1" is set. This is "0" at reset or by reading this register.	395	Bit description (bit 2) At receiving, 1 is set when the result of CRC is an error and the final data of the receive frame reaches the bottom of FIFO.	Changed
18.3.3.2 IrDA extended control register 2				
389	This is used for selecting the sizes of the transmit FIRO and receive FIRO.	400	This is used for selecting the sizes of the transmit FIFO and receive FIFO.	Changed
18.3.5.2 IrDA infrared control register 1				
393	Bit description (Bit 1) 0: Return the initial value during reading out the IrDA timer initial value lower register/IrDA timer initial value upper register. 1: Return the executing count value during reading out the IrDA timer initial value lower register/IrDA timer initial value upper register.	405	Bit description (Bit 1) 0: Return the executing count value during reading out the IrDA timer initial value lower register/IrDA timer initial value upper register. 1: Return the initial value during reading out the IrDA timer initial value lower register/IrDA timer initial value upper register.	Changed
18.3.5.3.1 IrDA transmit frame length lower count register				
394	Bit table (bits 7-0) Bit name: TFLU[7:0]	405	Bit table (bits 7-0) Bit name: TFLL[7:0]	Changed
18.3.5.4.1 IrDA transmit frame maximum-length lower count register				
395	Bit table (bits 7-0) Description: Reads 8 MSB of the number of the data bytes during transmitting the frame.	406	Bit table (bits 7-0) Description: Reads 8 LSB of the number of the data bytes during transmitting the frame.	Changed

