LAPIS SEMICONDUCTOR

FEUL696201-05

ML696201/69Q6203
User's Manual

32-bit General Purpose Single-chip Microcontroller

Issue Date: March 27, 2009

NOTICE

No copying or reproduction of this document, in part or in whole, is permitted without the consent of LAPIS Semiconductor Co., Ltd.

The content specified herein is subject to change for improvement without notice.

The content specified herein is for the purpose of introducing LAPIS Semiconductor's products (hereinafter "Products"). If you wish to use any such Product, please be sure to refer to the specifications, which can be obtained from LAPIS Semiconductor upon request.

Examples of application circuits, circuit constants and any other information contained herein illustrate the standard usage and operations of the Products. The peripheral conditions must be taken into account when designing circuits for mass production.

Great care was taken in ensuring the accuracy of the information specified in this document. However, should you incur any damage arising from any inaccuracy or misprint of such information, LAPIS Semiconductor shall bear no responsibility for such damage.

The technical information specified herein is intended only to show the typical functions of and examples of application circuits for the Products. LAPIS Semiconductor does not grant you, explicitly or implicitly, any license to use or exercise intellectual property or other rights held by LAPIS Semiconductor and other parties. LAPIS Semiconductor shall bear no responsibility whatsoever for any dispute arising from the use of such technical information.

The Products specified in this document are intended to be used with general-use electronic equipment or devices (such as audio visual equipment, office-automation equipment, communication devices, electronic appliances and amusement devices).

The Products specified in this document are not designed to be radiation tolerant.

While LAPIS Semiconductor always makes efforts to enhance the quality and reliability of its Products, a Product may fail or malfunction for a variety of reasons.

Please be sure to implement in your equipment using the Products safety measures to guard against the possibility of physical injury, fire or any other damage caused in the event of the failure of any Product, such as derating, redundancy, fire control and fail-safe designs. LAPIS Semiconductor shall bear no responsibility whatsoever for your use of any Product outside of the prescribed scope or not in accordance with the instruction manual.

The Products are not designed or manufactured to be used with any equipment, device or system which requires an extremely high level of reliability the failure or malfunction of which may result in a direct threat to human life or create a risk of human injury (such as a medical instrument, transportation equipment, aerospace machinery, nuclear-reactor controller, fuel-controller or other safety device). LAPIS Semiconductor shall bear no responsibility in any way for use of any of the Products for the above special purposes. If a Product is intended to be used for any such special purpose, please contact a ROHM sales representative before purchasing.

If you intend to export or ship overseas any Product or technology specified herein that may be controlled under the Foreign Exchange and the Foreign Trade Law, you will be required to obtain a license or permit under the Law.

Preface

This manual describes the hardware, software of the 32-bit microcontroller ML696201/69Q6203. The manuals shown below are also available, and should be consulted as necessary.

ARM Architecture Reference Manual

* Description of ARM instruction set architecture

ARM946E-S Technical Reference Manual

The above documents are published by ARM Corporation.
Please ensure that you refer to the latest versions.

Notation

Classification	Notation	Description
♦ Numeric value	xxh, xxH	Indicates a hexadecimal number. x: Any value in the range of 0 to F
	xxb	Indicates a binary number; "b" may be omitted. x: A value 0 or 1
♦ Unit	word, W	1 word = 16 bits
	byte, B	1 byte = 2 nibbles = 8 bits
	nibble, N	1 nibble = 4 bits
	maga-, M	10^6
	kilo-, K	$2^{10} = 1024$
	kilo-, k	$10^3 = 1000$
	milli-, m	10^{-3}
	micro-, μ	10^{-6}
	nano-, n	10^{-9}
	second, s (lower case)	second
♦ Symbol	x0hex	x indicates any value in the range of 0 to F of the high-order 4 bits.
♦ Terminology	"H" level, "1" level	Indicates high voltage signal levels V_{IH} and V_{OH} as specified by the electrical characteristics.
	"L" level, "0" level	Indicates low voltage signal levels V_{IL} and V_{OL} as specified by the electrical characteristics.

Table of Contents

Chapter 1 Overview

Chapter 2 CPU

Chapter 3 Address Map

Chapter 4 Mode Settings

Chapter 5 Clock Control

Chapter 6 Reset Control

Chapter 7 Power Consumption Control and System Control

Chapter 8 Interrupt Controller

Chapter 9 Built-in Memory

Chapter 10 External Memory Controller

Chapter 11 Direct Memory Access Controller

Chapter 12 GPIO

Chapter 13 Watchdog Timer (WDT)

Chapter 14 Timers

Chapter 15 PWM

Chapter 16 µPLAT-SIO (UART)

Chapter 17 A/D Converter

Chapter 18 Synchronous SIO

Chapter 19 I2C

Chapter 20 RTC

Chapter 21 Configuration Registers

Chapter 22 I2S Transmission

Chapter 23 I2S Reception

Chapter 24 NAND Flash Controller

Chapter 25 ATA IP (IDE Host Controller)

Chapter 26 HS USB

Chapter 27 Built-In Flash ROM

Chapter 28 JTAG

Chapter 29 Electrical Characteristics

Chapter 30 List of Registers

Chapter 31 Package Dimensions

Revision History

Chapter 1

Overview

Chapter 1 Overview

1.1 Overview

This LSI uses the 32-bit RISC CPU ARM946E™ developed by ARM as its core. It is a high performance general-purpose CMOS 32-bit single chip microcontroller embedded with integrated peripheral functions, including a high speed USB, an IDE controller, a DMA controller.
Its functionality can be enhanced by directly connecting memory such as ROM, SRAM and DRAM and peripheral devices using an external memory controller.
It has the most suitable configuration for secure or multimedia processing such as storage or digital equipment.
The ML696201 series microcontrollers are available in two models; ML696201 without built-in ROMs and ML69Q6203 with a built-in 512KB flash memory.
The features of the LSI are listed below.

Features:

- CPU
  - 32-bit RISC CPU (ARM946E)
  - Built-in 8KB instruction cache and 8KB data cache
  - Little endian format
  - Maximum operating frequency of 120 MHz
  - Instruction structure: Highly dense 32-bit long instructions and their subset 16-bit long instructions with high object code efficiency can be executed by switching between them.
  - General-purpose registers: 32 bits x 31
  - Built-in barrel shifter (The operations of the ALU and barrel shift can be executed by one instruction.)
  - Built-in multiplier (32 bits x 16 bits)
  - Built-in debug function (JTAG interface)
- Internal memory
  - Built-in 128KB SRAM (32K x 32 bits)
  - AHB bus connection
  - Built-in 16KB ROM for boot up (4K x 32 bits)
- µPLAT external memory controller
  - ROM (Flash) access function
  - Supporting 16-bit devices
  - Supporting Flash memory
  - SRAM access function
  - Supporting 16-bit devices
  - Supporting asynchronous SRAM
  - DRAM access function
  - Supporting 16-bit devices
  - Supporting SDRAM
  - Supporting distributed CBR
  - External I/O access function
  - I/O space of 2 banks
  - Supporting 16-bit devices
- µPLAT interrupt controller/extended interrupt controller
  - FIQ: One source (external source)
  - IRQ: 27 sources (internal sources: 23, external sources: 4)
  - Seven interrupt priority levels can be set for each interrupt source as priority.
- µPLAT system timer
  - 16-bit auto reload timer x 1 channel
  - Cycle: 2.133 µs to 139.5 ms
- uPLAT-SIO (UART)
  - Full-duplex start-stop synchronization method
  - Built-in baud rate generator

- DMA controller.
  - 4 channels
  - Fixed mode or round-robbin mode can be selected as the priority of the channels.
  - The cycle steal mode or burst mode can be selected as the bus access privilege request method.
  - Software requests and external requests are supported as DMA transfer requests.
  - Maximum transfer count: 65,536
  - Data transfer size: 8/16/32 bits
- High speed USB
  - Connectable to USB 2.0 and USB 1.1
  - Bulk transfer, control transfer, and interrupt transfer are supported.
  - Up to 4 levels of USB hub are supported.
- IDE Controller
  - PIO mode, multi-word DMA mode, and UltraDMA66 are suppported.
  - PIO mode: Supports up to mode 4.
  - Multi-word DMA mode: Supports up to mode 2.
  - UltraDMA mode: Supports up to mode 4.
  - Scatter-gather DMA function
  - Set as primary function (Secondary function cannot be used)
  - Only device 0 (master) can be used (Device 1 (slave) cannot be used)
  - AHB master and slave interface
  - Built-in data transfer FIFO: 8 × 32 bits (32 bytes)
- PWM
  - 16-bit resolution PWM x 1 channel
- Watchdog timer
  - 16-bit timer
  - Interval mode or watchdog timer mode can be selected.
  - An interrupt or a reset can be generated by setting.
  - Cycles that can be set: 8.7, 35.0, 140.0 or 559.2 ms
- Analog-to-digital converter
  - 10-bit sequential compare type x 4 channels
  - Sample hold function
  - Shortest conversion time: 6.7 μs
- I^2C bus controller
  - I^2C bus standard compliant controller x 1 channel
  - Can be operated only as a master device.
  - Communication speed: 100/400 kbps
- Timer
  - 16-bit auto reload timer x 3 channels
  - A different clock can be set for each channel.
  - One-shot mode or interval mode can be set for each channel.
  - Cycle: 0.133 μs to 2.237 sec
- Synchronous SIO
  - 8-bit clock synchronous serial port x 2 channels
  - Clock polarity can be selected.
  - LSB first or MSB first can be selected.
  - Master or Slave mode can be selected.
- Configuration register
  - Four 8-bit internal status setup registers
- NAND Flash memory controller
  - SmartMedia Standard 2000 compliant (512 bytes/sector)
  - Advanced NAND flash memories are supported (2048 bytes/sector).
  - SmartMedia of 8MB to 128MB is supported.
  - Built-in ECC circuit.
  - 512B/2048B auto write/read function
  - Switchable to the IDE controller using the IDEMODE pin

- RTC
  - 1-second clock generation function from 32.768 KHz
  - Built-in 32-bit 1-second clock counter
  - 32-bit compare interrupt function

- GPIO
  - Built-in GPIO of 16 x 5 channels (GPIOA, GPIOB, GPIOC, GPIOD, GPIOE) and 7 x 1 channel (GPIOF)
  - Input/output can be set in units of bits, and interrupt input can be set in units of bits.
  - GPIOA15 to GPIOA1500, GPIOB15 to GPIOB00 and GPIOC15 to GPIOC00 can be set to an external bus by setting the EXTBUS pin.
  - GPIOD15 to GPIOD00 can be used to select a secondary function in units of bits.
  - GPIOE15 to GPIOE11 can be used as external interrupts. Only PIOE15 is 5V tolerant input.
  - GPIOF06 to GPIOF00 function as IDE data if the IDE controller mode is set using the IDEMODE pin.
- I^2S transmission/reception
  - 32/44.1/48 kHz as well as 1/2 and 1/4 of these frequencies are supported as sampling frequencies.
  - System clock is 256 times the sampling rate.
  - Channel data length: 16 bits
  - With or without 1-bit delay, left/right reversible
- Flash memory
  - ML69Q6203: 512KB Flash ROM is embedded in the MCP.
  - ML696201: No Flash ROMs
- Clocks
  - Can connect a 48 MHz crystal oscillator and input an external clock directly.
  - The RTC section can connect a 32.768 kHz crystal oscillator.
  - Can connect an 11.2896 MHz crystal oscillator for I2S and input an external clock directly.
- Power management
  Power down mode: Can stop the power supply other than to the RTC section.
  - Stop mode: Stops clock supply to the main section including the processor by software.
  - Halt mode: Stops clock supply partially.
  - Clock gear: Can dynamically change clock to 1/1, 1/2, 1/4, 1/8, 1/16 or 1/32 of the clock input frequency by software.
  - Clock stop in units of functions: Can stop clock supply for each function by software.
- Package
  - 272-pin LFBGA, 0.65 mm ball pitch

1.2 Functional Block Configuration

1.2.1 Block Diagram

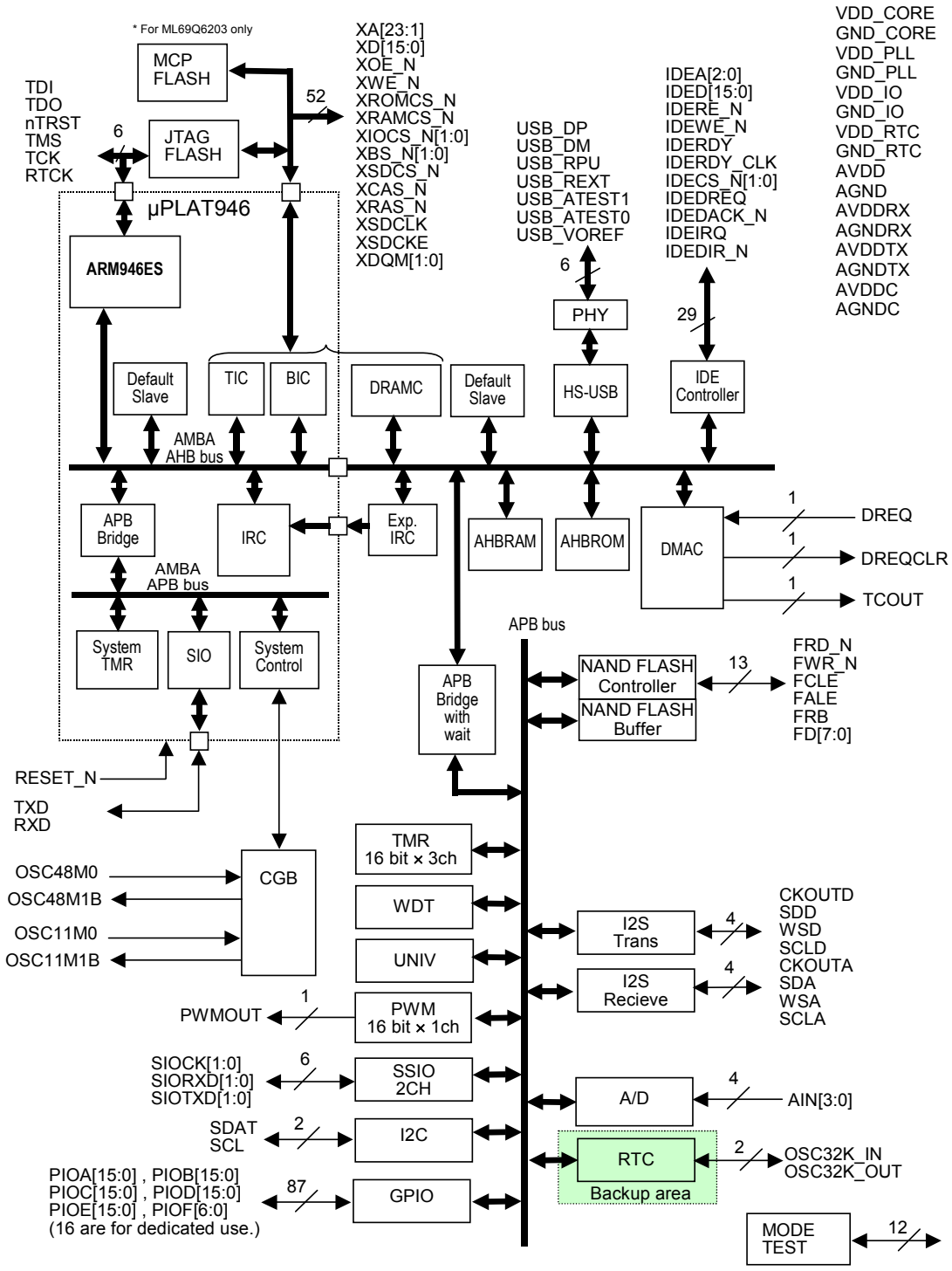


Figure 1-1 Block Diagram

1.2.2 MCP Connection Diagram

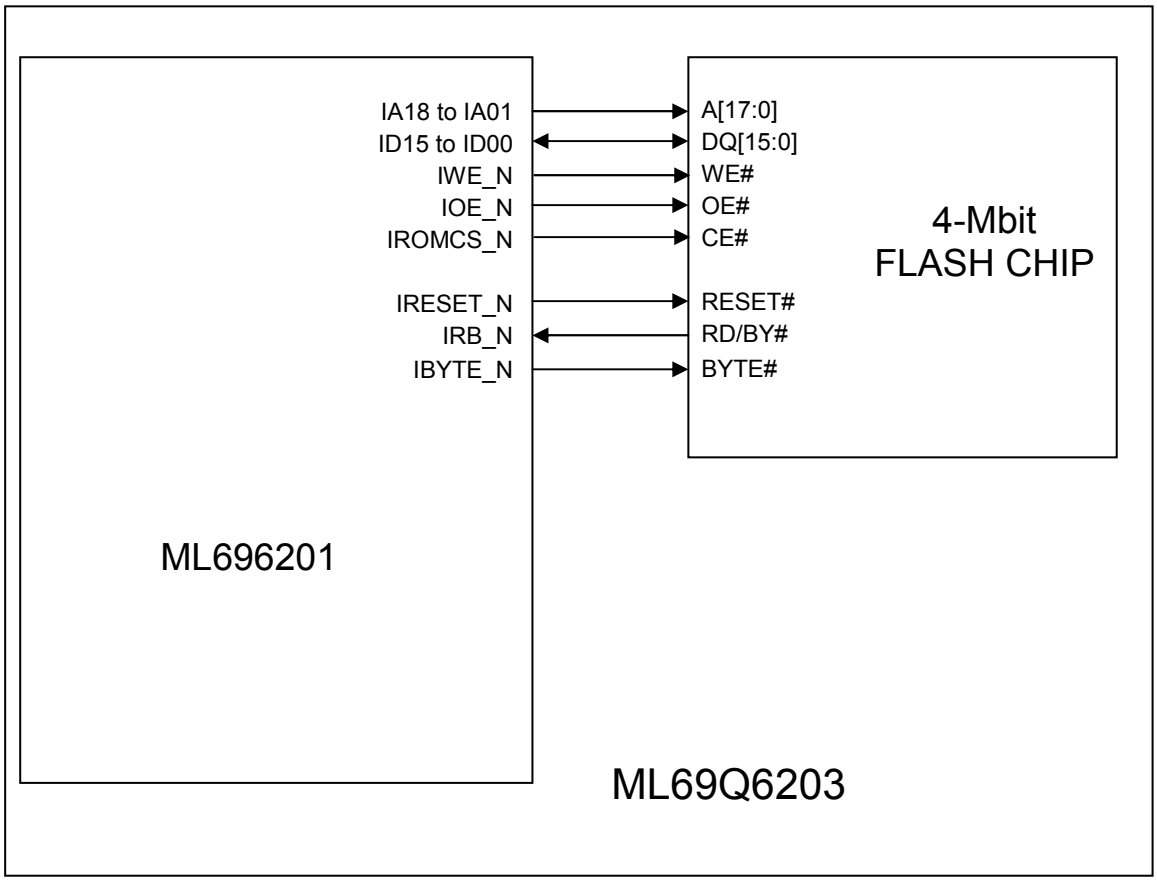


Figure 1-2 MCP Connection Diagram

1.3 Pins

1.3.1 Pin Configuration

1.3.1.1 Pin Configuration

AA21	Y21	W21	V21	U21	T21	R21	P21	N21	M21	L21	K21	J21	H21	G21	F21	E21	D21	C21	B21	A21
AA20	Y20	W20	V20	U20	T20	R20	P20	N20	M20	L20	K20	J20	H20	G20	F20	E20	D20	C20	B20	A20
AA19	Y19	W19	V19	U19	T19	R19	P19	N19	M19	L19	K19	J19	H19	G19	F19	E19	D19	C19	B19	A19
AA18	Y18	W18	V18	U18	T18	R18	P18	N18	M18	L18	K18	J18	H18	G18	F18	E18	D18	C18	B18	A18
AA17	Y17	W17	V17														D17	C17	B17	A17
AA16	Y16	W16	V16														D16	C16	B16	A16
AA15	Y15	W15	V15														D15	C15	B15	A15
AA14	Y14	W14	V14														D14	C14	B14	A14
AA13	Y13	W13	V13														D13	C13	B13	A13
AA12	Y12	W12	V12														D12	C12	B12	A12
AA11	Y11	W11	V11														D11	C11	B11	A11
AA10	Y10	W10	V10														D10	C10	B10	A10
AA9	Y9	W9	V9														D9	C9	B9	A9
AA8	Y8	W8	V8														D8	C8	B8	A8
AA7	Y7	W7	V7														D7	C7	B7	A7
AA6	Y6	W6	V6														D6	C6	B6	A6
AA5	Y5	W5	V5														D5	C5	B5	A5
AA4	Y4	W4	V4	U4	T4	R4	P4	N4	M4	L4	K4	J4	H4	G4	F4	E4	D4	C4	B4	A4
AA3	Y3	W3	V3	U3	T3	R3	P3	N3	M3	L3	K3	J3	H3	G3	F3	E3	D3	C3	B3	A3
AA2	Y2	W2	V2	U2	T2	R2	P2	N2	M2	L2	K2	J2	H2	G2	F2	E2	D2	C2	B2	A2
AA1	Y1	W1	V1	U1	T1	R1	P1	N1	M1	L1	K1	J1	H1	G1	F1	E1	D1	C1	B1	A1

ML696201/ML69Q6203
272-PIN LFBGA
BOTTOM VIEW
0.65-mm Ball Pitch

Figure 1-3 Pin Configuration (272-pin Plastic LFBGA)

1.3.1.2 Pin Assignment Table

Table 1-1 Pin Assignment

BGA Pin	Symbol	BGA Pin	Symbol
A1	GNDIO	C1	PIOE07/-
A2	PIOE11/-	C2	PIOE08/-
A3	PIOE14/-	C3	GNDIO
A4	-/XA22	C4	PIOE04/-
A5	PIOC15/XA20	C5	PIOE06/-
A6	PIOC13/XA18	C6	PIOE15/-
A7	PIOA14/XA15	C7	-/XA21
A8	PIOA13/XA14	C8	PIOC12/XA17
A9	PIOA11/XA12	C9	PIOA10/XA11
A10	PIOA08/XA09	C10	PIOA07/XA08
A11	PIOA06/XA07	C11	PIOA03/XA04
A12	PIOA04/XA05	C12	PIOB13/XD13
A13	PIOA02/XA03	C13	PIOB09/XD09
A14	PIOB14/XD14	C14	PIOB05/XD05
A15	PIOB11/XD11	C15	PIOC08/XRAMCS_N
A16	PIOB08/XD08	C16	PIOC07/XIOCS11_N
A17	PIOB06/XD06	C17	PIOC05/XIOCS01_N
A18	PIOB04/XD04	C18	PIOC06/XIOCS10_N
A19	PIOB03/XD03	C19	GNDIO
A20	PIOB01/XD01	C20	PIOC10/XWE_N
A21	GNDIO	C21	PIOC09/XROMCS_N
B1	PIOE09/-	D1	PIOE02/-
B2	GNDIO	D2	PIOE03/-
B3	PIOE10/-	D3	PIOE05/-
B4	PIOE13/-	D4	GNDIO
B5	PIOE12/-	D5	GNDIO
B6	-/XA23	D6	VDDCORE
B7	PIOC14/XA19	D7	VDDCORE
B8	PIOA15/XA16	D8	VDDIO
B9	PIOA12/XA13	D9	VDDIO
B10	PIOA09/XA10	D10	GNDCORE
B11	PIOA05/XA06	D11	VDDCORE
B12	PIOA01/XA02	D12	VDDIO
B13	PIOB15/XD15	D13	VDDIO
B14	PIOA00/XA01	D14	VDDIO
B15	PIOB12/XD12	D15	GNDCORE
B16	PIOB10/XD10	D16	GNDCORE
B17	PIOB07/XD07	D17	GNDCORE
B18	PIOB02/XD02	D18	GNDIO
B19	PIOB00/XD00	D19	VDDIO
B20	GNDIO	D20	GNDIO
B21	PIOC11/XOE_N	D21	GNDIO

BGA Pin	Symbol	BGA Pin	Symbol	BGA Pin	Symbol
E1	PIOE00/-	K1	-/IDEIRQ	R1	PIOF02/IDED10
E2	PIOE01/-	K2	-/IDENPCBRID	R2	PIOF03/IDED11
E3	GNDCORE	K3	32K_TESTMODE/-	R3	TMS/-
E4	GNDCORE	K4	VDDIO	R4	VDDIO
E18	VDDIO	K18	-/TCOUT	R18	VDDIO
E19	PIOC03/XBS1_N	K19	-/DREQCLR	R19	GNDFLA
E20	PIOC04/XIOCS00_N	K20	AGNDTX	R20	AGND
E21	PIOC02/XBS0_N	K21	USB_DP/-	R21	ADIN0
F1	-/IDEA0	L1	OSC32K0/-	T1	FD7/IDED07
F2	-/IDEA2	L2	RTC_TESTMODE/-	T2	PIOF00/IDED08
F3	-/IDEA1	L3	PIOF01/IDED09	T3	TCK/-
F4	VDDCORE	L4	VDDIO	T4	VDDIO
F18	-/XSDCLK	L18	GNDPLL	T18	GNDCORE
F19	PIOC00/XWAIT0	L19	USB_ATEST1/-	T19	ADIN3/-
F20	-/XSDCS_N	L20	AGNDC	T20	ADIN2/-
F21	PIOC01/XWAIT1	L21	AVDDRX	T21	NC
G1	FWR_N/IDEWR_N	M1	GNDRTC	U1	FD4/IDED04
G2	FRD_N/IDERE_N	M2	OSC32K1B/-	U2	FD6/IDED06
G3	-/IDERST_N	M3	FD5/IDED05	U3	RTCK/-
G4	VDDCORE	M4	GNDCORE	U4	GNDIO
G18	-/XSYSCLK	M18	OSC11M1B/-	U18	AVDD
G19	VDDIO	M19	GNDPLL	U19	ADIN1/-
G20	-/XRAS_N	M20	AVDDC	U20	NC
G21	-/XCAS_N	M21	USB_REXT/-	U21	NC
H1	FALE/IDECS1_N	N1	PIOF06/IDED14		
H2	FRB/IDERDY	N2	-/IDED15		
H3	FCLE/IDECS0_N	N3	FD3/IDED03		
H4	VDDIO	N4	VDDCORE		
H18	-/XDQM1	N18	OSC11M0/-		
H19	-/XSDCKE	N19	VDDPLL		
H20	AGNDRX	N20	USB_ATEST0/-		
H21	AVDDTX	N21	USB_VOREF/-		
J1	-/IDEDREQ	P1	PIOF04/IDED12		
J2	-/IDEDACK_N	P2	PIOF05/IDED13		
J3	VDDRTC	P3	TDI/-		
J4	VDDIO	P4	VDDCORE		
J18	-/DREQ	P18	VDDCORE		
J19	-/XDQM0	P19	OSC48M1B/-		
J20	AGNDTX	P20	OSC48M0/-		
J21	USB_DM/-	P21	VDDPLL		

BGA Pin	Symbol	BGA Pin	Symbol
V1	FD1/IDED01	Y1	TDO/-
V2	FD2/IDED02	Y2	GNDIO
V3	nTRST/-	Y3	GNDCORE
V4	GNDIO	Y4	BOOT0/-
V5	VDDIO	Y5	EXTBUS/-
V6	VDDIO	Y6	TMODE1/-
V7	VDDCORE	Y7	TEST_RSV/-
V8	VDDIO	Y8	SCL/-
V9	VDDIO	Y9	PIOD13/SSIOCK1
V10	GNDCORE	Y10	PIOD09/SSIORXD0
V11	GNDCORE	Y11	VDDFLA
V12	VDDIO	Y12	PIOD04/SCLA/SCL
V13	VDDIO	Y13	PIOD02/SDD
V14	VDDCORE	Y14	VDDCORE
V15	GNDIO	Y15	GNDIO
V16	GNDIO	Y16	NC
V17	GNDIO	Y17	GNDIO
V18	NC	Y18	NC
V19	NC	Y19	NC
V20	NC	Y20	NC
V21	NC	Y21	NC
W1	GNDFLA	AA1	GNDIO
W2	FD0/IDED00	AA2	GNDIO
W3	GNDIO	AA3	GNDCORE
W4	GNDCORE	AA4	BOOT1/-
W5	IDEMODE/-	AA5	TMODE0/-
W6	TMODE3/-	AA6	TMODE2/-
W7	PLLBYPASS/-	AA7	PWMOUT/-
W8	SDAT/-	AA8	PIOD14/UP_TXD
W9	PIOD15/UP_RXD	AA9	PIOD12/SSIORXD1
W10	PIOD10/SSIOCK0	AA10	PIOD11/SSIOTXD1
W11	PIOD07/CKOUTA/CLKOUT	AA11	PIOD08/SSIOTXD0
W12	PIOD05/WSA	AA12	PIOD06/SDA
W13	VDDCORE	AA13	PIOD03/CKOUTD
W14	VDDCORE	AA14	PIOD01/WSD
W15	GNDIO	AA15	PIOD00/SCLD
W16	RESET_N/-	AA16	NC
W17	GNDIO	AA17	NC
W18	NC	AA18	EFIQ_N
W19	NC	AA19	NC
W20	NC	AA20	NC
W21	NC	AA21	NC

1.3.2 Pin List

Table 1-2 shows a list of pins.

Table 1-2 Pin List

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
1	B6	—	O	"L" level output	XA23	O	External memory access address	(d)	4mA
2	A4	—	O	"L" level output	XA22	O	External memory access address	(d)	4mA
3	C7	—	O	"L" level output	XA21	O	External memory access address	(d)	4mA
4	A5	PIOC15	I/O	General-purpose port	XA20	O	External memory access address	(d)	4mA
5	B7	PIOC14	I/O	General-purpose port	XA19	O	External memory access address	(d)	4mA
6	A6	PIOC13	I/O	General-purpose port	XA18	O	External memory access address	(d)	4mA
7	C8	PIOC12	I/O	General-purpose port	XA17	O	External memory access address	(d)	4mA
8	B8	PIOA15	I/O	General-purpose port	XA16	O	External memory access address	(d)	4mA
9	A7	PIOA14	I/O	General-purpose port	XA15	O	External memory access address	(d)	4mA
10	A8	PIOA13	I/O	General-purpose port	XA14	O	External memory access address	(d)	4mA
11	B9	PIOA12	I/O	General-purpose port	XA13	O	External memory access address	(d)	4mA
12	A9	PIOA11	I/O	General-purpose port	XA12	O	External memory access address	(d)	4mA
13	C9	PIOA10	I/O	General-purpose port	XA11	O	External memory access address	(d)	4mA
14	B10	PIOA09	I/O	General-purpose port	XA10	O	External memory access address	(d)	4mA
15	A10	PIOA08	I/O	General-purpose port	XA09	O	External memory access address	(d)	4mA
16	C10	PIOA07	I/O	General-purpose port	XA08	O	External memory access address	(d)	4mA
17	A11	PIOA06	I/O	General-purpose port	XA07	O	External memory access address	(d)	4mA
18	B11	PIOA05	I/O	General-purpose port	XA06	O	External memory access address	(d)	4mA
19	A12	PIOA04	I/O	General-purpose port	XA05	O	External memory access address	(d)	4mA
20	C11	PIOA03	I/O	General-purpose port	XA04	O	External memory access address	(d)	4mA
21	A13	PIOA02	I/O	General-purpose port	XA03	O	External memory access address	(d)	4mA
22	B12	PIOA01	I/O	General-purpose port	XA02	O	External memory access address	(d)	4mA
23	B14	PIOA00	I/O	General-purpose port	XA01	O	External memory access address	(d)	4mA
24	B13	PIOB15	I/O	General-purpose port	XD15	I/O	External memory access data	(d)	4mA
25	A14	PIOB14	I/O	General-purpose port	XD14	I/O	External memory access data	(d)	4mA
26	C12	PIOB13	I/O	General-purpose port	XD13	I/O	External memory access data	(d)	4mA

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
27	B15	PIOB12	I/O	General-purpose port	XD12	I/O	External memory access data	(d)	4mA
28	A15	PIOB11	I/O	General-purpose port	XD11	I/O	External memory access data	(d)	4mA
29	B16	PIOB10	I/O	General-purpose port	XD10	I/O	External memory access data	(d)	4mA
30	C13	PIOB09	I/O	General-purpose port	XD09	I/O	External memory access data	(d)	4mA
31	A16	PIOB08	I/O	General-purpose port	XD08	I/O	External memory access data	(d)	4mA
32	B17	PIOB07	I/O	General-purpose port	XD07	I/O	External memory access data	(d)	4mA
33	A17	PIOB06	I/O	General-purpose port	XD06	I/O	External memory access data	(d)	4mA
34	C14	PIOB05	I/O	General-purpose port	XD05	I/O	External memory access data	(d)	4mA
35	A18	PIOB04	I/O	General-purpose port	XD04	I/O	External memory access data	(d)	4mA
36	A19	PIOB03	I/O	General-purpose port	XD03	I/O	External memory access data	(d)	4mA
37	B18	PIOB02	I/O	General-purpose port	XD02	I/O	External memory access data	(d)	4mA
38	A20	PIOB01	I/O	General-purpose port	XD01	I/O	External memory access data	(d)	4mA
39	B19	PIOB00	I/O	General-purpose port	XD00	I/O	External memory access data	(d)	4mA
40	B21	PIOC11	I/O	General-purpose port	XOE_N	O	External memory access read enable	(d)	4mA
41	C20	PIOC10	I/O	General-purpose port	XWE_N	O	External memory access write enable	(d)	4mA
42	C21	PIOC09	I/O	General-purpose port	XROMCS_N	O	External ROM chip select	(d)	4mA
43	C15	PIOC08	I/O	General-purpose port	XRAMCS_N	O	External RAM chip select	(d)	4mA
44	C16	PIOC07	I/O	General-purpose port	XIOCS11_N	O	I/O bank 1 chip select 1	(d)	4mA
45	C18	PIOC06	I/O	General-purpose port	XIOCS10_N	O	I/O bank 1 chip select 0	(d)	4mA
46	C17	PIOC05	I/O	General-purpose port	XIOCS01_N	O	I/O bank 0 chip select 1	(d)	4mA
47	E20	PIOC04	I/O	General-purpose port	XIOCS00_N	O	I/O bank 0 chip select 0	(d)	4mA
48	E19	PIOC03	I/O	General-purpose port	XBS1_N	O	External memory byte select (MSB)	(d)	4mA
49	E21	PIOC02	I/O	General-purpose port	XBS0_N	O	External memory byte select (LSB)	(d)	4mA
50	F21	PIOC01	I/O	General-purpose port	XWAIT1	I	WAIT signal for I/O bank 1 ("1": WAIT)	(d)	4mA
51	F19	PIOC00	I/O	General-purpose port	XWAIT0	I	WAIT signal for I/O bank 0 ("1": WAIT)	(d)	4mA
52	G18	—	O	"L" level output	XSYSCLK	O	AHB clock for external bus	(d)	4mA
53	F20	—	O	"L" level output	XSDCS_N	O	SDRAM chip select	(d)	4mA
54	G21	—	O	"L" level output	XCAS_N	O	Column address strobe (SDRAM)	(d)	4mA
55	G20	—	O	"L" level output	XRAS_N	O	Row address strobe (SDRAM)	(d)	4mA
56	F18	—	O	"L" level output	XSDCLK	O	Clock for SDRAM	(d)	6mA
57	H19	—	O	"L" level output	XSDCKE	O	Clock enable (SDRAM)	(d)	4mA
58	H18	—	O	"L" level output	XDQM1	O	INPUT/OUTPUT mask (MSB)	(d)	4mA

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
59	J19	—	O	"L" level output	XDQM0	O	INPUT/OUTPUT mask (LSB)	(d)	4mA
60	J18	—	O	"L" level output	DREQ	I	DMA request for external I/O	(d)	4mA
61	K19	—	O	"L" level output	DREQCLR	O	DMA clear for external I/O	(d)	4mA
62	K18	—	O	"L" level output	TCOUT	O	DMAC Terminal Count	(d)	4mA
63	C6	PIOE15	I/O	General-purpose port (5V tolerant and with IRQ interrupt)	—	—	—	(e)	2mA
64	A3	PIOE14	I/O	General-purpose port (with IRQ interrupt)	—	—	—	(d)	2mA
65	B4	PIOE13	I/O	General-purpose port (with IRQ interrupt)	—	—	—	(d)	2mA
66	B5	PIOE12	I/O	General-purpose port (with IRQ interrupt)	—	—	—	(d)	2mA
67	A2	PIOE11	I/O	General-purpose port	—	—	—	(d)	2mA
68	B3	PIOE10	I/O	General-purpose port	—	—	—	(d)	2mA
69	B1	PIOE09	I/O	General-purpose port	—	—	—	(d)	2mA
70	C2	PIOE08	I/O	General-purpose port	—	—	—	(d)	2mA
71	C1	PIOE07	I/O	General-purpose port	—	—	—	(d)	2mA
72	C5	PIOE06	I/O	General-purpose port	—	—	—	(d)	2mA
73	D3	PIOE05	I/O	General-purpose port	—	—	—	(d)	2mA
74	C4	PIOE04	I/O	General-purpose port	—	—	—	(d)	2mA
75	D2	PIOE03	I/O	General-purpose port	—	—	—	(d)	2mA
76	D1	PIOE02	I/O	General-purpose port	—	—	—	(d)	2mA
77	E2	PIOE01	I/O	General-purpose port	—	—	—	(d)	2mA
78	E1	PIOE00	I/O	General-purpose port	—	—	—	(d)	2mA
79	F2	—	O	"L" level output	IDEA2	O	IDE controller address	(e)	4mA
80	F3	—	O	"L" level output	IDEA1	O	IDE controller address	(e)	4mA
81	F1	—	O	"L" level output	IDEA0	O	IDE controller address	(e)	4mA
82	N2	—	O	"L" level output	IDED15	I/O	IDE controller data	(e)	4mA
83	N1	PIOF06	I/O	General-purpose port	IDED14	I/O	IDE controller data	(e)	4mA
84	P2	PIOF05	I/O	General-purpose port	IDED13	I/O	IDE controller data	(e)	4mA
85	P1	PIOF04	I/O	General-purpose port	IDED12	I/O	IDE controller data	(e)	4mA
86	R2	PIOF03	I/O	General-purpose port	IDED11	I/O	IDE controller data	(e)	4mA
87	R1	PIOF02	I/O	General-purpose port	IDED10	I/O	IDE controller data	(e)	4mA
88	L3	PIOF01	I/O	General-purpose port	IDED09	I/O	IDE controller data	(e)	4mA
89	T2	PIOF00	I/O	General-purpose port	IDED08	I/O	IDE controller data	(e)	4mA
90	T1	FD7	I/O	NAND FLASH data	IDED07	I/O	IDE controller data	(e)	4mA
91	U2	FD6	I/O	NAND FLASH data	IDED06	I/O	IDE controller data	(e)	4mA
92	M3	FD5	I/O	NAND FLASH data	IDED05	I/O	IDE controller data	(e)	4mA
93	U1	FD4	I/O	NAND FLASH data	IDED04	I/O	IDE controller data	(e)	4mA
94	N3	FD3	I/O	NAND FLASH data	IDED03	I/O	IDE controller data	(e)	4mA
95	V2	FD2	I/O	NAND FLASH data	IDED02	I/O	IDE controller data	(e)	4mA
96	V1	FD1	I/O	NAND FLASH data	IDED01	I/O	IDE controller data	(e)	4mA
97	W2	FD0	I/O	NAND FLASH data	IDED00	I/O	IDE controller data	(e)	4mA
98	G2	FRD_N	O	NAND FLASH read enable	IDERE_N	O	IDE controller read eanable	(e)	4mA
99	G1	FWR_N	O	NAND FLASH write enable	IDEWR_N	O	IDE controller write eanable	(e)	4mA

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
100	H2	FRB	I	NAND FLASH ready/busy (1: ready, 0: busy)	IDERDY	I	IDE controller I/O cycle extend enable	(e)	4mA
101	G3	—	O	"L" level output	IDERST_N	O	Reset signal (L: Reset)	(e)	4mA
102	H1	FALE	O	NAND FLASH address latch enable	IDECS1_N	O	IDE controller status/ controller select signal	(e)	4mA
103	H3	FCLE	O	NAND FLASH command latch enable	IDECS0_N	O	IDE controller data/ command select signal	(e)	4mA
104	J1	—	O	"L" level output	IDEDREQ	I	IDE controller DMA request	(e)	4mA
105	J2	—	O	"L" level output	IDEDACK_N	O	IDE controller DMA acknowledge	(e)	4mA
106	K1	—	O	"L" level output	IDEIRQ	I	IDE controller interrupt input	(e)	4mA
107	K2	—	O	"L" level output	IDENPCBRID	I	IDE primary cable ID detection	(e)	4mA
108	K21	USB_DP	I/O	USB DP input/output	—	—	—	USB	—
109	J21	USB_DM	I/O	USB DM input/output	—	—	—	USB	—
110	M21	USB_REXT	I	USB REXT pin	—	—	—	USB	—
111	N21	USB_VOREF	O	VoRef pin for USB TEST	—	—	—	USB	—
112	L19	USB_ ATES T1	O	USB ANALOG TEST pin 1	—	—	—	USB	—
113	N20	USB_ ATES T0	O	USB ANALOG TEST pin 0	—	—	—	USB	—
114	AA7	PWMOUT	O	PWM output	—	—	—	(d)	2mA
115	W8	SDAT	I/O	I^2C transmit/receive data	—	—	—	(d)	2mA
116	Y8	SCL	O	I^2C clock output	—	—	—	(d)	2mA
117	W9	PIOD15	I/O	General-purpose port	UP_RXD	I	uPLAT SIO (UART) receive data	(d)	2mA
118	AA8	PIOD14	I/O	General-purpose port	UP_TXD	O	uPLAT SIO (UART)transmit data	(d)	2mA
119	Y9	PIOD13	I/O	General-purpose port	SSIOCK1	I/O	SSIO clock pin 1	(d)	2mA
120	AA9	PIOD12	I/O	General-purpose port	SSIORXD1	I	SSIO receive data input 1	(d)	2mA
121	AA10	PIOD11	I/O	General-purpose port	SSIOTXD1	O	SSIO transmit data output 1	(d)	2mA
122	W10	PIOD10	I/O	General-purpose port	SSIOCK0	I/O	SSIO clock pin 0	(d)	2mA
123	Y10	PIOD09	I/O	General-purpose port	SSIORXD0	I	SSIO receive data input 0	(d)	2mA
124	AA11	PIOD08	I/O	General-purpose port	SSIOTXD0	O	SSIO transmit data output 0	(d)	2mA
125	W11	PIOD07	I/O	General-purpose port	CKOUTA/ CLKOUT	O	I^2S receive system clock	(d)	2mA
126	AA12	PIOD06	I/O	General-purpose port	SDA	I	I^2S receive data	(d)	2mA
127	W12	PIOD05	I/O	General-purpose port	WSA	I/O	I^2S receive channel select	(d)	2mA
128	Y12	PIOD04	I/O	General-purpose port	SCLA/SCL	I/O	I^2S receive transfer clock	(d)	2mA
129	AA13	PIOD03	I/O	General-purpose port	CKOUTD	O	I^2S transmit system clock	(d)	2mA
130	Y13	PIOD02	I/O	General-purpose port	SDD	O	I^2S transmit data	(d)	2mA
131	AA14	PIOD01	I/O	General-purpose port	WSD	I/O	I^2S transmit channel select	(d)	2mA
132	AA15	PIOD00	I/O	General-purpose port	SCLD	I/O	I^2S transmit transfer clock	(d)	2mA
133	T19	ADIN3	I	A/D C input (CH3)	—	—	—	(b)	2mA
134	T20	ADIN2	I	A/D C input (CH2)	—	—	—	(b)	2mA
135	U19	ADIN1	I	A/D C input (CH1)	—	—	—	(b)	2mA
136	R21	ADIN0	I	A/D C input (CH0)	—	—	—	(b)	2mA
137	L1	OSC32K0	I	32 kHz oscillation input	—	—	—	WOSC2	—
138	M2	OSC32K1B	O	32kHz oscillation output	—	—	—	WOSC2	—