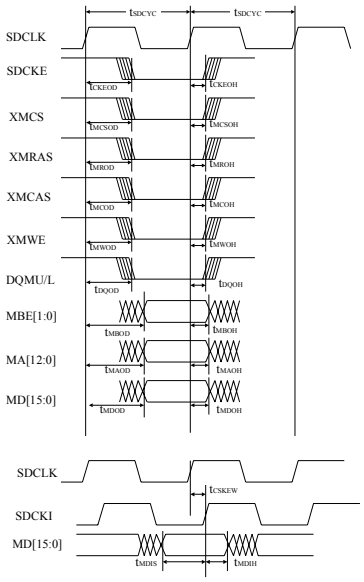
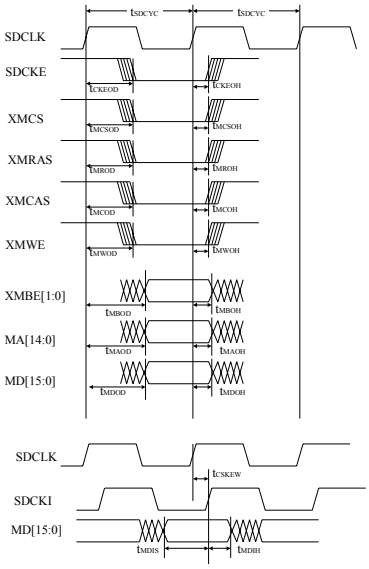
Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
18.3.6.2 IrDA infrared control register 2				
398	Bit description (bit 6) Bit name : SRTH	409	Bit description (bit 6) Bit name : SFTH	Changed
18.3.8.3 IrDA extended control register 4				
408	Bit : 3	419	Bit : 5	Changed
18.4.6.1 Interrupt priority				
416	This is negated through reading the RDR.	426	This is negated through reading the IrDA receive data register.	Changed
18.4.11 Interaction pulse (SIP) transmission				
420	…of the IrDA pipeline mode register and is …	431	…of the IrDA mode control register and is …	Changed
19.1 General				
422	… includes two independent I2C interfaces that support …	434	… includes two independent I2C controllers that support …	Changed
19.2 Features				
422	Open drain output (SDA/SDL)	434	Open drain output (SDA/SCL)	Changed
20.2.1 I/O port 0				
436	Port 0 is an 8-bit I/O port. Port 0 is also used for timer clock I/O (TM0-TM6B) and test signal output (EYECLK, EYED) for the analog front end.	448	Port 0 is an 8-bit I/O port. Port 0 is also used for timer clock I/O (TM0IO-TM6IOB) and test signal output (EYECLK, EYED) for the analog front end interface.	Addition
436	Table 82 Configuration of port 0 TM0 TM2 TM3 TM4 TM5 TM6A TM6B	448	Configuration of port 0 TM0IO TM2IO TM3IO TM4IO TM5IO TM6IOA TM6IOB	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
20.2.2	I/O port 1			
436	Configuration of port 0 TM7 TM8 TM9 TM10 TM11	449	Configuration of port 0 TM7IO TM8IO TM9IO TM10IO TM11IO	Changed
20.2.3	I/O port 2			
437	…DRAM CAS signal (XSCAS[3:0]), address buffer enable (XSABOE), and for initial settings pins at time of reset (BOOTBW, BOOTSEL, CMOD, CLKIO).	449	…DRAM CAS signal (XSCAS[3:0]) and for initial settings pins at time of reset (BOOTBW, BOOTSEL, CMOD, CKIO).	Deleted/ Changed
20.2.4	I/O port 3			
437	… is also used for analog front end pins …	449	… is also used for analog front end interface pins …	Addition
20.2.5	I/O port 4			
437	… is also used for I2C interface pins …	449	… is also used for I2C controller pins …	Changed
20.2.6	I/O port 5			
438	… is also used for IrDA interface pins …	450	… is also used for IrDA controller pins …	Changed
20.3.4	Port 0 Timer Pin Input/Output Control Register			
443	Bit description (bit 7) PIO0[7] Timer Pin Input/Output Setting …and functions as TM6B pin.	455	Bit description (bit 7) TM6IOB Pin Input/Output Setting …and functions as TM6IOB pin.	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
443	Bit description (bit 6) PIO0[6] Timer Pin Input/Output Setting …and functions as TM6A pin.	455	Bit description (bit 6) TM6IOA Pin Input/Output Setting …and functions as TM6IOA pin.	Changed
443	Bit description (bit 5) PIO0[5] Timer Pin Input/Output Setting …and functions as TM5 pin.	455	Bit description (bit 5) TM5IO Pin Input/Output Setting …and functions as TM5IO pin.	Changed
443	Bit description (bit 4) PIO0[4] Timer Pin Input/Output Setting …and functions as TM4 pin.	455	Bit description (bit 4) TM4IO Pin Input/Output Setting …and functions as TM4IO pin.	Changed
443	Bit description (bit 3) PIO0[3] Timer Pin Input/Output Setting …and functions as TM3 pin.	455	Bit description (bit 3) TM3IO Pin Input/Output Setting …and functions as TM3IO pin.	Changed