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
139	L2	RTC_TESTMODE	I	SCAN mode pin of RTCsection	—	—	—	(g) (1.5V I/F)	—
140	K3	32K_TESTMODE	I	Test mode pin of 32 kHz oscillation circuit	—	—	—	(g) (1.5V I/F)	—
141	P20	OSC48M0	I	48 MHz oscillation input	—	—	—	WOAC8	—
142	P19	OSC48M1B	O	48 MHz oscillation output	—	—	—	WOAC8	—
143	N18	OSC11M0	I	11MHz oscillation input	—	—	—	WOSC5	—
144	M18	OSC11M1B	O	11 MHz oscillation output	—	—	—	WOSC5	—
145	W16	RESET_N	I	System reset (L: Reset)	—	—	—	(a)	2mA
146	P3	TDI	I	JTAG TDI input	—	—	—	(f)	2mA
147	Y1	TDO	O	JTAG TDO output	—	—	—	(c)	2mA
148	V3	nTRST	I	JTAG nTRST input	—	—	—	(d)	2mA
149	R3	TMS	I	JTAG TDI input	—	—	—	(f)	2mA
150	T3	TCK	I	JTAG TMS input	—	—	—	(d)	2mA
151	U3	RTCK	O	JTAG RTCK output	—	—	—	(d)	2mA
152	W21	NC	—	—	—	—	—	—	—
153	V19	NC	—	—	—	—	—	—	—
154	AA19	NC	—	—	—	—	—	—	—
155	Y19	NC	—	—	—	—	—	—	—
156	AA18	EFIQ_N	I	External FIQ input	—	—	—	(d)	2mA
157	AA21	NC	—	—	—	—	—	—	—
158	Y21	NC	—	—	—	—	—	—	—
159	V21	NC	—	—	—	—	—	—	—
160	U21	NC	—	—	—	—	—	—	—
161	T21	NC	—	—	—	—	—	—	—
162	Y20	NC	—	—	—	—	—	—	—
163	W19	NC	—	—	—	—	—	—	—
164	W18	NC	—	—	—	—	—	—	—
165	Y4	BOOT0	I	BOOT switching	—	—	—	(d)	2mA
166	AA4	BOOT1	I	BOOT switching	—	—	—	(d)	2mA
167	Y5	EXTBUS	I	External bus/GPIO switching (H: External bus, L: GPIO)	—	—	—	(d)	2mA
168	W5	IDEMODE	I	NAND FLASH/IDE switching "H": IDEcont, "L": NAND FLASH	—	—	—	(d)	2mA
169	Y7	TEST_RSV		Reserved pin	—	—	—	(g)	2mA
170	W7	PLLBYPASS	I	PLL bypass mode	—	—	—	(g)	2mA
171	AA5	TMODE0	I	Test mode switching	—	—	—	(g)	2mA
172	Y6	TMODE1	I	Same as above	—	—	—	(g)	2mA
173	AA6	TMODE2	I	Same as above	—	—	—	(g)	2mA
174	W6	TMODE3	I	Same as above	—	—	—	(g)	2mA
175	D6	VDDCORE	—	Core logic power supply 1	—	—	—	1.5V power supply	—
176	D7	VDDCORE	—	Core logic power supply 2	—	—	—	1.5V power supply	—

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
177	D11	VDDCORE	—	Core logic power supply 3	—	—	—	1.5V power supply	—
178	F4	VDDCORE	—	Core logic power supply 4	—	—	—	1.5V power supply	—
179	G4	VDDCORE	—	Core logic power supply 5	—	—	—	1.5V power supply	—
180	N4	VDDCORE	—	Core logic power supply 6	—	—	—	1.5V power supply	—
181	P4	VDDCORE	—	Core logic power supply 7	—	—	—	1.5V power supply	—
182	P18	VDDCORE	—	Core logic power supply 8	—	—	—	1.5V power supply	—
183	V7	VDDCORE	—	Core logic power supply 9	—	—	—	1.5V power supply	—
184	V14	VDDCORE	—	Core logic power supply 10	—	—	—	1.5V power supply	—
185	W13	VDDCORE	—	Core logic power supply 11	—	—	—	1.5V power supply	—
186	W14	VDDCORE	—	Core logic power supply 12	—	—	—	1.5V power supply	—
187	Y14	VDDCORE	—	Core logic power supply 13	—	—	—	1.5V power supply	—
188	D10	GNDCORE	—	Core logic GND pin 1	—	—	—	GND	—
189	D15	GNDCORE	—	Core logic GND pin 2	—	—	—	GND	—
190	D16	GNDCORE	—	Core logic GND pin 3	—	—	—	GND	—
191	D17	GNDCORE	—	Core logic GND pin 4	—	—	—	GND	—
192	E3	GNDCORE	—	Core logic GND pin 5	—	—	—	GND	—
193	E4	GNDCORE	—	Core logic GND pin 6	—	—	—	GND	—
194	M4	GNDCORE	—	Core logic GND pin 7	—	—	—	GND	—
195	T18	GNDCORE	—	Core logic GND pin 8	—	—	—	GND	—
196	V10	GNDCORE	—	Core logic GND pin 9	—	—	—	GND	—
197	V11	GNDCORE	—	Core logic GND pin 10	—	—	—	GND	—
198	W4	GNDCORE	—	Core logic GND pin 11	—	—	—	GND	—
199	Y3	GNDCORE	—	Core logic GND pin 12	—	—	—	GND	—
200	AA3	GNDCORE	—	Core logic GND pin 13	—	—	—	GND	—
201	N19	VDDPLL	—	PLL power supply pin 1 (Core level)	—	—	—	1.5V power supply	—
202	P21	VDDPLL	—	PLL power supply pin 2 (Core level)	—	—	—	1.5V power supply	—
203	L18	GNDPLL	—	PLL GND pin 1	—	—	—	GND	—
204	M19	GNDPLL	—	PLL GND pin 2	—	—	—	GND	—

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
205	D8	VDDIO	—	IO power supply pin 1	—	—	—	3.3V power supply	—
206	D9	VDDIO	—	IO power supply pin 2	—	—	—	3.3V power supply	—
207	D12	VDDIO	—	IO power supply pin 3	—	—	—	3.3V power supply	—
208	D13	VDDIO	—	IO power supply pin 4	—	—	—	3.3V power supply	—
209	D14	VDDIO	—	IO power supply pin 5	—	—	—	3.3V power supply	—
210	D19	VDDIO	—	IO power supply pin 6	—	—	—	3.3V power supply	—
211	E18	VDDIO	—	IO power supply pin 7	—	—	—	3.3V power supply	—
212	G19	VDDIO	—	IO power supply pin 8	—	—	—	3.3V power supply	—
213	H4	VDDIO	—	IO power supply pin 9	—	—	—	3.3V power supply	—
214	J4	VDDIO	—	IO power supply pin 10	—	—	—	3.3V power supply	—
215	K4	VDDIO	—	IO power supply pin 11	—	—	—	3.3V power supply	—
216	L4	VDDIO	—	IO power supply pin 12	—	—	—	3.3V power supply	—
217	R4	VDDIO	—	IO power supply pin 13	—	—	—	3.3V power supply	—
218	R18	VDDIO	—	IO power supply pin 14	—	—	—	3.3V power supply	—
219	T4	VDDIO	—	IO power supply pin 15	—	—	—	3.3V power supply	—
220	V5	VDDIO	—	IO power supply pin 16	—	—	—	3.3V power supply	—
221	V6	VDDIO	—	IO power supply pin 17	—	—	—	3.3V power supply	—
222	V8	VDDIO	—	IO power supply pin 18	—	—	—	3.3V power supply	—
223	V9	VDDIO	—	IO power supply pin 19	—	—	—	3.3V power supply	—
224	V12	VDDIO	—	IO power supply pin 20	—	—	—	3.3V power supply	—

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
225	V13	VDDIO	—	IO power supply pin 21	—	—	—	3.3V power supply	—
226	A1	GNDIO	—	IO GND pin 1	—	—	—	GND	—
227	A21	GNDIO	—	IO GND pin 2	—	—	—	GND	—
228	B2	GNDIO	—	IO GND pin 3	—	—	—	GND	—
229	B20	GNDIO	—	IO GND pin 4	—	—	—	GND	—
230	C3	GNDIO	—	IO GND pin 5	—	—	—	GND	—
231	C19	GNDIO	—	IO GND pin 6	—	—	—	GND	—
232	D4	GNDIO	—	IO GND pin 7	—	—	—	GND	—
233	D5	GNDIO	—	IO GND pin 8	—	—	—	GND	—
234	D18	GNDIO	—	IO GND pin 9	—	—	—	GND	—
235	D20	GNDIO	—	IO GND pin 10	—	—	—	GND	—
236	D21	GNDIO	—	IO GND pin 11	—	—	—	GND	—
237	U4	GNDIO	—	IO GND pin 12	—	—	—	GND	—
238	V4	GNDIO	—	IO GND pin 13	—	—	—	GND	—
239	V15	GNDIO	—	IO GND pin 14	—	—	—	GND	—
240	V16	GNDIO	—	IO GND pin 15	—	—	—	GND	—
241	V17	GNDIO	—	IO GND pin 16	—	—	—	GND	—
242	W3	GNDIO	—	IO GND pin 17	—	—	—	GND	—
243	W15	GNDIO	—	IO GND pin 18	—	—	—	GND	—
244	W17	GNDIO	—	IO GND pin 19	—	—	—	GND	—
245	Y2	GNDIO	—	IO GND pin 20	—	—	—	GND	—
246	Y15	GNDIO	—	IO GND pin 21	—	—	—	GND	—
247	Y17	GNDIO	—	IO GND pin 22	—	—	—	GND	—
248	AA1	GNDIO	—	IO GND pin 23	—	—	—	GND	—
249	AA2	GNDIO	—	IO GND pin 24	—	—	—	GND	—
250	U18	AVDD	—	Power supply for 10-bit A/D C (IO level)	—	—	—	3.3V power supply	—
251	R20	AGND	—	GND for 10-bit A/D C	—	—	—	GND	—
252	J3	VDDRTC	—	Power supply for RTC (Core level)	—	—	—	1.5V power supply	—
253	M1	GNDRTC	—	GND for RTC	—	—	—	GND	—
254	L21	AVDDRX	—	Power supply for USB (IO level)	—	—	—	3.3V power supply	—
255	H20	AGNDRX	—	GND for USB	—	—	—	GND	—
256	H21	AVDDTX	—	Power supply for USB (IO level)	—	—	—	3.3V power supply	—
257	J20	AGNDTX	—	GND for USB	—	—	—	GND	—
258	K20	AGNDTX	—	GND for USB	—	—	—	GND	—
259	M20	AVDDC	—	Power supply for USB (IO level)	—	—	—	3.3V power supply	—
260	L20	AGNDC	—	GND for USB	—	—	—	GND	—
261	W20	NC	—	NC	—	—	—	—	—

Pin	BGA Pin	Primary			Secondary			Pin type	Output current
		Symbol	I/O		Symbol	I/O	Function		
262	V18	NC	—	NC	—	—	—	—	—
263	AA20	NC	—	NC	—	—	—	—	—
264	Y18	NC	—	NC	—	—	—	—	—
265	V20	NC	—	NC	—	—	—	—	—
266	U20	NC	—	NC	—	—	—	—	—
267	AA17	NC	—	NC	—	—	—	—	—
268	AA16	NC	—	NC	—	—	—	—	—
269	Y16	NC	—	NC	—	—	—	—	—
270	Y11	VDDFLA	—	Power supply for FLASH (IO level)	—	—	—	FLASH	—
271	R19	GNDFLA	—	GND for FLASH	—	—	—	FLASH	—
272	W1	GNDFLA	—	GND for FLASH	—	—	—	FLASH	—

1.3.3 Pin Description

<table>
<tr><th>Symbol</th><th>I/O</th><th>Description</th><th>Primary/ secondary</th><th>Logic</th></tr>
<tr><td colspan="5">System</td></tr>
<tr><td>OSC32K0</td><td>I</td><td>32 kHz oscillation input</td><td>—</td><td>—</td></tr>
<tr><td>OSC32K1B</td><td>O</td><td>32 kHz oscillation output</td><td>—</td><td>—</td></tr>
<tr><td>OSC48M0</td><td>I</td><td>48 MHz oscillation input</td><td>—</td><td>—</td></tr>
<tr><td>OSC48M1B</td><td>O</td><td>48 MHz oscillation output</td><td>—</td><td>—</td></tr>
<tr><td>OSC11M0</td><td>I</td><td>11 MHz oscillation input</td><td>—</td><td>—</td></tr>
<tr><td>OSC11M1B</td><td>O</td><td>11 MHz oscillation output</td><td>—</td><td>—</td></tr>
<tr><td>RESET_N</td><td>I</td><td>System reset (“L”: Reset)</td><td>—</td><td>Nagative</td></tr>
<tr><td colspan="5">Mode</td></tr>
<tr><td>BOOT[1:0]</td><td>I</td><td>Controls boot devices.
<table>
<tr><th>BOOT [1]</th><th>BOOT [0]</th><th>Boot device</th></tr>
<tr><td>0</td><td>0</td><td>MCP FLSAH</td></tr>
<tr><td>0</td><td>1</td><td>External FLASH</td></tr>
<tr><td>1</td><td>0</td><td>AHBROM</td></tr>
<tr><td>1</td><td>1</td><td>AHBROM</td></tr>
</table></td><td>—</td><td>—</td></tr>
<tr><td>EXTBUS</td><td>I</td><td>Switches between external bus pin and GPIO.
<table>
<tr><th>EXTBUS</th><th>GPIO/external bus</th></tr>
<tr><td>0</td><td>GPIO</td></tr>
<tr><td>1</td><td>External bus</td></tr>
</table></td><td>—</td><td>—</td></tr>
<tr><td>IDEMODE</td><td>I</td><td>Switches between IDE and NAND Flash pin
<table>
<tr><th>IDEMODE</th><th>NAND Flash/IDE controller</th></tr>
<tr><td>0</td><td>NAND Flash</td></tr>
<tr><td>1</td><td>IDE controller</td></tr>
</table></td><td>—</td><td>—</td></tr>
<tr><td>PLLBYPASS</td><td>I</td><td>Sets PLL bypass mode.
<table>
<tr><th>PLLBYPASS</th><th>PLL mode</th></tr>
<tr><td>0</td><td>Use PLL</td></tr>
<tr><td>1</td><td>Bypass PLL</td></tr>
</table>
Always use PLLBYPASS at ground level.</td><td>—</td><td>—</td></tr>
<tr><td>TEST_RSV</td><td></td><td>This pin is used for testing. Always use TEST_RSV at ground level.</td><td>—</td><td>—</td></tr>
<tr><td>TMODE [3:0]</td><td>I</td><td>These pins are used to switch test modes such as SCAN, TIC, JTAG-FLASH, AUDIO A/D and D/A, and PLL.
Always use TMODE at ground level.</td><td>—</td><td>Positive</td></tr>
<tr><td>RTC_ TESTMODE</td><td>I</td><td>This pin switches the Scan mode pin of the RTC section.</td><td>—</td><td>Positive</td></tr>
<tr><td>32K_ TESTMODE</td><td>I</td><td>This pin switches the test mode pin of a 32 kHz oscillator circuit.</td><td>—</td><td>Positive</td></tr>
</table>

Symbol	I/O	Description	Primary/ secondary	Logic
Debug support				
TCK	I	This pin is used during debugging. Normally connect this pin to ground.*	—	—
TMS	I	This pin is used during debugging. Normally input a high level signal to this bin.*	—	Positive
nTRST	I	This pin is used during debugging. Normally connect this pin to ground.*	—	Negative
TDI	I	This pin is used during debugging. Normally input a high level signal to this bin.*	—	Positive
TDO	O	This pin is used during debugging. Normally set this pin to open.*	—	Positive
RTCK	O	This pin is used during debugging. Normally set this pin to open.*	—	—
		*: See "Example of connection circuit with JTAG-ICE" in Chapter 29 before debugging.		
External bus				
XA [23:1]	O	Address of the bus that connects external RAM, external ROM, external IO and external DRAM	Secondary	—
XD [15:0]	I/O	Data bus that connects external RAM, external ROM, external IO and external DRAM	Secondary	—
External bus control signal				
XOE_N	O	External memory access read enable	Secondary	Negative
XWE_N	O	External memory access write enable	Secondary	Negative
XROMCS_N	O	External ROM chip select	Secondary	Negative
XRAMCS_N	O	External RAM chip select	Secondary	Negative
XBS1_N	O	External memory byte select (MSB)	Secondary	Negative
XBS0_N	O	External memory byte select (LSB)	Secondary	Negative
XIOCS11_N	O	I/O bank 1 chip select 1	Secondary	Negative
XIOCS10_N	O	I/O bank 1 chip select 0	Secondary	Negative
XIOCS01_N	O	I/O bank 0 chip select 1	Secondary	Negative
XIOCS00_N	O	I/O bank 0 chip select 0	Secondary	Negative
XWAIT [1:0]	I	Wait signal for I/O bank 0/1. A device slower than the register set value can be connected by inputting this signal (wait when 1).	Secondary	Positive
XSYSCLK	O	AHB clock for external bus	Secondary	—
External bus control signal (DRAM)				
XSDCS_N	O	SDRAM chip select	Secondary	Negative
XCAS_N	O	Column address strobe (SDRAM)	Secondary	Negative
XRAS_N	O	Row address strobe (SDRAM)	Secondary	Negative
XSDCLK	O	Clock for SDRAM	Secondary	—
XSDCKE	O	Clock enable (SDRAM)	Secondary	—
XDQM1	O	Input/output mask (MSB)	Secondary	Positive
XDQM0	O	Input/output mask (LSB)	Secondary	Positive
DMA control				
DREQ	I	DMA request signal. This signal is used if the DREQ type is set by the DMA controller.	Secondary	Positive
DREQCLR	O	DREQ signal clear request. The DMA device turns off the DREQ signal when this signal is output.	Secondary	Positive
TCOUT	O	This signal notifies the DAM device that the last transfer has been started.	Secondary	Positive

Symbol	I/O	Description	Primary/ secondary	Logic
General-purpose I/O port				
PIOA[15:0]	I/O	This is a general-purpose port. Because this port has a secondary function, it cannot be used as a port if its secondary function is used.	Primary	—
PIOB[15:0]	I/O	This is a general-purpose port. Because this port has a secondary function, it cannot be used as a port if its secondary function is used.	Primary	—
PIOC[15:0]	I/O	This is a general-purpose port. Because this port has a secondary function, it cannot be used as a port if its secondary function is used.	Primary	—
PIOD [15:0]	I/O	This is a general-purpose port. Because this port has a secondary function, it cannot be used as a port if its secondary function is used.	Primary	—
PIOE[15:0]	I/O	This is a general-purpose port. PIOE[15] is 5V tolerant. PIOE[15:12] are with an IRQ interrupt.	Primary	—
PIOF[6:0]	I/O	This is a general-purpose port. Since this port has a secondary function, it cannot be used as a port if its secondary function is used.	Primary	—
μPLAT-SIO				
UP_RXD	I	μPLAT SIO (UART) receive data	Secondary	—
UP_TXD	O	μPLAT SIO (UART) transmit data	Secondary	—
IDE				
IDEA [2:0]	O	IDE controller address	Secondary	—
IDED [15:0]	I/O	IDE controller data	Secondary	—
IDERE_N	O	IDE controller read enable	Secondary	Negative
IDEWR_N	O	IDE controller write enable	Secondary	Negative
IDERDY	I	This allows the IDE controller's I/O cycle extension.	Secondary	Positive
IDERST_N	O	Reset signal ("L": Reset)	Secondary	Negative
IDECS1_N	O	IDE controller status/control select signal	Secondary	Negative
IDECS0_N	O	IDE controller data/command select signal	Secondary	Negative
IDEDREQ	I	IDE controller DMA request	Secondary	Positive
IDEDACK_N	O	IDE controller DMA acknowledge	Secondary	Negative
IDEIRQ	I	IDE controller interrupt input	Secondary	Positive
IDENPCBRID	I	This detects the IDE primary cable ID.	Secondary	Positive
FLASH				
FD[7:0]	I/O	NAND FLASH data	Primary	—
FRD_N	O	NAND FLASH read enable	Primary	Negative
FWR_N	O	NAND FLASH write enable	Primary	Negative
FRB	I	NAND FLASH ready/busy (1:ready, 0:busy)	Primary	—
FALE	O	NAND FLASH address latch enable	Primary	—
FCLE	O	NAND FLASH command latch enable	Primary	—

Symbol	I/O	Description	Primary/ secondary	Logic
USB				
USB_DP	I/O	USB DP input/output	Primary	—
USB_DM	I/O	USB DM input/output	Primary	—
USB_REXT	I	USB REXT pin	Primary	—
USB_VOREF	O	VoRef pin for USB TEST	Primary	—
USB_ATEST [1:0]	O	USB ANALOG TEST pin 1-0	Primary	—
I^2S reception				
ICLKOUTA/CLKOUT	O	I^2S receive system clock	Secondary	—
SDA	I	I^2S receive data	Secondary	—
WSA	I/O	I^2S receive channel select	Secondary	—
SCLA/SCL	I/O	I^2S receive transfer clock	Secondary	—
I^2S transmission				
CKOUTD	O	I^2S transmit system clock	Secondary	—
SDD	O	I^2S transmit data	Secondary	—
WSD	I/O	I^2S transmit channel select	Secondary	—
SCLD	I/O	I^2S transmit transfer clock	Secondary	—
I^2C				
SDAT	I/O	I^2C transmit/receive data	Primary	—
SCL	O	I^2C clock output	Primary	—
SSIO				
SSIOCK [1:0]	I/O	SSIO clock pins 0 and 1	Secondary	—
SSIORXD [1:0]	I	SSIO receive data input 1	Secondary	—
SSIOTXD [1:0]	O	SSIO transmit data output 1	Secondary	—
PWM				
PWMOUT	O	PWM output	Primary	—
A/D converter				
ADIN [3:0]	I	ADC input (CH3/2/1/0)	Primary	—
Interrupt				
EFIQ_N	I	External FIQ input. (L: Interrupt)	—	Negative

Symbol	I/O	Description		Primary/ secondary	Logic
Power supply					
VDDCORE		Core logic power supply pin		—	—
GNDCORE		Core logic GND pin		—	—
VDDPLL		PLL power supply pin (core level)		—	—
GNDPLL		PLL GND pin		—	—
VDDIO		IO power supply pin		—	—
GNDIO		IO GND pin		—	—
AVDD		Power supply for 10-bit A/D C (IO level)		—	—
AGND		GND for 10-bit A/D C		—	—
VDDRTC		Power supply for RTC (core level)		—	—
GNDRTC		GND for RTC		—	—
AVDDRX		Power supply for USB (IO level)		—	—
AGNDRX		GND for USB		—	—
AVDDTX		Power supply for USB (IO level)		—	—
AGNDTX		GND for USB		—	—
AVDDC		Power supply for USB (IO level)		—	—
AGNDC		GND for USB		—	—
VDDFLA		Power supply for FLASH (IO level)	*: These pins can be left open for the ML696201 (the ROMless version).	—	—
GNDFLA		GND for FLASH		—	—

Internal Pins Connected with the Flash ROM Module

Symbol (Internal pins)	I/O	Function	Pin type	Symbol (Internal pins)	I/O	Function	Pin type
IA18	O	FLASH address	(d)	ID13	O	FLASH address	(d)
IA17	O	FLASH address	(d)	ID12	O	FLASH address	(d)
IA16	O	FLASH address	(d)	ID11	O	FLASH address	(d)
IA15	O	FLASH address	(d)	ID10	O	FLASH address	(d)
IA14	O	FLASH address	(d)	ID09	O	FLASH address	(d)
IA13	O	FLASH address	(d)	ID08	O	FLASH address	(d)
IA12	O	FLASH address	(d)	ID07	O	FLASH address	(d)
IA11	O	FLASH address	(d)	ID06	O	FLASH address	(d)
IA10	O	FLASH address	(d)	ID05	O	FLASH address	(d)
IA09	O	FLASH address	(d)	ID04	O	FLASH address	(d)
IA08	O	FLASH address	(d)	ID03	O	FLASH address	(g)
IA07	O	FLASH address	(d)	ID02	O	FLASH address	(g)
IA06	O	FLASH address	(d)	ID01	O	FLASH address	(g)
IA05	O	FLASH address	(d)	ID00	O	FLASH address	(g)
IA04	O	FLASH address	(d)	IOE_N	O	FLASH output enable	(g)
IA03	O	FLASH address	(d)	IWE_N	O	FLASH write enable	(g)
IA02	O	FLASH address	(d)	IROMCS_N	O	FLASH chip enable	(g)
IA01	O	FLASH address	(d)	IBYTE_N	O	FLASH byte signal (Fixed to word setting)	(g)
ID15	O	FLASH address	(d)	IRB_N	I	FLASH ready/busy signal	(g)
ID14	O	FLASH address	(d)	IRESET_N	O	FLASH reset signal	(g)

1.3.4 Status of Pins

Table 1-3 lists the status of the output pins (including input/output pins) during the reset operation of the LSI. The pins that have a primary/secondary function are listed using the pin names on the side selected.

Table 1-3 Status of Output Pins (Including Input/Output Pins) During a Reset Operation

Symbol	Switching between IDEMODE and EXTBUS pin settings	Pin status during reset	Pin status after releasing reset
PIOA15 to PIOA00	Note 1	High impedance	←
PIOB15 to PIOB00	Note 1	High impedance	←
PIOC15 to PIOC00	Note 1	High impedance	←
PIOD15 to PIOD00		High impedance	←
PIOE15 to PIOE00		High impedance	←
PIOF06 to PIOF00	Note 3	High impedance	←
XA23 to XA01	Note 2	"L" level	←
XD15 to XD00	Note 2	High impedance	←
XOE_N	Note 2	"H" level	←
XWE_N	Note 2	"H" level	←
XROMCS_N	Note 2	"H" level	←
XRAMCS_N	Note 2	"H" level	←
XIOCS11_N	Note 2	"H" level	←
XIOCS10_N	Note 2	"H" level	←
XIOCS01_N	Note 2	"H" level	←
XIOCS00_N	Note 2	"H" level	←
XBS1_N	Note 2	"H" level	←
XBS0_N	Note 2	"H" level	←
XSYSCLK		"H" level	←
XSDCD_N		"H" level	←
XCAS_N		"H" level	←
XRAS_N		"H" level	←
XSDCLK		"H" level	←
XSDCKE		"H" level	←
XDQM1		"H" level	←
XDQM0		"H" level	←
DREQCLR		"L" level	←
TCOUT		"L" level	←
IDEA2 to IDEA0	Note 4	"L" level	←
IDED15 to IDED08	Note 4	High impedance	←
IDED07 to IDED00	Note 4	High impedance	←
IDERE_N	Note 4	"H" level	←
IDEWR_N	Note 4	"H" level	←
IDECS1_N	Note 4	"H" level	←
IDECS0_N	Note 4	"H" level	←
IDEDACK_N	Note 4	"H" level	←
FD7 to FD0	Note 3	"L" level	←
FRD_N	Note 3	"H" level	←
FWR_N	Note 3	"H" level	←

Symbol	Switching between IDEMODE and EXTBUS pin settings	Pin status during reset	Pin status after releasing reset
FALE	Note 3	"L" level	←
FCLE	Note 3	"L" level	←
USB_DP		"L" level output	←
USB_DN		"L" level output	←
USB_VOREF		High impedance	←
USB_ATES T1/ USB_ATES T0		High impedance	←
PWMOUT		Undefined	←
SDAT		High impedance	←
SCL		High impedance	←
TDO		High impedance	←
RTCK		"L" level output	←

Note 1: Valid when EXTBUS = 0.
Note 2: Valid when EXTBUS = 1.
Note 3: Valid when IDEMODE = 0.
Note 4: Valid when IDEMODE = 1.

Table 1-4 Status of Output Pins (Including Input/Output Pins) in Standby Mode

Symbol	Pin status during reset
PIOA15 to PIOA00	High impedance
PIOB15 to PIOB00	High impedance
PIOC15 to PIOC00	High impedance
PIOD15 to PIOD00	High impedance
PIOE15 to PIOE00	High impedance
PIOF06 to PIOF00	High impedance
XA23 to XA01	"L" level
XD15 to XD00	High impedance
XOE_N	"H" level
XWE_N	"H" level
XROMCS_N	"H" level
XRAMCS_N	"H" level
XIOCS11_N	"H" level
XIOCS10_N	"H" level
XIOCS01_N	"H" level
XIOCS00_N	"H" level
XBS1_N	"H" level
XBS0_N	"H" level
XSYSCLK	"H" level
XSDCD_N	"H" level
XCAS_N	"H" level
XRAS_N	"H" level
XSDCLK	"H" level
XSDCKE	"H" level
XDQM1	"H" level
XDQM0	"H" level
DREQCLR	
TCOUT	
IDEA2 to IDEA0	
IDED15 to IDED00	High impedance
IDERE_N	
IDEWR_N	
IDECS1_N	
IDECS0_N	
IDEDACK_N	
FD7 to FD0	
FRD_N	
FWR_N	
FALE	
FCLE	
USB_DP	
USB_DN	
USB_ANALOG_TEST	
PWMOUT	

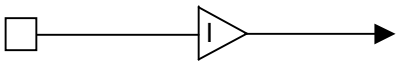
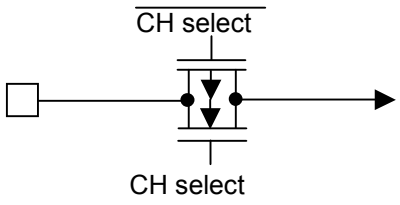
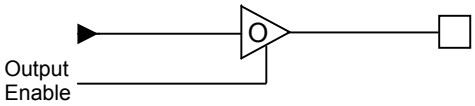
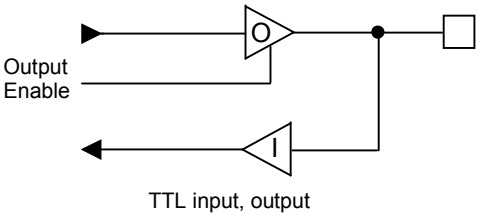
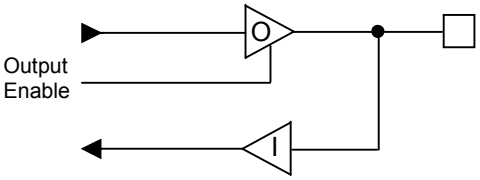
Symbol	Pin status during reset
SDAT	
SCL	
UP_TXD	
SSIOCK1	Clock output
SSIOTXD1	"L" level output
SSIOCK0	Clock output
SSIOTXD0	"L" level output
CKOUTA/CKOUT	
WSA	
SCLA/SCL	
CKOUTD	
SDD	
SCLD	
TDO	High impedance
RTCK	"L" level output

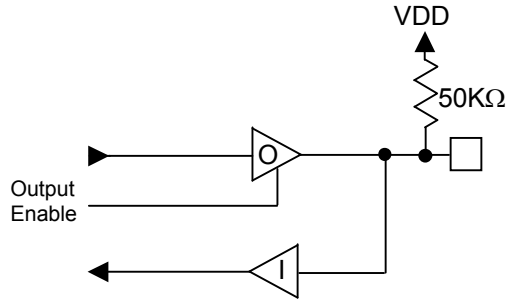
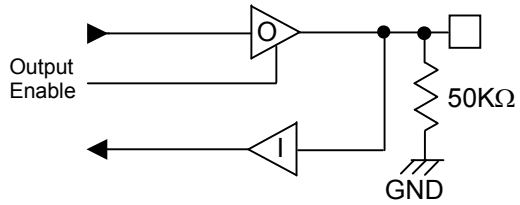
1.3.5 Pin Structure and Processing

1.3.5.1 Pin Structure

Table 1-5 shows the simplified structures of the pins used in the LSI.

Table 1-5 Pin Structure Types

I/O	Symbol	Internal circuit	Type
I	RESET_N	 TTL Schmidt trigger input	(a)
	ADIN0 to ADIN3	 A/D input	(b)
O	TDO		(c)
I/O	PIOA00 to PIOA15, PIOB15 to PIOB00, PIOC00 to PIOC15, PIOE00 to PIOE14, PIOD00 to PIOD15, PWMOUT, SDAT, SCL, nTRST, TCK, RTCK, BOOT1 to BOOT0, EXTBUS, IDEMODE, XA23 to XA01, XD15to XD00, XOE_N, XWE_N, XROMCS_N, XRAMCS_N, XIOCS11_N, XIOCS10_N, XIOCS01_N to XIOCS00_N, XBS1_N to XBS0_N, XWAIT1 to XWAIT0, XSYSCLK, XSDCS_N, XCAS_N, XRAS_N, XSDCLK, XSDCKE, XDQM1 to XDQM0, DREQ, DREQCLR, TCOUT, UP_RXD, UP_TXD, SSIOCK1 to SSIOCK0, SSIORXD1 to SSIORXD0, SSIOTXD1 to SSIOTXD0, CKOUTA/CLKOUT, SDA, WSA, SCLA/SCL, CKOUTD, SDD, WSD, SCLD,EFIQ_N	 TTL input, output	(d)
I/O	PIOE15, PIOF06 to PIOF00, FD7 to FD0, FRD_N, FWR_N, FRB, FALE, FCLE, IDEA2 to IDEA0, IDED15 to IDED00, IDERE_N, IDEWR_N, IDERDY, IDERST_N, IDECS1_N to IDECS0_N, IDEDREQ, IDEDACK_N, IDEIRQ, IDENPCBRID	 5V tolerant TTL input, output	(e)

I/O	Symbol	Internal circuit	Type
	TDI, TMS	 TTL input, output with pull-up	(f)
	TEST_RSV, PLLBYPASS, TMODE3 to TMODE0, RTC_TESTMODE*Note), 32K_TESTMODE* Note)	 TTL input, output with pull-down	(g)

*Note) 1.5V I/F

1.3.5.2 Connection Method of Unused Pins

Table 1-6 lists the connection methods of unused pins of the LSI.

Table 1-6 Connection Methods of Unused Pins

Symbol	Connection method
PIOA, PIOB, PIOC, PIOD, PIOE, PIOF	Output mode
XD15 to XD00	PIO mode and output mode of primary function
XWAIT1,XWAIT0	"L" level
DREQ	"L" level
IDED15 to IDED00	"H" or "L" level
IDERDY	"H" or "L" level
IDEDREQ	"L" level
IDEIRQ	"L" level
IDENPCBRID	"H" or "L" level
FRB	"H" or "L" level
SDAT	Output mode
SSIOCK0 to 2	Open
SDA	"H" or "L" level
WSA, WSD, SCLA/SCL, SCLD	Output mode
ADIN3 to ADIN0	VREF or AGND
OSC32K0	"L" level
OSC48M0	"L" level
OSC11M0	"L" level
TDI	Open
TCK	"L" level
nTRST	"L" level *1
TMS	Open
USB_DP/USB_DM/USB_REXT	Open
USB_VOREF/USB_ATEST	Open

* It is necessary to input a reset pulse to the nTRST pin after turning on the power. If not used, set it as ground level.

Chapter 2

CPU

Chapter 2 CPU

2.1 Overview

This LSI uses the 32-bit RISC CPU ARM946E developed by ARM as its CPU core. 8-Kbyte instruction cache and 8-Kbyte data cache are built in this LSI. There are two operation states in ARM946E: the ARM state that executes a 32-bit long instruction set (ARM instructions), and the Thumb state that executes a 16-bit long instruction set (Thumb instructions) that is a subset of the ARM instructions. A program can be executed by switching these two operation states as necessary.

2.2 Features

CPU core	: ARM946E_88 (Rev. rlpl, TAP ID code 0x1594605D)
Architecture	: ARMv5TE
Instructions	: ARM instruction (32-bit length) and THUMB instruction (16-bit length) can be mixed. Enhanced DSP instructions
General-purpose register bank	: 31 x 32 bits
Cache memory	: I-cache 8 KB, D-cache 8KB Write-through and write-back settings are possible.
Protection unit	: 8 protect areas can be set.
Write buffer	: 16-entry FIFO Non-bufferable or bufferable can be set when non-cacheable. Fixed at bufferable (no other setting allowed) when cacheable.
Built-in debug function	: JTAG interface, break point register

Note 1: This LSI is fixed to the little endian format.

Note 2: ARM946E switches to abort processing if an error response (HRESP = "ERROR") is received by AHB access. However, ARM ignores error responses in the following cases:

1. ARM receives data (an instruction) for which an error response is received during a cache line-fill operation, even if that data is invalid. Also, invalid data (an instruction) is loaded into cache memory.
2. An error response received during a write operation to an area set as bufferable.

2.3 AHB Arbitration

The priority of arbitration is a fixed priority, in which the test interface controller is assigned the highest priority and the ARM CPU the lowest priority. The priority is as follows:

Priority	Master	Note
Highest	Test Interface Controller (TIC)	
|	IDE Controller	Bus Master 001
|	DMA Controller	Bus Master 002
Lowest	ARM CPU 0(Low priority ARM CPU)	arm946E-S

2.4 Details of the CPU Core and the Cache Memory

For the details of the CPU core (ARM946E_88) used in this LSI, refer to the following documents issued by ARM Limited.

ARM v5TE Architecture Reference Manual (ARM DDI 0100E): Mainly describes the CPU.
ARM946 Rev.r1p1 Technical Reference Manual: Mainly describes the cache memories.

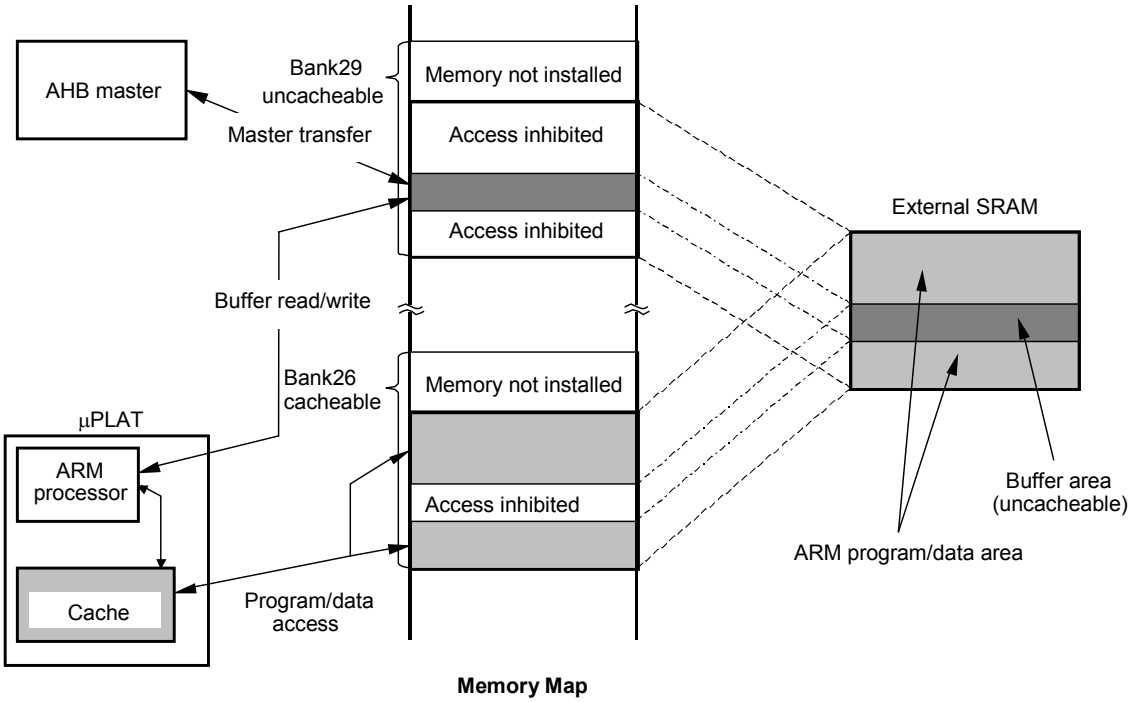
2.5 Precautions When Using Caches

2.5.1 Precautions in Using AHB Master Functions such as DMA (See the Following Figure)

When locating buffer areas in which AHB master access, such as DMA, is executed in AHB memory, a buffer area for AHB master access cannot be located in a bank in which a cache is also located. This is because AHB master access is executed to actual memory and ARM processor access is executed to cache memory, so that a mismatch in memory contents can occur. Therefore, the buffer area for AHB master access should be located in a bank in which a cache is not enabled.

In the case of LSI in which a buffer area for AHB master access cannot be provided in the bank of cache in valid setting, the use of memory mirroring allows for preparing a buffer area for the AHB master access in the memory of cacheable bank. This LSI allows mirroring of Bank26 (external SRAM area) to Bank29 and mirroring of Bank25 (external ROM area) to Bank28. (Note)

For example, one external SRAM can be accessed from separate addresses of Bank26 and Bank29, physically. By setting one of the banks (e.g., Bank26) to cache valid and the other bank (e.g., Bank29) to cache invalid, it is possible to provide a cacheable area (address area of Bank26) and uncacheable area (address area of Bank29) on one external SRAM.
In this example, the program and data of ARM to be accessed at high speed using cache is placed in the address of Bank26 while the buffer for AHB master of DMA etc., is placed in the address of Bank29. By doing so, Bank26 becomes cacheable because the ARM program and data is in Bank26. And on the other hand, Bank29 becomes uncacheable because the buffer that accesses the AHB master and ARM of DMA, etc. is in Bank29.



Memory Map

Note 1: The boundaries of each area must be allocated to the cache line address boundaries (32Byte boundaries).
Note 2: ITCM and DTM are not used in this LSI.

2.5.2 Precautions for Remapping

When Bank0 is set cacheable, implement remapping according to the following procedure to maintain the coherency of the memory device and cache memory.

< Remapping procedure when Bank0 of the address space is set cacheable >

① Flash the instruction cache of Bank0's address space.
② When the data cache of the Bank0's address space is set write-back, implement Flash & Clean. When it is set write-through, implement Flash.
③ To empty the write buffer from data, execute the Drain write buffer operation.
④ Implement remapping.
Moreover, make sure to execute the above instuctions ① to ④ outside the Bank0.

When the cache is set valid in a state in which the same memory is visible in multiple banks by remapping, make sure to set valid only one cache, and to set invalid the other. Also make sure to access only the bank which has been set to cache valid. Otherwise, a mismatch of the memory content might occur for the reason described below.
For instance, when remapping Bank0 an external SRAM of Bank26 set to Cache invalid and set Bank0 to Cache valid, do not access Bank26, the remap source. This is because since the cache is used in accessing Bank0, the memory content (inside the cache) would be updated at any time, while the newest data might not be reflected yet on Bank26, and consequently, Bank26 may still hold the old data. Similarly, the access to the remap source is inhibited when Bank0 is set to Cache valid while other bank is remapped.

2.5.3 Precautions in Setting Cacheable and Bufferable Areas

Cacheable and bufferable settings should only be made for address areas to which memory devices are allocated. This is because if an I/O area or a control register area is set to cacheable or bufferable, the store operation for the I/O or control register will not be executed immediately. The memory map of this LSI assumes that memory devices are allocated to the following areas.

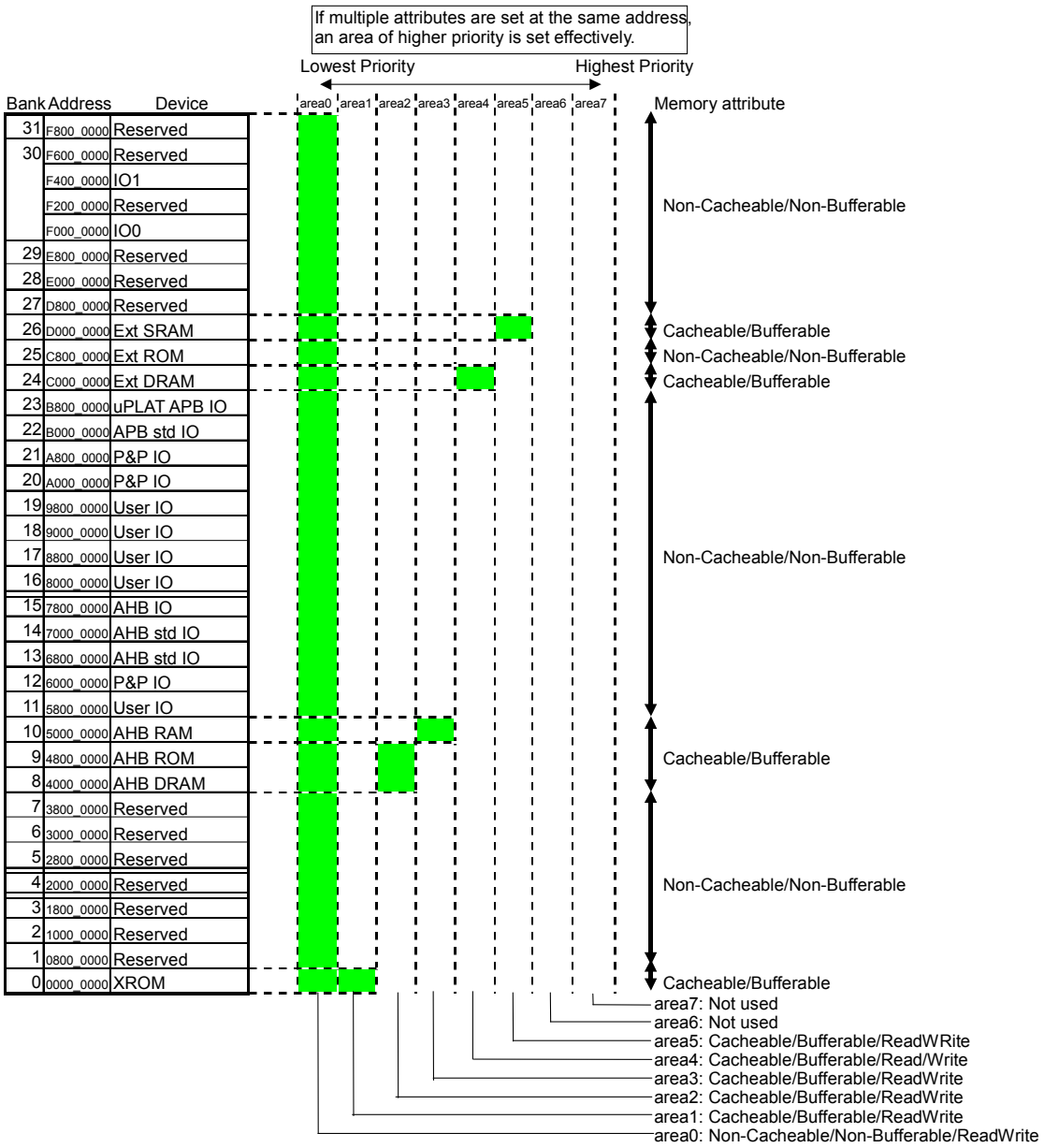
Banks that are Cacheable and Bufferable

Usable Banks	Address
Bank 0	0x0000_0000 to 0x07FF_FFFF
Bank 8 to Bank 10	0x4000_0000 to 0x57FF_FFFF
Bank 24 to Bank 29	0xC000_0000 to 0xEFFF_FFFF

This LSI uses a protection unit with built-in ARM946E-S and attributes for up to eight designated memory areas can be set. The items that can be set for each area are as follows:

- Base address of the area
- Size of the area
- Cacheable/non-cacheable attribute
- Bufferable/non-bufferable attribute (only in data areas)
- Read/write attribute

If a memory area is accessed by a means other than that set in the protection unit attributes (e.g., if a write is executed to a read-only memory area), then ARM will go to the abort exception process.
The base address of the memory area can be set only in units of the area size. The area size can be set in units of a power of two with 4 KB as a minimum unit.
Since the memory Banks of this LSI are provided in units of 128 MB, setting the same memory attributes to 3 consecutive Banks (128 MB x 3 = 384 MB) requires allocation of 2 memory areas.
An example of protection unit settings for the memory map of this LSI is given on the following page. This example shows protection unit settings in which external ROM is remapped to BANK 0 and external memory mirroring is not set in the protection unit.



Protection Unit

Eight areas can be set (area 0 to area 7).
The size of an area is 4 KB to 4 GB. (The size can be set as a power of two.)
The base address of an area is allocated to the boundary of the size that is set.
(For example: If the size of the area is 1 MB, then the base address is set at the 1 MB boundary.)
Individual areas can be set overlapped.
Areas have priority settings (area 0: Lowest priority ⇔ area 7: Highest priority).
The following can be set separately for each area:

- Cacheable
- Bufferable
- Priviledged/User mode access disabled/Read-only/Read-write settings
- Area base address

Protection Unit Setting Example

For further details of the functions of the protection unit, refer to Chapter 4 "Protection Unit" in the "ARM946 Rev.r1p1 Technical Reference Manual".

2.5.4 Notes on Memory Cache and Write Buffer

2.5.4.1 Setting of Cache Policy

Attributes bufferable/non-bufferable and cacheable/non-cacheable, as well as the cache policy (write back, write through) can be set by the bufferable bit and cacheable bit in the protection unit as follows:

Cacheable bit	Bufferable bit	Access mode
0	0	NCNB (Non-Cacheable, Non-Bufferable)
0	1	NCB (Non-Cacheable, Bufferable)
1	0	WT (Write Through) (*)
1	1	WB (Write Back) (*)

(*) If the cacheable bit is set to "1", the bufferable bit specifies the cache policy. In this case, the access mode is always bufferable, enabling the write buffer irrespective of the cache policy.

For further details, refer to Section 2.3.6 "Register 2, Cache configuration register" and Section 2.3.7 "Register 3, Write buffer control register" in the "ARM946 Rev.r1p1 Technical Reference Manual".

2.5.4.2 DMA Region

Since the ARM946E-S incorporated in this LSI does not support the snoop function [(*1)], the memory contents in the cache may become inconsistent with the main memory if an AHB master, such as DMAC, accesses a cacheable DMA region. Therefore, do not set the DMA region cacheable. To avoid data inconsistency, set the DMA region to non-cacheable or take the following procedures. Note that this LSI does not use DTCM.

1. When providing a DMA region in a cacheable area, use the mirroring function [(*2)] of this LSI and set only this DMA region to non-cacheable [(*3)].
2. If DTCM supports the DMA function, assign the DMA region to it [(*4)].

(*1) A function to monitor the bus connected to the main memory to verify whether the cache contents are identical with the contents in the main memory.
(*2) The mirroring function of this LSI maps a memory block to two different address spaces.
(*3) Map a memory to two address spaces using the mirroring function of this LSI. Set one of them cacheable, the other non-cacheable, and assign the DMA region to the non-cacheable memory region. The DMA region is then set non-cacheable without setting the entire memory region non-cacheable allowing avoidance of data inconsistency in cache and memory.
Moreover, since memory access of DMA region from ARM946E-S with non-cacheable setting is slower than cache hit access, use the block transfer instructions (LDM and STM instructions) for high-speed performance.
(*4) No data inconsistency occurs, since the DTCM region cannot be cacheable.

For further information, refer to Section 2.5.1 "Precautions in using AHB master functions such as DMA".

2.5.4.3 Prevention of Data Loss during Power OFF and Reset

In some cases, such as accessing cacheable region from an AHB master, data exist only in the cache memory and write buffer of the ARM946E-S built in this LSI. In these cases, if this LSI is reset or its power is turned OFF, valid data might be lost. To prevent such data loss, the next procedures must be taken to drain the cache memory and write buffer of ARM946E-S.

[Case of a write-back data cache]

1. Disable interrupts.
2. Execute the Clean and Flush operation to all data cache lines (8KB 4 sets x 64 lines).
3. Drain the write buffer with the Drain Write Buffer operation [(*1)]. When the execution of the Drain Write Buffer operation is completed, the write buffer becomes empty. At this point, there is no risk of data loss.

[Case of a write-through data cache]

1. Disable interrupts.
2. Execute the Flush operation (*2).
3. Drain the write buffer with the Drain Write Buffer operation. When the execution of the Drain Write Buffer operation is completed, the write buffer becomes empty. At this point, there is no risk of data loss.

(*1) The ARM946E-S is has the following operation related to cache memory operation.

- Flush operation: This invalidates the cache contents.
- Clean operation: This performs a write back and keeps the memory and cache contents consistent, when the specified cache line is Dirty Data.
- Clean and Flush operation: This invalidates the cache contents after performing a write back and keeping the memory and cache contents consistent, when the specified cache line is Dirty Data.
- Drain Write Buffer operation: This drains the write buffer. This operation stalls the ARM946E-S core until any outstanding accesses in the write buffer have been completed.

(*2) The Flush operation is not necessary in processes requiring initialization such as power-off and reset. However, in a write access to cacheable region from an AHB master, the old data in the cache memory must be invalidated.

The cycles required for the drain of the cache memory and write buffer depend on the number of lines including the Dirty data in the cache memory, access time of the external memory and the clock ratio between ARM946E-S and AHB. Described below is the number of AHB clock cycles required for the drain of the cache memory and write buffer.

AHB clock cycles required for the drain of the cache memory and write buffer

Cache setting	Number of AHB clock cycles	
	Instruction cache ON	Instruction cache OFF
Write back	3200	17000
Write through	100	100

Conditions:

ARM946E-S clock:	AHB clock = 1:2
Data memory:	SDRAM
Program memory:	ROM
SDRAM controller:	z0248a0_dramc
SDRAM timing setting:	DRPC register= 0x2 (tRCD: 2, tRAS: 3, tRP: 2, tDPL: 1) Column address length: 8 bits, CL = 3 External memory bus width: 32 bits
Refresh cycle:	16 µs
ROM timing setting:	ROMAC register = 0x10 (OE/WE pulse width: 1)
Data cache setting:	
Write back:	All cache lines Dirty, write buffer full
Write through:	Write buffer full

For more details, refer to Chapter 3, Section 2.3.10 "Register 7, Cache operation register" in the "ARM946 Rev.r1p1 Technical Reference Manual (ARMDDI0201B)".
Refer also to Section 3.3.5 "Data cache clean and flush" of the same manual for information on a sample program of the Clean and Flush process.

2.5.4.4 Prevention of Data Loss During Use of a Memory Controller with Memory Access Disable Mode

When using a memory controller with a memory access disable mode such as the DRAMC self-refresh mode, a memory access might occur unintentionally by software and valid data might be lost regardless that the memory controller is disabling the relevant memory.

The following describes the two cases when valid data in the relevant memory might be lost.

[Case 1]

Case of data write access to the relevant memory region during and directly before the memory controller disables the memory access, having the memory region set WB (Write Back), or case of data read access to arbitrary cacheable memory regions.

In the above case, when the memory controller goes to the memory access disable mode, the ARM processor might not have completed draining the write buffer for the relevant memory region. When the write access is executed after the memory controller disables the memory access (*1), valid data of the write buffer will be lost.
In addition, a cache read miss of the ARM processor occurs, giving a possibility of a write back to the relevant memory while the memory controller is disabling the memory access. Since this write access is not done correctly (*2), valid data of the relevant memory region will be lost if it is located only in the cache.

(*1) Since the write to the relevant memory region (bufferable attribute) is done through the ARM processor write buffer, a write to the mode setting registers (non-bufferable attribute) might be done first. Beware the access order might be reversed even if writes to the relevant memory region were to be executed first by software.

(*2) For the write back access of the ARM processor, AHB error response will be ignored by the ARM processor even if it is returned by the memory controller. Note that, in this case, the ARM processor will not perform a write back again, and not go to the abort exception routine.

[Case 2]

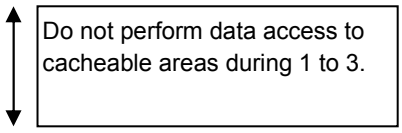
Case of data write access to a relevant memory region directly before the memory controller, having the relevant memory region set WT (Write Through) or NCB (Non-Cacheable, Bufferable), goes to the memory access disable mode.

The operation in the above case is the same as the data write access to the relevant memory regions in Case 1. In this case, no malfunction due to data read access to the relevant memory region occurs.

To prevent such data loss, execute the Drain Write Buffer operation directly before the memory controller goes to the memory access disable mode, and do not perform data access to arbitrary memory regions set on cacheable areas (*3) while the memory controller is disabling the memory access.

Described below is a control sequence until the memory controller ends the memory access disable mode.

1. Execute the Drain Write Buffer operation.
2. Go to the memory access disable mode.
3. End the memory access disable mode.



Described below is a control sequence example setting DRAMC to self-refresh mode during CPUGHALT mode when DRAMC is used as the memory controller. Do not perform data access to arbitrary memory regions set on cacheable areas (*3) during the DRAMC self-refresh mode.

[Transition to the CPUGHALT mode]

1. Mask the ARM processor interrupt.
2. Execute the Drain Write Buffer operation
3. Enter the self-refresh mode.
4. Set the CPUGHALT mode.
5. Execute the WFI operation.

↓

(CPUGHALT mode)

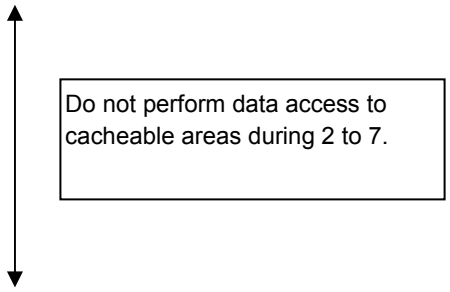
↓

[Recovery from the CPUGHALT mode]

6. Recover with a recovery source (Interrupt or debug request).
7. End the DRAM self-refresh mode.
8. Release the ARM processor interrupt mask.

↓

(Servicing the interrupt handler)



(*3) Beware the memory regions set on all cacheable areas including the relevant memory region are also included.

Described below is a sample program to execute the above sequence 2 to 7. Before executing this program, the ARM processor interrupts must be masked, and the stack area used in this sample (memory area of the address SP) must be set to the attribute non-cacheable.

```
halt:
        STMDB   SP!, {R4-R8,LR}

        ; disable all interrupts before this function is called so that interrupt handler does not process.
        LDR     R4, =SYS_BASEADDR
        LDR     R3, [R4, #CGBCNT0]
        MOV     R2, #0x3C

        MOV     R0, #0x0                ; execute drain write buffer operation.
        MCR     p15, 0, r0, c7, c10, 4

        MOV     R0, #DRAM_SELF_VALUE        ; sdram gets into the self refreshment mode. (0x06)
        LDR     R5, =DRAMC_BASEADDR
        STR     R0, [R5, #DCMD]

        LDR     R12, =0x100             ; wait 10us
halt_1:
        SUB     R12, R12, #0x01
        CMP     R12, #0x00
        BNE     halt_1

        LDR     R0, [R4, #CLKSTP]   ; cpu gets into halt mode.
        ORR     R0, R0,#CPUG_HALT_VALUE ; (0x04)
        STR     R2, [R4, #CLKSTP]
        STR     R0, [R4, #CLKSTP]

        MOV     R0, #0x0                ; execute WFI operation.
        MCR     p15, 0, r0, c7, c0, 4

        LDR     R12, =0x100             ; wait 10us
```

For further details, refer to Chapter 3 "Caches" and Section 6.5.1 "Write Buffer operation" in the "ARM946 Rev.r1p1 Technical Reference Manual (ARMDDI0201B)."

2.5.4.5 Rewriting the Program

Since the ARM946E-S built in this LSI does not support the snoop function, rewriting the instruction code in a cacheable region during the program execution results in inconsistency between the main memory and instruction/data cache. In this case, dirty data in the data cache must be flushed into the main memory, and old program in the instruction cache must be invalidated. This procedure is described below.

[Case of a write-back data cache]

1. Disable interrupts.
2. Rewrite the program. (The ARM946E-S core loads and writes the new program provided beforehand as data (*1). At this time, if a write access hits the data cache, the written data become dirty data existing only in the data cache. This will be solved in the following steps.)
3. Execute the Clean operation to empty the data cache from the dirty data.
4. Drain the write buffer with the Drain Write Buffer operation. At this point, the entire new program will be placed in the main memory.
5. Execute the Flush operation to the instruction cache to prevent the old program in the instruction cache from executing. At this point, rewriting the program is complete, allowing the new program to be executed.

(*1) The program can be rewritten by bypassing the ARM946E-S core, using an AHB master DMA.
In this case, the processes 3 and 4 in the procedure above are not necessary, and the new program can be executed with only the Flush operation after the rewrite process.

[Case of a write-through data cache]

1. Disable interrupts.
2. Rewrite the program (*2).
3. Drain the write buffer with the Drain Write Buffer operation.
4. Execute the Flush operation to the instruction cache to prevent the old program in the instruction cache from executing. At this point, rewriting the program is complete, allowing the new program to be executed.

(*2) The program can be rewritten by bypassing the ARM946E-S core, using an AHB master DMA.
In this case, the process 3 in the procedure above is not necessary, and the new program can be executed with only the Flush operation after the rewrite process.

For further details, refer to Chapter 3 "Caches" in the "ARM946 Rev.r1p1 Technical Reference Manual (ARMDDI0201B)" and Section 5.5.2 "Instruction cache coherency" in the "ARM Architecture Reference Manual (ARMDDI0100E)."

2.5.4.6 Order of Write Access to a Bufferable Region

Since the write access to a bufferable region is executed as an AHB write access through the ARM946E-S write buffer, accesses on the AHB following the write access to a bufferable region might be executed first.
The following access sequence gives an example where the program access order on ARM946E-S differs from the access order on the AHB.

● Write and read access to a non-cacheable and non-bufferable region after a write access to a bufferable region (*).

(*) Write to a bufferable region occurs in the following settings:

- Write to a NCB (Non-Cacheable, Bufferable) region
- Write to a WT (Write Through) region
- Write to a WB (Write Back) region (not generated as an AHB cycle when hit to a cache)

Since a read from a bufferable region after a write in the above-mentioned access mode is executed after the write buffer drain is complete, the program access order in this case will be the same.

To execute the program on ARM946E-S with the same access order as on the AHB, the access destination should be non-cacheable and non- bufferable.

For further information, refer to Section 6.5.1 "Write Buffer operation" in the "ARM946 Rev.r1p1 Technical Reference Manual (ARMDDI0201B)."

Chapter 3

Address Map

Chapter 3 Address Map

3.1 Overall Address Map

Bank No.	Address		
31	F800_0000	Reserved	External
30	F600_0000	Reserved	
	F500_0000	Ext-IO11 (corresponds to XIOCS11_N)	
	F400_0000	Ext-IO10 (corresponds to XIOCS10_N)	
	F200_0000	Reserved	
	F100_0000	Ext-IO01 (corresponds to XIOCS01_N)	
	F000_0000	Ext-IO00 (corresponds to XIOCS00_N)	
29	E800_0000	Ext-SRAM (mirroring)	
28	E000_0000	Ext-ROM (mirroring)	
27	D800_0000	Ext-DRAM (mirroring)	
26	D000_0000	Ext-SRAM	
25	C800_0000	Ext-ROM	
24	C000_0000	Ext-DRAM	
23	B800_0000	μPLAT Core	APB
22	B000_0000	μPLAT Standard for legacy	
21	A800_0000	User5 (μPLAT Standard for Plug & Play)	
20	A000_0000	User4 (μPLAT Standard for Plug & Play)	
19	9800_0000	User3	
18	9000_0000	User2	
17	8800_0000	User1	
16	8000_0000	User0	
15	7800_0000	μPLAT Core	AHB
14	7000_0000	μPLAT Standard	
13	6800_0000	PCI bus (not installed)	
12	6000_0000	Reserved	
11	5800_0000	Reserved	
10	5000_0000	AHB RAM	
9	4800_0000	AHB ROM	
8	4000_0000	AHB DRAM (not installed)	
7	3800_0000	Reserved	Processor
6	3000_0000	Reserved	
5	2800_0000	Reserved	
4	2000_0000	Reserved	
3	1800_0000	Reserved	
2	1000_0000	prc RAM (not installed)	
1	0800_0000	prc ROM (not installed)	
0	0000_0000	Remappable space	

∗ The table above lists basic address assignment, including AHB DRAM, prc RAM and prc ROM that are not installed.

3.2 Detailed Address Map of the AHB Section

Base Address	
7800_0000	μPLAT core
7000_0000	μPLAT standard (not used)
6800_0000	PCI bus (not used)
6000_0000	User2 (not used)
5E00_0000	User11 (not used)
5C00_0000	User10 (not used)
5A00_0000	User01 (not used)
5800_0000	User00 (not used)
5000_0000	AHB RAM (128kByte)
4800_0000	AHB ROM (16kByte)
4000_0000	AHB DRAM (not included)

8000_0000	
7C00_0000	Reserved
7BF0_0000	Extended IRC
7BE0_0000	DMAC
7BC0_0000	Reserved
7BB0_0000	HS-USB
7BA0_0000	Reserved
7B90_0000	IDE controller
7820_0000	Reserved
7810_0000	BIC/DRAMC
7800_0000	IRC

3.3 Detailed Address Map of the APB Section

Base Address	
B800_0000	μPLAT core
B000_0000	μPLAT standard for legacy
A800_0000	User5
A000_0000	User4
9800_0000	User3
9000_0000	User2
8800_0000	User1
8000_0000	User0

B800_3000	Reserved
B800_2000	μPLAT -SIO
B800_1000	μPLAT -TIMER
B800_0000	SYSCON
B600_0000	
B400_0000	PSELSL064
B200_0000	PSELSL032
B000_0000	PSELSL000
A800_0000	PSELSP128
A000_0000	PSELSP000
9E00_0000	PSELU15
9C00_0000	PSELU14
9A00_0000	PSELU13
9800_0000	PSELU12
9600_0000	PSELU11
9400_0000	PSELU10
9200_0000	PSELU09
9000_0000	PSELU08
8E00_0000	PSELU07
8C00_0000	PSELU06
8A00_0000	PSELU05
8800_0000	NAND Flash buffer
8600_0000	NAND Flash controller
8400_0000	I2S-recieve
8200_0000	I2S-trans + FIFO RAM
8000_0000	CONFIG

B7F0_0000	Timer
B7E0_0000	WDT
B7D0_0000	PWM
B7C0_0000	RTC
B7B0_0000	SSIO/I2C
B7A0_0000	GPIO
B790_0000	PSELSL121
B780_0000	PSELSL120
B770_0000	PSELSL119
B760_0000	PSELSL118
B750_0000	PSELSL117
B740_0000	PSELSL116
B730_0000	PSELSL115
B720_0000	PSELSL114
B710_0000	PSELSL113
B700_0000	PSELSL112
B6F0_0000	PSELSL111
B6E0_0000	PSELSL110
B6D0_0000	PSELSL109
B6C0_0000	PSELSL108
B6B0_0000	PSELSL107
B6A0_0000	PSELSL106
B690_0000	PSELSL105
B680_0000	PSELSL104
B670_0000	PSELSL103
B660_0000	PSELSL102
B650_0000	PSELSL101
B640_0000	PSELSL100
B630_0000	PSELSL099
B620_0000	PSELSL098
B610_0000	PSELSL097
B600_0000	ADC

3.4 Remapping

RMPM[3:0]				BOOT[1:0]		Bank 0	Bank 9	Bank 10	Bank 25	Bank 26	Bank 24
3	2	1	0	1	0						
0	x	x	x	0	0	MCP Flash + External ROM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
				0	1	External ROM	AHBROM	AHBRAM	external ROM	External SRAM	External SDRAM
				1	0	AHBROM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
				1	1	AHBROM	AHBROM	AHBRAM	External ROM	External SRAM	External SDRAM
1	0	0	0	x	0	External SRAM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
					1				External ROM		
1	0	0	1		0	External SDRAM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
					1				External ROM		
1	0	1	0		0	AHBRAM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
					1				External ROM		
1	0	1	1		x	Cannot be used.					
1	1	x	x								

[Cautions]

- In the sections notated as MCP Flash + external ROM, MCP Flash is accessed for the first 512 Kbytes (4M bits) of addresses, and the external ROM area is accessed for the subsequent addresses. Therefore, the address area of the first 512 Kbytes in external ROM cannot be used.
- Also, because the access timing via the external bus controller is common to MCP Flash and external ROM, and thus cannot be set independently. Set it to match the slower one.
- If AHBRAM is remapped, it can also be accessed from the source bank. In this case, the values cannot be guaranteed if access is made by setting one bank with cache and the other without cache.

Chapter 4

Mode Settings

Chapter 4 Mode Settings

4.1 Overview

This LSI has the mode setting pins to control the enabling/disabling of functions in units of functional blocks. All the mode pins determine the applied voltage level at power on; do not change it afterward.

4.1.1 Pin List

Symbol	I/O	Function
BOOT0 to BOOT1	I	Controls the boot device.
EXTBUS	I	Switches between the external bus pin and GPIO.
IDEMODE	I	IDE and NAND Flash pin switch signal
PLLBYPASS	I	Sets the PLL bypass mode.
TEST_RSV	I	Reserved. Always leave this pin at "0" or open.
TMODE0 to TMODE3	I	Sets the test mode.

Always use the TMODE0 to TMODE3 pins at ground level. Always use the PLLBYPASS pin at ground level. For all mode pins, the input levels cannot be changed after power is turned off since levels are set up when power is turned on.

4.2 Mode Setting Pins

- Mode setting by BOOT1 and BOOT0

BOOT1	BOOT0	Boot device	Address range of MCP Flash	Address range of external ROM	Remark
0	0	MCP Flash	C800_0000 to C807_FFFF	C808_0000 to C8FF_FFFF	
0	1	External Flash	Cannot be accessed	C800_0000 to C8FF_FFFF	Always set EXTBUS to "1".
1	0	AHBROM	C800_0000 to C807_FFFF	C808_0000 to C8FF_FFFF	
1	1	AHBROM	Cannot be accessed	C800_0000 to C8FF_FFFF	Always set EXTBUS to "1".

To use only external Flash without using internal Flash, set the BOOT0 pin to "1". At that time, always set EXTBUS to "1". C800_0000 to C8FF_FFFF are assigned to external ROM, and it becomes possible to boot up from external ROM by remapping this bank.

The BOOT1 pin switches to boot up using Flash (MCP Flash or external Flash) or AHBROM. If the BOOT1 pin is set to "0", the ML696201/69Q6203 boots from Flash (MCP Flash or external Flash). If the BOOT1 pin I set to "1", the ML69Q6500 boots from AHBROM.

In the case of the ML696201, set the BOOT0 pin to "1". At that time, always set EXTBUS to "1". C800_0000 to C8FF_FFFF are assigned to external ROM, and it becomes possible to boot up from external ROM by remapping this bank.

In the case of the ML69Q6203, set the BOOT0 pin to "0". C800_0000 to C807_FFFF are assigned to MCP Flash, and it becomes possible to boot up from MCP Flash by remapping this bank. If EXTBUS is set to "1", an external ROM area can be used, and C8008_0000 to C8FF_FFFF are assigned.

In the case of the ML69Q6203, to use an external ROM area by setting the BOOT0 pin to "0", the chip select signal for external ROM becomes active when C8008_0000 to C83F_FFFF are accessed.
Therefore, if ROM is directly connected to an external device, MCP Flash is accessed for the first 512 Kbytes (4 Mbits) of addresses, and an external ROM area is accessed for the subsequent addresses. As a result, note that the address area of the first 512 Kbytes in external ROM cannot be used.

- Mode setting by EXTBUS

EXTBUS	External bus/GPIO	Remarks
0	GPIO	Sets PIOA15 – PIOA00, PIOB15 – PIOB00 and PIOC15 – PIOC00 to the GPIO of the primary function.
1	External bus	Sets PIOA, PIOB and PIOC to the external bus of the secondary function.

- Mode setting by IDEMODE

IDEMODE	NAND FLASH/IDE controller	Remarks
0	NAND Flash	Sets GPIOF and the NAND Flash function to valid pins.
1	IDE controller	Sets the IDE controller function to a valid pin.

- Mode setting by PLLBYPASS

PLLBYPASS	PLL usage mode	Remarks
0	Use PLL	48 MHz x 5/2 is the maximum clock frequency of the CPU.
1	Bypass PLL	Setting is not allowed.

- Mode setting by TEST_RSV

TEST_RSV	Test mode	Remarks
0	Normal operation mode	—
1	Reserved	Setting is not allowed.

- Mode setting by TMODE3 to TMODE0

These are test mode settings. Always use the test modes at "L". Operation cannot be guaranteed if a level other than "L" is set.

TMODE3	TMODE2	TMODE1	TMODE0	Mode name
0	0	0	0	Normal operation mode
0	0	0	1	Setting is not allowed.
0	0	1	0	
:	:	:	:	
1	1	1	0	
1	1	1	1	

Chapter 5

Clock Control

Chapter 5 Clock Control

5.1 Overview

This LSI has three clocks - the FCLK supplied to the CPU, the HCLK supplied to the AHB bus, and the PCLK supplied to the APB bus, each of which has a clock gear (clock division select function).

5.1.1 Configuration

FCLK : 120 MHz maximum. 120 MHz or 32.768 kHz of PLL output can be selected for the source clock. The clock gear allows to switch among 1/2, 1/4, 1/8, 1/16 and 1/32.

HCLK : 60 MHz maximum (1/2 of FCLK). The source clock is the FCLK. The clock gear allows to set either 1/2 or 1/4.

PCLK : 30 MHz maximum (1/2 of HCLK). The source clock is the HCLK. The clock frequency is always 1/2 the HCLK.

5.2 Clock System Diagram

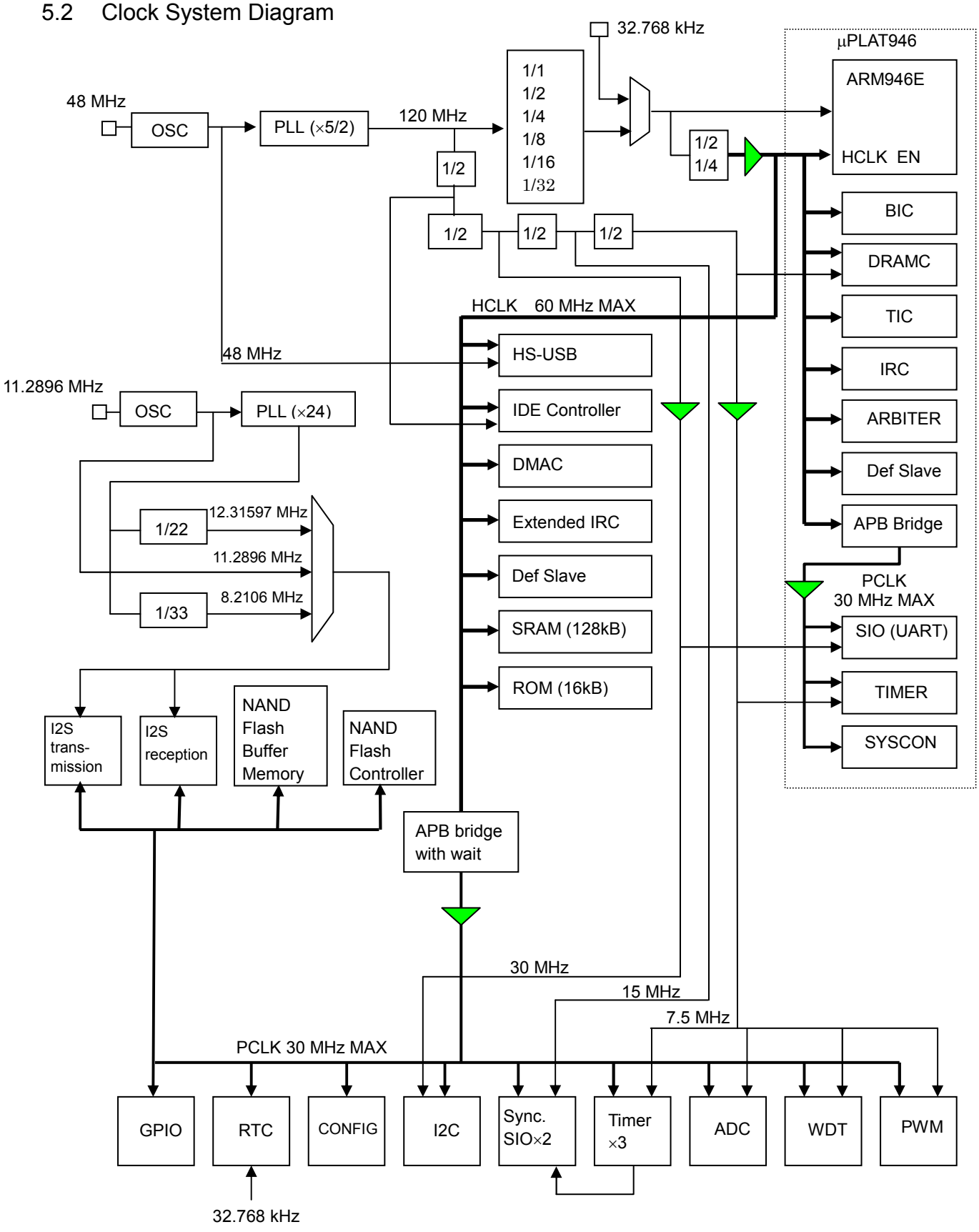


Figure 5-1 Clock System Diagram

Chapter 6

Reset Control

Chapter 6 Reset Control

6.1 Overview

This LSI has two type of resets: a reset by the external reset pin (RESET_N) and a reset by an overflow of the watchdog timer. In the even of a reset, it is possible to determine whether it is a reset by the RESET_N pin or a reset by an overflow of the watchdog timer by reading the content of the RSTSTATUS bit of the WDSTAT register.
The minimum reset pulse width required for system initialization is 10 μs. Operation cannot be guaranteed if a reset of less than 10 μs is input.

6.1.1 Pin List

Symbol	I/O	Function
RESET_N	I	Reset input

6.2 Reset Methods

6.2.1 Reset by an External Pin

This LSI can be reset by inputting "0" (low) to the RESET_N pin for 10 μs or more. However, it is necessary to maintain the RESET_N pin at low level for 10 ms or more, because the time to wait for the crystal oscillation to stabilize is required at power on or if a recovery from standby mode is performed by an external reset.

6.2.2 Resetting by Watchdog Timer Overflow

When the watchdog timer mode ("0") is selected by the ITM bit of the WDTBCON register, a system reset ("1") is selected by the OFINTMODE bit, and then the watchdog timer is operated, a system reset occurs if the watchdog timer overflows.
When a watchdog timer overflow occurs, an internal reset signal is generated. Note that this reset is the same as the reset by an external pin, except that the WDSTAT register is set to 0x0001.

6.3 Reset Sequence

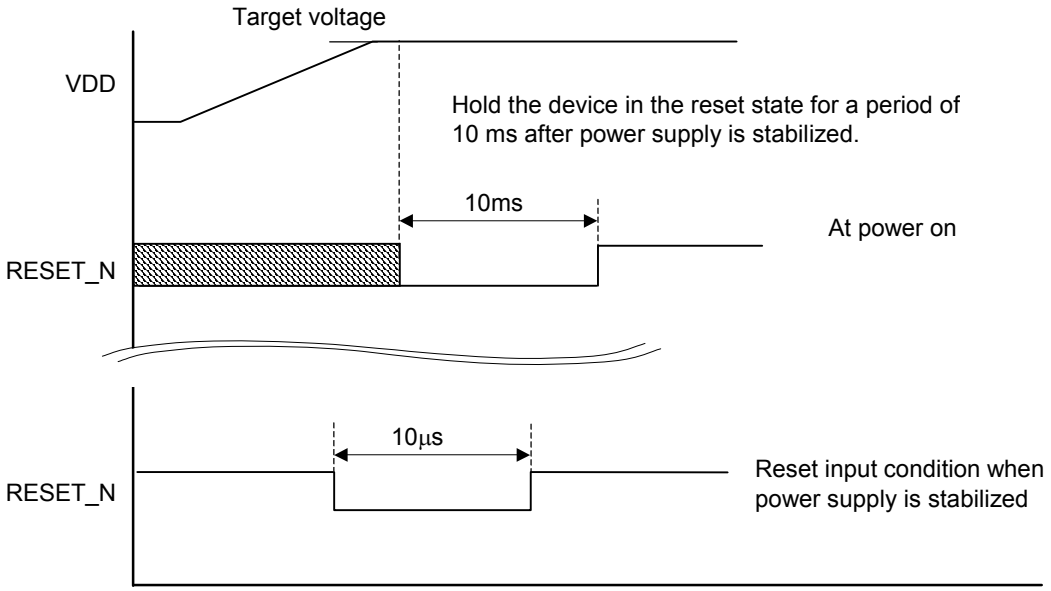


Figure 6-1 Reset Signal Timing

Chapter 7

Power Consumption Control and System Control

Chapter 7 Power Consumption Control and System Control

7.1 Overview

This LSI can be set to the Backup mode that stops power supply to modules other than to the RTC (Real Time Clock) in order to reduce power consumption. Also, power saving is supported by software using the Stop mode that stops clock supply, the HALT mode that controls clock supply in function block units, and the Clock Gear function that controls cock frequency division in function block units.
For the source clock of the CPU clock (FCLK), either a 120 MHz clock, which is 5/2 multiplication of the 48 MHz clock supplied to the LSI or a 32.768 kHz clock can be selected. For the frequency divide value of the 120 MHz clock system, 1/1, 1/2, 1/4, 1/8, 1/16 or 1/32 can be selected. For the source clocks of the AHB module and bus clock (HCLK), either 1/2 or 1/4 of the CPU clock (FCLK) can be selected by the clock gear.
The bus clock (PCLK) of the APB slave is fixed to 1/2 of the bus clock (HCLK).

7.2 Power Consumption Control Function

The following describes the power consumption control method.

7.2.1 RTC (Real Time Clock) Module Backup Mode

It is possible to supply power to only the RTC module, and shut down power to other modules. When implementing this, the following transition and recovery sequences must be observed. However, it is not possible to shut down power to only the RTC module, and supply power to other modules. Be sure to supply power to the RTC module even when the RTC module is not used.

7.2.1.1 Procedure for Transiting to the Backup Mode

1. Set the RTCEN bit of the PDMASK register in the configuration register module to "0".
2. Wait for the time longer than the designated period (150 μs). Or, access the RTC scratch pad register, and consecutively read/write an arbitrary value and its inverted value several times. If "0" is read all the time, determine that the RTC module can transit to the Backup mode.
3. Shut down all power supplies other than VDD_RTC at the same time.
   In the transient state when powering off VDD_IO and VDD_CORE, power off VDD_IO, holding the VDD_IO level equal to or higher than the VDD_CORE level.

7.2.1.2 Procedure for Recovering from the Backup Mode

1. Turn on all power supplies other than VDD_RTC at the same time.
2. Assert the external reset for 10 ms or more.
3. Set the RTCEN bit of the PDMASK register in the configuration register module to "1".
4. Wait for the time longer than the designated period (150 μs). Or, access the RTC scratch pad register, and consecutively read/write an arbitrary value and its inverted value several times. If all values are the same as the expected values, determine that the RTC module can be accessed.
5. The RTC module is ready to be accessed.

7.2.2 Clock Stop Modes

There are two clock stop control modes: STOP mode, and clock stop mode in function block units.

7.2.2.1 STOP Mode

The STOP mode stops the oscillation of clocks by setting a register using software. Because all internal clocks of the LSI stop, power consumption can be reduced in this mode. While recovering from the STOP mode, clocks are not supplied internally until clock oscillation stabilizes. Operation resumes after the oscillation stabilization wait time has elapsed according to the value set in the CKWT register. Since the RTC module's count operation functions normally even in the STOP mode, it is possible to recover by using an interrupt of the RTC module.

The following lists the sources for transiting to and recovering from STOP mode.

Source for transiting to STOP mode	Sources for recovering from STOP mode
Writing to register	RTC interrupt GPIO interrupt (asynchronous level mode) External reset

[Caution]
When using a GPIO interrupt as a recovery source from the STOP mode, it can only be used in the level mode. Be sure to maintain the interrupt state until a recovery from the STOP mode completes.

7.2.2.1.1 Method of Recovering from the STOP Mode

1. Recovering by the interrupt
   The oscillation stabilization wait time of 10 ms or more should be programmed in the CKWT register (See Section 7.4.4) before the LSI is set in the STOP
   mode.
   When the LSI is recovered from the STOP mode by the interrupt, the LSI waits automatically for the oscillation stabilization wait time programmed in the CKWT register until the clock stabilizes. After that, the clock is supplied to CPU.

2. Recovering by the external reset
   The external reset signal should be asserted for 10 ms or more.
   It is not necessary to set the CKWT register.

7.2.2.1.2 Method of Recovering from the STOP Mode When the 32 kHz Clock is Selected for the CPU Clock.

1. Recovering by the interrupt
   The oscillation stabilization wait time programmed in the CKWT register should be set to a maximum value before the LSI is set in the STOP mode.
   When the LSI is recovered from the STOP mode by the interrupt, the LSI waits automatically for the oscillation stabilization wait time programmed in the CKWT register until the clock t stabilizes. After that, the clock is supplied.
   Switching to the high speed clock should be made after the PLL and OSC oscillation stabilization wait times of 10 ms or more have elapsed.

2. Recovering from the external reset
   The 32 kHz clock is switched to the 120 MHz clock at the time of reset.
   So the external reset should be asserted for 10 ms or more.
   It is not necessary to set the CKWT register.

7.2.2.2 Clock STOP Mode in Functional Block Units

Each of the HS-USB and IDE controller has its own power down mode. If these modules are not used, low power consumption can be achieved by placing these modules in the power down mode.
This LSI can control clock supply/stop for each functional block. When a block is accessed while the clock supply to the block is stopped, the function does not operate normally. Do not access the registers other than the HS-USB system control register (SysCtl) used for releasing the Power-Down mode and IDE clock control (CLKCTL) while the clock supply is stopped.
When the mode is changed to the HALT or STANDBY mode and the control is returned due to a source other than reset while the clock for each functional block is stopped, this LSI restores control while maintaining the clock stop state of each functional block.

The following shows the blocks to which block supply can be stopped.

Block	Stopping clock supply by software	Low power consumption mode in module	Remarks
uPLAT-SIO	○	—	Control using the SYSCON register in μPLAT.
uPLAT-TIC	○	—	Control using the SYSCON register in μPLAT.
DMAC	○	—	Do not stop clock supply when starting via the DMA. If an attempt is made to access this module while clock supply is being stopped, it is handled as an error and a data abort operation will be performed.
DRAMC	○	—	To control clock supply by software, self-refresh must always be executed. If an attempt is made to access this module while clock supply is being stopped, it is handled as an error and a data abort operation will be performed.
HS-USB	—	○	If an attempt is made to access this module during power down, it will not be handled as an error and a regular operation will be performed. For more information about operation during power down, see the chapter on USB.
IDE controller	—	○	If an attempt is made to access this module during power down, it will not be handled as an error (data abort) and a regular operation will be performed. For more information about operation during power down, see the chapter on IDE.
I^2S transmission	○	—	If an attempt is made to access this module while clock supply is being stopped, access is denied but an error (data abort) will not occur.
I^2S reception	○	—	If an attempt is made to access this module while clock supply is being stopped, access is denied but an error (data abort) will not occur.
NAND FLASH controller	○	—	If an attempt is made to access this module while clock supply is being stopped, an error (data abort) will not occur.
NAND FLASH Buffer	○	—	If an attempt is made to access this module while clock supply is being stopped, an error (data abort) will not occur.

All of the above clock stop controls are performed using the configuration register module.
For more information, see a detailed description of the configuration register module.

7.2.2.3 Clock Gear

While in operation, the clock gear can change the frequency divide value of a clock by writing during operation into the CGBCNT0/CGBCNT1/CGBCNT2 register of the SYSCON module by software.

For the source clock of the CPU clock (FCLK), either a 120 MHz clock, which is 5/2 multiplication of the 48 MHz clock supplied to the LSI or a 32.768 kHz clock can be selected. For the frequency divide value of the 120 MHz clock system, 1/1, 1/2, 1/4, 1/8, 1/16 or 1/32 can be selected.
For the source clocks of the AHB module and bus clock (HCLK), either 1/2 or 1/4 of the CPU clock (FCLK) can be selected by the clock gear.
The bus clock (PCLK) of the APB is fixed to 1/2 of the bus clock (HCLK).

The refresh clocks of µPLAT-TIMER, µPLAT-SIO(UART), WDT, TIMER, PWM, SSIO, I2C, I2S and DRAM are not affected by frequency divide values changed by the clock gear.

Also, using the clock gear, it is possible to transit to the HALT or STOP mode while the operating frequency is being reduced. In this case, it recovers to the state before transiting to the HALT or STOP mode.

When changing a supply clock using the clock gear while DRAM is connected, if the clock frequency after change will be below the minimum frequency of the DRAM operation, it is necessary to transit to the Self Refresh mode by software.

7.3 List of Registers

7.3.1 SYSCON Module

Address	Register name	Symbol	R/W	Initial value [H]
0xB800_0000	ID register	IDR	R	0x0000_0100
0xB800_0004	Clock stop register	CLKSTP	R/W	0x0000_0000
0xB800_0008	Clock (CGB) control register 0	CGBCNT0	R/W	0x0000_0000
0xB800_000C	Clock supply wait time control register	CKWT	R/W	0x0000_00FF
0xB800_0010	Remap control register	RMPCON	R/W	0x0000_0000
0xB800_0014	<Reserved>	—	—	—
0xB800_0018	Clock (CGB) control register 1	CGBCNT1	—	0x0000_0000
0xB800_001C	Clock (CGB) control register 2	CGBCNT2	R/W	0x0000_0000

When writing into this register, write an applicable value immediately after writing "0x3C" to prevent erroneous writing..