443	Bit description (bit 2) PIO0[2] Timer Pin Input/Output Setting …and functions as TM2 pin.	455	Bit description (bit 2) TM2IO Pin Input/Output Setting …and functions as TM2IO pin.	Changed
443	Bit description (bit 1) PIO0[1] Timer Pin Input/Output Setting …and functions as TM1 pin.	455	Bit description (bit 1) TM1IO Pin Input/Output Setting …and functions as TM1IO pin.	Changed
443	Bit description (bit 0) PIO0[0] Timer Pin Input/Output Setting …and functions as TM0 pin.	455	Bit description (bit 0) TM0IO Pin Input/Output Setting …and functions as TM0IO pin.	Changed
20.3.8 Port 1 timer Input/Output Control Register				
446	Bit description (bit 4) PIO1[4] Timer Pin Input/Output Setting …and functions as TM7 pin.	458	Bit description (bit 4) TM7IO Pin Input/Output Setting …and functions as TM7IO pin.	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
446	Bit description (bit 3) PIO1[3] Timer Pin Input/Output Setting …and functions as TM8 pin.	458	Bit description (bit 3) TM8IO Pin Input/Output Setting …and functions as TM8IO pin.	Changed
446	Bit description (bit 2) PIO1[2] Timer Pin Input/Output Setting …and functions as TM9 pin.	458	Bit description (bit 2) TM9IO Pin Input/Output Setting …and functions as TM9IO pin.	Changed
446	Bit description (bit 1) PIO1[1] Timer Pin Input/Output Setting …and functions as TM10 pin.	458	Bit description (bit 1) TM10IO Pin Input/Output Setting …and functions as TM10IO pin.	Changed
446	Bit description (bit 0) PIO1[0] Timer Pin Input/Output Setting …and functions as TM11 pin.	458	Bit description (bit 0) TM11IO Pin Input/Output Setting …and functions as TM11IO pin.	Changed
20.1 Absolute maximum ratings				
458	• In MN103E010H, 3.3V must be provided for PAD, ADC, PLL, and 1.8V must be provided for RTC internal (digital) circuit.	470	• In this LSI, 3.3V must be provided for PAD, ADC, PLL and RTC, and 1.8V must be provided for an internal (digital) circuit.	Changed
20.2 Operation al requirements				
459	Operational requirements Item: B4, Acceptable value (minimum): 1.71	471	Operational requirements Item: B4, Acceptable value (minimum): 3.175	Changed
459	(Note) Equal voltage levels must be supplied to V_{DD33}, AV_{DD}, PV_{DD} and RV_{DD}.	471	(Note) Equal voltage levels must be supplied to V_{DD33}, AV_{DD}, and PV_{DD}.	
21.3 DC characteristics				
468	C67	481	C66	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
21.5 AC characteristics				
471	21.5.1 Reset signal timing These tables show AC characteristics.	483	These following tables show AC characteristics. 21.5.1 Reset signal timing	Transferred/Addition
21.5.2 Clock timing				
472	AC characteristics (2) Item: E12, E13 Condition: C_L=50 pF	484	AC characteristics (2) Item: E12, E13 Condition: C_L=50 pF	Changed
21.5.3 System bus signal timing				
475	AC characteristics (3)	487	AC characteristics (3) Heading item: System bus signal input timing	Addition
482	AC characteristics (5) (Item: E64) output hold time (SMWE)	494	AC characteristics (5) (Item: E64) output hold time (XMWE)	Changed
482	AC characteristics (5) (Item: E65) DQMU/DQML output delay time	494		Deleted
482	AC characteristics (5) (Item: E66) DQMU/DQML output hold time	494		Deleted
482	AC characteristics (5) (Item: E67) output delay time (MBE)	494	AC characteristics (5) (Item: E67) output delay time (XMBE)	Changed
482	AC characteristics (5) (Item: E68) output hold time (MBE)	494	AC characteristics (5) (Item: E68) output hold time (XMBE)	Changed

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
483	Figure 101 Memory bus signal input/output timing 	495	Figure Memory bus signal input/output timing 	Changed
21.5.8 Analog front end signal timing				
487	AC characteristics (9) AFE	499	AC characteristics (9) Analog front end	Changed
21.5.10 I2C controller signal timing				
489	I2C interface signal timing	501	I2C controller signal timing	Changed
Chapter 22 Appendix				
-	Appendix Appendix A Pin list Appendix B Address map Appendix C Instruction set list Appendix D Outer dimensions		Chapter 22 Appendix 22.1 Pin list 22.2 Address map 22.3 Instruction set list 22.4 Outer dimensions	Changed
22.1 Pin list				
-	PIO2[0]/XSCAS[0]/BOOTBW PIO2[1]/XSCAS[1]/BOOTSEL PIO2[2]/XSCAS[2] PIO2[3]/XSCAS[3]/CKIO PIO2[4]/XSABOE/CMOD	508	PIO2[0]/BOOTBW PIO2[1]/BOOTSEL PIO2[2] PIO2[3]/CKIO PIO2[4]/CMOD	Deleted

Page	Old version (1.0)	Page	Revised Version (2.0)	Definition
22.2 Address Map				
-	INTC 5-4: G41CR	512	INTC 5-4: G4ICR	Changed
-	16-bit timer 5-4: TM6BR	512		Deleted
-	IO port registers 0xDB00000C 0xDB00010C	512	IO port registers 0xDB00000C: P0TMIO 0xDB00010C: P0TMIO	Addition
22.4 Outer dimensions				
-	MN103E010HYB	515	MN103E040HYB	Changed
Colophon				
-	MN103E010H LSI User's Manual April, 2002 1st Edition	-	MN103E010H/040H LSI User's Manual August, 2002 2nd Edition	Changed

MN103E010H/040H
LSI User's Manual

August, 2002 2nd Edition

Issued by Matsushita Electric Industrial Co., Ltd.

Semiconductor Company, Matsushita Electric Industrial Co., Ltd.

Nagaokakyo, Kyoto 617-8520, Japan
Tel: (075) 951-8151
http://www.panasonic.co.jp/semicon/

SALES OFFICES

■ NORTH AMERICA

●U.S.A. Sales Office:

Panasonic Industrial Company [PIC]

• New Jersey Office:
Two Panasonic Way Secaucus, New Jersey 07094 U.S.A.
Tel: 1-201-348-5257 Fax:1-201-392-4652

• Chicago Office:
1707 N. Randall Road Elgin, Illinois 60123-7847 U.S.A.
Tel: 1-847-468-5720 Fax:1-847-468-5725

• Milpitas Office:
1600 McCandless Drive Milpitas, California 95035 U.S.A.
Tel: 1-408-942-2912 Fax:1-408-946-9063

• Atlanta Office:
1225 Northbrook Parkway Suite 1-151 Suwanee, GA 30024 U.S.A.
Tel: 1-770-338-6953 Fax:1-770-338-6849

• San Diego Office:
9444 Balboa Avenue, Suite 185, San Diego, California 92123 U.S.A.
Tel: 1-619-503-2903 Fax:1-858-715-5545

●Canada Sales Office:

Panasonic Canada Inc. [PCI]
5770 Ambler Drive 27 Mississauga, Ontario, L4W 2T3 CANADA
Tel: 1-905-238-2315 Fax:1-905-238-2414

■ LATIN AMERICA

●Mexico Sales Office:

Panasonic de Mexico, S.A. de C.V. [PANAMEX]
Amores 1120 Col. Del Valle Delegacion Benito Juarez C.P. 03100 Mexico, D.F. MEXICO
Tel: 52-5-488-1000 Fax:52-5-488-1073

• Guadalajara Office:
SUCURSAL GUADALAJARA
Av. Lazaro Cardenas 2305 Local G-102 Plaza Comercial Abastos; Col. Las Torres Guadalajara, Jal. 44920 MEXICO
Tel: 52-3-671-1205 Fax:52-3-671-1256

●Brazil Sales Office:

Panasonic do Brasil Ltda. [PANABRAS]
Caixa Postal 1641, Sao Jose dos Campos, Estado de Sao Paulo
Tel: 55-12-335-9000 Fax:55-12-331-3789

■ EUROPE

●Europe Sales Office:

Panasonic Industrial Europe GmbH [PIE]

• U.K. Sales Office:
Willoughby Road, Bracknell, Berks., RG12 8FP, THE UNITED KINGDOM
Tel: 44-1344-85-3671 Fax:44-1344-85-3853

• Germany Sales Office:
Hans-Pinsel-Strasse 2 85540 Haar, GERMANY
Tel: 49-89-46159-119 Fax:49-89-46159-195

■ ASIA

●Singapore Sales Office:

Panasonic Semiconductor of South Asia [PSSA]
300 Beach Road, #16-01, The Concourse, Singapore 199555 THE REPUBLIC OF SINGAPORE
Tel: 65-6390-3688 Fax:65-6390-3689

●Malaysia Sales Office:

Panasonic Industrial Company (M) Sdn. Bhd. [PICM]