7.3.2 Configuration Register Module

The following shows a list of registers pertaining to system control. For a description of each register, see the chapter on the configuration register module.

Address	Register name	Symbol	R/W	Initial value [H]
0x8000_0000	Power-down master register	PDMASK	R/W	0x0000_0000
0x8000_0004	Module clock stop register	MCKST	R/W	0x0000_0000
0x8000_0008	PIO pin switching register	PIOCTL	R/W	0x0000_0000
0x8000_000C	I^2S control register	I2SCNTL	R/W	0x0000_0000
0x8000_0010	OTHER configuration	OCONFIG	R/W	0x0000_0000
0x8000_0014	ADC test register	ADCTST	R/W	0x0000_0000

When writing into this register, write an applicable value immediately after writing "0x3C" to prevent erroneous writing..

7.4 Description of Registers

7.4.1 ID Register (IDR)

The IDR is a read-only register that indicates the type and revision of a (PLAT product). This register is for 32-bit access.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	TYPE[9:0]										REVISION[2:0]		
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Address: 0xB8000000
Access: R
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
By giving consideration to future expansion, it is recommended to write "0" when writing, not expecting "0" when reading.

[Bit Description]

- **REVISION[2:0]** (bits 0 to 2)
  In this LSI, "000" is read.

- **TYPE[9:0]** (bits 3 to 12)
  In this LSI, "000010000" is read.

7.4.2 Clock Stop Register (CLKSTP)

The CLKSTP register controls the clocks (STOP or HALT specification) inside µPLAT. The STOP mode stops the clocks of the CPU group (CPUG: ARM CPU, ARBITER, INTRC, APB IF, DEFSLV, BIC) (can also stop the clocks of the entire LSI using the stopack signal output), and the HALT mode stops individual clocks of each module.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLKSTP	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKSTP	—*	—*	—*	—*	—*	—*	—*	—*	STOP	—*	—*	—*	—*	CPUG	TIC	SIO
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8000004
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
By giving consideration to future expansion, it is recommended to write "0" when writing, not expecting "0" when reading.
When writing into this register, write an applicable value immediately after writing "0x3C" to prevent erroneous writing.

[Bit Description]

- **SIO** (bit 0)
  This bit specifies to halt (HALT) the clocks to be input to the µPLAT-SIO(UART).

SIO	Description
0	Clock supply
1	Clock halt (HALT)

- **TIC** (bit 1)
  This bit specifies to halt (HALT) the clocks to be input to the TIC. The initial value is "0 (clock supply)." Generally, the TIC is not used. Therefore, by setting this bit to "1" after power on, power saving can be achieved.

TIC	Description
0	Clock supply
1	Clock halt (HALT)

- **CPUG** (bit 2)
  This bit specifies to halt (HALT) the clocks of the CPU group (CPUG: ARM CPU, ARBITER, INTRC, APB IF, DEFSLV, BIC). It is possible to recover from this CPUG HALT mode by an interrupt. When in this mode, 48 MHz oscillation and PLL operation are not stopped. Therefore, there is no need to wait for oscillation and PLL operation to stabilize when recovering.
  After setting the CPUG bit to "1," be sure to set the Wait for Interrupt state by controlling the CP15 register of ARM. Otherwise, the mode will not transit to the CPUG HALT mode. Also, after setting this bit to "1," be sure to perform control using a sequence that controls the CP15 register of ARM.

CPUG	Description
0	Clock supply
1	Clock halt (HALT)

- **STOP** (bit 7)
  This bit sets the STOP mode. It stops the CPUG clocks and sends a stopack signal.
  When in the STOP mode, 48 MHz oscillation and PLL operation are stopped, achieving further low power consumption. However, since all clocks stop, including the WDT clock, CPU runaway cannot be detected by the WDT.
  It is possible to recover from this CPUG HALT mode by an interrupt. When recovering, operation starts after waiting for 48 MHz oscillation and PLL operation to stabilize for the duration of the time set in the clock supply wait time control register.
  However, operation starts at 32 kHz without waiting for 48 MHz oscillation and PLL operation to stabilize if the CPUCLK has been operated at 32 kHz, and the 48 MHz oscillation and PLL operation have been stopped in advance.
  After setting the STOP bit to "1," be sure to set the Wait for Interrupt state by controlling the CP15 register of ARM. Otherwise, the mode will not transit to the STOP mode. Also, after setting this bit to "1," be sure to perform control using a sequence in which the CP15 register of ARM is controlled.

CPUG	Description
0	Clock supply
1	Clock halt (HALT)

7.4.3 Clock (CGB) Control Register 0 (CGBCNT0)

This register controls the output signal cgbcnt0 [7:0] used to control the CGB. This LSI sets the operating frequencies of (PLAT/ARM/AHB BUS.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CGBCNT0	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	HCLK GEAR1	CCLKGEAR[2:0]			HCLK GEAR0	FCLKGEAR[2:0]		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8000008
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
By giving consideration to future expansion, it is recommended to write "0" when writing, not expecting "0" when reading.
When writing into this register, write an applicable value immediately after writing "0x3C" to prevent erroneous writing.
The initial value of this register is 32h0000_0000; however, the actual setting of the CGB is equivalent to HCLKGEAR1="1", and this register value and the clock setting are different after a reset. Therefore, be sure to write 32h'0000_0080 after canceling a reset to match this register setting and the actual operation.
[Bit Description]

- **FCLKGEAR[2:0]** (bits 0 to 2)
  These bits specify the operating clock frequency (CPUCLK) of the ARM946.
  This initial value is 120 MHz.

FCLKGEAR[2]	FCLKGEAR[1]	FCLKGEAR[0]	Description
0	0	0	120 MHz (1/1)
0	0	1	60 MHz (1/2)
0	1	0	30 MHz (1/4)
0	1	1	15 MHz (1/8)
1	0	0	7.5 MHz (1/16)
1	0	1	3.75 MHz (1/32)
1	1	0	Cannot be set.
1	1	1	32 kHz

Set FCLK to 32 kHz after setting the RTCEN bit of the PDMASK register of the configuration register module to "1" and then verifying, using the scratch pad register of the RTC module, that RTC is not left masked. If RTC is masked, no 32 kHz clock is output externally from the RTC module. If the 32 kHz mode is set with RTC left masked, CPU is hung up.

- **HCLKGEAR0, HCLKGEAR1** (bits 3 and 7)
  These bits set the operating clock frequency of AHB BUS. The HCLKGEAR1 bit sets whether or not the frequency can be divided. If the frequency can be divided, the HCLKGEAR0 is used to set the number of frequency divisions. The initial values of this register are "0" for both HCLKGEAR0 and HCLKGEAR1. However, the initial value of the HCLK frequency is CPUCLK/2.

HCLKGEAR1	HCLKGEAR0	Description
0	0	CPUCLK
0	1	CPUCLK
1	0	CPUCLK / 2
1	1	CPUCLK / 4

- **CCLKGEAR[2:0]** (bits 3 to 5)
  These bits set a clock for various peripherals. Generally, set this bit field to 120 MHz. If 32 kHz has been selected as the CPU clock, select 32 kHz. At that time, the frequency of the TIMER clock is set to 2.048 kHz.
  The initial value is 120 MHz.

CCLKGEAR[2]	CCLKGEAR[1]	CCLKGEAR[0]	Description
0	0	0	120 MHz (1/1)
0	0	1	Cannot be set.
0	1	0	Cannot be set.
0	1	1	Cannot be set.
1	0	0	Cannot be set.
1	0	1	Cannot be set.
1	1	0	Cannot be set.
1	1	1	32 kHz

7.4.4 Clock Supply Wait Time Control Register (CKWT)

This register outputs the signal CKWT [3:0] that sets the value of the CLKWT counter (the stabilization wait counter when starting PLL and OSC) in the CGB. This register sets the PLL and OSC stabilization wait value when recovering from the STOP mode of the clock stop register.
When the CPUCLK is set to 32 kHz, be aware that this register does not function to set the PLL and OSC stabilization wait value when recovering after stopping PLL and OSC by controlling the clock (CGB) control register 2 or when the power is turned on. For the oscillation and PLL stabilization wait when recovering after stopping PLL and OSC in 32 KHz mode, stability time waiting is performed by the auto reload timer that uses a 32 KHz clock. PLL and OSC stabilization waiting when the power is turned on is performed by allocating a sufficient reset period until PLL and OSC can be stabilized after the power is turned on.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKWT	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	CKWT1[3:0]				CKWT0[3:0]			
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Address: 0xB800000C
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
By giving consideration to future expansion, it is recommended to write "0" when writing, not expecting "0" when reading.
When writing into this register, write an applicable value immediately after writing "0x3C" to prevent erroneous writing.

[Bit Description]

- **CKWT0[3:0]** (bits 0 to 3)
  These bits specify the oscillation and PLL stabilization wait time when recovering from the clock STOP mode. The initial value is "1111", which is 87.4 ms.

CKWT0[3:0]	Wait Cycle	Wait time [s] at OCS = 48 MHz
0000	1	0.0000000208
0001	256	0.0000053333
0010	512	0.0000106667
0011	1024	0.0000213333
0100	2048	0.0000426667
0101	4096	0.0000853333
0110	8129	0.0001693542
0111	16385	0.0003413542
1000	32786	0.0006830417
1001	65536	0.0013653333
1010	131072	0.0027306667
1011	262144	0.0054613333
1100	524288	0.0109226667
1101	1048576	0.0218453333
1110	2097152	0.0436906667
1111	4194304	0.0873813333

- **CKWT1[3:0]** (bits 4 to 7)
  These bits are not used. Always set this bit field to "0000".

7.4.5 Remap Control Register (RMPCON)

This register controls the remapping of AHB and external memory (ROM/DRAM/SRAM) for bank0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RMPCON	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	RMPM[3:0]			
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8000010
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
By giving consideration to future expansion, it is recommended to write "0" when writing, not expecting "0" when reading.
When writing into this register, write an applicable value immediately after writing "0x3C" to prevent erroneous writing.

[Bit Description]

- **RMPM[3:0]** (bits 0 to 3)
  These bits can be set as shown below in combination with the BMODE [1:0] pin.

RMPM[3:0]				BOOT[1:0]		BANK0	BANK9	BANK10	BANK25	BANK26	BANK24
3	2	1	0	1	0						
0	x	x	x	0	0	MCP FLASH+ External ROM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
				0	1	External ROM	AHBROM	AHBRAM	External ROM	External SRAM	External SDRAM
				1	0	AHBROM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
				1	1	AHBROM	AHBROM	AHBRAM	External ROM	External SRAM	External SDRAM
1	0	0	0	x	0	External SRAM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
					1				External ROM		
1	0	0	1		0	External SDRAM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
					1				External ROM		
1	0	1	0		0	AHBRAM	AHBROM	AHBRAM	MCP FLASH + External ROM	External SRAM	External SDRAM
					1				External ROM		
1	0	1	1		x	Cannot be used.					
1	1	x	x								

[Note]

- In the sections notated as MCP FLASH + external ROM, MCP FLASH is accessed for the first 512K bytes (4M bits) of addresses, and the external ROM area is accessed for the subsequent addresses. Therefore, the address area of the first 512K bytes in external ROM cannot be used.
- Also, because the access timing via the external bus controller is common to MCP FLASH and external ROM, and thus cannot be set independently. Set it to match the slower one.
- If AHBRAM is remapped, it can also be accessed from the source bank. In this case, the values cannot be guaranteed if access is made by setting one bank with cache and the other without cache.

7.4.6 Clock (CGB) Control Register 1 (CGBCNT1)

This register is not used. Do not access this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CGBCNT1	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8000018
Access:
Access size: 32 bits

[Bit Description]

- This register is not used. Do not access this register.

7.4.7 Clock (CGB) Control Register 2 (CGBCNT2)

This register stops 48 MHz/PLL when the CPUCLK is set to 32 kHz mode by the CGBCNT0 register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CGBCNT2	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	STOP
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB800001C
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
By giving consideration to future expansion, it is recommended to write "0" when writing, not expecting "0" when reading.
When writing into this register, write an applicable value immediately after writing "0x3C" to prevent erroneous writing.

[Bit Description]

- **STOP** (bit 0)
  This bit stops 48 MHz/PLL. This bit can be used only if the CPUCLK is set to 32 kHz mode. If this bit is set to "1" in mode other than 32 kHz mode, this LSI will hang up. After setting this bit to "1", do not reset it to "0" for at least 50 clock periods. After setting this bit to "0", it is necessary to wait for 48 MHz/PLL to stabilize.

  The auto reload timer can be used for the stabilization wait time. In 32 kHz mode, the CCLK frequency of the auto reload timer is 32 kHz/16 = 2.048 kHz.

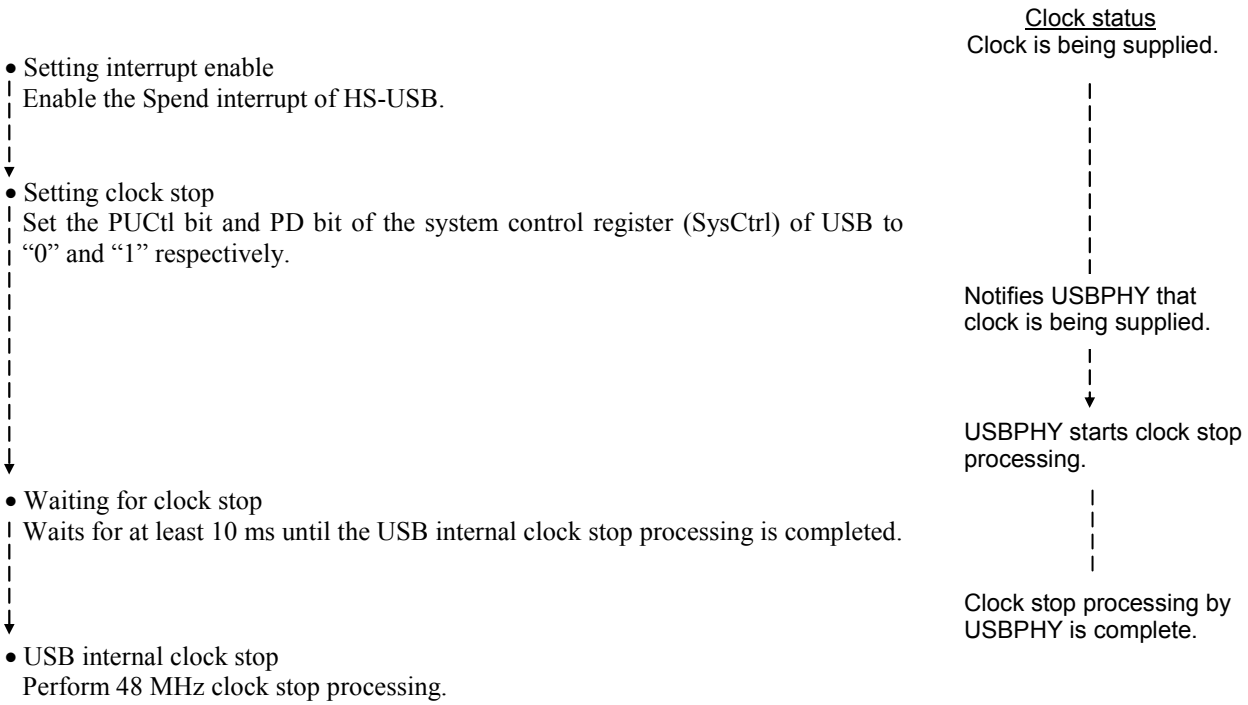
STOP	Description
0	Operates 48 MHz/PLL
1	Stops 48 MHz/PLL.

7.5 Procedures, Cautions, and Restrictions on the Use of Power Management

7.5.1 On Power Management by Stopping the 48 MHz Clock

The following restriction applies when stopping the clock by setting the STOP bit of the CLKSTP register to "1" or when stopping the 48 MHz oscillation clock after switching the CPUCLK to the 32 kHz mode (See Section 26.5.1 for details):

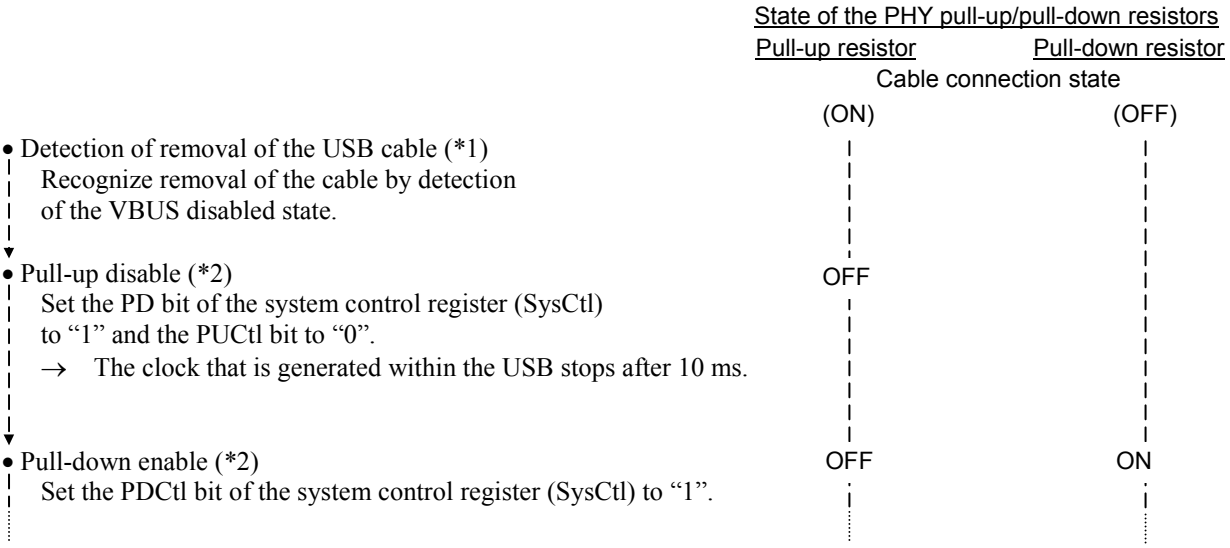
It is required to suspend the HS-USB so as to stop the HS-USB internal clock before stopping the 48 MHz clock. Following shows the procedure for stopping the USB internal clock.



7.5.2 Power Management by Stopping USB Clock While USB Is Idle

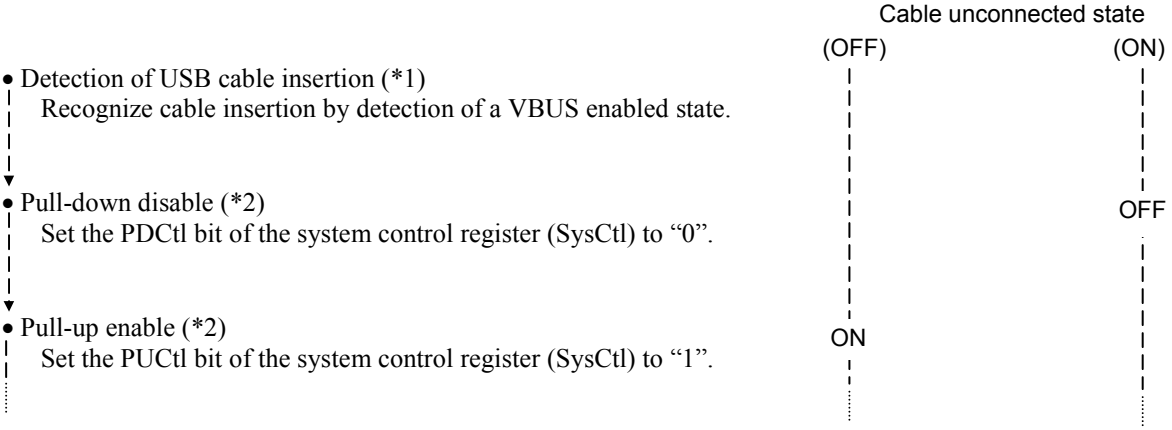
When the USB is not used, the clock that is generated within the USB can be stopped. The penetration current of the USB input section that is generated when the USB cable is not connected can be prevented through pull-down control of the USB pin.

The procedure is shown below.



As a result, the current that is supplied when the USB is not used can be reduced. Since the PD bit of the system control register (SysCtl) is a mode setting bit, in the USB idle state, the clock can be stopped by setting the bit only once after resetting. When writing other bits of the system control register, write PD = "1".

The operation of the USB can be restarted in the following procedure.



(*1): Removal of the USB cable: Recognize insertion or removal of the USB cable based on whether VBUS is enabled or disabled. When VBUS is in an enabled state, recognize this as cable insertion and when VBUS is in a disabled state, recognize this as cable removal.

(*2): The PCUtl bit and PDCtl bit are assigned in the same system control register (SysCtl), however, each bit must be set individually.

For instance, to execute pull-up disable → pull-down enable of the sequence indicated above, set the relevant bit of the PUCtl bit to "0", write to the system control register (SysCtl), then set the relevant bit of the PDCtl bit to "1", and write to the system control register (SysCtl), instead of setting the relevant bit of the PUCtl bit to "0"' and the relevant bit of the PDCtl bit to "1", and writing them to the system control register (SysCtl).

7.5.3 SDRAM Power Management

7.5.3.1 Power Down due to Stopping of the XSDCLK when SDRAM Is Not Accessed

When SDRAM is not frequently accessed, the XSDCLK signal can be stopped to reduce the power consumption of this LSI and SDRAM.

1. Set the DRAM power down control register (PDWC). *
2. Set the PDWN bit of the DRAM control register (DRMC) to "1".

When the number of accesses to SDRAM that resulted idle exceeds the specified number, XSDCLK is stopped by the setting above, reducing the power that is consumed. However, when SDRAM is accessed after XSDCLK is stopped, the stopping of XSDCLK under an unnecessarily shorter cycle causes performance deterioration since it takes 3 clock cycles to restore SDRAM operation. Therefore, the value that is set in the DRAM Power Down control register must be determined through experiments.
(The recommended value for this register in the SDK is 0x0000_0002 [changed to power down mode when the SDRAM accesses are idle for 3 cycles or more.])

7.5.3.2 Power Down due to Transition to the Self-Refresh State

To stop the clocks in the DRAM control module using power management, it is necessary to set or cancel the self-refresh state by software as hardware does not support automatic transition to the self-refresh state.
The self-refresh state is set by a self-refresh start request by the DCMD register, and cancelled by a self-refresh cancel request by the DCMD register. Moreover, if SDRAM is connected, the output from XSDCLK is stopped when in the Self-Refresh mode. If access is attempted to a DRAM space while self-refreshing, an abort exception will occur.

1) Procedure to Stop Clock Supply/Transit to the Standby Mode When DRAM Is Used
   1. Write "110" into the DRCMD bit of the DCMD register, and transit to the self-refresh state.
   2. DRAMC does not execute distributed CBR refresh until the self-refresh state is cancelled, and stops the output of the XSDCLK clock of SDRAM.
   3. Stops clock supply by software, or transits to the STANDBY mode.

2) Procedure to Recover from Clock STOP/STANDBY Mode
   1. Resume clock supply by software or an external source.
   2. Write "111" into the DRCMD bit of the DCMD register, and cancel the self-refresh state.
   3. The output of the XSDCLK clock of SDRAM is resumed.

7.5.4 Power Management by Stopping the Clock of an Unused Module

When the modules that are described in the clock stop mode section in functional block units in Section 7.2.2.3 are not used, the clock supply can be stopped. When these modules are not used, the power consumption can be reduced by stopping the clock for these modules.

7.5.5 Power Management by Stopping CPU Operation

Power consumption can be reduced by stopping the CPU. If the oscillator circuit is stopped by using the STOP bit in the clock stop register of the SYSCON module, a wait time for oscillator stabilizing time (ten milliseconds) is required. Therefore, the CPUG HALT mode in which the oscillator circuit is not stopped should be used if the oscillator stabilizing time cannot be obtained.

In this case, although the CPU clock is stopped in CPUG HALT mode, the BUS clock is operating. Therefore, before changing the mode to the CPUG HALT mode, access clock (CGB) control register 0 to reduce the clock frequency as much as possible, and then change the mode to the CPUG HALT mode to reduce the power consumption.
Refer to the description of the clock stop register of the SYSCON module for the method of changing the mode to the CPUG HALT mode.

Note: When the HALT mode is set after data is written to FIFO of I2S and transmission/reception is performed in the HALT mode, the following specification restrictions apply.

Modes that cannot be used when HCLK = 0.398 MHz/PCLK = 0.469 MHz

Sampling rate		48 kHz	44.1 kHz	33 kHz	24 kHz	22.05 kHz	16.5 kHz
Transmission	Without dly	×	○	○	○	○	○
	With dly	○	○	○	○	○	○
Reception	Without dly	×	×	×	○	○	○
	With dly	×	×	○	○	○	○

* The combinations of the sampling rates and modes indicated by "Xs" in the table above cannot be used when HCLK is 0.938 MHz and PCLK is 0.469 MHz.
  Should be used with HCLK = 1.875 MHz/PCLK = 0.938 MHz.
* HCLK should be 1.875 MHz or higher and PCLK should be 0.938 MHz or higher to use the combinations of the sampling rates and modes indicated by "Xs" in the table above.
* The table above is not applicable when CPU clock is 32 kHz.

Refer to the configuration register/I2S register section for detailed settings such as change of the sampling rate and with or without dly.

Chapter 8

Interrupt Controller

Chapter 8 Interrupt Controller

8.1 Overview

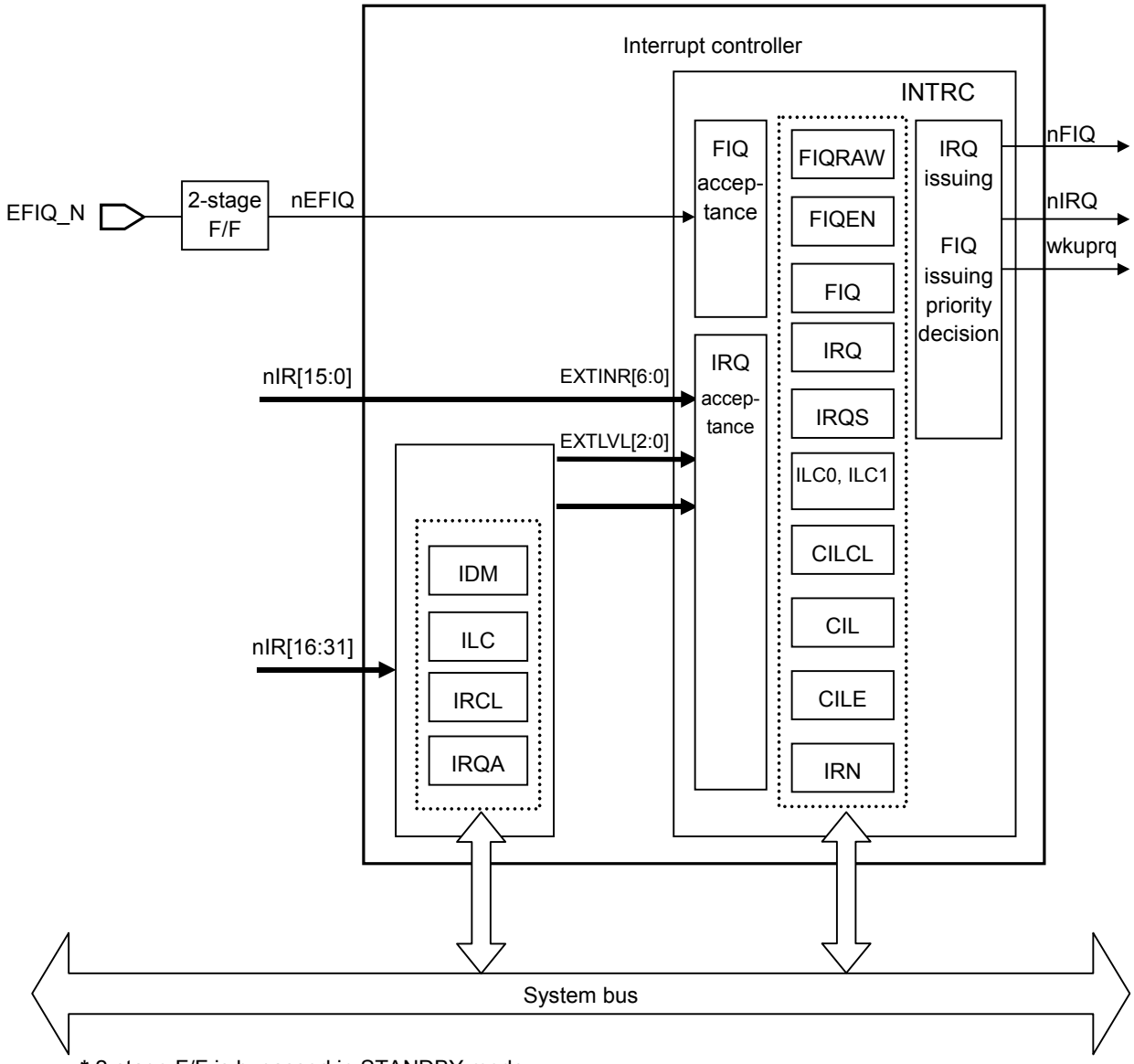
The interrupt controller (INTRC) controls the masks and priority sequences for fast interrupt (FIQ) inputs and interrupt (IRQ) inputs, and outputs FIQ exception requests and IRQ exception requests.
An edge interrupt mode can be set through the setting of the interrupt controller, however, all the sources in level interrupt mode should be used.

Features:

- FIQ: 1 source (external)
- IRQ: 27 sources (internal: 23, external: 4)
- Interrupt mask for IRQ interrupt
- Enables the setting of an interrupt priroity level for each IRQ interrupt source
- Function for prohibiting acceptance of interrupts with interrupt levels lower than the level currently being processed for IRQ interrupts
- Applies a level sense for the interrupt sense mode
- Generates a clock stop release request at clock stop

8.1.1 Configuration

Figure 8-1 shows the configuration of the interrupt controller.



* 2-stage F/F is bypassed in STANDBY mode.

Figure 8-1 Configuration of the Interrupt Controller

8.1.2 List of Pins

Pin	I/O	Function
PIOE15	I	Interrupt handling PIO pin
PIOE14	I	Interrupt handling PIO pin
PIOE13	I	Interrupt handling PIO pin
PIOE12	I	Interrupt handling PIO pin
EFIQ_N	I	FIQ input (negative logic)

8.1.3 List of Registers

Address	Register name	Symbol	R/W	Size	Initial value
0x7800_0000	IRQ register	IRQ	R	32	0x0000_0000
0x7800_0004	IRQ software register	IRQS	R/W	32	0x0000_0000
0x7800_0008	FIQ register	FIQ	R	32	0x0000_0000
0x7800_000C	FIQRAW status register	FIQRAW	R	32	Depends on FIQ_N interrupt input
0x7800_0010	FIQ enable register	FIQEN	R/W	32	0x0000_0000
0x7800_0014	IRQ number register	IRN	R	32	0x0000_0000
0x7800_0018	Current IRQ level register	CIL	R/W	32	0x0000_0000
0x7800_001C	(Reserved)				
0x7800_0020	IRQ level control register 0	ILC0	R/W	32	0x0000_0000
0x7800_0024	IRQ level control register 1	ILC1	R/W	32	0x0000_0000
0x7800_0028	Current IRQ level clear register	CILCL	W	32	—
0x7800_002C	Current IRQ level control register	CILE	R	32	0x0000_0000
0x7BF0_0000	(Reserved)				
0x7BF0_0004	IRQ clear register	IRCL	W	32	—
0x7BF0_0010	IRQA register	IRQA	R/W	32	0x0000_0000
0x7BF0_0014	IRQ detect mode setting register	IDM	R/W	32	0x0000_0000
0x7BF0_0018	IRQ level control register	ILC	R/W	32	0x0000_0000

8.2 Interrupt Sources

The following table lists the interrupts.

Interrupt number	IRQ group	Interrupt source	Comment
nfiq	—	FIQ_N (EFIQ_N)	
nir0	ILR0	μPLAT System-Timer	μPLAT Timer
nir1	ILR1	Synchronous SIO0	Serial I/F ①
nir2		Synchronous SIO1	
nir3		Synchronous SIO2	
nir4	ILR4	WDT	WDT/TIMER ①
nir5		RTC	
nir6	ILR6	PIOE12	GPIO
nir7		—	
nir8	ILR8	Software interrupt (IRQS)	Software
nir9	ILR9	PWM	TIMER ②
nir10	ILR10	μPLAT Serial I/F (UART)	μPALT SIO
nir11	ILR11	I^2C	Serial ②
nir12	ILR12	ADC	ADC/DAC
nir13	ILR13	NAND FLASH	Others ①
nir14	ILR14	IDE controller	
nir15	ILR15	Not used	Others/External ①
nir16	ILR16	TIMER0	TIMER ③
nir17		TIMER1	
nir18	ILR18	TIMER2	
nir19		Not used	
nir20	ILR20	I^2S-trans	Others ②
nir21		I^2S-receive	
nir22	ILR22	DMAC0	DMAC
nir23		DMAC1	
nir24	ILR24	DMAC2	
nir25		DMAC3	
nir26	ILR26	PIOE15(USB-VBUS)	USB
nir27		USB	
nir28	ILR28	Not used	Others ②
nir29		PIOE14	
nir30	ILR30	Not used	
nir31		PIOE13	

*1 Set a level detect mode for the interrupt mode of the extended interrupt of nir16 to nir31.
*2 At return from a clock stop mode, set a level mode for the interrupt mode in PIO for the interrupt from PIO. If an edge mode is set, an interrupt does not occur since the PIO clock is stopped.

8.3 Interrupt Levels

An interrupt level can be set for each interrupt source. The higher the interrupt level setting value, the higher is the interrupt priority. When the interrupt level setting value is 0, the interrupt is masked.

Interrupt levels and the priorities

Interrupt level setting value	Priority
7	High
↕	↕
	Low
0	Interrupt mask

8.4 Description of Registers

8.4.1 IRQ Register (IRQ)

The IRQ register indicates the presence or absence of the interrupt request that is not masked to an IRQ interrupt request signal nIR[15:0].
The bit number corresponds to the interrupt number.
The IRQ register can only be read by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78000000
Access: R
Access size: 32 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation, however, it is recommended that the program do not assume "0" (don't care).

[Bit Description]

- **IRQ[15:0]** (bits 0 to 5)
  These bits indicate the presence or the absence of the interrupt request that is not masked.

IRQ[15:0]	Description]
0	No interrupt request exists
1	An Interrupt request exists

The following table shows the relationship among IRQ input signal nIR[N], the interrupt level setting value corresponding to interrupt number N, and bit N of the IRQ register when N is assumed as the interrupt number.

IRQ input signal: nIR[N]	Interrupt level setting value	IRQ register: IRQ[N]
X	0	0
1	1 or higher value	0
0	1 or higher value	1

[Note]
Set the interrupt levels in registers ILC0 to ILC1.
IRQ input signal: nIR[N] is a negative logic.

8.4.2 IRQ Software Register (IRQS)

The IRQS register generates a software IRQ interrupt request signal by writing "1" to the IRQS[1] bit.
This register performs the function equivalent to the function performed when "0" is input in interrupt request signal nIR[8]. That is, this register generates an IRQ interrupt of the level that is set in the ILC1 register. Writing "0" to the IRQS[1] bit cancels the interrupt request. The IRQS register can be read/written by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQS	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	IRQS	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78000004
Access: R/W
Access size: 32 bits

[Note]
*: For the bit data that is indicated by "—"
- Although "0" is returned at read operation, it is recommended that the program do not assume "0" (don't care).
- Write "0" at write operation.

[Bit Description]
- **IRQS** (bit 1)
  This bit is used for generating/canceling a software interrupt request signal. When "1" is written to this bit, an interrupt request of nir8 is generated.

IRQS	Description
0	Cancels a software interrupt request.
1	Generates a software interrupt request.

8.4.3 FIQ Register (FIQ)

The FIQ register indicates whether a FIQ exception request has been issued.
The FIQ register can only be read by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIQ	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	FIQ
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78000008
Access: R
Access size: 32 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation, however, it is recommended that the program do not assume "0" (don't care).

[Bit Description]

- **FIQ** (bit 9)
  This bit indicates whether a FIQ exception request has been issued.

FIQ	Description
0	FIQ exception request not issued
1	FIQ exception request issued

The following table shows the status of FIQ (bit 0) of the FIQ register according to the statuses of FIQRAW (bit 0) of the FIQRAW register and FIQEN (bit 0) of the FIQEN register.

FIQRAW register [0]: FIQ	FIQEN register [0]: FIQEN	FIQ register [0]: FIQ
X	0	0
0	X	0
1	1	1

8.4.4 FIQRAW Register (FIQRAW)

The FIQRAW register indicates the status of the FIQ interrupt input signal.
The status of this register is not affected by FIQEN (bit 0) of the FIQEN register.
The FIQRAW register can only be read by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIQRAW	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	FIQRAW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1

Address: 0x7800000C
Access: R
Access size: 32 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation, however, it is recommended that the program do not assume "0" (don't care).
*1 The initial value depends on the status of the EFIQ_N pin.

[Bit Description]

- **FIQRAW** (bit 0)
  This bit indicates the status of the FIQ interrupt input signal.

FIQRAW	Description
0	FIQ interrupt request not issued
1	FIQ interrupt request issued

8.4.5 FIQ Enable Register (FIQEN)

The FIQEN register enables/masks an FIQ interrupt request.
The FIQEN register can be read/written by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIQEN	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	FIQEN
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78000010
Access: R/W
Access size: 32 bits

[Note]
*: For the bit data that is indicated by "—"
- Although "0" is returned at read operation, it is recommended that the program do not assume "0" (don't care).
- Write "0" at write operation.

[Bit Description]
- **FIQEN** (bit 0)
  This bit controls to enable/mask an FIQ interrupt input signal.

FIQEN	Description
0	Masks an FIQ interrupt request.
1	Enables an FIQ interrupt request.

8.4.6 IRQ Number Register (IRN)

The IRN register indicates a source number of the highest priority interrupt at IRQ request output.
When the IRN register is read, "1" is set in the bit position of the CIL register corresponding to the interrupt level of this interrupt. This setting allows the interrupt priority decision circuit to be operated and the notification of the interrupts lower than this level to be held, so the IRN register is set to "0".
When an interrupt with a higher priority occurs, the interrupt number is set in the IRN register and an IRQ exception request is output to the CPU.
The IRN register can only be read by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRN	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	IRN[6:0]						
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78000014
Access: R
Access size: 32 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation, however, it is recommended that the program do not assume "0" (don't care).

[Bit Description]

- **IRN[6:0]** (bits 0 to 6)
  These bits indicate an interrupt source number of the highest priority. See "8.2 Interrupt Sources" for interrupt source numbers and the corresponding interrupt sources.

8.4.7 Current IRQ Level Register (CIL)

The CIL register reads an IRN register and indicates the level of the interrupt whose interrupt number was posted to the CPU.
The CIL register can be read/written by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIL	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	CIL[7:1]							—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78000018
Access: R/W
Access size: 32 bits

[Note]
*: For the bit data that is indicated by "—"
- Although "0" is returned at read operation, it is recommended that the program do not assume "0" (don't care).
- Write "0" at write operation.

[Bit Description]
- **CIL[7:1]** (bits 1 to 7)
  These bits indicate that the interrupt processing of the level corresponding to the bit position where "1" is set is being executed.

CIL[7]	CIL[6]	CIL[5]	CIL[4]	CIL[3]	CIL[2]	CIL[1]
Interrupt level 7	Interrupt level 6	Interrupt level 5	Interrupt level 4	Interrupt level 3	Interrupt level 2	Interrupt level 1

When "1" is set in any of the bits CIL[7:1], notification of the interrupts of the levels lower than the interrupt level indicated by the position of the highest "1" bit is held.

Setting condition: When the IRN register is read, "1" is set in the bit position of the CIL register corresponding to the interrupt level of the interrupt.
Clearing condition: When "1" is written to CIL[N], CIL[N] = "0" is set.
When "1" is written to the CILCL register, only the highest "1" bit of the CIL register is cleared.

At termination of interrupt processing, clear the highest "1" bit of the CIL register by executing only once the write operation to the CIL register or CILCL register.

8.4.8 IRQ Level Control Register 0 (ILC0)

The ILC0 register is used to set an interrupt level of each source of IRQ interrupt nIR[7:0].
The ILC0 register can be read/written by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ILC0	—*	—*	—*	—*	—*		ILR6		—*	—*	—*	—*	—*		ILR4	
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ILC0	—*	—*	—*	—*	—*	—*	—*	—*	—*		ILR1		—*		ILR0	
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78000020
Access: R/W
Access size: 32 bits

[Note]
*: For the bit data that is indicated by "—"
- Although "0" is returned at read operation, it is recommended that the program do not assume "0" (don't care).
- Write "0" at write operation.

[Bit Description]
- **ILR0** (bits 0 to 2)
  ILR1 (bits 4 to 6)
  ILR4 (bits 16 to 18)
  ILR6 (bits 24 to 26)
  These bits set an interrupt level of each source of IRQ interrupt nIR[7:0].

ILR**			Interrupt level
2	1	0	
1	1	1	7 (High)
1	1	0	↑
1	0	1	↕
1	0	0	↕
0	1	1	↕
0	1	0	↓
0	0	1	1 (Low)
0	0	0	Interrupt mask

The following table shows the correspondence between interrupt sources and interrupt levels.

nIR[0]	ILR0	ILC0[2:0]
nIR[1]	ILR1	ILC0[6:4]
nIR[2]		
nIR[3]		
nIR[4]	ILR4	ILC0[18:16]
nIR[5]		
nIR[6]	ILR6	ILC0[26:24]
nIR[7]		

8.4.9 IRQ Level Control Register 1 (ILC1)

The ILC1 register is used to set an interrupt level of each source of IRQ interrupt nIR[15:8].
The ILC1 register can be read/written by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ILC1	—*		ILR15		—*		ILR14		—*		ILR13		—*		ILR12	
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*		ILR11		—*		ILR10		—*		ILR9		—*		ILR8	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78000024
Access: R/W
Access size: 32 bits

[Note]
*: For the bit data that is indicated by "—"
- Although "0" is returned at read operation, it is recommended that the program do not assume "0" (don't care).
- Write "0" at write operation.

[Bit Description]
- **ILR8** (bits 0 to 2)
  ILR9 (bits 4 to 6)
  ILR10 (bits 8 to 10)
  ILR11 (bits 12 to 14)
  ILR12 (bits 16 to 18)
  ILR13 (bits 20 to 22)
  ILR14 (bits 24 to 26)
  ILR15 (bits 28 to 30) (ILR15: Unused in this LSI. Must be always set to "0".)
  These bits set an interrupt level of each source of IRQ interrupt nIR[15:8].

ILR8 to 15			Interrupt level
2	1	0	
1	1	1	7 (High)
1	1	0	↑
1	0	1	
1	0	0	
0	1	1	
0	1	0	↓
0	0	1	1 (Low)
0	0	0	Interrupt mask

The following table shows the correspondence between interrupt sources and interrupt levels.

nIR[8]	ILR8	ILC1[2:0]
nIR[9]	ILR9	ILC1[6:4]
nIR[10]	ILR10	ILC1[10:8]
nIR[11]	ILR11	ILC1[14:12]
nIR[12]	ILR12	ILC1[18:16]
nIR[13]	ILR13	ILC1[22:20]
nIR[14]	ILR14	ILC1[26:24]
nIR[15]	ILR15	ILC1[30:28]

8.4.10 Current IRQ Level Clear Register (CILCL)

When a write operation is executed for the CILCL register, only the bit where "1" is set on the MSB side of the CIL register is cleared. Any value can be written.
Consequently, the interrupt notification of the interrupt level that is currently being processed is released from the hold state.
The CILCL register can only be written by program.

Address: 0x78000028
Access: W
Access size: 32 bits

[Note]
At termination of interrupt processing, clear only the bit in which "1" is set on the MSB side of the CIL register by executing only once write operation for either this register or the CIL register.

8.4.11 Current IRQ Level Encode Register (CILE)

The CILE register indicates, in binary mode, the interrupt level that is indicated by the bit position in which "1" is set on the MSB side of the CIL register.
The CILE register can only be read by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CILE	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	CILE[2:0]		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7800002C
Access: R
Access size: 32bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation, however, it is recommended that the program do not assume "0" (don't care).