• Head Office:
Tingkat 16B, Menara PKNS Petaling Jaya, No.17, Jalan Yong Shook Lin 46050 Petaling Jaya, Selangor Darul Ehsan, MALAYSIA
Tel: 60-3-7951-6601 Fax:60-3-7954-5968

• Penang Office:
Suite 20-07,20th Floor, MWE Plaza, No.8, Lebuh Farquhar,10200 Penang, MALAYSIA
Tel: 60-4-201-5113 Fax:60-4-261-9989

• Johore Sales Office:
Menara Pelangi, Suite8.3A, Level8, No.2, Jalan Kuning Taman Pelangi, 80400 Johor Bahru, Johor, MALAYSIA
Tel: 60-7-331-3822 Fax:60-7-355-3996

●Thailand Sales Office:

Panasonic Industrial (THAILAND) Ltd. [PICT]
252-133 Muang Thai-Phatra Complex Building, 31st Fl. Rachadaphisek Rd., Huaykwang, Bangkok 10320, THAILAND
Tel: 66-2-693-3428 Fax:66-2-693-3422

●Philippines Sales Office: [PISP]

Panasonic Indsutrial Sales Philippines Division of Matsushita Electric Philippines Corporation
102 Laguna Boulevard,Bo.Don Jose Laguna Technopark, Santa. Rosa, Laguna 4026 PHILIPPINES
Tel: 63-2-520-8615 Fax:63-2-520-8629

●India Sales Office:

National Panasonic India Ltd. [NPI]
E Block, 510, International Trade Tower Nehru Place, New Delhi_110019 INDIA
Tel: 91-11-629-2870 Fax:91-11-629-2877

●Indonesia Sales Office:

P.T.MET & Gobel [M&G]
JL. Dewi Sartika (Cawang 2) Jakarta 13630, INDONESIA
Tel: 62-21-801-5666 Fax:62-21-801-5675

●China Sales Office:

Panasonic Industrial (Shanghai) Co., Ltd. [PI(SH)]
Floor 6, Zhong Bao Mansion, 166 East Road Lujian Zui, PU Dong New District, Shanghai, 200120 CHINA
Tel: 86-21-5866-6114 Fax:86-21-5866-8000

Panasonic Industrial (Tianjin) Co., Ltd. [PI(TJ)]
Room No.1001, Tianjin International Building 75, Nanjin Road, Tianjin 300050, CHINA
Tel: 86-22-2313-9771 Fax:86-22-2313-9770

Panasonic SH Industrial Sales (Shenzhen) Co., Ltd. [PSI(SZ)]
7A-107, International Bussiness & Exhibition Centre, Futian Free Trade Zone, Shenzhen 518048, CHINA
Tel: 86-755-8359-8500 Fax:86-755-8359-8516

Panasonic Shun Hing Industrial Sales (Hong Kong) Co., Ltd. [PSI(HK)]
11th Floor, Great Eagle Center 23 Harbour Road, Wanchai, HONG KONG
Tel: 852-2529-7322 Fax:852-2865-3697

●Taiwan Sales Office:

Panasonic Industrial Sales (Taiwan) Co.,Ltd. [PIST]

• Head Office:
6F, 550, Sec. 4, Chung Hsiao E. RD. Taipei, 110, TAIWAN
Tel: 886-2-2757-1900 Fax:886-2-2757-1906

• Kaohsiung Office:
6th Floor, Hsin Kong Bldg. No.251, Chi Hsien 1st Road Kaohsiung 800, TAIWAN
Tel: 886-7-346-3815 Fax:886-7-236-8362

●Korea Sales Office:

Panasonic Industrial Korea Co., Ltd. [PIKL]
Kukje Center Bldg. 11th Fl., 191 Hangangro 2ga, Youngsan-ku, Seoul 140-702, KOREA
Tel: 82-2-795-9600 Fax:82-2-795-1542

050702

Printed in JAPAN