[Bit Description]

- **CILE[2:0]** (bits 0 to 2)
  The CILE register indicates, in binary mode, the interrupt level that is indicated by the bit position in which "1" is set on the MSB side of the CIL register.

CIL[7:1]	CILE[2:0]	Interrupt level
0000000	000	Not being interrupted
0000001	001	1
000001X	010	2
00001XX	011	3
0001XXX	100	4
001XXXX	101	5
01XXXXX	110	6
1XXXXXX	111	7

8.4.12 IRQ Clear Register (IRCL)

The IRCL register clears the interrupt request bit when the interrupt detect mode of the specified interrupt number is an edge detect mode.
When the interrupt detect mode of the specified interrupt number is a level detect mode, no action is taken.
The IRCL register can only be written by program.
This register must not be accessed since this LSI external interrupt pin handles a level detect mode only.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRCL	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	IRCL[6:0]						

Address: 0x7BF00004
Access: W
Access size: 32bits

[Note]
*: For the bit data that is indicated by "—", write "0" at write operation.

[Bit Description]

- **IRCL[6:0]** (bits 0 to 6)
  When the interrupt detect mode is an edge detect mode, the interrupt request bit related to the interrupt number that was specified in the write data is cleared.

8.4.13 IRQA Register (IRQA)

The IRQA register indicates the presence or absence of a valid interrupt request when it is read. When the interrupt detect mode is an edge detect mode, the register clears the interrupt request bit at write operation. Write operation for the bit whose interrupt detect mode is set to a level detect mode does not have any effect. Do not access this register since this LSI external interrupt pin handles a level detect mode only.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQA	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ[31:16]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BF00010
Access: R/W
Access size: 32bits

[Note]
*: For the bit data that is indicated by "—"
- Although "0" is returned at read operation, it is recommended that the program do not assume "0" (don't care).
- Write "0" at write operation.

[Bit Description]
- **IRQ[31:16]** (bits 0 to 15)
  At read: Indicates the presence or absence of a valid interrupt request.

IRQ[31:16]	Description
0	Interrupt request not issued
1	Interrupt request issued

At write: Clears the interrupt request detect bit when the interrupt detect mode is an edge detect mode. In level detect mode, writing to this register does not have any effect (No effect on the IRCL register either).

IRQ[31:16]	Description
0	No action
1	Clears the interrupt request bit.

IRQ[31:16] bit setting/clearing condition:

- When the interrupt detect mode is a level detect mode (each bit is independent)

Interrupt request signal	IRQ level setting value	IRQ register
"H"	—	"0"
"L"	0	"0"
"L"	7 to 1	"1"

[Note]
The IRQ internal register cannot be accessed externally and indicates the presence or absence of an interrupt request signal regardless of any IRQ level setting value.

When the interrupt detect mode is a level detect mode (each bit is independent)

Event	IRQ register
Write "1" in the IRQ register	"0"
Write the applicable interrupt number in the IRCL register	"0"
Interrupt request signal "H"→"L"	"1"

IRQ level setting value	IRQ register
—	"0"
0	"0"
7 to 1	"1"

8.4.14 IRQ Detect Mode Setting Register (IDM)

The IDM register sets an IRQ detect mode and the polarity of the interrupt.
A level detect mode and an edge detect mode are available as the detect modes.
The IDM register can be read/written by program.
Always use this LSI with the IDM register set to "0" since it handles a level mode only.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDM	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	IDM30	—*	IDM28	—*	IDM26	—*	IDM24	—*	IDM22	—*	IDM20	—*	IDM18	—*	IDM16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BF00014
Access: R/W
Access size: 32 bits

[Notes]
1. Always use this LSI with the IDM register set to "0" since it handles a level mode only.
*. Reserved bit for the future extension. In this LSI, "0" is read at read operation and "0" is written at write operation.

[Bit Description]
- **IDM16** (bit 0)
  This bit sets the interrupt detect mode of IRQ17 and IRQ16.
  When this bit is set to "0", the mode is set to a level detect mode and when the bit is set to "1", an edge detect mode is set. Use in level detect mode.
- **IDM18** (bit 2)
  This bit sets the interrupt detect mode of IRQ19 and IRQ18.
  When this bit is set to "0", the mode is set to a level detect mode and when the bit is set to "1", an edge detect mode is set. Use in level detect mode.
- **IDM20** (bit 4)
  This bit sets the interrupt detect mode of IRQ21 and IRQ20.
  When this bit is set to "0", the mode is set to a level detect mode and when the bit is set to "1", an edge detect mode is set. Use in level detect mode.
- **IDM22** (bit 6)
  This bit sets the interrupt detect mode of IRQ23 and IRQ22.
  When this bit is set to "0", the mode is set to a level detect mode and when the bit is set to "1", an edge detect mode is set. Use in level detect mode.
- **IDM24** (bit 8)
  This bit sets the interrupt detect mode of IRQ25 and IRQ24.
  When this bit is set to "0", the mode is set to a level detect mode and when the bit is set to "1", an edge detect mode is set. Use in level detect mode.
- **IDM26** (bit 10)
  This bit sets the interrupt detect mode of IRQ27 and IRQ26.
  When this bit is set to "0", the mode is set to a level detect mode and when the bit is set to "1", an edge detect mode is set. Use in level detect mode.
- **IDM28** (bit 12)
  This bit sets the interrupt detect mode of IRQ29 and IRQ28.
  When this bit is set to "0", the mode is set to a level detect mode and when the bit is set to "1", an edge detect mode is set. Use in level detect mode.

- **IDM30** (bit 14)
  This bit sets the interrupt detect mode of IRQ31 and IRQ30.
  When this bit is set to "0", the mode is set to a level detect mode and when the bit is set to "1", an edge detect mode is set. Use in level detect mode.

8.4.15 IRQ Level Control Register (ILC)

The ILC register sets an IRQ interrupt priority level.
The ILC register can be read/written by program.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ILC	—*		ILC30		—*		ILC28		—*		ILC26		—*		ILC24	
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*		ILC22		—*		ILC20		—*		ILC18		—*		ILC16	
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BF00018
Access: R/W
Access size: 32 bits

*: For the bit data that is indicated by "—"
- Although "0" is returned at read operation, it is recommended that the program do not assume "0" (don't care).
- Write "0" at write operation.

[Bit Description]

- **ILC16** (bits 0 to 2)
  ILC18 (bits 4 to 6)
  ILC20 (bits 8 to 10)
  ILC22 (bits 12 to 14)
  ILC24 (bits 16 to 18)
  ILC26 (bits 20 to 22)
  ILC28 (bits 24 to 26)
  ILC30 (bits 28 to 30)
  These bits set an interrupt priority level.

ILC**			Interrupt priority level
2	1	0	
1	1	1	7 (priority: High)
1	1	0	
1	0	1	
1	0	0	
0	1	1	
0	1	0	
0	0	1	1 (priority: Low)
0	0	0	Interrupt mask

Set an interrupt detect mode for two sources each.

Interrupt source	Interrupt priority level
IRQ16	ILC16
IRQ17	
IRQ18	ILC18
IRQ19	
IRQ20	ILC20
IRQ21	
IRQ22	ILC22
IRQ23	

Interrupt source	Interrupt priority level
IRQ24	ILC24
IRQ25	
IRQ26	ILC26
IRQ27	
IRQ28	ILC28
IRQ29	
IRQ30	ILC30
IRQ31	

8.4.16 Correspondence between Interrupt Sources and Registers that Are Set

Interrupt source number	Interrupt source	Interrupt level	
		Register	Bit
nFIQ	—	—	—
nIR0	μPLAT System-Timer	ILC0	ILR0
nIR1	SSIO0		ILR1
nIR2	SSIO1		
nIR3	SSIO2		
nIR4	WDT	ILC0	ILR4
nIR5	RTC		
nIR6	PIOE12		ILR6
nIR7	Not used		
nIR8	Software interrupt (IRQS)	ILC1	ILR8
nIR9	PWM		ILR9
nIR10	μPLAT Serial I/F (UART)		ILR10
nIR11	I^2C		ILR11
nIR12	ADC		ILR12
nIR13	NAND FLASH		ILR13
nIR14	IDE controller		ILR14
nIR15	Not used		ILR15
nIR16	TIMER0	ILC	ILC16
nIR17	TIMER1		
nIR18	TIMER2		ILC18
nIR19	Not used		
nIR20	I^2S-trans		ILC20
nIR21	I^2S-recieve		
nIR22	DMAC0		ILC22
nIR23	DMAC1		
nIR24	DMAC2		ILC24
nIR25	DMAC3		
nIR26	PIOE15 (USB-VBUS)		ILC26
nIR27	USB		
nIR28	Not used		ILC28
nIR29	PIOE14		
nIR30	Not used		ILC30
nIR31	PIOE13		

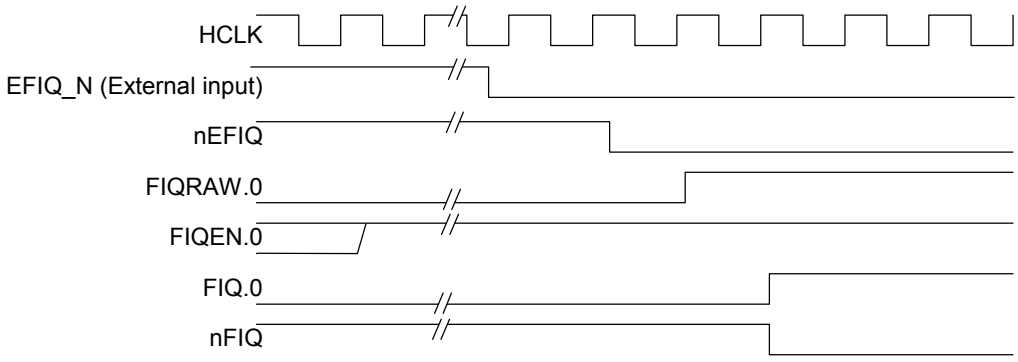
8.5 Description of Operation

8.5.1 External FIQ Interrupt

The FIQ interrupt is used for a high-priority interrupt that is assigned to a time-critical interrupt request. When an FIQ interrupt request occurs and it is not masked by the FIQEN register, FIQ exception processing is requested (nFIQ) to the CPU.
The FIQ interrupt is connected to an interrupt output pin of the AUDIO CODEC chip.
In this LSI, this FIQ interrupt is named EFIQ_N.

8.5.1.1 FIQ External Interrupt Sequence

1. Canceling a FIQ interrupt mask
   An FIQ interrupt mask can be canceled by clearing the SPSR register F flag (bit 6) of the CPU to "0" and setting the FIQEN register bit 0 to "1".
2. Occurrence of an interrupt
   When an FIQ interrupt request signal is input to the external input EFIQ_N of this LSI, bit 0 of the FIQRAW register is set to "1".
3. Requesting exception to the CPU
   When bit 0 of the FIQ register is set, the nFIQ signal is notified at the same time to request an FIQ exception to the CPU.



4. Acceptance of exceptions by the CPU (hardware)
   If the CPU is in the exception acceptable state (not masked by the F flag of the CPSR), it proceeds to IRQ exception processing mode.
   The CPU saves the address of the next instruction in r14_irq, and saves the value of the CPSR in SPSR_fiq.
   The CPU sets "1" in bit 6 (F flag) and bit 7 (I flag) of the CPSR, disabling the acceptance of FIQ and IRQ exceptions.
5. Interrupt processing
   The FIQ handler must cancel an FIQ interrupt request to the request source of the FIQ interrupt before termination of the interrupt processing.
6. Returning from an interrupt
   By executing the return instruction from the interrupt at the end of the FIQ handler, return the instruction address and the CPSR value that have been saved in r14_fiq and SPSR_fiq respectively.

[Note] Refer to the DataSheet of ARM946E for details of the CPU operation in FIQ exception processing mode.

8.5.2 External Interrupt and Internal Interrupt

IRQ interrupts are used as regular interrupts that have lower priority than FIQ interrupts. The interrupt controller determines the priority of multiple interrupt requests. If the interrupt controller finds an interrupt with higher priority than the interrupt level it is currently processing, it issues an IRQ exception processing request (nIRQ) to the CPU. It is possible to determine interrupt sources by reading the registers inside the interrupt controller.

8.5.2.1 Description of the Interrupt Priority Control Method

1. An interrupt with a higher interrupt level is given priority.
2. An interrupt with a larger interrupt number takes precedence within the same interrupt level.
3. After an IRQ exception has been notified by reading the IRN register, interrupts below the same level are masked

8.5.2.2 IRQ Interrupt Sequence (IRQ external interrupt sequence)

1. Setting an interrupt level (software)
   Interrupt mask is cancled by clearing bit 7 (I flag) of the SPSR register of the CPU to "0" and setting the interrupt level of the interrupt to be used to other than "0". However, if the interrupt level is set to "0," interrupts are still masked.

2. Generating an interrupt (hardware)
   When an interrupt request occurs, the interrupt signal is notified to the interrupt controller.

3. Requesting exception processing to the CPU (hardware)
   If there is an interrupt request at an interrupt level higher than the current interrupt level retained in the CIL register (and the CILE register) among interrupt requests, the interrupt controller notifies the nIRQ signal and requests IRQ exception processing to the CPU.
   At this time, the interrupt controller first detects the highest interrupt level among interrupt requests, detects the largest interrupt number among them, and then sets it in the IRN register.
   If there is no interrupt request at an interrupt level higher than the current interrupt level retained in the CIL register (and the CILE register) among interrupt requests, no IRQ exception processing request is generated and the IRN register value becomes "0".

4. Acceptance of exceptions by the CPU (hardware)
   If the CPU is in the exception acceptable state (not masked by the I flag of the CPSR), it proceeds to IRQ exception processing.
   The CPU saves the address of the next instruction in r14_irq, and saves the value of the CPSR in SPSR_irq.
   The CPU sets "1" in bit 7 (I flag) of the CPSR, disabling the acceptance of IRQ exceptions.

5. Interrupt processing (software and hardware)
   (a) Software: The IRQ handler acquires an interrupt number by reading the IRN register, and executes the interrupt processing corresponding to the acquired interrupt number.
   (b) Hardware: When the IRN register is read, the INTRC sets the applicable bit of the CIL register corresponding to that interrupt level to "1". Once the applicable bit of the CIL register is set, all interrupt requests below that interrupt level are masked, the nIRQ signal to the CPU is stopped, and the binary data of the interrupt level is set in the CILE register.
   (c) Software: The IRQ handler must cancel an IRQ interrupt request of the IRQ interrupt request source before finishing interrupt processing. For the sources whose detect mode is set to an edge detection, clear the IRQA register.

6. Returning from an interrupt (software)
   (a) Execute a write operation to the CILCL register (any data can be written) and clear the bit of the CIL register corresponding to the interrupt level of the interrupt processed.
   (b) Execute return processing from an interrupt.

[Note]
Refer to the DataSheet of ARM946E for details of the CPU operation in IRQ exception processing mode.

8.5.3 Reentrant Interrupt Control

Reentrant interrupt control is the processing that suspends the currently executing interrupt processing if there is an interrupt request at an interrupt level higher than the currently processing interrupt level, and processes that interrupt request at a higher interrupt level.
In reentrant interrupt control, the IRQ handler saves the contents of the link register (R14_irq) and the save program status register (SPSR_irq) that are overwritten upon accepting an IRQ exception, and clears bit I of the current program status register (CPSR) so as to enable the acceptance of IRQ exceptions.
Furthermore, if a subroutine is used within the IRQ handler, the IRQ handler makes a transition from the IRQ mode to the System mode in order to prevent the return address of the subroutine saved in the link register (R14_irq) in the IRQ mode from being overwritten when an interrupt request at a higher interrupt level is accepted. Although the System mode uses the same register banks as those used in the User mode, access to the status registers and others can be made in the same way as in the IRQ mode. Note that if the IRQ handler is executed in the System mode, it is necessary to save the link register in the System mode (User mode) as well as the registers used by the IRQ handler.

The processing procedure of the IRQ handler that performs reentrant interrupt control is as follows:

1. Saves the R14_irq, SPSR_irq and other necessary working registers.
2. Reads the IRN register.
3. Switches to the System mode and enables IRQs.
4. Saves the link register in the System mode (User mode) and other necessary working registers.
5. Executes the main unit of the IRQ handler.
6. Restores the registers in the System mode (User mode).
7. Prohibits IRQs, and switches to the IRQ mode.
8. Clears the CIL register.
9. Restores the R14_irq, SPSR_irq and working registers.
10. Returns from an IRQ.

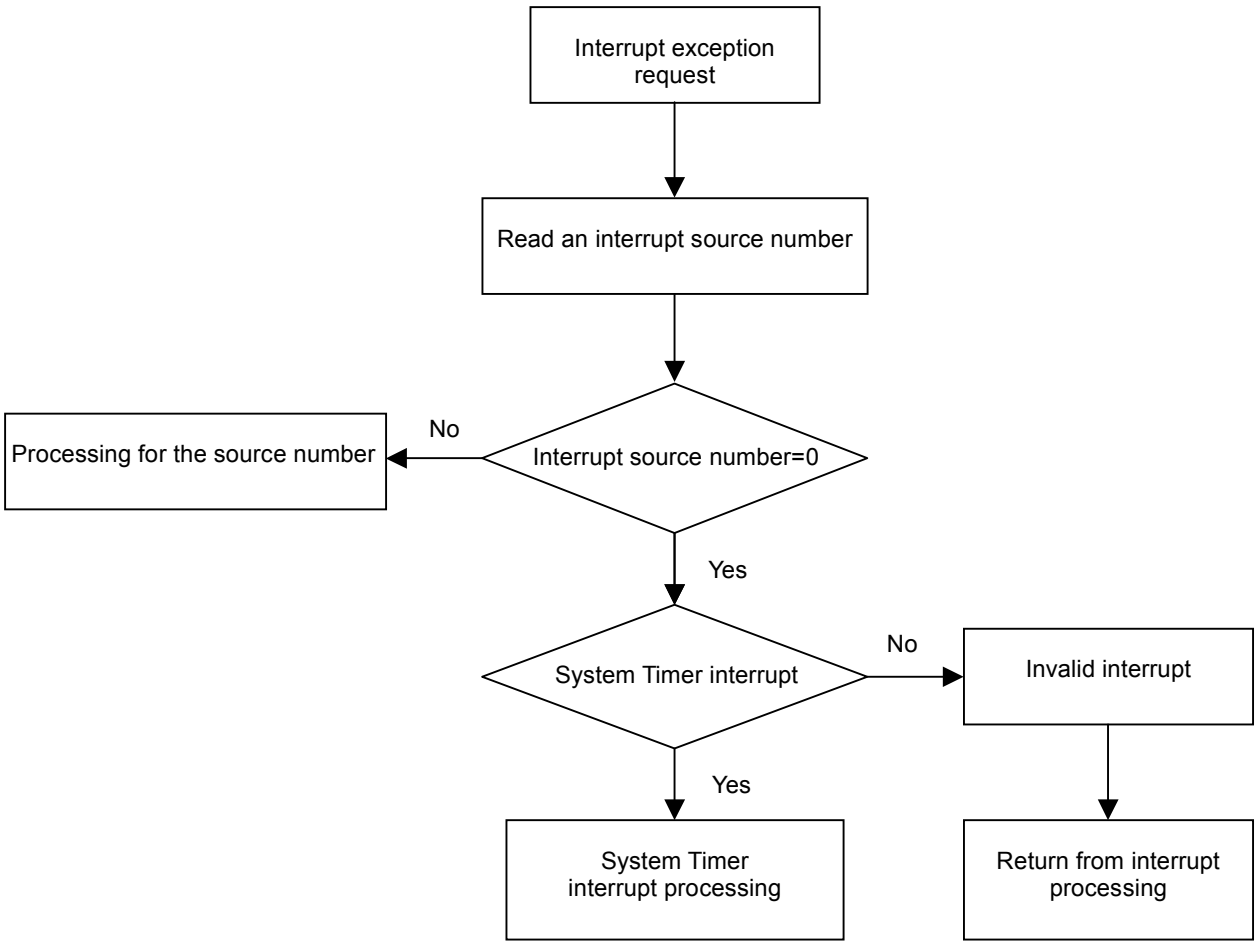
8.5.4 Cautions Pertaining to Interrupts

8.5.4.1 Reading the IRN register

The IRN register is cleared by a read operation.

8.5.4.2 Invalid interrupts

Even though an exception processing request has occurred, interrupt source number 0 may be read because of a lost interrupt source. However, because interrupt source number 0 has been assigned to the system timer interrupt, the system timer interrupt status will be read if interrupt source number 0 is read for an exception processing request. It is thus necessary for the interrupt handler to perform system timer interrupt processing if a system timer interrupt has occurred, and to handle as an invalid interrupt if a system timer interrupt has not occurred. Since no processing is required when an invalid interrupt occurs, it simply returns from interrupt processing.



8.5.4.3 Notes on Interrupt Handler

An incorrect interrupt may occur if an instruction that releases the interrupt mask is executed immediately of several instructions after the instruction clearing the interrupt in an interrupt handler routine.
Samples are given below.

Case 1:

If an incorrect interrupt occurs when performing the mask release of the IRQ mask flag (I flag) of the ARM CPSR register after a read of the INTRC IRN register, the interrupt routine will be repeated indefinitely.

Program code sample:

```
MRS   R2,CPSR                ; Read CPSR register
AND   R2,R2,#0xFFFFFF7F      ; Clear only IRQ mask flag
LDR   R0,[PC,#IRND-.-8]      ; R0=IRN register address
LDR   R1,[R0,#0x00]          ; Read IRN register
MSR   CPSR_F,R2              ; Clear ARM IRQ mask flag
```

Case 2:

If an incorrect interrupt occurs when writing to the INTRC CIL or CILCL register (release of interrupt mask level) after a clear of the APB peripheral interrupt source register, the timer interrupt handling is performed excessively.

Program code sample:

```
LDR   R0,[PC,#CILCLD-.-8]    ; R0=CILCL register address
MOV   R1,#0x00               ; R1= Write value to CILCL
LDR   R2,[PC,#APBID-.-8]     ; R2=Interrupt source of APB peripheral
                             ; Clear register address
MOV   R3,#0x01               ; R3=Interrupt source clear value
STR   R3,[R2,#0x00]          ; Clear APB peripheral interrupt source
STR   R1,[R0,#0x00]          ; Write to CILCL register
```

Correct actions for the cases above are described below.

For case 1:

This problem can be avoided by inserting a wait time as dummy instructions between the interrupt clear instruction and the mask release instruction.

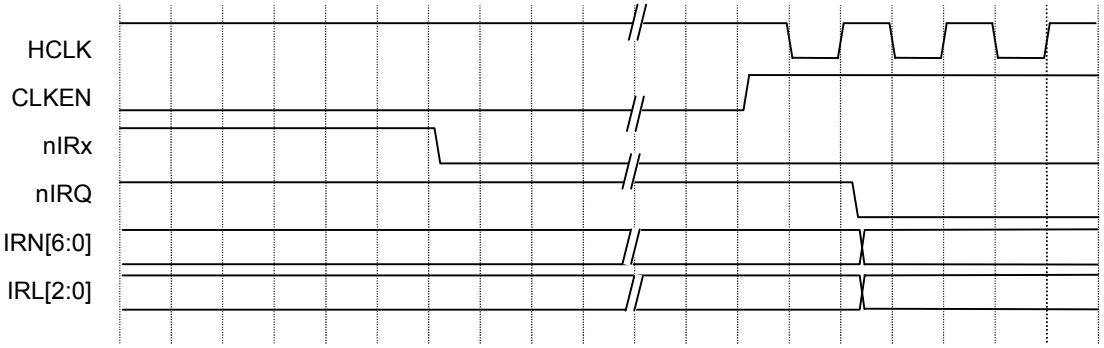
For HCLK = CPUCLK/2 NPO*2
For HCLK = CPUCLK/4 NPO*4

For case 2:

Read the interrupt status register of the relevant peripheral after executing the interrupt source clear instruction of each peripheral to check that the interrupt source has been cleared.

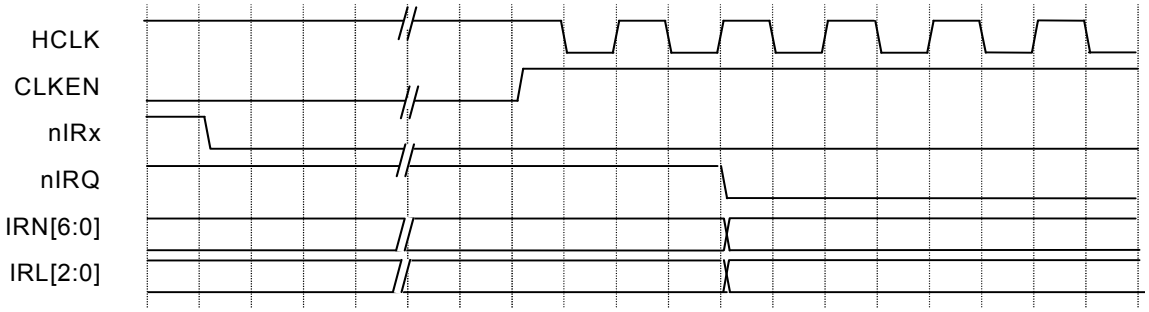
8.5.5 Returning from the HALT/STANDBY Mode (Hardware Requirement)

See below for the timing for returning from the HALT/STANDBY mode.



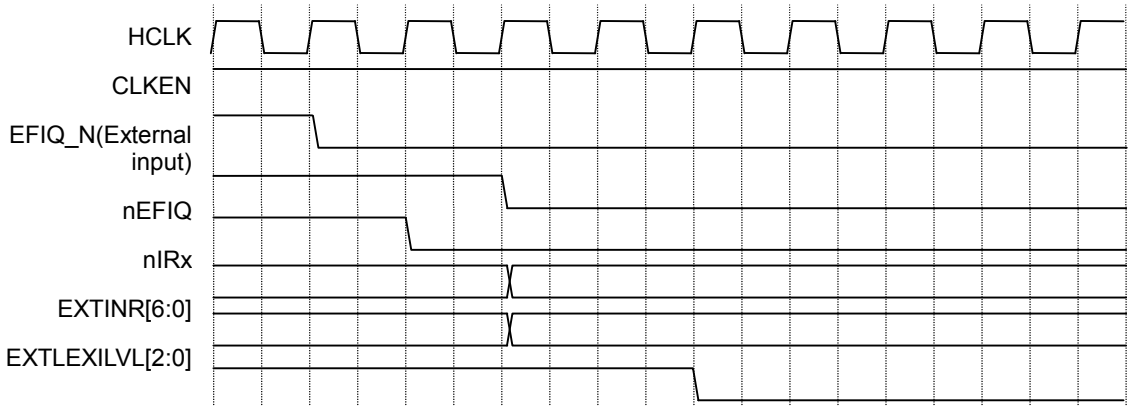
(a) HALT/STANDBY/STOP→RUN (Clock asynchronous)

(IRQ interrupt of nIR0 to nIR15)



(b) HALT/STANDBY/STOP→RUN (Clock asynchronous)

(IRQ interrupt of nIR16 to nIR31)



During RUN processing (Clock synchronous)

8.5.6 Generating Error Responses

When an access is made to 0x78000030 to 0x780FFFFF in the space that is allocated to the interrupt controller (INTRC), a data abort exception occurs.

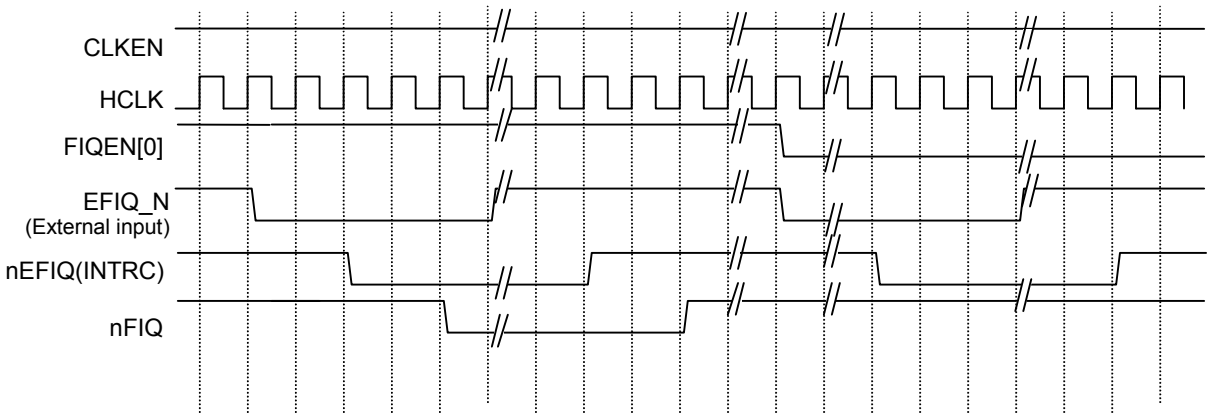
8.5.7 Interrupt Response Time

FIQ interrupt (nFIQ)	4 clock cycles from the input of EFIQ_N external interrupt to ARM FIQ input
IRQ interrupt (nIR0 to nIR15)	2 clock cycles from the occurrence of an IRQ interrupt to ARM IRQ input
IRQ interrupt (nIR16 to nIR31)	3 clock cycles from the occurrence of IRQ interrupt to ARM IRQ input

[Note]
Refer to the DataSheet of ARM946E for the time from notification of IRQ and FIQ exception requests to the CPU to the start of actual interrupt processing.

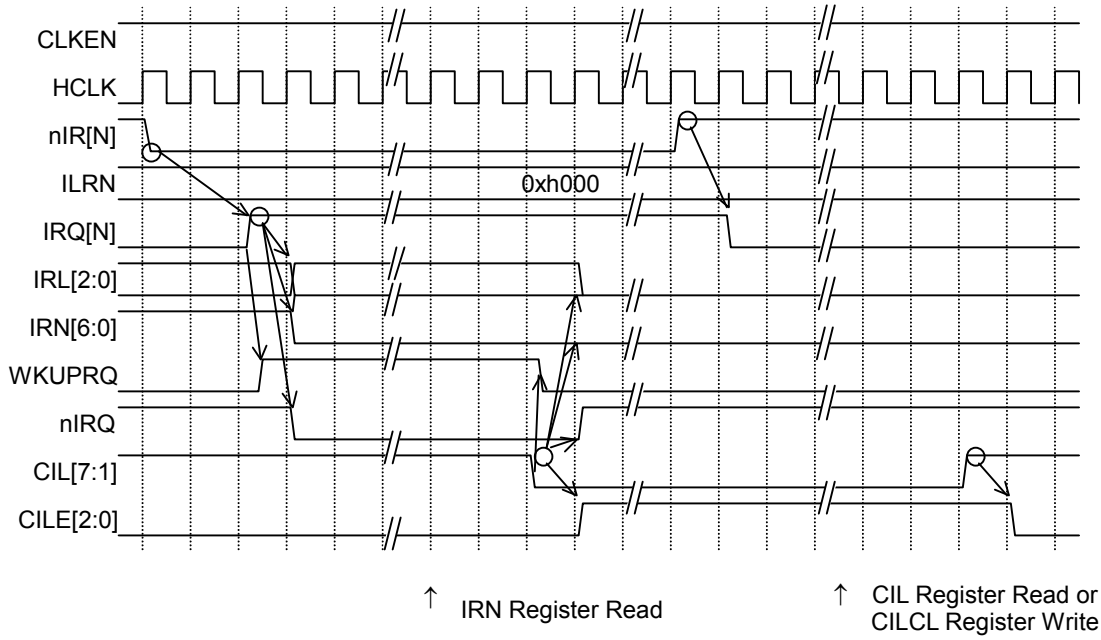
8.6 Interrupt Acceptance Timing Diagram

8.6.1 FIQ Interrupt Timing

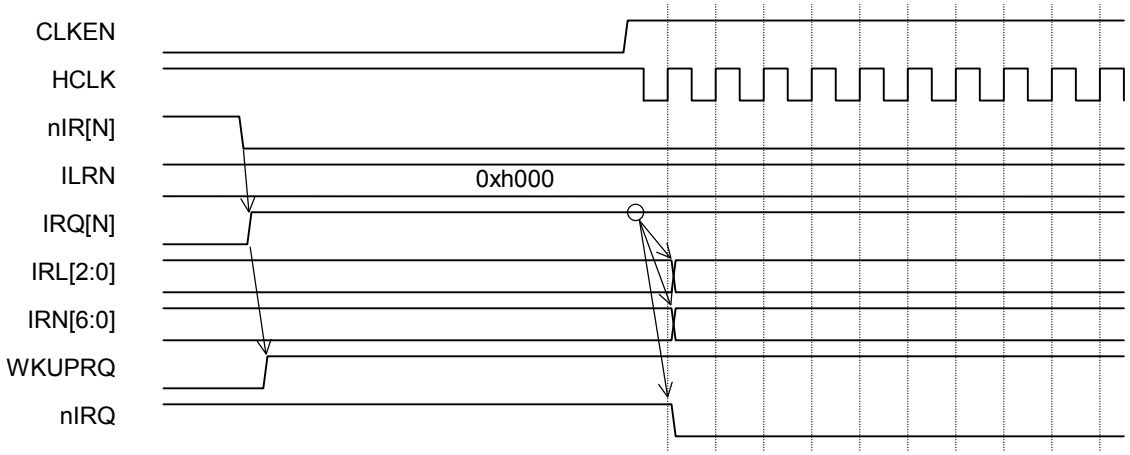


At clock operation

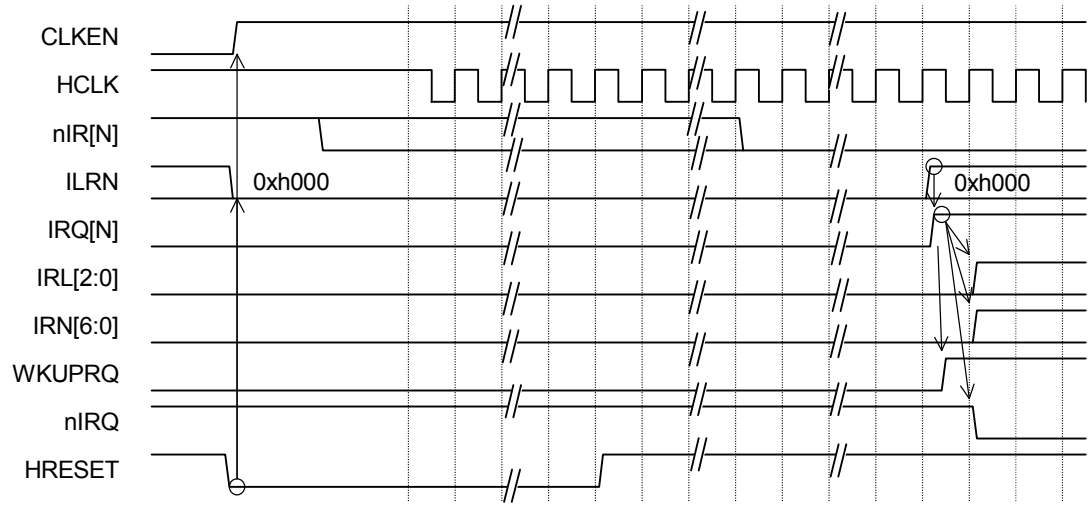
8.6.2 IRQ Interrupt Timing (nIR0 to nIR15)



During clock operation



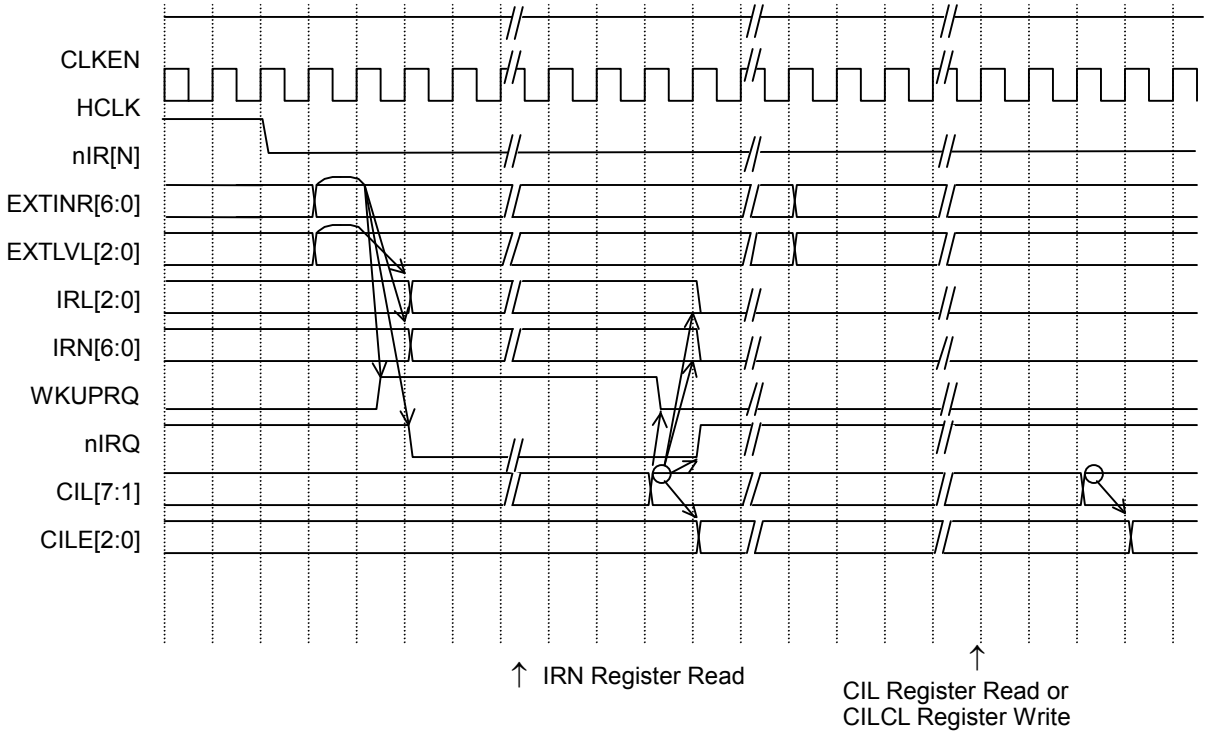
STANDBY→RUN



↑ILC0,ILC1 Register Write

Sleep → RUN

8.6.3 IRQ Interrupt Timing (nIR16 to nIR31)



During clock operation (nIR 16 to nIR 31)

Chapter 9

Built-in Memory

Chapter 9 Built-in Memory

9.1 Overview

The ML696201 has 16KB (4K × 32 bits) of MASK-ROM (address: 0x48000000–0x48003FFC) and 128KB (32K × 32 bits) of SRAM (address: 0x50000000–0x5001FFFC).
The ML69Q6203 has 4KB (256K × 16 bits) of a Flash ROM module in addition to the memory described above. The Flash ROM module is connected to the external memory bus inside of the LFBGA package.

9.2 Features

The built-in MASK-ROM has the following features.

- Bus width: 32 bits
- Allows 32-bit Read access.
- Wait: 0 Wait at Read access

The built-in SRAM has the following features.

- Bus width: 32 bits
- Allows 32/16/8-bit Read/Write access.
- Wait: 0 wait at Read access. 0 Wait at Write access.

The Flash-ROM has the following features.

- Bus width: 16 bits
- Allows 16-bit Read access.
- Three types of Erase commands (enables batched erase and erase in small sector units)
  Sector erase: 2KB per sector
  Block erase: 64KB per block
  Chip erase: All the sectors in batch
- Protection function
  Top address side: : 16KB Block Protect
  : 512KB Chip Protect
- Self Time Erase/Program
- Write termination detection function: Toggle bit, DATA# polling, and RD/BY# pin
- Data protection function by hardware and software

Note: Built-in Flashes are accessed through the external memory controller since they are in a multichip connection. For the access timing setting for the external memory controller, see Chapter 10, External Memory Controller.

Chapter 10

External Memory Controller

Chapter 10 External Memory Controller

10.1 Overview

This LSI incorporates into it a controller for connecting external SDRAM, ROM, SRAM, and I/O.
The controller supports the following devices.

- Asynchronous type ROM
- Asynchronous type SRAM
- Flash memory
- I/O device
- SDRAM (EDO-DRAM cannot be handled)

10.1.1 List of Pins

The following table shows the pin utilization modes for connection with the devices.

Pin name	I/O	Function	Primary function
XA23 to XA21	O	External address bus	—
XA20 to XA17	O	External address bus	PIOC[15:12]
XA16 to XA01	O	External address bus	PIOA[15:0]
XD15 to XD00	I/O	External data bus	PIOB[15:0]
XROMCS_N	O	External ROM chip select	PIOC[9]
XRAMCS_N	O	External RAM chip select	PIOC[8]
XIOCS00_N	O	I/O bank 0 chip select 0	PIOC[4]
XIOCS01_N	O	I/O bank 0 chip select 1	PIOC[5]
XIOCS10_N	O	I/O bank 1 chip select 0	PIOC[6]
XIOCS11_N	O	I/O bank 1 chip select 1	PIOC[7]
XOE_N	O	Output enable	PIOC[11]
XWE_N	O	Write enable	PIOC[10]
XBS1_N	O	External bus byte select (MSB)	PIOC[3]
XBS0_N	O	External bus byte select (LSB)	PIOC[2]
XSDCS_N	O	SDRAM chip select	—
XSDCLK	O	Clock for SDRAM	—
XSDCKE	O	Clock enable for SDRAM	—
XCAS_N	O	Row address strobe for SDRAM	—
XRAS_N	O	Row address strobe	—
XDQM1	O	Data input/output mask for SDRAM	—
XDQM0	O	Data input/output mask for SDRAM	—
XWAIT1	I	WAIT signal for I/O bank 1	PIOC[1]
XWAIT0	I	WAIT signal for I/O bank 0	PIOC[0]
XSYSCLK	O	AHB clock for external bus	

10.1.2 List of Registers

Address	Register name	Symbol	R/W	Size	Initial value
0x7810_0000	Bus width control register	BWC	R/W	32	0x0000_0008
0x7810_0004	External ROM access control register	ROMAC	R/W	32	0x0000_0007
0x7810_0008	External SRAM access control register	RAMAC	R/W	32	0x0000_0007
0x7810_000C	External I/O bank 0 access control register	IO0AC	R/W	32	0x0000_0007
0x7810_0010	External I/O bank 1 access control register	IO1AC	R/W	32	0x0000_0007
0x7818_0000	DRAM bus width control register	DBWC	R/W	32	0x0000_0000
0x7818_0004	DRAM control register	DRMC	R/W	32	0x0000_0000
0x7818_0008	DRAM property parameter control register	DRPC	R/W	32	0x0000_0000
0x7818_000C	SDRAM mode register	SDMD	R/W	32	0x0000_0001
0x7818_0010	DRAM command register	DCMD	R/W	32	0x0000_0000
0x7818_0014	DRAM refresh cycle control register 0	RFSH0	R/W	32	0x0000_0000
0x7818_0018	DRAM power down control register	PDWC	W	32	0x0000_0003
0x7818_001C	DRAM refresh cycle control register 1	RFSH1	R/W	32	0x0000_0000

10.2 Registers

10.2.1 Bus Width Control Register (BWC)

The BWC register is used for setting an external data bus width of each bank.
Read/Write operation is enabled for the BWC register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BWC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	IO1BW[1:0]		IO0BW[1:0]		RAMBW [1:0]		ROMBW [1:0]		—*	—*
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Address: 0x78100000
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.

[Description of Bits]

- **ROMBW[1:0]** (Bits 2 – 3)
  These bits are used for setting a ROM area bus.
  Set parameters according to this setting when using the built-in Flash memories. Be sure to use the 16-bit width.

ROMBW[1:0]		Description
3	2	
0	0	Not mounted (ROM area is disabled. An abort exception is generated for the access from the CPU.)
0	1	(Reserved)
1	0	16-bit width
1	1	(Reserved)

(Note)
If any of the values that are indicated as "(Reserved)" is set, access to the ROM area is not guaranteed.

- **RAMBW[1:0]** (Bits 4 – 5)
  These bits are used for setting an SRAM area bus.

RAMBW[1:0]		Description
5	4	
0	0	Not installed (SRAM area is disabled. An abort exception is generated for the access from the CPU. Access from the DMA controller results in an error response).
0	1	(Reserved)
1	0	16-bit width
1	1	(Reserved)

  (Note)
  If any of the values that are indicated as "(Reserved)" is set, access to the SRAM area is not guaranteed.

- **IO0BW[1:0]** (Bits 6 – 7)
  These bits are used for setting an I/O bank 0 (Ext-IO00, Ext-IO01 space in the memory map) area bus.

IO0BW[1:0]		Description
7	6	
0	0	Not mounted (I/O bank 0 area is disabled. Access from the DMA controller results in an error response.)
0	1	(Reserved)
1	0	16-bit width
1	1	(Reserved)

  (Note)
  If any of the values that are indicated as "(Reserved)" is set, access to the I/O bank 0 area is not guaranteed.

- **IO1BW[1:0]** (Bits 8 – 9)
  These bits are used for setting an I/O bank 1 (Ext-IO10, Ext-IO11 space in the memory map) area bus.

IO1BW[1:0]		Description
9	8	
0	0	Not mounted (I/O bank 1 area is disabled. Access from the DMA controller results in an error response.)
0	1	(Reserved)
1	0	16-bit width
1	1	(Reserved)

  (Note)
  If any of the values that are indicated as "(Reserved)" is set, access to the I/O bank 1 area is not guaranteed.

10.2.2 External ROM Access Control Register (ROMAC)

The ROMAC register is used for controlling ROM access timing.
This setting determines the access timing for MCP FLASH ROM and external ROM. When MCP FLASH ROM and external ROM are used concurrently, each access timing cannot be set independently. Therefore, make their timing settings by tailoring them to the slower timing.
Read/Write operation is enabled for the ROMAC register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ROMAC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	BRST	—*	ROMTYPE[2:0]		
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Address: 0x78100004
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
1. When the operation frequency is changed from the high speed to the low speed, make a setting for this register after the frequency is changed.
2. When the operation frequency is changed from the low speed to the high speed, make a setting for this register before changing the frequency.

[Description of Bits]

- **ROMTYPE[2:0]** (Bits 0 – 2)
  These bits are used for setting ROM access timing. The following values indicate the clock counts.
  Set parameters according to this setting when using the built-in Flash memories.

ROMTYPE[2:0]			Address setup	OE/WE pulse width	Data Off timing	Burst timing	Remarks
2	1	0					
0	0	0	1	1	1	1	
0	0	1	1	2	2	2	
0	1	0	1	3	3	3	
0	1	1	1	5	3	3	
1	0	0	2	8	4	5	
1	0	1	2	10	5	6	
1	1	0	2	13	6	7	
1	1	1	2	16	7	9	

Setting of ROMTYPE when using the built-in Flash ROM is shown below. Set the BRST bit to 0 when using the built-in Flash ROM.

HCLK[MHz]	R/W	ROMTYPE	HCLK[MHz]	R/W	ROMTYPE
60	read	011	15	read	000
	write	100		write	001
30	read	001	7.5 or lower	read	000
	write	010		write	000

- **BRST** (Bit 4)
  Setting of Page Mode access to the ROM.

BRST	Description
0	Page Mode off
1	Page Mode on

10.2.3 External SRAM Access Control Register (RAMAC)

The RAMAC register is used for controlling SRAM access timing.
Read/Write operation is enabled for the RAMAC register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RAMAC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAMAC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	BRST	—*	RAMTYPE[2:0]		
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Address: 0x78100008
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
1. When the operation frequency is changed from the high speed to the low speed, make a setting for this register after the frequency is changed.
2. When the operation frequency is changed from the low speed to the high speed, make a setting for this register before changing the frequency.

[Description of Bits]

- **RAMTYPE[2:0]** (Bits 0 – 2)
  These bits are used for setting SRAM access timing. The following values indicate the clock counts.

ROMTYPE[2:0]			Address setup	OE/WE pulse width	Data Off timing	Burst timing	Remarks
2	1	0					
0	0	0	1	1	1	1	
0	0	1	1	2	2	2	
0	1	0	1	3	3	3	
0	1	1	1	5	3	3	
1	0	0	2	8	4	5	
1	0	1	2	10	5	6	
1	1	0	2	13	6	7	
1	1	1	2	16	7	9	

- **BRST** (Bit 4)
  Setting of Page Mode access to the SRAM.

BRST	Description
0	Page Mode off
1	Page Mode on

10.2.4 External I/O Bank 0 Access Control Register (IO0AC)

The IO0AC register is used for controlling access timing for the I/O bank 0 (Ext-IO00, Ext-IO01 space in the memory map) area.
Read/Write operation is enabled for the IO0AC register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IO0AC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	IO0TYPE[2:0]		
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Address: 0x7810000C
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
1. When the operation frequency is changed from the high speed to the low speed, make a setting for this register after the frequency is changed.
2. When the operation frequency is changed from the low speed to the high speed, make a setting for this register before changing the frequency.

[Description of Bits]

- **IO0TYPE[2:0]** (Bits 0 – 2)
  These bits are used for setting the IO0 area access timing. The following values indicate the clock counts.

ROMTYPE[2:0]			Address setup	OE/WE pulse width	Data Off timing	Remarks
2	1	0				
0	0	0	1	1	1	
0	0	1	1	4	3	
0	1	0	1	6	4	
0	1	1	2	8	5	
1	0	0	2	12	7	
1	0	1	2	16	8	
1	1	0	3	20	9	
1	1	1	4	24	11	

Access timing to the devices to be selected by the XIOCS01_N and XIOCS00_N pins are set. XWAIT0 is the WAIT signal to be used, and XWAIT0 is shared with respect to the two I/Os described above.

10.2.5 External I/O bank 1 Access Control Register (IO1AC)

The IO1AC register is used for controlling access timing for the I/O bank 1 (Ext-IO10, Ext-IO11 space in the memory map) area.
Read/Write operation is enabled for the IO1AC register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IO1AC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IO1AC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	IO1TYPE[2:0]		
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Address: 0x78100010
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
1. When the operation frequency is changed from the high speed to the low speed, make a setting for this register after the frequency is changed.
2. When the operation frequency is changed from the low speed to the high speed, make a setting for this register before changing the frequency.

[Description of Bits]

- **IO1TYPE[2:0]** (Bits 0 – 2)
  These bits are used for setting the IO1 area access timing. The following values indicate the clock counts.

ROMTYPE[2:0]			Address setup	OE/WE pulse width	Data Off timing	Remarks
2	1	0				
0	0	0	1	1	1	
0	0	1	1	4	3	
0	1	0	1	6	4	
0	1	1	2	8	5	
1	0	0	2	12	7	
1	0	1	2	16	8	
1	1	0	3	20	9	
1	1	1	4	24	11	

Access timing to the devices to be selected by the XIOCS11_N and XIOCS10_N pins are set. XWAIT1 is the WAIT signal to be used, and XWAIT1 is shared with respect to the two I/Os described above.

10.2.6 DRAM Bus Width Control Register (DBWC)

The DBWC register is used for setting a bus width of a DRAM area.
Read/Write operation is enabled for the DBWC register through program control.
* Since EDO-DRAM is not supported, do not set the EDO-DRAM use mode.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBWC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	BWDRAM [1:0]	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78180000
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
1. This register is not involved in the control of SDRAM clock output XSDCLK.

[Description of Bits]

- **BWDRAM** (Bits 0 – 1)
  These bits are used for setting a bus width of a DRAM area.

BWDRAM[1:0]		Description
1	0	
0	0	Not mounted (Access to the DRAM area results in an error response.)
0	1	8-bit width (used only at EDO-DRAM connection). Setting prohibited.
1	0	16-bit width
1	1	(Reserved)

10.2.7 DRAM Control Register (DRMC)

The DRMC register is used for setting a DRAM type and access timing.
Read/Write operation is enabled for the DRMC register through program control.
* Since EDO-DRAM is not supported, do not set this LSI to the EDO-DRAM use mode.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRMC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	RFRSH	PDWN	—*	PRELAT	—*	ARCH	AMUX[1:0]	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78180004
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.

[Description of Bits]

- **AMUX[1:0]** (Bits 0 – 1)
  These bits are used for setting the address multiplexing.

AMUX[1:0]		Description		
1	0	Column length	RAS address	CAS address
0	0	8bit	A[23:8]	A[7:0]
0	1	9bit	A[23:9]	A[8:0]
1	0	10bit	A[23:10]	A[9:0]
1	1	(Reserved)		

- **ARCH** (Bit 2)
  This bit is used for setting the DRAM type.

ARCH	Description
0	SDRAM
1	EDO-DRAM Cannot be set

- **PRELAT** (Bit 4)
  This bit is used for setting the precharge latency of SDRAM.

PRELAT	Description
0	Fix to 2 clocks
1	Same setting as the CAS latency

- **PDWN** (Bit 6)
  This bit is used for setting automatic transition to the SDRAM Power Down mode.

PDWN	Description
0	Disables automatic transition
1	Enables Automatic transition

- **RFRSH** (Bit 7)
  This bit is used for controlling execution of distributed CBR refresh.

RFRSH	Description
0	Stops distributed CBR refresh
1	Executes distributed CBR refresh

10.2.8 DRAM Property Parameter Control Register (DRPC)

The DRPC register is used for setting DRAM property parameters.
Read/Write operation is enabled for the DRPC register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRPC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRPC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	DRAMSPEC[3:0]			
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78180008
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.

[Description of Bits]

- **DRAMSPEC[3:0]** (Bits 0 – 3)
  These bits are used for setting DRAM property parameters.

DRAMSPEC[3:0]				When SDRAM used (Number of clocks)				
3	2	1	0	tRCD	tRAS	tRP	tDPL	
0	0	0	0	1	2	1	1	High speed / Low frequency
0	0	0	1	1	3	1	1	↑ / ↑
0	0	1	0	2	3	2	1	↑ / ↑
0	0	1	1	2	4	2	1	↑ / ↑
0	1	0	0	2	4	2	2	↑ / ↑
0	1	0	1	2	5	2	1	↑ / ↑
0	1	1	0	2	5	2	2	↑ / ↑
0	1	1	1	2	5	3	1	↑ / ↑
1	0	0	0	3	5	3	2	↑ / ↑
1	0	0	1	3	6	3	2	Low speed / High frequency
1	0	1	0	(Reserved)				
1	0	1	1					
1	1	0	0					
1	1	0	1					
1	1	1	0					
1	1	1	1					

(Note)
Do not set the values that are indicated as "(Reserved)". Otherwise, the operation will not be guaranteed.
Set the parameters (tRCD, tRAS and so on) so as to satisfy the DRAM standard by calculating the timing parameter based on the operation clock and by referencing the data sheet of the DRAM to be used.

10.2.9 SDRAM Mode Register (SDMD)

The SDMD register is used for setting the CAS latency of SDRAM.
Read/Write operation is enabled for the SDMD register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMD	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	MODE WR	—*	—*	—*	—*	—*	—*	LT MODE
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Address: 0x7818000C
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
1. The SDRAM mode setup cycle is executed only when this register is written as MODEWR=1. When read, MODEWR will always read "0".

[Description of Bits]

- **LTMODE** (Bit 0)
  This bit is used for setting the CAS latency of SDRAM.

LTMODE	Description
0	2 clocks
1	3 clocks

- **MODEWR** (Bit 7)
  This bit is used for executing the mode setting.

MODEWR	Description
0	Setting operation disabled.
1	Executes the setting operation.

The following setting cycle is issued.

A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
OPECODE					LTMODE			WT	BL		

*Axx indicates the input address signal to SDRAM.

Field	Function	Setting value	Meaning
OPCODE	Mode option	00000	Burst Read & Burst Write
LTMODE	CAS latency	010 011	2 clocks 3 clocks
WT	Wrap type	0	Sequential (Wrap Around)
BL	Burst length	011	Burst length: 8

Note: The setting values other than LTMODE are fixed.

10.2.10 DRAM Command Register (DCMD)

This register is used for executing the DRAM command.
When SDRAM is used, the specified command can be executed by writing values in this register.
Read/Write operation is enabled for the DCMD register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCMD	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	DRCMD[2:0]		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78180010
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.

[Description of Bits]

- **DRCMD** (Bits 0 – 2)
  These bits are used for executing the DRAM command.

DRCMD[2:0]			When SDRAM is used
2	1	0	
0	0	0	No Operation
0	0	1	(Reserved)
0	1	0	(Reserved)
0	1	1	(Reserved)
1	0	0	PALL (SDRAM all bank precharge)
1	0	1	REF (SDRAM CBR refresh)
1	1	0	SELF (Starts SDRAM self-refresh)
1	1	1	SREX (Ends SDRAM self-refresh)

(Notes)
1. Do not write the setting values that are indicated as "(Reserved)". Operation cannot be guaranteed if they are written.
2. In self-refresh mode, operation of the commands other than Self-refresh Termination cannot be guaranteed.
3. The output of SDRAM clock, SDCLK, will be:
   (a) Terminated by the SELF command ("L" level), or
   (b) Restarted by the SREX command.
4. Termination of access to this register is set to a wait state until the execution of the specified command terminates.

10.2.11 DRAM Refresh Cycle Control Register 0 (RFSH0)

The RFSH0 register is used for setting a DRAM refresh cycle.
Set whether refresh is performed in double or equal to the cycle that was set by the RFSH1 register.
Read/Write operation is enabled for the RFSH0 register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RFSH0	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFSH0	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	RC CON
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x78180014
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
1. If a write operation is performed to this register in the not-mounted (DBWC=0) state, operation of the DRAM controller will not be guaranteed.

[Description of Bits]

- **RCCON** (Bit 0)
  This bit is used for setting a DRAM refresh cycle.

RCCON	Description
0	Refresh with the double cycle of the clock that is generated by the RFSH1 register
1	Refresh with the equal cycle of the clock that is generated by the RFSH1 register

10.2.12 DRAM Refresh Cycle Control Register 1(RFSH1)

The RFSH1 register is used for setting a refresh cycle of DRAM.
Set a frequency division ratio generated from CCLK in this register.
Read/Write operation is enabled for the RFSH1 register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RFSH1	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	RFSEL1												
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7818001C
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.

[Description of Bits]

- **RFSEL1** (Bits 0 – 12)
  These bits are used for setting a refresh cycle (frequency division ratio generated from CCLK) of DRAM.
  The calculation expression of the refresh clock cycle is as follows.
  Refresh clock cycle = CCLK/RFSEL1[12:0]
  Set the RFSEL1 value within the range from 0x00000008 to 0x00000406.

  [Note]
  Set the RFSH1 register (refresh cycle setting of DRAM) before setting a frequency division value of CCLK. Since CCLK can be changed through the clock gear, the setting of this register needs to be changed when CCLK is changed dynamically.

  Since CCLK supplies 7.5 MHz in this LSI, the following value is set in CCLK.
  Set the frequency of the clock to be generated to 32 kHz or more or 64 kHz or more.

RFSEL1 setting value	Frequency division value	Refresh clock frequency that is set (CCLK = 7.5 MHz)
0075	117	64.10 kHz
00EA	234	32.05 kHz

10.2.13 DRAM Power Down Control Register (PDWC)

The PDWC register is used for setting the condition for changing a mode to the DRAM Power Down mode. Only Write operation is enabled for the PDWC register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDWC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*		PDCNT		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Address: 0x78180018
Access: W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.

[Description of Bits]

- **PDCNT** (Bits 0 – 3)
  These bits are used for setting the condition for transition of the mode to the DRAM Power Down mode.

PDCNT				Description
3	2	1	0	
0	0	0	0	Transition to the Power Down mode when the Idle state continued for 1 cycle or more
0	0	0	1	Transition to the Power Down mode when the Idle state continued for 2 cycles or more
0	0	1	0	Transition to the Power Down mode when the Idle state continued for 3 cycles or more
0	0	1	1	Transition to the Power Down mode when the Idle state continued for 4 cycles or more
0	1	0	0	Transition to the Power Down mode when the Idle state continued for 5 cycles or more
0	1	0	1	Transition to the Power Down mode when the Idle state continued for 6 cycles or more
0	1	1	0	Transition to the Power Down mode when the Idle state continued for 7 cycles or more
0	1	1	1	Transition to the Power Down mode when the Idle state continued for 8 cycles or more
1	0	0	0	Transition to the Power Down mode when the Idle state continued for 9 cycles or more
1	0	0	1	Transition to the Power Down mode when the Idle state continued for 10 cycles or more
1	0	1	0	Transition to the Power Down mode when the Idle state continued for 11 cycles or more
1	0	1	1	Transition to the Power Down mode when the Idle state continued for 12 cycles or more
1	1	0	0	Transition to the Power Down mode when the Idle state continued for 13 cycles or more
1	1	0	1	Transition to the Power Down mode when the Idle state continued for 14 cycles or more
1	1	1	0	Transition to the Power Down mode when the Idle state continued for 15 cycles or more
1	1	1	1	Transition to the Power Down mode when the Idle state continued for 16 cycles or more

10.3 Memory Access Control

10.3.1 Bus Width

The BWC register is used for setting "mounted" or "not mounted" for the external ROM and external SRAM areas. For the areas of external I/O bank 0 and external I/O bank 1, the BWC register is used for setting not mounted or the bus width. In SDRAM, the DBWC register is used for setting not mounted or the bus width
At reset, all the initial values are set to the not mounted state except the setting of the ROM area.
The initial value of the ROM area is set to 16-bit connection.

The table below shows each area and the bus width that can be set. When an inappropriate bus width is set, the operation will not be guaranteed.

Area	Bus width	
	8 bits	16 bits
ROM	Unavailable	Available
RAM	Unavailable	Available
IO0	Unavailable	Available
IO1	Unavailable	Available
SDRAM	Unavailable	Available

When the bus width and the data size to be accessed do not match, the memory is accessed according to the bus width.
For instance, if a 32-bit Read operation is performed for the external memory whose bus width is 16 bits, the memory controller realizes 32-bit access by accessing LSB16 bits first and then MSB16 bits of the data. As a result, the XOE_N signal or XWE_N signal is output twice for one word access. See the timing charts that are shown in Figure 10-1 and Figure 10-2.

10.3.2 ROM/SRAM Control Method

1. To use this LSI with SRAM mounted, set the SRAM bus width to 16 bits in the BWC register.
2. Set the XOE_N/XWE_N pulse width in the ROMAC or RAMAC register.

At reset, the bus width of the ROM area is set to 16 bits.

When using CPUCLK in 32 kHz mode and returning to the PLL mode, set in advance CPUCLK to the access control register's state that would be effect after returning to the PLL mode, then return the mode to the PLL mode.
If the setting of the access control register causes external bus timing violation even for a moment, this LSI hangs up.

10.3.3 I/O bank 0/1 Control Method

1. When using I/O bank 0 or I/O bank 1, set the bus width of the I/O bank to 16 bits in the BWC register.

2. Set the XOE_N/XWE_N pulse width in the IO0AC register or the IO1AC register.

The Chip Select signal is stored in each bank and CS00/CS01 corresponds to I/O bank 0 and CS10/CS11 to I/O bank 1. Only one type of access WAIT setting is allowed in the bank.

The access times of IO banks 0 and 1 can be extended by inputting the XWAIT signal.
The XWAIT signal is sampled 1HCLK before the XOE_N/XWE_N signal is deasserted and the this signal is extended while the XWAIT signal is "H". When the XWAIT signal is set to "L", access terminates. The access time is extended in the unit of the pulse width of XOE_N/XWE_N.

When using CPUCLK in 32 kHz mode and returning to the PLL mode, set in advance CPUCLK to the access control register's state that would be effect after returning to the PLL mode, then return the mode to the PLL mode.
If the setting of the access control register causes external bus timing violation even for a moment, this LSI hangs up.

10.3.4 XSYSCLK Output Enable Control Method

When a synchronous type external device is used and a bus clock is required, the AHB clock (HCLK) can be output to XSYSCLK.
This clock output is in an output disable state at Power ON. To enable the clock output, write "1" to the XSYSCLKEN bit of the Clock Stop register of the configuration register module.

10.3.5 DRAM Control Method

When using CPUCLK with 32 kHz, DRAMC does not operate. When data needs to be stored in the DRAM memory, set refresh before changing the mode to the 32 kHz mode.

(1) Enabling/disabling the DRAM controller using the Mode pin
To use the DRAM controller for DRAM connection, the clock supply to the DRAM controller must be enabled in the configuration register module.
When the DRAM controller is disabled, access to the DRAM controller registers, DBWC, DRMC, DRPC, SDMD, DCMD, RFSH0, RFSH1, and PDWC is disabled. When any of the registers is accessed, an error response is returned and data abort processing is performed.

(2) Initializing DRAM
Initialization of DRAM is necessary at Power ON. Perform the necessary DRAM initialization sequence using the following sequence as the reference.

1. When 200 µs or more has elapsed after the reset state is released, set the DRAM access size in the DBWC register. (Originally, a wait time of 200 µs is required after Power ON; however, to ensure allocation of 200 µs or more in the program, wait of 200 µs is coded after resetting.)
2. Set DRAM property parameters in the DRPC register.
3. Perform all bank precharge by writing 0x00000004 in the DCMD register.
4. Perform CBR by writing 0x00000005 in the DCMD register. Perform this operation eight times.
5. In the DRMC register, select address multiplex, SDRAM, and execution/stop of distributed CBR refresh. Set also the presence or absence of precharge latency of SDRAM and automatic execution to Power Down mode.
6. Set the CAS latency of SDRAM in the SDMD register. (When SDRAM is connected)
7. Set a refresh cycle of DRAM in the RFSH0 to RFSH1 registers.
8. Set the condition for changing the mode to DRAM Power Down mode in the PDWC register.
9. Execute various DRAM commands (precharge, CBR refresh, self-refresh start/end) using the DCMD register.

(3) Address Output and Bus Width
Correspondence between address output (XA) and the DRAM address pin (A[*])

	AMUX[1:0] = 00 Column length 8 bits (*2)		AMUX[1:0] = 01 Column length 9 bits (*2)		AMUX[1:0] = 10 Column length 10 bits (*2)		Connection relationship between bus width setting and DRAM address pin (A[*]) (*1)
	Row	Column	Row	Column	Row	Column	16bit
XA[23]	0	0	0	0	0	0	A[22]
XA[22]	0	0	0	0	0	0	A[21]
XA[21]	0	0	0	0	0	0	A[20]
XA[20]	0	0	0	0	0	0	A[19]
XA[19]	0	0	0	0	0	0	A[18]
XA[18]	Ha[26]	Ha[26](*4)	0	0	0	0	A[17]
XA[17]	Ha[25]	Ha[25](*4)	Ha[26]	Ha[26](*4)	0	0	A[16]
XA[16]	Ha[24]	Ha[24](*4)	Ha[25]	Ha[25](*4)	Ha[26]	Ha[26](*4)	A[15]
XA[15]	Ha[23]	Ha[23](*4)	Ha[24]	Ha[24](*4)	Ha[25]	Ha[25](*4)	A[14]
XA[14]	Ha[22]	Ha[22](*4)	Ha[23]	Ha[23](*4)	Ha[24]	Ha[24](*4)	A[13]
XA[13]	Ha[21]	Ha[21](*4)	Ha[22]	Ha[22](*4)	Ha[23]	Ha[23](*4)	A[12]
XA[12]	Ha[20]	Ha[20]	Ha[21]	Ha[21]	Ha[22]	Ha[22]	A[11]
XA[11]	Ha[19]	Ha[11](*3)	Ha[20]	ha[11](*3)	Ha[21]	Ha[11](*3)	A[10](*3)
XA[10]	Ha[18]	Ha[10]	Ha[19]	Ha[10]	Ha[20]	Ha[10]	A[9]
XA[9]	Ha[17]	Ha[9]	Ha[18]	Ha[9]	Ha[19]	Ha[9]	A[8]
XA[8]	Ha[16]	Ha[8]	Ha[17]	Ha[8]	Ha[18]	Ha[8]	A[7]
XA[7]	Ha[15]	Ha[7]	Ha[16]	Ha[7]	Ha[17]	Ha[7]	A[6]
XA[6]	Ha[14]	Ha[6]	Ha[15]	Ha[6]	Ha[16]	Ha[6]	A[5]
XA[5]	Ha[13]	Ha[5]	Ha[14]	Ha[5]	Ha[15]	Ha[5]	A[4]
XA[4]	Ha[12]	Ha[4]	Ha[13]	Ha[4]	Ha[14]	Ha[4]	A[3]
XA[3]	Ha[11]	Ha[3]	Ha[12]	Ha[3]	Ha[13]	Ha[3]	A[2]
XA[2]	Ha[10]	Ha[2]	Ha[11]	Ha[2]	Ha[12]	Ha[2]	A[1]
XA[1]	Ha[9]	Ha[1]	Ha[10]	Ha[1]	Ha[11]	Ha[1]	A[0]
XA[0]	Ha[8]	Ha[0]	Ha[9]	Ha[0]	Ha[10]	Ha[0]	N.C.

* Ha is the abbreviation of Host Address. It indicates the address that is used by the program.

*1: The connection between the XA[*] signal and DRAM address pin A[*] is determined according to the data bus width that is specified by the BWDRAM bit of the DBWC register. The column length is not related to the connection. When an 8-bit bus width is set, SDRAM cannot be used.

*2: The Row/Column addresses are output to the XA pin in multiplex mode according to the column length that is specified in the AMUX bit of the DRMC register. This is applicable to DRAM of column lengths 8 to 10 bits. DRAM of column length 11 bits or more cannot be used.

*3: SDRAM mode: Auto precharge specification bit of SDRAM is set. At Column output (execution of READ/WRITE command), zero is output to specify non-auto precharge.
EDO-DRAM mode: At Column output, the Column address is output.

*4: Since a bank address is assigned at execution of the ACT/READ/WRITE command of SDRAM, the same value is output to the XA[18:12] section at Row/Column output.

(4) CBR Refresh
The DRAM controller can execute distributed CBR (CAS Before RAS) refresh cyclically. The following registers are used for the control.

- Indication of stop/execution: FRESH bit of the DRMC register
- Specification of the distributed refresh cycle: RFSH0 register and RFSH1 register
- CBR refresh can be executed at any timing by writing the CBR refresh command in the DCMD register.

(5) Self-refresh

1. Use a self-refresh mode for power management operation.
2. The value set in the DRCMD bit in the DCMD register determines the transition/return to the SDRAM self-refresh mode.
3. The SDRAM clock (SDCLK) is stopped/restarted by the SDRAM self-refresh start/end command.
4. If DRAM access through the DCMD register access other than an access using the self-refresh end command is executed in self-refresh mode, the operation is not guaranteed.
5. If the SDRAM mode is set through ADMD access in self-refresh mode, the operation is not guaranteed.

(6) SDRAM command

The DRAM controller outputs the following SDRAM commands only.

Command	Abbre.	XSDCKE		XSDCS_N	XRAS_N	XCAS_N	XWE_N	XDQM	Address		
		n-1	n						A11	A10	A9-0
Mode Register Set	MRS	H	H	L	L	L	L	H	Operation code		
CBR (Auto) Refresh	REFH	H	H	L	L	L	H	H	X	X	X
Self-Refresh Start	SELF	H	L	L	L	L	H	H	X	X	X
Self-Refresh End	SREX	L	H	L	H	H	H	H	X	X	X
		L	H	H	H	H	H	H	X	X	X
All Bank Precharge	PALL	H	H	L	L	H	L	H	X	H	X
Bank Active	ACT	H	H	L	L	H	H	H	BA	RA	RA
Write(*2)	WRIT	H	H	L	H	L	L	L	BA	L	CA
Read(*2)	READ	H	H	L	H	L	H	L	BA	L	CA
No Operation	NOP	H	H	L	H	H	H	H	X	X	X
Device Non-selection	DESL	H	H	H	H	H	X	H	X	X	X
Power Down(*1)	—	X	L	H	H	H	X	H	X	X	X
Self-refresh Continue(*1)	—	L	L	L	L	L	X	H	X	X	X

H: High level, L: Low level, x: High level or Low level (Don't care)
BA: Bank address, RA: Row address, CA: Column address
*1: Pseudo command that is defined for convenience of description
*2: The DQM value of the byte lane that is not required goes to a High level (masked).

(7) SDRAM Power Down Mode

For power saving, SDRAM can be switched to the Power Down mode by the DRAM controller. The DRAM controller achieves it by outputting a Low level to CKE when there are no DRAM access requests including refresh.

Transition conditions	Makes a transition when the following conditions are ALL satisfied: -There is no DRAM space access request (including error response), -There is no DRAM register space access request, -There is no distributed CBR refresh execution request, and - The idle period that is set in PDWC has expired.
Return condition	Any of the transition conditions is not satisfied.
Penalty	Return → Overhead at memory access: 3 clocks
Signal output	SDRAM clock signal (SDCLK): Stop output (fixed to L level) Clock Enable signal (CKE): Disabled (fixed to L level)
Caution	After Power-ON Reset, do not change the mode to the SDRAM Power Down mode until the SDRM initialization sequence is executed.

10.3.6 Reference Parameters

The SDRAM parameters that are specified in the DRAMSPEC bit of the DRPC register conform to the following combinations of the operating frequencies and device speeds.

Operating frequency [MHz]	SDRAM				
	PC133	PC125	PC100	PC33	PC66
60.0	○	○	○	○	○
30.0	○	○	○	○	○
15.0	○	○	○	○	○
7.5	○	○	○	○	○

Specify DRAM parameter values based on the numbers of the combinations of the DRAM parameter (tRAS, tRCD, tRP, and so on) values (see the table of DRAMSPEC of the DRPC register). The tRAS, tRCD, and tRP parameters cannot be specified individually.

Lowest operating frequency
At SDRAM connection: 2.56 MHz

(1) SDRAM parameters and operating frequencies (60M/30M/15M/7.5M/3.25M)

Type	Memory parameter [ns]				Cycle count at 15 MHz				Cycle count at 30 MHz				Cycle count at 60 MHz			
PCxx	tRCD	tRAS	tRP	tDPL	tRCD	tRAS	tRP	tDPL	tRCD	tRAS	tRP	tDPL	tRCD	tRAS	tRP	tDPL
133	15	45	15	10	1	2	1	1	1	2	1	1	1	3	1	1
133	20	45	20	10	1	2	1	1	1	2	1	1	2	3	2	1
133	20	45	20	15	1	2	1	1	1	2	1	1	2	3	2	1
125	20	48	20	8	1	2	1	1	1	2	1	1	2	4	2	1
125	20	48	20	10	1	2	1	1	1	2	1	1	2	4	2	1
125	20	50	30	8	1	2	1	1	1	2	1	1	2	4	2	1
125	24	48	24	10	1	2	1	1	1	2	1	1	2	4	2	1
100	20	50	20	10	1	2	1	1	1	2	1	1	2	4	2	1
100	20	50	20	15	1	2	1	1	1	2	1	1	2	4	2	1
100	20	50	20	20	1	2	1	1	1	2	1	1	2	4	2	1
100	30	50	30	15	1	2	1	1	1	2	1	1	2	4	2	1
100	30	60	30	10	1	2	1	1	1	2	1	1	2	4	2	1
100	30	60	30	15	1	2	1	1	1	2	1	1	2	4	2	1
83	35	70	45	24	1	2	1	1	2	3	2	1	3	5	3	2
66	30	60	30	15	1	2	1	1	1	2	1	1	2	4	2	1
66	30	70	30	15	1	2	1	1	1	3	1	1	2	5	2	1

[Note]
The DRAM controller operates under tRC = tRAS + tRP. When using SDRAM under tRC > tRAS + tRP, select the SDRAM access parameter considering tRC.

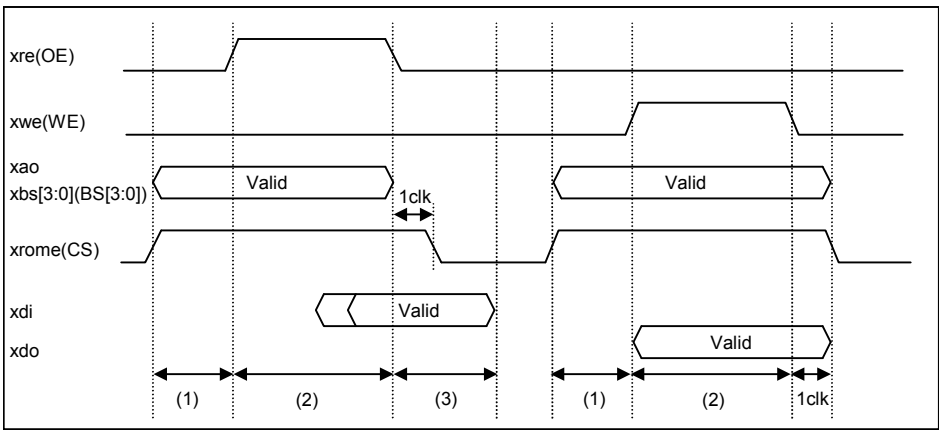
10.3.7 Timing Settings

10.3.7.1 External ROM access function

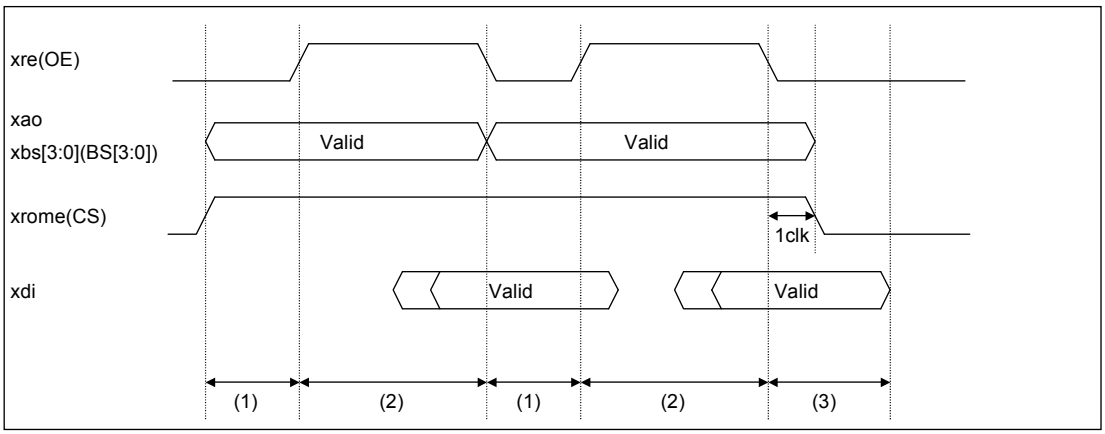
The access timing is settable in order to support various types of ROMs. The access timing is shown in the timing diagram below. (The access timing is determined by the setting value of ROMTYPE[2:0] in ROMAC register. The default setting value is ROMTYPE[2:0] = 111.)

(1) Address setup
(2) OE/WE pulse width
(3) Data-off timing
(Wait period for the ROM output floating delay time from OE deassert)
(4) Burst timing
(Period up to sampling of valid data after verifying the address on a page)

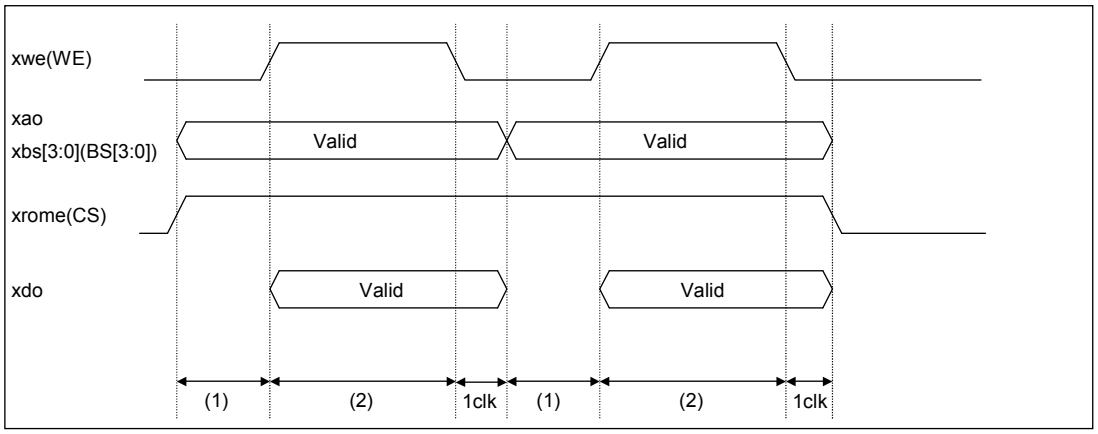
• Read and write cycles



• Successive (*) read cycle

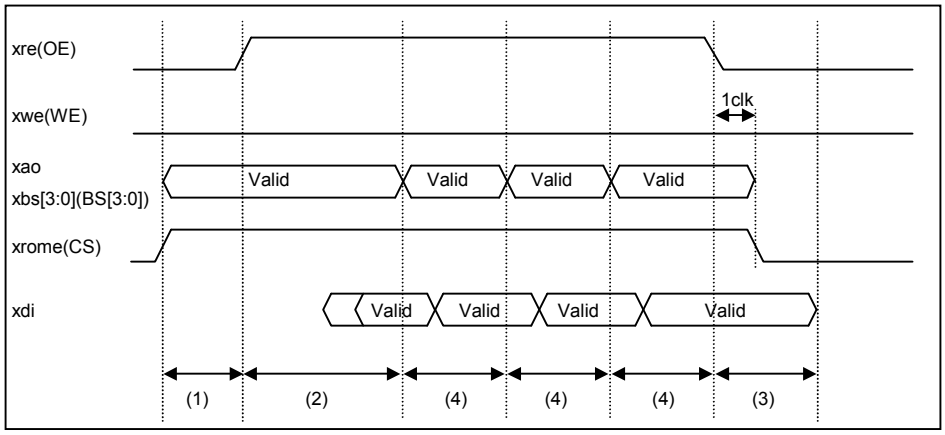


• Successive (*) write cycle



(*) Continuous cycle in the following types of access: (The above drawings show twice continuous accesses.)
- An access in transfer size larger than the ROM area bus width set.
- INCR4/8/16 or WRAP4/8/16 access from AHB master

• Page read cycle



(*) Page read cycles in the following read accesses to the external ROM in addition to setting the ROMAC register to ROMBRST="1". (The drawing when four page addresses are output is shown above.)
- An access in size larger than the ROM area bus width set.
  (SINGLE, INCR, INCR4/8/16, WRAP4/8/16)
- INCR4/8/16 or WRAP4/8/16 access of ROM area bus width of 32 bits and access size of 32 bits.

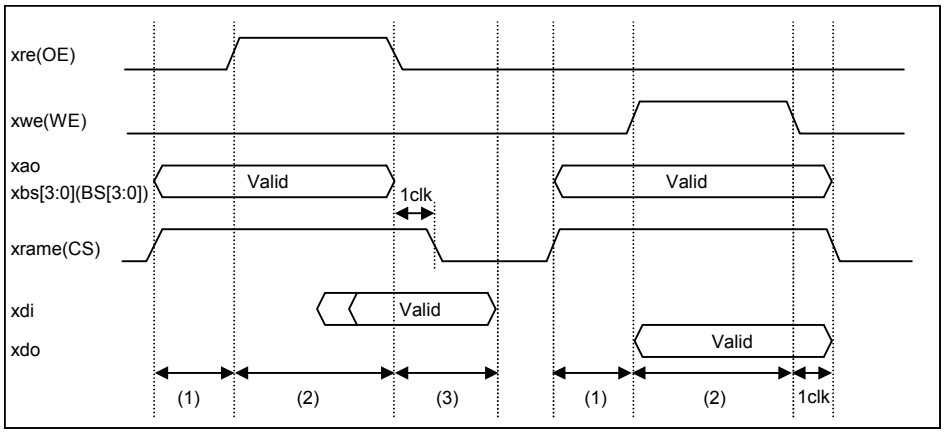
The page address is output for each read access, and one to four page addresses (a maximum of four) are output depending on the location of the start address of the page relative to the boundary between pages. When a read access is not completed in one page read cycle, page read cycles are succeeded and the address hold time between page read cycles is 0 clocks (when xre is deasserted, xao changes simultaneously).

10.3.7.2 External SRAM access function

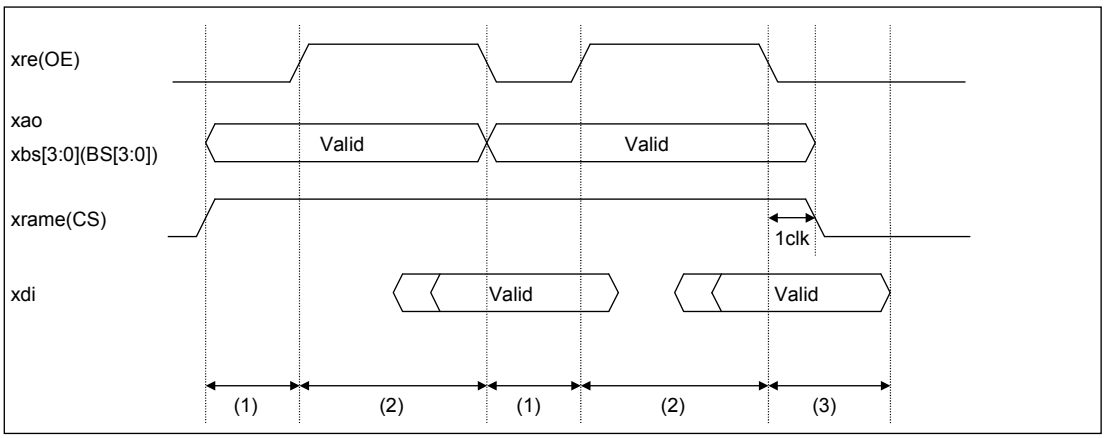
The access timing is settable in order to support various types of SRAMs. The timing diagrams indicating settable parameters are shown below. (The access timing is determined by the setting value of RAMTYPE[2:0] in RAMAC register. The default setting value is RAMTYPE[2:0] = 111.)

(1) Address setup
(2) OE/WE pulse width
(3) Data-off timing
(Wait period for the SRAM output floating delay time from OE deassert.)
(4) Burst timing
(Period up to sampling of valid data after verifying the address on a page.)

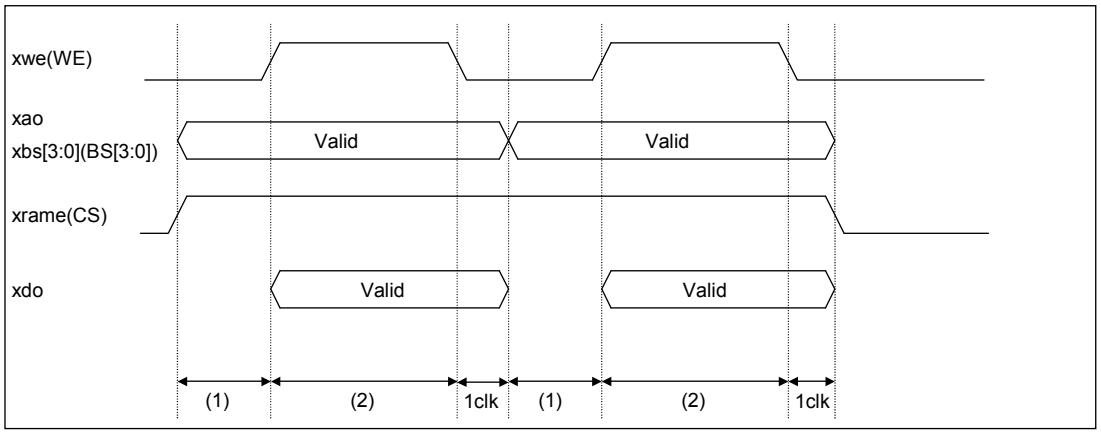
• Read and write cycles



• Successive (*) read cycle

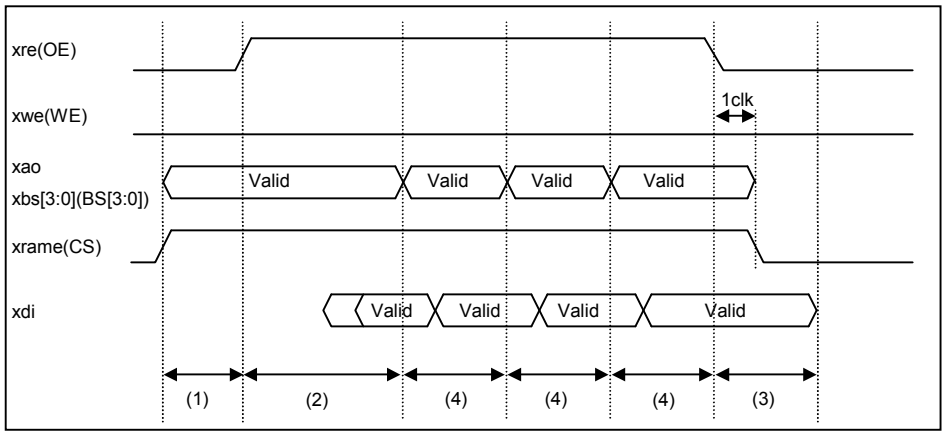


• **Successive (*) write cycle**



(*) Continuous cycles in the following types of access: (The above drawings show twice continuous accesses.)
- An access in transfer size larger than the SRAM area bus width set.
- INCR4/8/16 or WRAP4/8/16 access from AHB master

• **Page read cycle**



(*) Page read cycles in the following read accesses to the external RAM in addition to setting the RAMAC register to RAMBRST="1". (The drawing when four page addresses are output is shown above.)
- An access in size larger than the RAM area bus width set.
  (SINGLE, INCR, INCR4/8/16, WRAP4/8/16)
- INCR4/8/16 or WRAP4/8/16 access of RAM area bus width of 32 bits and access size of 32 bits.

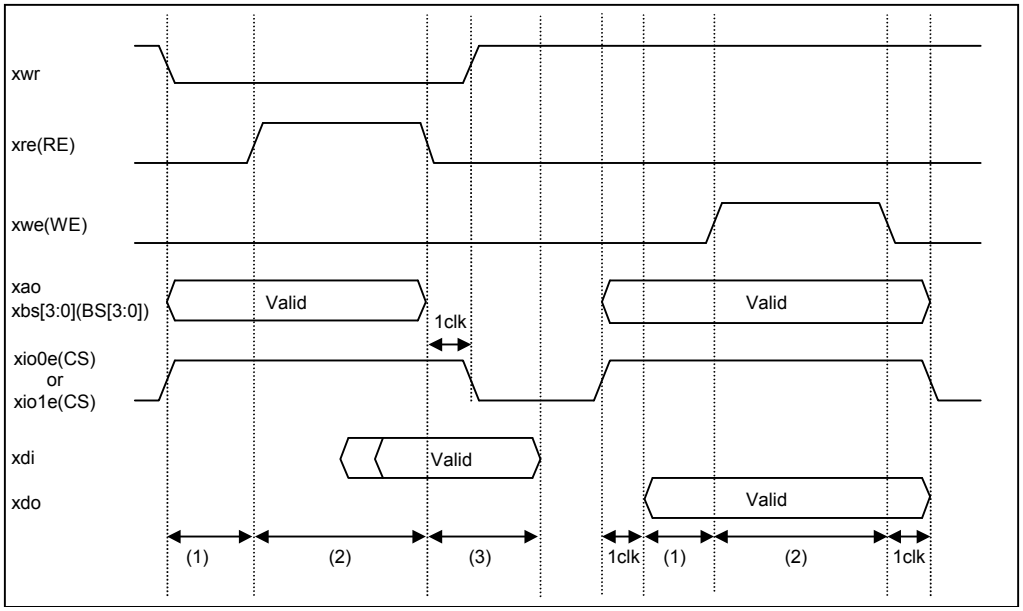
The page address is output for each read access, and one to four page addresses (a maximum of four) are output depending on the location of the start address of the page relative to the boundary between pages. When a read access is not completed in one page read cycle, page read cycles are succeeded and the address hold time between page read cycles is 0 clocks (when xre is deasserted, xao changes simultaneously).

10.3.7.3 External IO access function

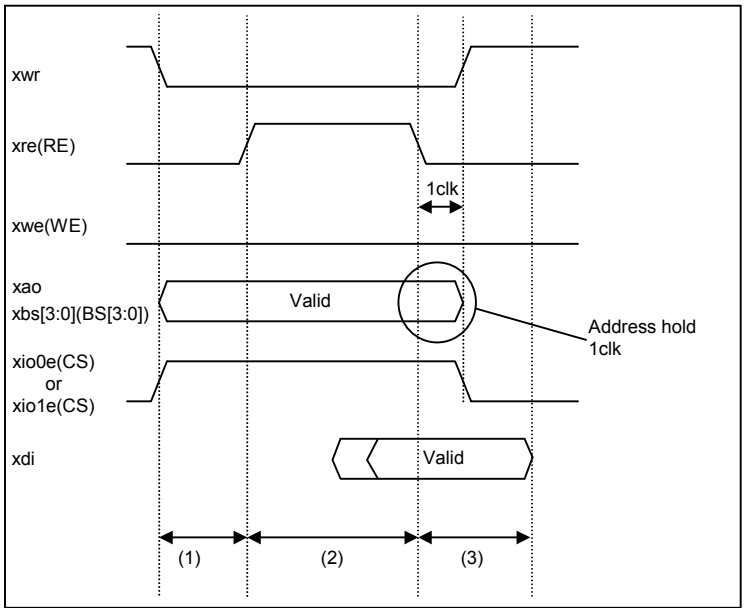
The access timing is settable in order to support various IOs. The timing diagrams indicating settable parameters are shown below. (The access timing is determined by the setting value of IO0TYPE[2:0]/IO1TYPE[2:0] in IO0AC/IO1AC register. The default setting value is IO0TYPE[2:0] = 111 or IO1TYPE[2:0] = 111.)

(1) Address setup
(2) RE/WE pulse width
(3) Data-off timing (Wait period for the IO output floating delay time from RE deassert.)

• Read and write cycles



• Read cycle (Address hold valid cycle)



When reading IO from RAM:

It is possible to ensure address hold time for 1 clock if a read access is performed at a data size below the IO area bus width (8-bit/16-bit/32-bit) set by IOBW0[1:0]/IOBW1[1:0] bit of the bus width control register BWC. On the other hand, if the read access is performed at a data size larger than the IO area, the address hold time is 0 clocks (when xre is deasserted, xao changes simultaneously) because the read access will be continuous.

When reading IO from AHB master (DMA controller, etc.):

In the case of SINGLE (single) transfer or INCR (incrementing burst of unspecified length) transfer

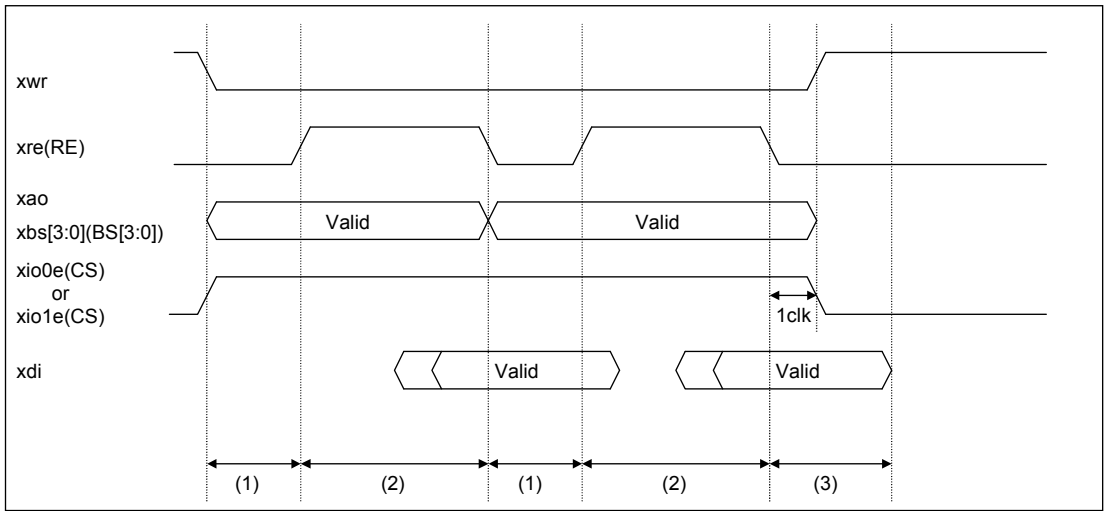
It is possible to ensure the address hold time for 1 clock if an AHB master read access is performed at a data size below the IO area bus width (8-bit/16-bit/32-bit) set by IOBW0[1:0]/IOBW1[1:0] bit in the bus width control register (BWC). On the other hand, if the read access is performed at a data size larger than the IO area, the address hold time is 0 clocks (when xre is deasserted, xao changes simultaneously) because the read access will be continuous.

In the case of INCR4/8/16 (4-beat/8-beat/16-beat incrementing burst) transfer or WRAP4/8/16 (4-beat/8-beat/16-beat wrapping burst) transfer

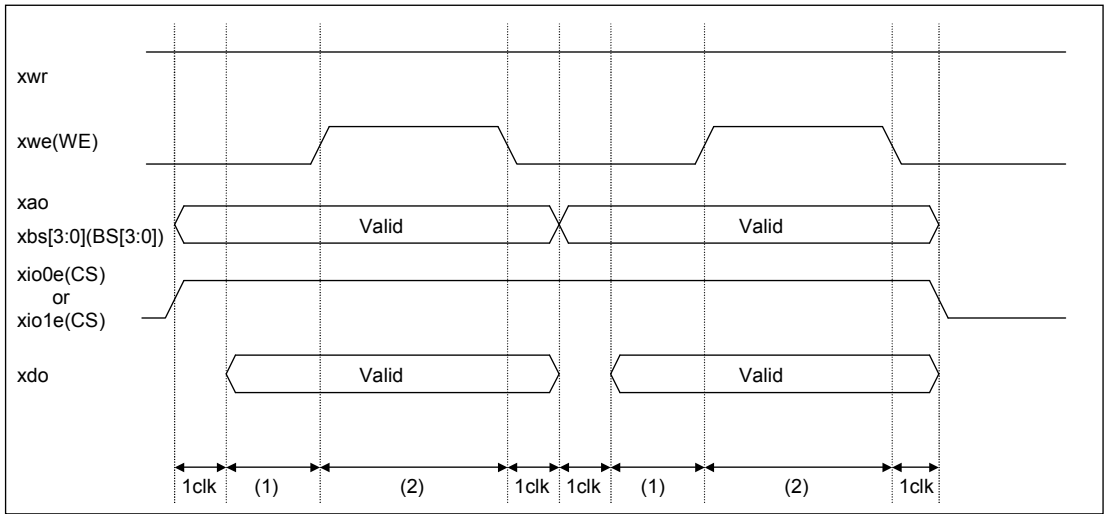
The address hold time is 0 clocks (when xre is deasserted, xao changes simultaneously) if read access (even in any of 32-bit/16-bit/8-bit) is performed from AHB master.

Access master	Access type	Data size at read access	IO area bus width	Address hold time (Clock)
ARM	SINGLE/INCR	8/16/32	32	1
		8/16	16	1
		32	16	0
		8	8	1
		16/32	8	0
	INCR4/8	8/16/32	8/16/32	0
AHB master	SINGLE/INCR	8/16/32	32	1
		8/16	16	1
		32	16	0
		8	8	1
		16/32	8	0
	INCR4/8/16, WRAP4/8/16	8/16/32	8/16/32	0

• **Successive (*) read cycle**



• **Successive (*) write cycle**

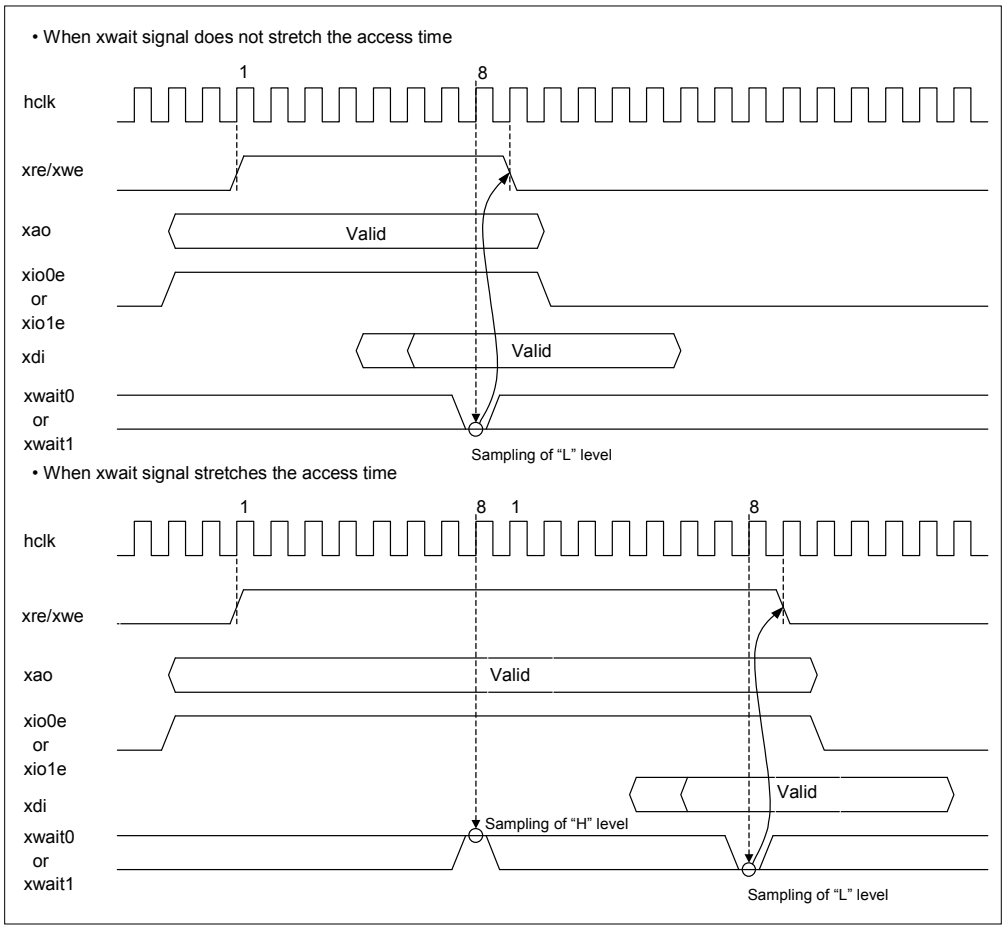


(*) An access will become a successive cycle in the following cases of an access: (The above figure represents 2 successive cycles of an access.)

- An access in transfer size larger than the IO area bus width set.
- INCR4/8/16, WRAP4/8/16 access from AHB master

• Stretching an access time by the external xwait (xwait0, xwait1) signal input

It is possible to stretch the RE/WE pulse width, in the unit of number of clock setting, at the time of access by using the external xwait (xwait0, xwait1) signal input. Whether the access time will be stretched or not is determined by sampling the status of xwait signal at a timing of 1 clock before the RE/WE pulse deasserts (falling edge). If the xwait signal is "H" level, the xwait signal status is again sampled after having further stretched for the number of clock of the RE/WE pulse width. As a result of sampling, the RE/WE pulse width stretching and the xwait signal status sampling are repeated while the xwait signal holds "H" level, and the access will finish at a point of time the xwait signal becomes "L" level as a result of the sampling.
However, if using at the write timing in which the write is performed at the rising edge of WE pulse (xwe), the access time will not be prolonged because the rising edge of WE pulse does not stretch even if xwait signal is used.
Below is illustrated an example in which the RE/WE pulse width has been set to 8 clocks.



10.3.8 Data Off Time Control

To avoid XD data collision, the external memory controller sets Data Off time automatically when external ROM, external SRAM, external I/O bank 0/1, or external DRAM is accessed continuously.

The following table shows the Data Off time that is inserted when external ROM, external SRAM, external I/O bank 0/1, or external DRAM is accessed.

Current cycle			Next cycle									
			ARM				DMA		ARM & DMA			
			ROM		RAM		RAM		IO		SDRAM	
			R	W	R	W	R	W	R	W	R	W
ARM	ROM	Read		○	○	○	○	○	○	○	○	○
		Write										
	RAM	Read	○	○		○	○	○	○	○	○	○
		Write										
	IO	Read	○	○	○	○	○	○		○	○	○
		Write										
DMA	RAM	Read	○	○	○	○		○	○	○	○	○
		Write										
	IO	Read	○	○	○	○	○	○		○	○	○
		Write										
ARM & DMA	SDRAM	Read										
		Write										

[Note]
Set the Data Off time of external ROM, external RAM, and external I/O bank 0/1 using the ROMAC, RAMAC, IO0AC, and IO1AC registers. Data Off time cannot be set for SDRAM.

10.4 Access Timing Examples

10.4.1 Accessing External Devices

(1) Accessing external ROM/RAM

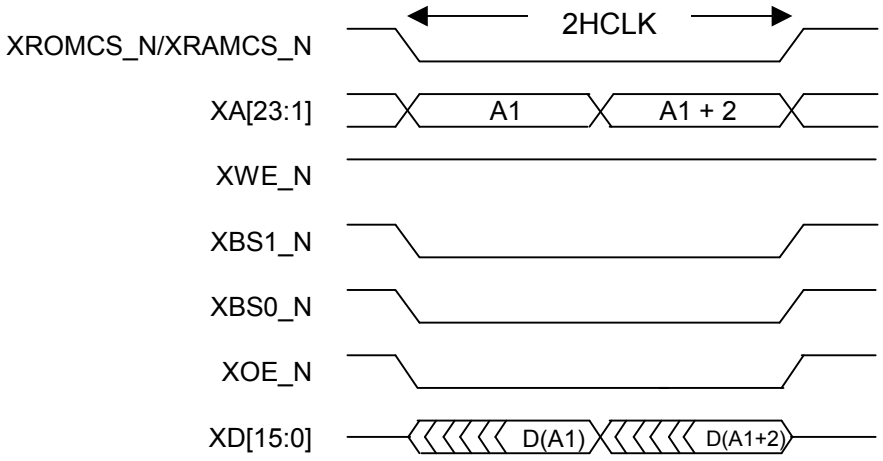


Figure 10-1 ROM/RAM, Word Read, OE/WE Pulse Width = 1 (minimum setting)

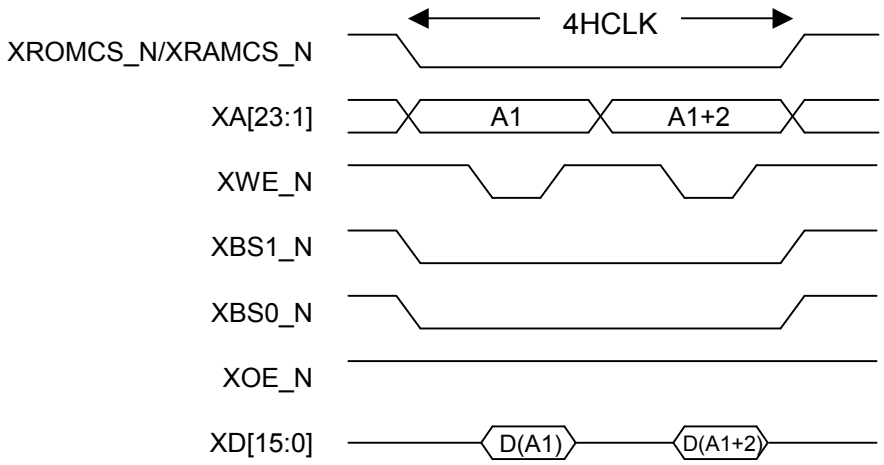


Figure 10-2 ROM/RAM, Word Write, OE/WE Pulse Width = 1 (minimum setting)

(2) Accessing the external I/O bank

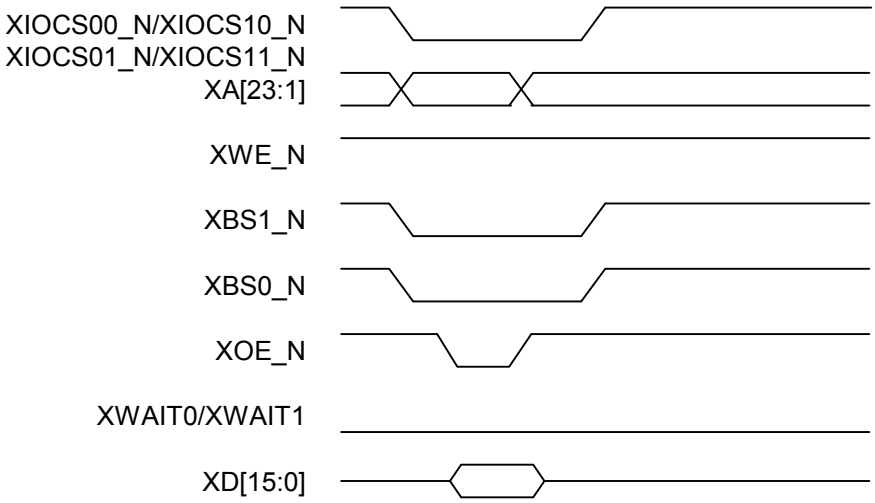


Figure 10-3 External I/O Bank, 16-bit Bus Width, Half-Word Read, OE/WE Pulse Width = 1 (minimum setting), Without wait by XWAIT

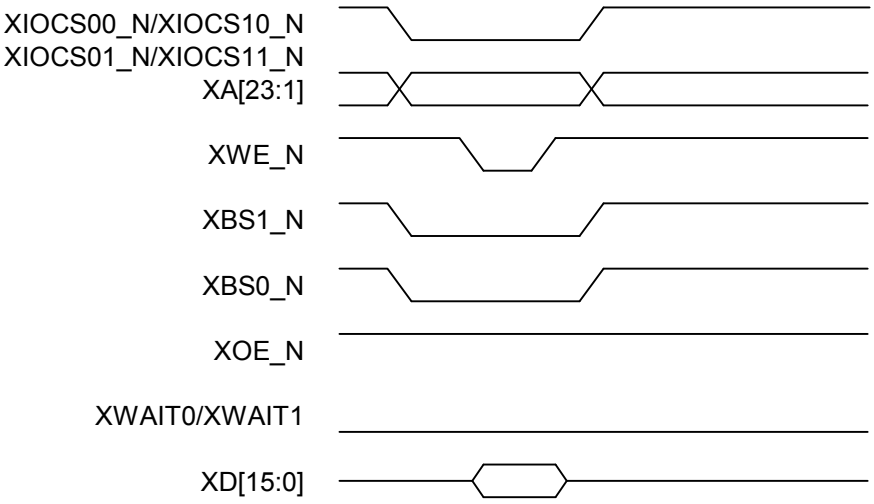


Figure 10-4 External I/O Bank, 16-bit Bus Width, Half-Word Write, OE/WE Pulse Width = 1 (minimum setting), Without wait by XWAIT

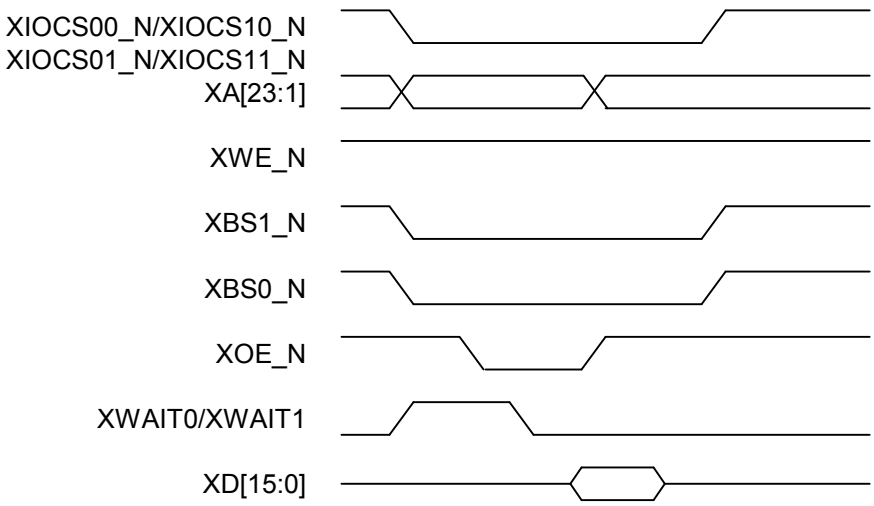


Figure 10-5 External I/O Bank, 16-bit Bus Width, Half-Word Read, OE/WE Pulse Width =1 (minimum setting), With Wait by XWAIT

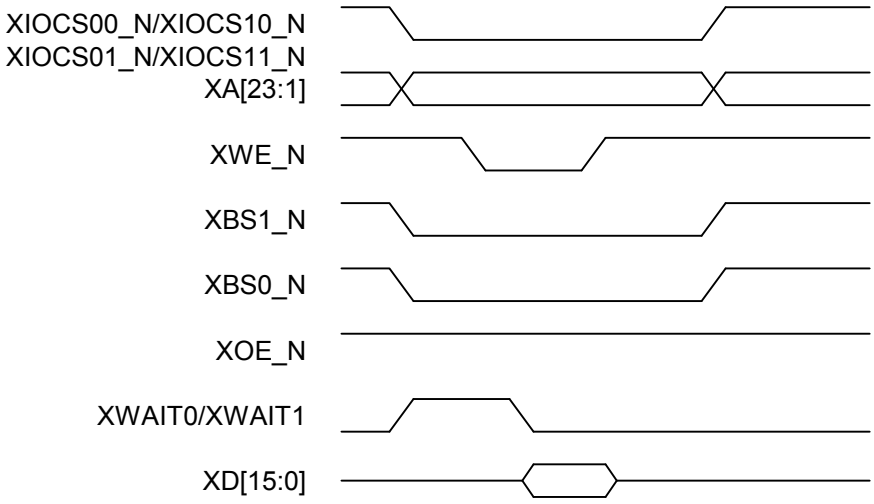


Figure 10-6 External I/O Bank, 16-bit Bus Width, Half-Word Write, OE/WE Pulse Width =1 (minimum setting), With Wait by XWAIT

(3) Accessing SDRAM

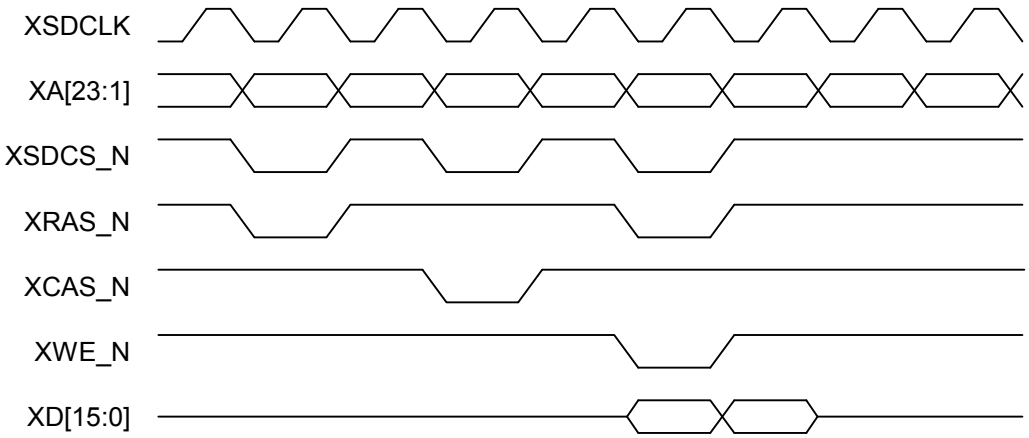


Figure 10-7 SDRAM Word Read

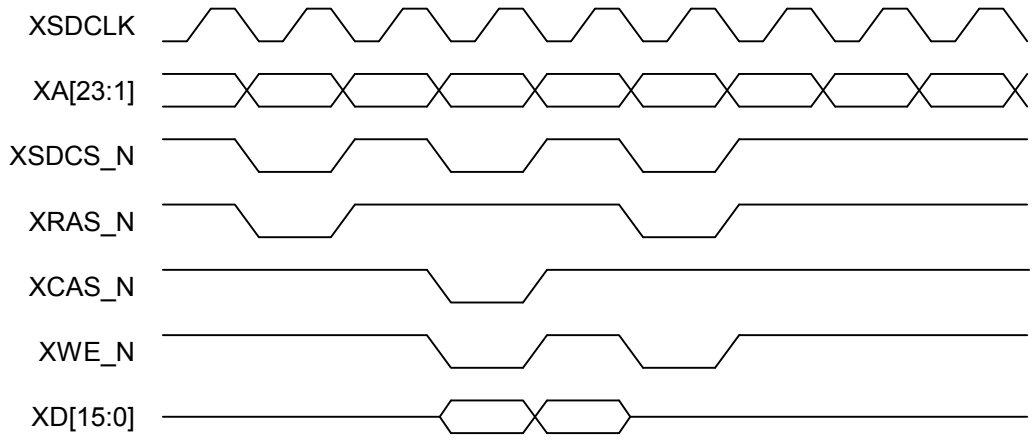


Figure 10-8 SDRAM Word Write

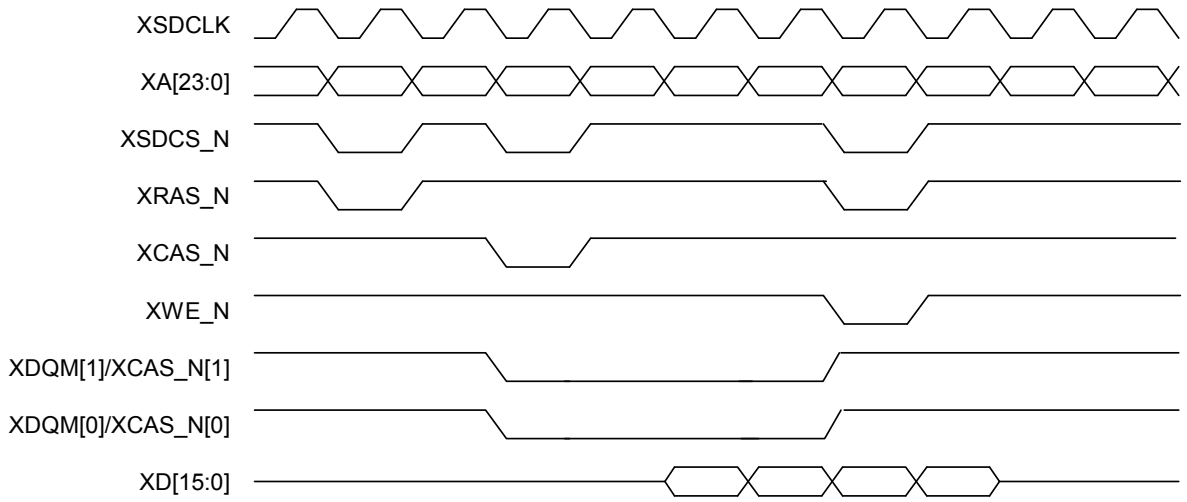


Figure 10-9 SDRAM 2-Word Burst Read

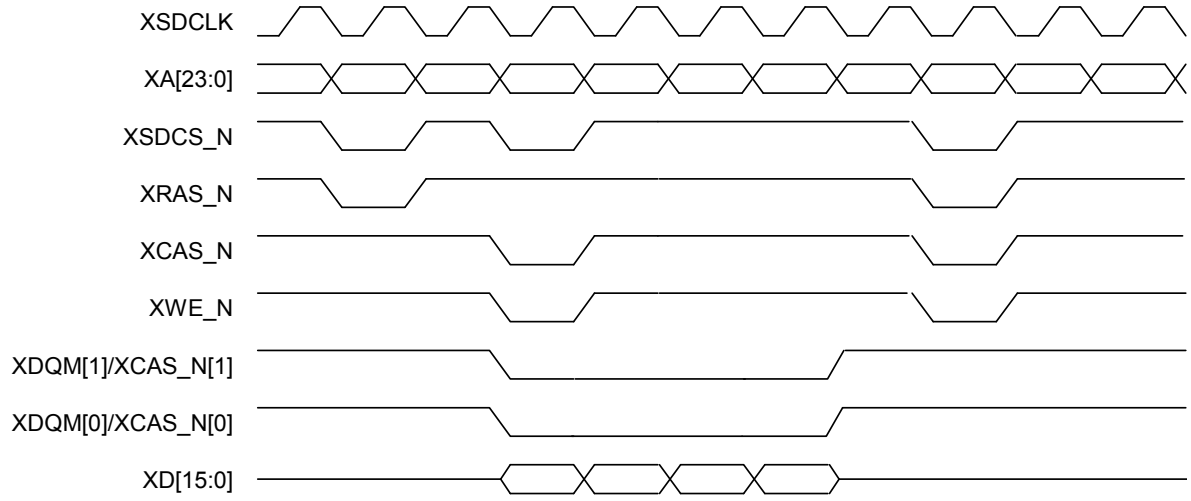


Figure 10-10 SDRAM 2-Word Burst Write

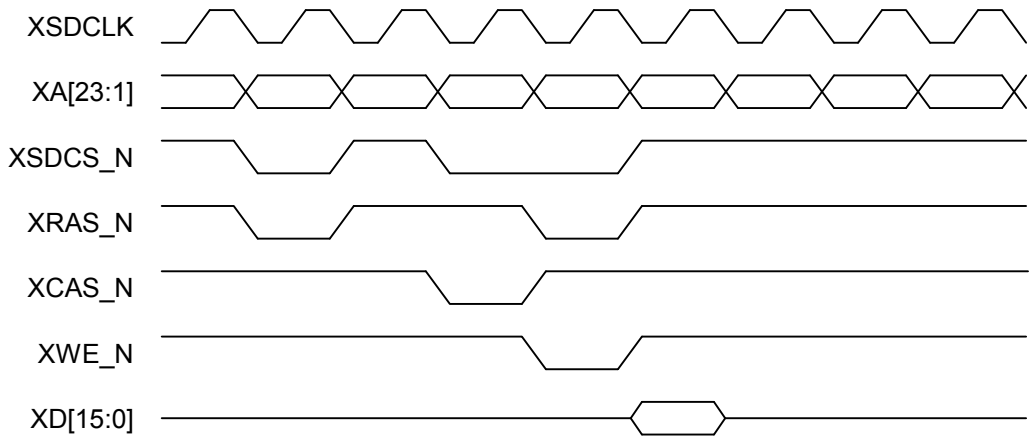


Figure 10-11 SDRAM Half-Word Read

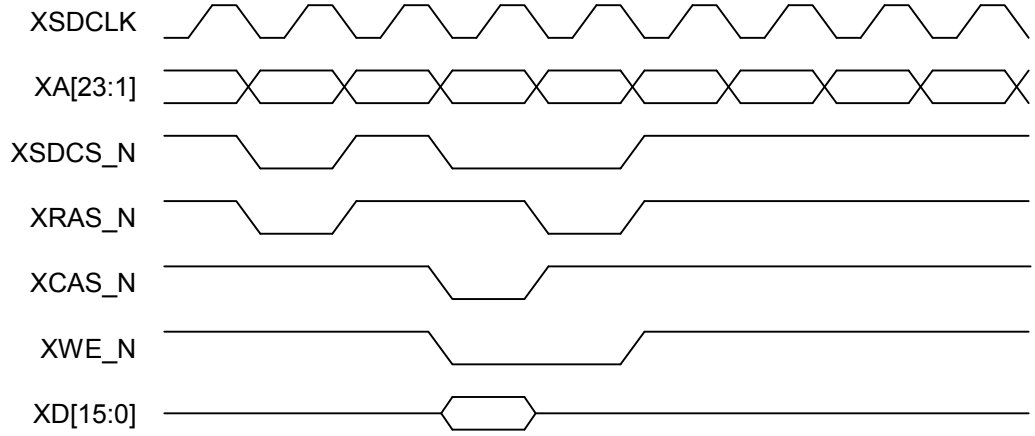


Figure 10-12 SDRAM Half-Word Write

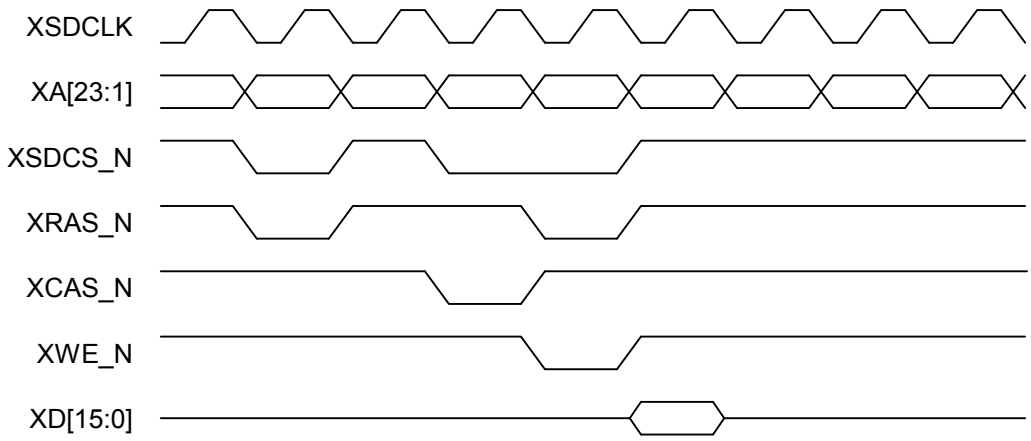


Figure 10-13 SDRAM Byte Read (LSB)

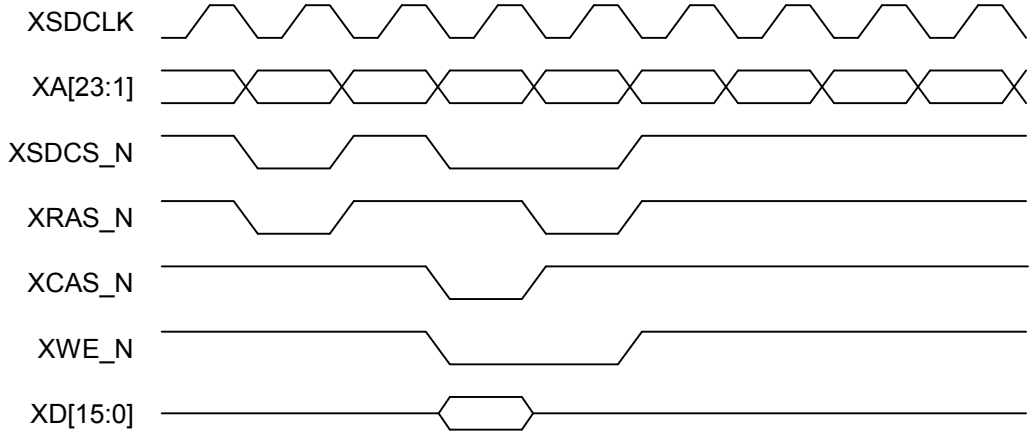


Figure 10-14 SDRAM Byte Write (LSB)

10.5 Power Management of DRAM

To use the STANDBY mode or the HALT mode, the DRAM controller must be controlled by a program. See Chapter 7, Power Management, for details.

10.6 Precautions in Using SDRAM with Self-Refresh Setting

While setting SDRAM to self-refresh mode, a memory access may occur unintentionally by software and valid data might be lost. This problem will occur under the following conditions if the attribute of the memory region of the external memory controller is set to WB (Write Back) or NCB (Non Cacheable, Bufferable).

Case 1:
Case of data write access to the relevant memory region during and directly before setting of SDRAM to self-refresh mode or case of data read access to arbitrary cacheable memory regions.
When the memory controller goes to the memory access disable mode, the ARM processor might not have completed draining the write buffer for the relevant memory region. When the write access is executed after the memory controller disables the memory access (*1), valid data of the write buffer will be lost.
A cache read miss of the ARM processor occurs, giving a possibility of a write back to the relevant memory while the memory controller is disabling the memory access. Since this write access is not done correctly (*2), valid data of the relevant memory region will be lost if it is located only in the cache.

*1: Since the write to the relevant memory region (Bufferable attribute) is done through the ARM processor write buffer, the write of the mode setup registers (Non Bufferable attribute) might be done first. Beware the access order might be reversed even if writes to the relevant memory region were to be executed first by software.

*2: For the write back access of the ARM processor, AHB error response will be ignored by the ARM processor even if it is returned by the memory controller. Note that, in this case, the ARM processor will not perform a write back again, and not go to the abort exception routine.

Case 2:
Case of data write access to a relevant memory region directly before setting SDRAM to self-refresh mode, having the relevant memory region set to NCB (Non Cacheable, Bufferable)
The operation in the above case is the same as the data write access to the relevant memory regions in Case 1. In this case, no malfunction occurs when data read access to arbitrary memory region is executed.

Corrections for Cases 1 and 2:
Execute the Drain Write Buffer instruction directly before the memory controller goes to the memory access disable mode, and do not perform data access for arbitrary memory regions set on cacheable areas while the memory controller is disabling the memory access.
Described below is a control sequence example for setting SDRAM to self-refresh mode during CPUGHALT mode. Do not perform data access to arbitrary memory regions set on cacheable areas during the SDRAM self-refresh mode

[Transition to the CPUGHALT mode]
① Mask the ARM processor interrupt.
② Execute the Drain Write Buffer instruction.
③ Enter the SDRAM self-refresh mode.
④ Set the CPUGHALT mode.
⑤ Execute the WFI instruction.
↓
(CPUGHALT mode)
↓
[Recovery from the CPUGHALT mode]
⑥ Recover with a recovery source (Interrupt or debug request).
⑦ End the SDRAM self-refresh mode.
⑧ Release the ARM processor interrupt mask.
↓
(Servicing the interrupt handler)

Do not perform data access to cacheable areas during ② to ⑦.

10.7 Examples of Connection with External Memory

The table below shows examples of the connection with each device.
At actual device connection, refer to the data sheet of the device also since there are some differences such as signal names.
When many devices are to be connected, use buffer devices considering the driver capacity and load. Connect a pull-up resistor externally to XD.

Pin name	I/O	Function	ROM	SRAM	I/O (Bank0)	I/O (Bank1)	SDRAM
XA23 to XA01	O	External address bus	○	○	○	○	○
XD15 to XD00	I/O	External data bus	○	○	○	○	○
XROMCS_N	O	External ROM chip select	○				
XRAMCS_N	O	External RAM chip select		○			
XIOCS00_N, XIOCS01_N	O	I/O bank 0 chip select			○		
XIOCS10_N, XIOCS11_N	O	I/O bank 1 chip select				○	
XOE_N	O	Output enable	○	○	○	○	
XWE_N	O	Write enable	○	○	○	○	○
XBS1_N, XBS0_N	O	External memory byte select	○	○	○	○	
XSDCS_N	O	SDRAM chip select					○
XSDCLK	O	Clock to SDRAM					○
XSDCKE	O	Clock enable to SDRAM					○
XCAS_N	O	Column address strobe of SDRAM					○
XRAS_N	O	Low address strobe					○
XDQM1	O	SDRAM data input/output mask					○
XDQM0	O	SDRAM data input/output mask					○
XWAIT1	I	I/O bank 0 WAIT signal				○	
XWAIT0	I	I/O bank 1 WAIT signal			○		
XSYSCLK	O	External bus AHB clock					

10.7.1 Connection of ROM

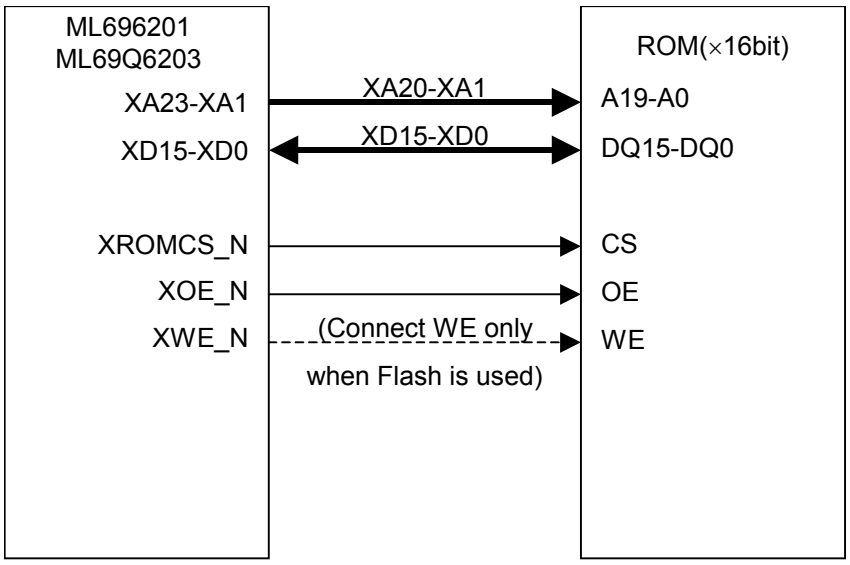


Figure 10-15 Connecting 1M × 16-bit (2-Mbyte) ROM

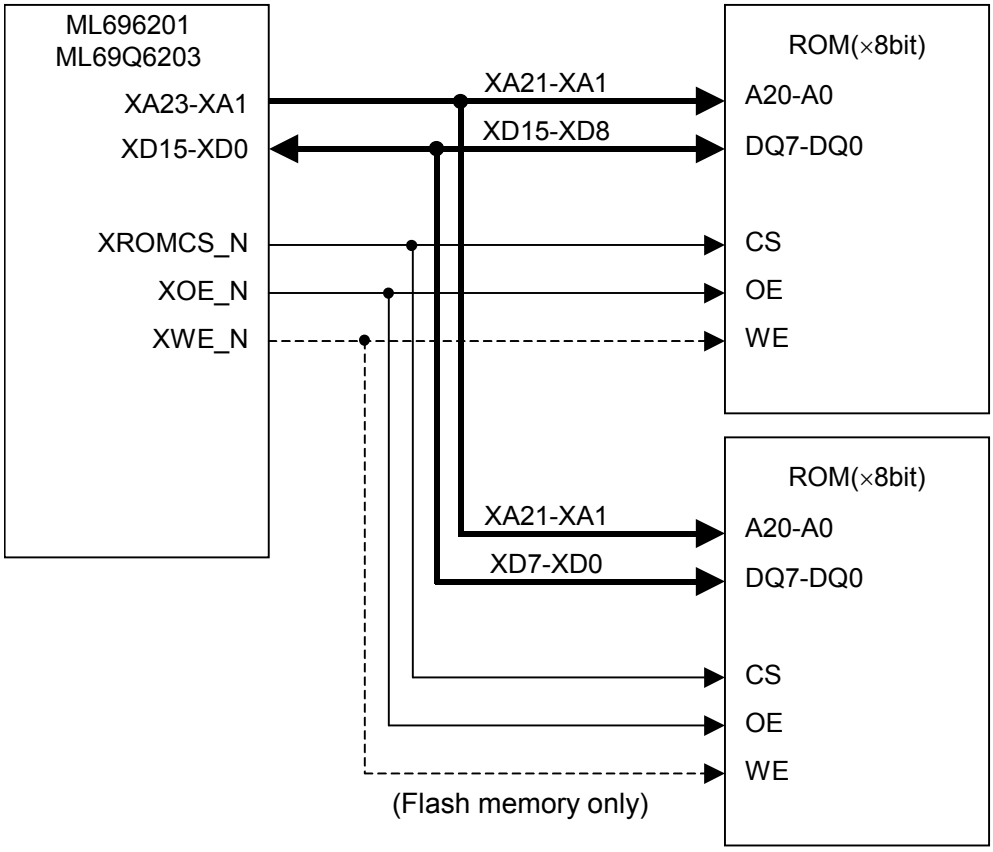


Figure 10-16 Connecting Two 4M × 8-bit (4-Mbyte) ROMs

10.7.2 Connection of SRAM

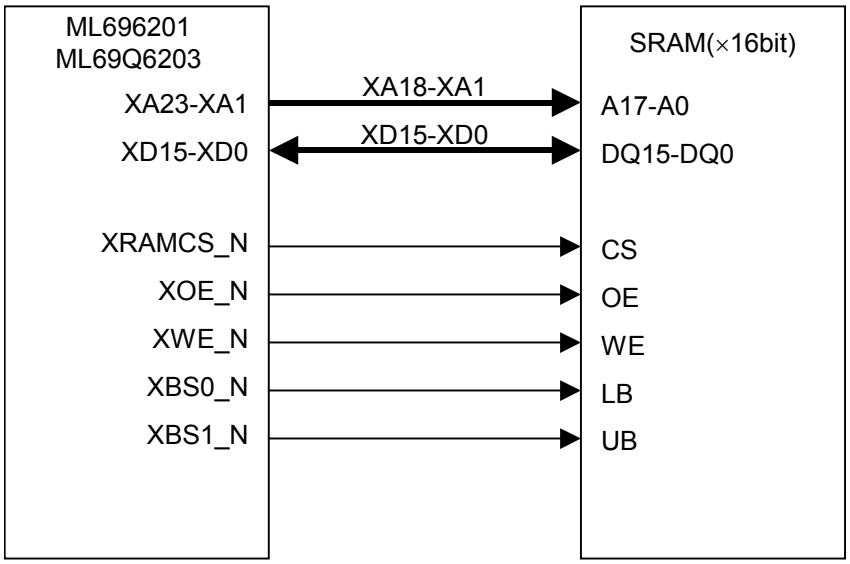


Figure 10-17 Connecting 256K × 16-bit (512-Kbyte) SRAM (byte select type)

10.7.3 Connection of SDRAM

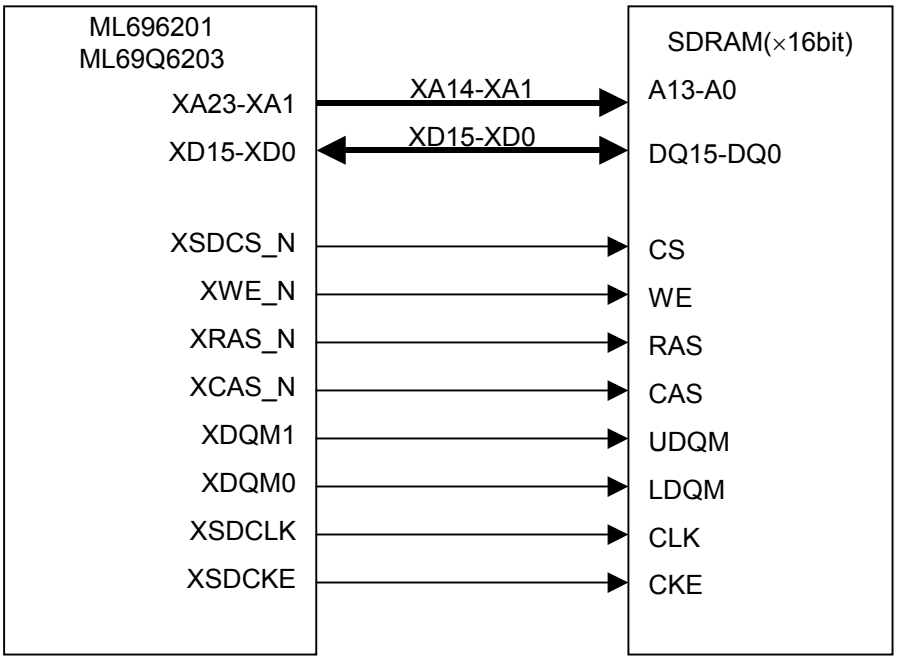


Figure 10-18 Connecting 4-bank × 1M × 16-bit (8-Mbyte) SDRAM

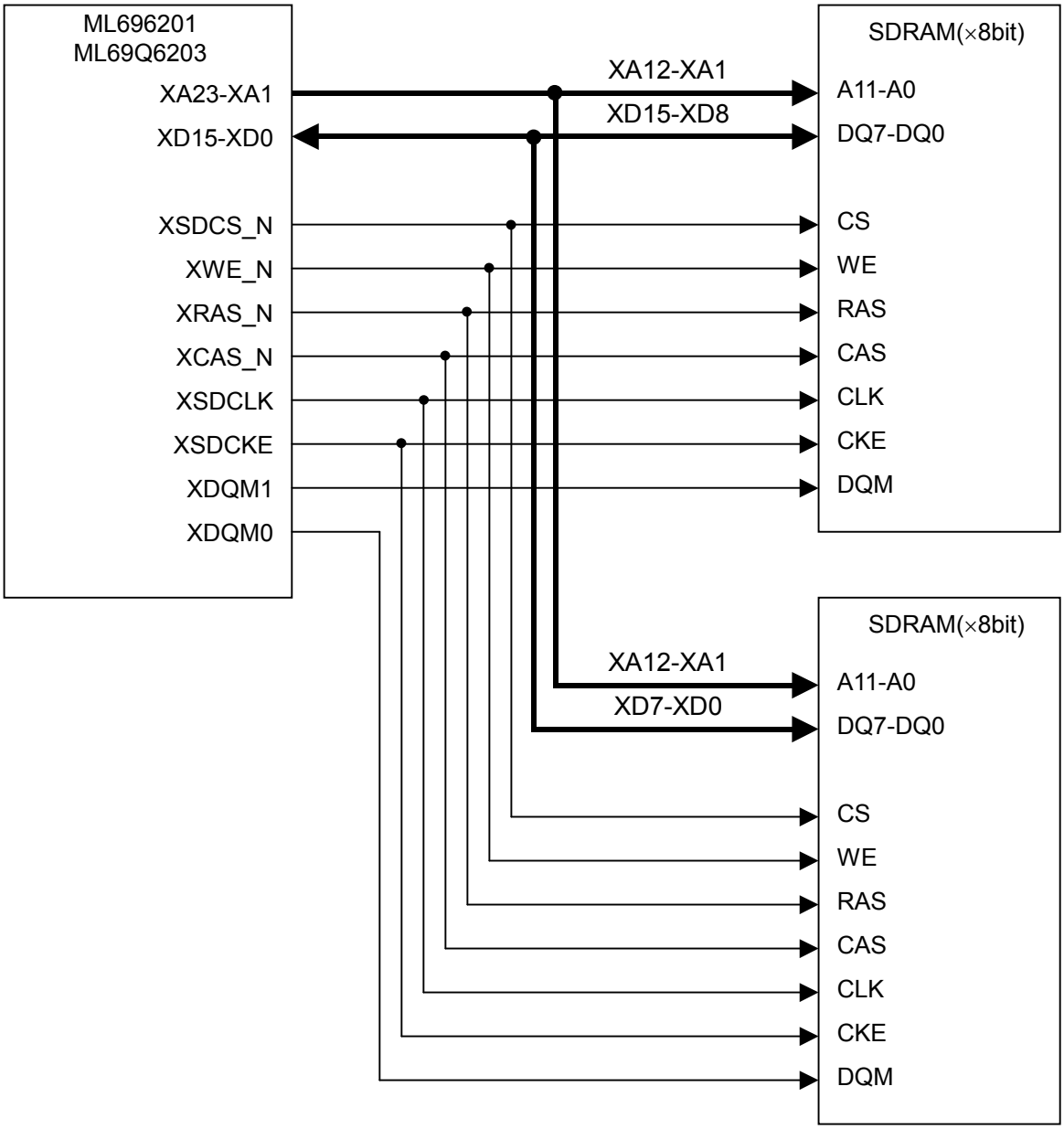


Figure 10-19 Connecting 4-bank × 1M × 8-bit (4-Mbyte) SDRAM

Chapter 11

Direct Memory Access Controller

Chapter 11 Direct Memory Access Controller

11.1 Overview

This LSI incorporates a 4-channel direct memory access controller (DMAC)
This DMAC, instead of the CPU, can perform data transfer between a memory and another memory, a I/O and a memory, and an I/O and another I/O, thereby reducing the load on the CPU and improving the operating efficiency of the LSI.

Features:

- Number of channels: 4
- Channel priority order
  - Fixed mode (Channel priority is permanently fixed to CH0 > CH1 > CH2 > CH3.)
  - Round Robin mode (Lowest priority is given to the channel that receives a transfer request.)
- Maximum number of transfers: 65536 times
- Data transfer size: Byte (8 bits)/Half-word (16 bits)/word (32 bits)
- Dual address access (Data Read from a transfer source and Data Write to a transfer destination are executed individually.)
- Bus authorization request mode
  - Cycle steal mode (releases the bus authorization every 1 time of DMA transfer)
  - Burst mode (does not release the bus authorization until the completion of transfer for the specified number of transfers)
- DMA transfer request
  - Auto request (generates transfer requests within DMAC until the completion of transfer for the specified number of transfers)
  - External request (accepts an external request by the DREQ signal sent from each channel; Rising edge of the DREQ signal is detected)
- Interrupt request: DMAC generates an interrupt request to the CPU at the end of DMA transfer for the specified number of transfers or after the occurrence of an error.
- An interrupt request signal is output separately for each channel.
- An interrupt request signal can mask the output in each channel.
- The error that is detected by DMAC indicates that "the specified address does not have the transfer destination or transfer source".

11.1.1 Configuration

Figure 11-1 shows the configuration of DMAC and Figure 11-2 shows the peripheral configuration of DMAC.

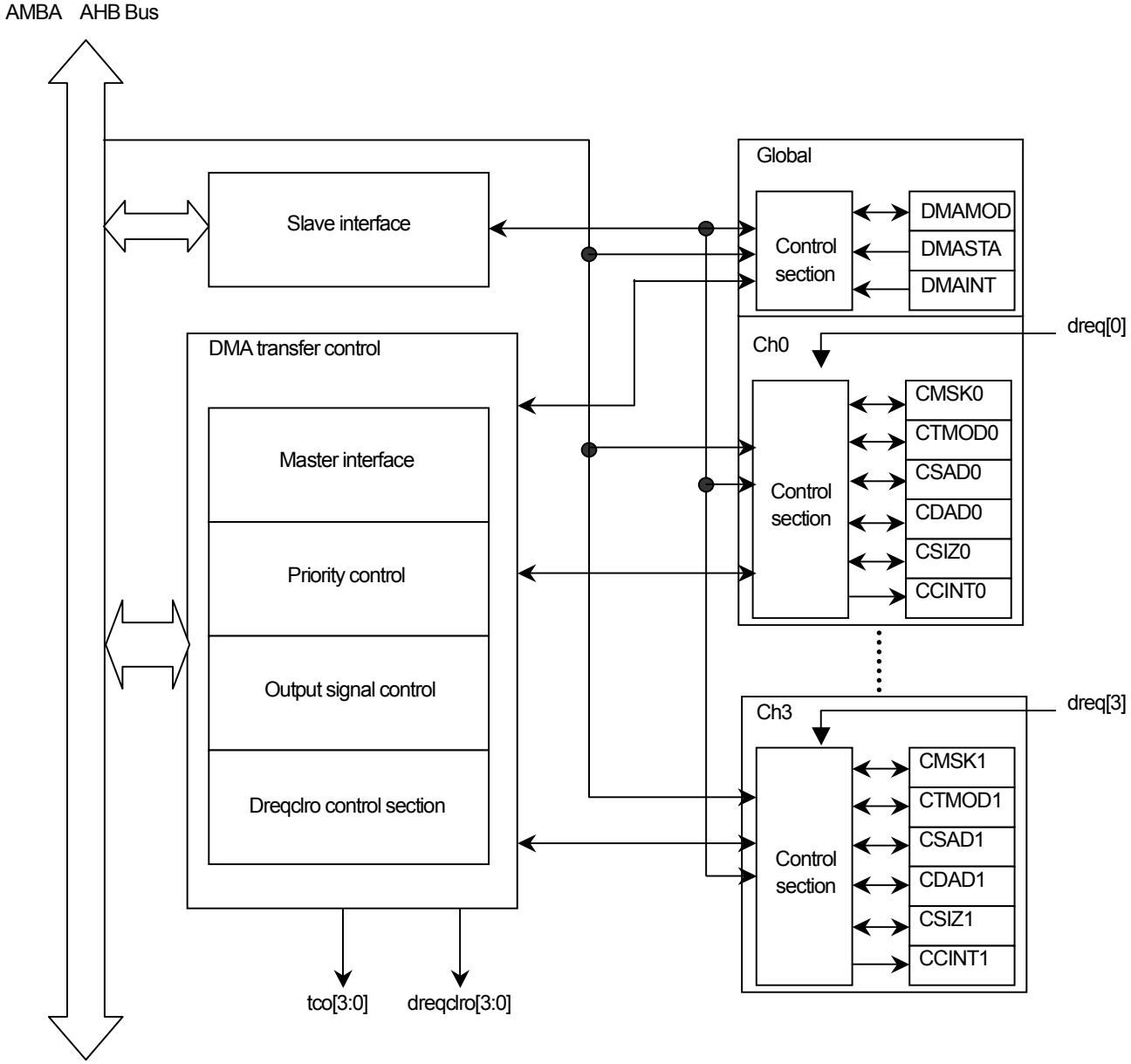


Figure 11-1 DMAC Block Diagram

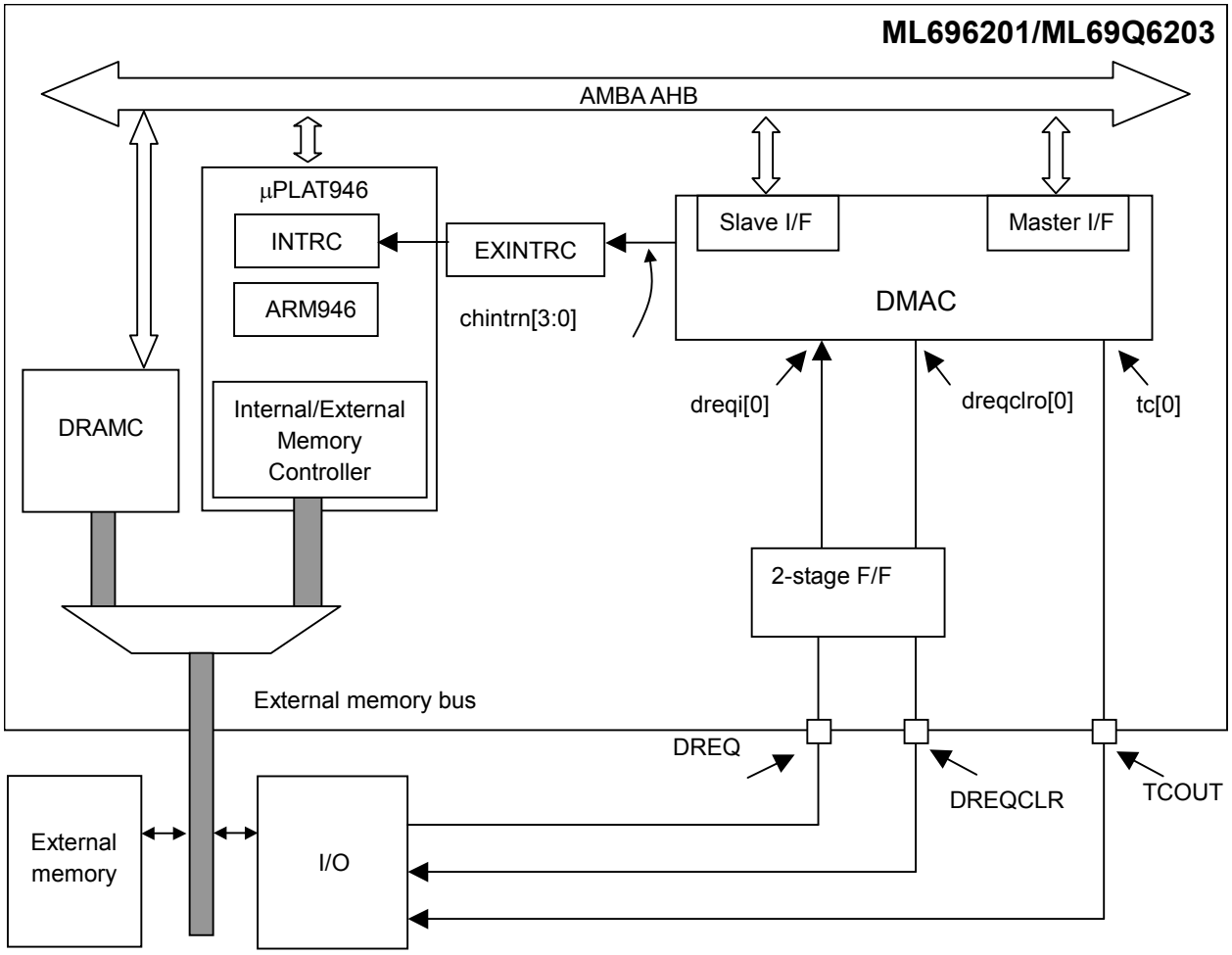


Figure 11-2 DMAC Peripheral Block Diagram

(a) AHB interface
AMBA AHB is connected with 32-bit data width. DMAC is equipped with slave and master interfaces.

- Slave interface
  Used when the processor accesses the DMAC register (DMAC Global/Channel).
- Master interface
  Used at DMA transfer.

11.1.2 List of Pins

Pin name	I/O	Function
DREQ	I	DMA request signal
DREQCLR	O	DREQ clear request signal
TCOUT	O	DMA final transfer notification signal

11.1.3 List of Registers

CH	Address	Name	Symbol	R/W	Size	Initial value
Com-mon	0x7BE0_0000	DMA mode register	DMAMOD	R/W	32	0x0000_0000
	0x7BE0_0004	DMA status register	DMASTA	R	32	0x0000_0000
	0x7BE0_0008	DMA termination status register	DMAINT	R	32	0x0000_0000
Channel 0	0x7BE0_0100	DMA channel mask register	DMACMSK0	R/W	32	0x0000_0001
	0x7BE0_0104	DMA transfer mode register	DMACTMOD0	R/W	32	0x0000_0040
	0x7BE0_0108	DMA transfer source address register	DMACSAD0	R/W	32	0x0000_0000
	0x7BE0_010C	DMA transfer destination address register	DMACDAD0	R/W	32	0x0000_0000
	0x7BE0_0110	DMA transfer count register	DMACSIZ0	R/W	32	0x0000_0000
	0x7BE0_0114	DMA termination status clear register	DMACCINT0	W	32	—
Channel 1	0x7BE0_0200	DMA channel mask register	DMACMSK1	R/W	32	0x0000_0001
	0x7BE0_0204	DMA transfer mode register	DMACTMOD1	R/W	32	0x0000_0040
	0x7BE0_0208	DMA transfer source address register	DMACSAD1	R/W	32	0x0000_0000
	0x7BE0_020C	DMA transfer destination address register	DMACDAD1	R/W	32	0x0000_0000
	0x7BE0_0210	DMA transfer count register	DMACSIZ1	R/W	32	0x0000_0000
	0x7BE0_0214	DMA termination status clear register	DMACCINT1	W	32	—
Channel 2	0x7BE0_0300	DMA channel mask register	DMACMSK2	R/W	32	0x0000_0001
	0x7BE0_0304	DMA transfer mode register	DMACTMOD2	R/W	32	0x0000_0040
	0x7BE0_0308	DMA transfer source address register	DMACSAD2	R/W	32	0x0000_0000
	0x7BE0_030C	DMA transfer destination address register	DMACDAD2	R/W	32	0x0000_0000
	0x7BE0_0310	DMA transfer count register	DMACSIZ2	R/W	32	0x0000_0000
	0x7BE0_0314	DMA termination status clear register	DMACCINT2	W	32	—
Channel 3	0x7BE0_0400	DMA channel mask register	DMACMSK3	R/W	32	0x0000_0001
	0x7BE0_0404	DMA transfer mode register	DMACTMOD3	R/W	32	0x0000_0040
	0x7BE0_0408	DMA transfer source address register	DMACSAD3	R/W	32	0x0000_0000
	0x7BE0_040C	DMA transfer destination address register	DMACDAD3	R/W	32	0x0000_0000
	0x7BE0_0410	DMA transfer count register	DMACSIZ3	R/W	32	0x0000_0000
	0x7BE0_0414	DMA termination status clear register	DMACCINT3	W	32	—

[Note]
Access all the registers in word units (32 bits). If accessed in byte units (8 bits) or Half-word units (16 bits), the operation is not guaranteed.

11.2 Registers

11.2.1 DMA Mode Register (DMAMOD)

The DMAMOD register is used for setting the priority sequence of a DMA channel.
Read/Write operation is enabled for the DMAMOD register through program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAMOD	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	PRI
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BE00000
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **PRI** (Bit 0)
  This bit is used for setting the DMA channel priority.

PRI	Description
0	Fixed (CH0 .> CH1 > CH2 > CH3)
1	Round Robin (the channel that was used last is given the lowest priority.)

11.2.2 DMA Status Register (DMASTA)

The DMASTA register indicates the status of DMA transfer.
When a value other than "0" is set in the transfer count register (CSIZ), the corresponding channel bit is set to "1". And the bit is reset to "0" upon normal end of DMA transfer for the specified number of transfers or if the DMA transfer failed in the middle.
Only Read operation is enabled for the DMASTA register through program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASTA	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASTA	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	STA3	STA2	STA1	STA0
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BE00004
Access: R
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read.

[Description of Bits]

- **STA0** (Bit 0)
  This bit indicates the transfer status of CH0.

STA0	Description
0	No CH0 untransferred data, or abnormal end occurred during transfer
1	Untransferred CH0 data exists

- STA1 (Bit 1)
  This bit indicates the transfer status of CH1.

STA1	Description
0	No CH1 untransferred data, or abnormal end occurred during transfer
1	Untransferred CH1 data exists

- STA2 (Bit 2)
  This bit indicates the transfer status of CH2.

STA2	Description
0	No CH2 untransferred data, or abnormal end occurred during transfer
1	Untransferred CH2 data exists

- STA3 (Bit 3)
  This bit indicates the transfer status of CH3.

STA3	Description
0	No CH3 untransferred data, or abnormal end occurred during transfer
1	Untransferred CH3 data exists

11.2.3 DMA Termination Status Register (DMAINT)

The DMAINT register indicates the interrupt request channel at the end of DMA transfer, its source (normal end/abnormal end), and cycle of abnormal end (Read cycle of the transfer source/Write cycle of the transfer destination).
If arbitrary data is written to the interrupt clear registers (DMACCINT0, DMACCINT1, DMACCINT2, and DMACCINT3) that are allocated to each channel, IREQ, ISTA, and ISTP in respective channel are cleared to "0".
Only Read operation is enabled for the DMAINT register through program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAINT	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	ISTP3	ISTP2	ISTP1	ISTP0
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	ISTA3	ISTA2	ISTA1	ISTA0	—*	—*	—*	—*	IREQ3	IREQ2	IREQ1	IREQ0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BE00008
Access: R
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read.
Each status bit can only be cleared by a program.

[Description of Bits]

- **IREQ0** (Bit 0)
  This bit indicates the DMA termination status of CH0.

IREQ0	Description
0	DMA transfer of CH0 has not terminated or has not been activated.
1	DMA transfer of CH0 has terminated.

- **IREQ1** (Bit 1)
  This bit indicates the DMA termination status of CH1.

IREQ1	Description
0	DMA transfer of CH1 has not terminated or has not been activated.
1	DMA transfer of CH1 has terminated.

- **IREQ2** (Bit 2)
  This bit indicates the DMA termination status of CH2.

IREQ2	Description
0	DMA transfer of CH2 has not terminated or has not been activated.
1	DMA transfer of CH2 has terminated.

- **IREQ3** (Bit 3)
  This bit indicates the DMA termination status of CH3.

IREQ3	Description
0	DMA transfer of CH3 has not terminated or has not been activated.
1	DMA transfer of CH3 has terminated.

- **ISTA0** (Bit 8)
  This bit indicates the termination status of CH0.

ISTA0	Description
0	CH0 terminated normally.
1	CH0 terminated abnormally.

- **ISTA1** (Bit 9)
  This bit indicates the termination status of CH1.

ISTA1	Description
0	CH1 terminated normally.
1	CH1 terminated abnormally.

- **ISTA2** (Bit 10)
  This bit indicates the termination status of CH2.

ISTA2	Description
0	CH2 terminated normally.
1	CH2 terminated abnormally.

- **ISTA3** (Bit 11)
  This bit indicates the termination status of CH3.

ISTA3	Description
0	CH3 terminated normally.
1	CH3 terminated abnormally.

- **ISTP0** (Bit 16)
  This bit indicates the abnormal end occurrence status of CH0. This bit is valid only when ISTA0 = 1.

ISTP0	Description
0	CH0 terminated abnormally at the Read cycle of the transfer source.
1	CH0 terminated abnormally at the Write cycle of the transfer destination.

- **ISTP1** (Bit 17)
  This bit indicates the abnormal end occurrence status of CH1. This bit is valid only when ISTA1 = 1.

ISTP1	Description
0	CH1 terminated abnormally at the Read cycle of the transfer source.
1	CH1 terminated abnormally at the Write cycle of the transfer destination.

- **ISTP2** (Bit 18)
  This bit indicates the abnormal end occurrence status of CH2. This bit is valid only when ISTA2 = 1.

ISTP2	Description
0	CH2 terminated abnormally at the Read cycle of the transfer source.
1	CH2 terminated abnormally at the Write cycle of the transfer destination.

- **ISTP3** (Bit 19)
  This bit indicates the abnormal end occurrence status of CH3. This bit is valid only when ISTA3 = 1.

ISTP3	Description
0	CH3 terminated abnormally at the Read cycle of the transfer source.
1	CH3 terminated abnormally at the Write cycle of the transfer destination.

11.2.4 DMA Channel Mask Registers (DMACMSK0 to 3)

The registers, DMACMSK0 to DMACMSK3 are used for setting a mask of each channel.
When the channel is masked, transfer for the channel is not performed.
Read/Write operation is enabled for the DMACMSK0 to 3 registers through program control.
At reset, these registers are set to 0x00000000 and all channels are masked.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACMSK0 to 3	—	—*	—*	—*	—*	—*	—*	—*	—	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	MSK
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Address: 0x7BE00100 (CH0), 0x7BE00200 (CH1), 0x7BE00300 (CH2), 0x7BE00400 (CH3)
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **MSK** (Bit 0)
  This bit is used for setting a channel mask.

MSK	Description
0	Releases the mask (enables/restarts DMA operation)
1	Masks (stops DMA operation)

11.2.5 DMA Transfer Mode Registers (DMACTMOD0 to 3)

The registers, DMACTMOD0 to DMACTMOD3, are used for setting the DMA transfer mode (request/transfer size/device type/the generation or non-generation of an interrupt after transfer termination) of each channel.
Read/Write operation is enabled for the DMACTMOD 0 to 3 registers through program control.
At reset, these registers are set to 0x00000040.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACTMOD0 to 3	—	—*	—*	—*	—*	—*	—*	—*	—	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	IMK	BRQ	DDP	SDP	TSIZ		ARQ
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Address: 0x7BE00104 (CH0), 0x7BE00204 (CH1), 0x7BE00304 (CH2), 0x7BE00404 (CH3)
Access: R/W
Access size: 32 bits

[Note]
Refer to the "Notes on use" since there are restrictions on setting.
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **ARQ** (Bit 0)
  This bit is used for setting a transfer request source.

ARQ	Description
0	Transfer request by DREQ
1	Transfer request by auto request

- **TSIZ** (Bit 2 to 1)
  These bits are used for setting a transfer size.

TSIZ		Description
2	1	
0	0	Byte (8 bits) transfer
0	1	Half-word (16 bits) transfer
1	0	Word (32 bits) transfer
1	1	Setting inhibited

- **SDP** (Bit 3)
  This bit is used for setting a transfer source device type.

SDP	Description
0	Fixed address device (always read from the same address)
1	Increment address device (an address is incremented according to the transfer size after normal end of transfer source Read operation)

- **DDP** (Bit 4)
  This bit is used for setting a transfer destination device type.

DDP	Description
0	Fixed address device (always written to the same address)
1	Increment address device (an address is incremented according to the transfer size after normal end of Write operation to the transfer destination)

- **BRQ** (Bit 5)
  This bit is used for setting a bus authorization request mode.

BRQ	Description
0	Burst mode (The bus authorization is not released until the end of transfer for the specified number of transfers)
1	Cycle steal mode (The bus authorization is released every 1 time of DMA transfer)

- IMK (Bit 6)
  This bit used for setting an interrupt mask.

IMK	Description
0	Releases an interrupt mask
1	Masks an interrupt

11.2.6 DMA Transfer Source Address Register (DMACSAD0 to 3)

The registers, DMACSAD0 to DMACSAD3, are used for setting the transfer source address of each channel. DMA transfer (Read) starts from the address set in this register.
When the transfer source is an increment address device, the address is incremented according to the transfer size. The address is updated when Read operation from the transfer source terminates normally. The transfer source address of a fixed address device remains unchanged.
Read/Write operation is enabled for the DMACSAD0 to 3 registers through program control.
At reset, these registers are set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACSAD0 to 3	CSAD[31:16]															
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSAD[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BE00108 (CH0), 0x7BE00208 (CH1), 0x7BE00308 (CH2), 0x7BE00408 (CH3)
Access: R/W
Access size: 32 bits

[Note]
DMAC increments the address according to the device type or the transfer size. In this case, the low-order bits of the address are internally ignored according to the setting as indicated below; however, at Read operation of this register, the low-order bits will output the written values.
Example: When 0x10000001 is written.
In word (32 bits) transfer: The low-order 2 bits are "00" internally.
When this register is read, 0x10000001 is read.
In half-word (16 bits) transfer: The low-order bit is "0" internally.
When this register is read, 0x10000001 is read.

11.2.7 DMA Transfer Destination Address Registers (DMACDAD0 to 3)

The registers, DMACDAD0 to DMACDAD3, are used for setting the transfer source address of each channel. DMA transfer (Write) starts from the address set in this register.
When the transfer source is an increment address device, the address is incremented according to the transfer size. The address is updated when Write operation from the transfer source terminates normally. The transfer source address of a fixed address device remains unchanged.
Read/Write operation is enabled for the DMACDAD0 to 3 registers through program control.
At reset, these registers are set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACDAD0 to 3	CDAD[31:16]															
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDAD[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BE0010C (CH0), 0x7BE0020C (CH1), 0x7BE0030C (CH2), 0x7BE0040C (CH3)
Access: R/W
Access size: 32 bits

[Note]
DMAC increments the address according to the device type or the transfer size. In this case, the low-order bits of the address are internally ignored according to the setting as indicated below; however, at Read operation of this register, the low-order bits will output the written values.
Example: When 0x10000001 is written.
In word (32 bits) transfer: The low-order 2 bits are "00" internally.
When this register is read, 0x10000001 is read.
In half-word (16 bits) transfer: The low-order bit is "0" internally.
When this register is read, 0x10000001 is read.

11.2.8 DMA Transfer Count Register (DMACSIZ0 to 3)

The registers, DMACSIZ0 to DMACSIZ1 are used for setting the number of transfers of each channel.
When Read operation from the transfer source terminates normally, the register value is decremented. Because this is for setting not the transfer data size but the number of transfers, so when performing DMA transfer n times, set "n" in this register.
The maximum value that can be set in these registers is "0x00010000" (65536 times).
Read/Write operation is enabled for the DMACSIZ0 to 3 registers through program control.
At reset, these registers are set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACSIZ0 to 3	—	—*	—*	—*	—*	—*	—*	—*	—	—*	—*	—*	—*	—*	—*	CSIZ16
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIZ[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BE00110 (CH0), 0x7BE00210 (CH1), 0x7BE00310 (CH2), 0x7BE00410 (CH3)
Access: R/W
Access size: 32 bits

[Note]
The maximum value that can be set in these registers is "0x00010000" (65536 times). When a value greater than this value is set, the operation is not guaranteed.
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

11.2.9 DMA Termination Status Clear Registers (DMACCINT0 to 3)

The registers, DMACCINT0 to DMACCINT3, clear the DMA termination statuses of the termination status register (DMAINT).
Writing an arbitrary data (32 bits) to this register will clear the corresponding DMA termination status bit (IREQ0 to IREQ3), termination source bit (ISTA0 to ISTA3), and abnormal end cycle bit (ISTP0 to ISTP3) of the DMAINT register to "0".
Only Write operation is enabled for the DMACCINT 0 to 3 register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACCINT0 to 3	CCINT[31:16]															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCINT[15:0]															

Address: 0x7BE00114(CH0), 0x7BE00214(CH1), 0x7BE00314(CH2), 0x7BE00414(CH3)
Access: W
Access size: 32 bits

11.3 Operation Description

The DMA controller starts DMA transfer when a transfer request is generated and terminates DMA transfer at completion of the specified number of transfers.

11.3.1 DMA Transfer Mode

A cycle steal mode and a burst mode are available as the transfer modes. Specify a transfer mode in the BRQ bit (bit 5) of the DMA transfer mode register of each channel (DMACTMOD0 to DMACTMOD3).

1. Cycle steal mode
   In cycle steal mode, data transfer is performed by repeating acquisition and release of the bus authorization every 1 time of DMA transfer.
   When transfer is requested in cycle steal mode, the DMA controller performs data transfer once by acquiring the bus authorization, and then returns the bus authorization to the CPU. After that, the DMA controller repeats acquisition of the bus authorization, transfer of 1 transfer unit, and bus atuthority release until completion of the specified number of transfers or the occurrence of an error.

2. Burst mode
   In Burst mode, once the bus authorization has been acquired, the DMA controller performs data transfer continuously until completion of the transfer for the specified number of transfers or the occurrence of an error.

11.3.2 DMA Activation Factors

DMA transfer is activated by the following factors (transfer request).
1. Transfer request in external request mode
2. Transfer request by an auto request

1. Transfer request in external request mode
   In this mode, a transfer request is issued by the external input signal (DREQ).
   This DMAC executes DMA transfer by detecting the rising edge of DREQ. In cycle steal mode, DREQ must be negated at each transfer and in Burst mode, at transfer termination.
   Since the DMAC outputs DREQCLR, which is the signal indicating Assert or Negate timing of DREQ, make the DREQ generation source Assert or Negate DREQ at the following timing.
   (a) Assert DREQ when DREQCLR is "0".
   (b) Negate DREQ when DREQCLR is "1".
   Note that the output timing differs depending on the mode, Cycle Steal mode or Burst mode in DREQCLR.
   In cycle steal mode, DREQCLR is output at each transfer in cycle steal mode (byte, half-word, or word). Also, since TCOUT (last transfer start signal) is output at the same timing as for DREQCLR at the last transfer of the specified number of transfers, make the DREQ generation source perform transfer termination processing if necessary.
   In burst mode, DREQCLR and TCOUT are output at the same time at execution of the last transfer of the set transfer count. Even if DREQ is negated after the start of DMA transfer, the bus authorization is retained until the transfer reaches the set transfer count without interrupting DMA transfer . Although DREQCLR and TCOUT will be output at execution of the last transfer even if DREQ is negated, it is recommended to negate DREQ when DREQCLR is "1".

2. Transfer request by the auto request
   In this mode, in the case of transfer between a memory and a memory or between an I/O module that cannot generate a transfer request and a memory, the processor will automatically generate a transfer request within DMAC until completion of the transfer for the specified number of transfers by setting the auto request bit in DMAC.
   In this case, once the auto request bit is set, a DMA transfer will take place automatically whenever DMAC is activated.

11.3.3 Activating DMA Transfer

DMA transfer can be activated in the following procedure.

1. Make a setting for the DMA channel mask register (DMACMSK0 to 3) of the applicable channel.
   - Mask a channel (this setting is not required when the channel is already masked).
2. Clear the interrupt flag that was generated at the previous DMA transfer by writing in the DMA Interrupt Clear register (DMACCINT0 to 3) of the applicable channel. (Not required if the flag has been cleared).
3. Set the DMA transfer mode register (DMACTMOD0 to 3) of the applicable channel.
4. Set the DMA transfer source address register (DMACSAD0 to 3) of the applicable channel.
5. Set the DMA transfer destination address register (DMACDAD0 to 3) of the applicable channel.
6. Set the DMA transfer count register (DMACSIZ0 to 1) of the applicable channel.
7. Set the DMA channel mask register (DMACMSK0 to 3) of the applicable channel.
   - DMAC can be activated by releasing the channel mask.

[Note]
DMA cannot be activated unless DMA termination status clear processing is performed.

11.3.4 End of DMA Transfer

Following are the conditions to end a DMA transfer.

1. Normal end (data transfer has been performed for the specified number of times)
   When data transfer has been performed for the number of times set in the DMA transfer count register (DMACSIZ0 to 3) of each channel, the DMA transfer of the relevant channel is terminated. At this time, the specified bit IREQ0–2 of the DMA termination status register (DMAINT) is set to "1". In this case the specified bit ISTA0–3 does not change. When the IMK bit of the DMA transfer mode register (DMACTMOD0 to 3) has been cleared to "0", an interrupt request is generated.

2. Abnormal end
   During DMA transfer for each channel, if an error response is received during the Read cycle from the transfer source or the Write cycle to the transfer destination, the DMAC terminates the DMA transfer immediately and the specified bit IREQ0–1 of the DMA interrupt status register (DMAINT) is set to "1". In this case, the specified bit of ISTA–1 is set to "1" and the error information is stored in the specified bit of ISTP–1. This information indicates the channel from which the error occurred and whether the error occurred during the Read cycle from the transfer source or the Write cycle to the transfer destination.
   When the IMK bit of the DMA transfer mode register (DMACTMOD0 to 3) is set to "0", an interrupt request is generated.

3. Forced discontinuation
   During DMA transfer for each channel, if the MSK bit of the DMA channel mask register (DMACMSK0 to 3) to "1" by the program, the transfer is discontinued after the transfer (Write operation to the transfer destination) at that point of time is terminated. In this case, no interrupt will be generated. The transfer can be resumed by releasing the mask.

DMAC checks DMA termination status by interrupt processing or polling and performs processing according to the termination status, as shown below.

1. Normal end (data transfer for the specified number of times has been completed)
   Sets the MSK bit of the DMA channel mask register (DMACMSK0 to3) of the relevant channel to 1 to mask that channel. Then clears the termination status.

2. Abnormal end
Sets the MSK bit of the DMA channel mask register (DMACMSK0 to3) of the relevant channel to 1 to mask that channel. Then clears the termination status. Performs necessary processing according to the cause of the abnormal end. (The contents of the necessary processing depend on the application.)

Table 11-1 shows the register values that are set during DMA transfer and after the end of DMA transfer

Table 11-1 Register values set during DMA transfer and after the end of DMA transfer

Register		1. During operation	2. After normal end	3. After abnormal end	4. After forced discontinuation
Status register (DMASTA)		1 (untransferred data exists)	0 (no untransferred data)	0 (no untransferred data)	1 (untransferred data exists)
Interrupt status register (DMAINT)	IREQ	0 (No influence)	1(*1)	1(*1)	0 (No influence)
	ISTA	0 (No influence)	0 (Normal end)	1 (Abnormal end)	0 (No influence)
	ISTP	0 (No influence)	0 (Initial state)	0 (In transfer source error) 1 (In transfer destination error)	0 (No influence)
Transfer source/transfer destination address register (DMACSAD0 to 3, DMACDAD0 to 3)		Stores the address currently being transferred	Stores the next address of the last transfer address	Stores the error address	Stores the next address of the discontinued transfer address
Transfer count register (DMACSIZ0 to3)		Stores the remaining transfer count value.	0	(*2)	Stores the remaining transfer count value.

*1: The IREQ bit is set to "1" regardless of the setting in IMK (interrupt mask) of the transfer mode register that corresponds.
*2: Note that the value varies depending on whether the error occurred in the transfer source or transfer destination.
The DMAC will decrement the transfer count register (CSIZ) at the point of time the access to the transfer source ended normally.

11.3.5 DMA Channel Priority Order

In the case of simultaneous transfer request to multiple channels, DMAC determines the channel for which transfer operation is performed according to the channel priority mode that is specified in the PRI bit of the DMA mode register (DMAMOD). A fixed mode or a round robin mode can be specified in the PRI bit as the priority mode.

When a fixed mode is specified, channel 0 always has the highest priority:
CH0 > CH1 > CH2 > CH3

When a round robin mode is specified, the channel for which transfer has ended will become the lowest priority. In the initial status, channel 0 has the highest priority.

Initial status	CH0 > CH1 > CH2 > CH3	
	↓	Transfer by CH0
Priority order after CH0 transfer	CH1 > CH2 > CH3 > CH0	
	↓	Transfer by CH2
Priority order after CH2 transfer	CH3 > CH0 > CH1 > CH2	
	↓	Transfer by CH1
Priority order after CH1 transfer	CH2 > CH3 > CH0 > CH1	
	↓	Transfer by CH3
Priority order after CH3 transfer	CH0 > CH1 > CH2 > CH3	

Figure 11-3 Operation in round robin mode

The channel priority order is determined when DMAC acquires the bus authorization from CPU. Therefore, while data transfer is being performed in burst mode in any channel, data transfer in burst mode is continued even if a transfer request occurs in the channel of the higher priority.

11.3.6 Notes on Use

1. Note the access restrictions on the devices in the transfer source and transfer destination when making various settings.
2. DMA transfer is not performed during HALT mode. The reason is that during HALT mode, the bus authorization cannot be acquired even if transfer is requested.
3. DMAC increments the transfer source/transfer destination addresses according to the setting of the device type and the transfer size. In this case, the low-order bit of the address values that are set in the DMACSAD0–3 and DMACDAD0–3 registers are ignored internally according to the setting as shown below. However, when these registers are read, the values that has been written are output from the low-order bits.
   Example: 0x10000001 is written:
   For word (32 bits) transfer: The low-order 2 bits will be configured to "00" internally.
   When this register is read, 0x10000001 is read.
   For half-word (16 bits) transfer: The low-order 1 bit will be configured to "0" internally.
   When this register is read, 0x10000001 is read.

4. DMA transfer may not be performed depending on the combination of the transfer source and the transfer destination. Note the following combinations when setting the transfer size (TSIZ) of the transfer mode register (DMAMOD0–3).

		Transfer destination	Internal device		External device	
			Incremental device (Internal SRAM)	Fixed address device	Incremental device	Fixed address device
Transfer source		Bus width	32 bits	32 bits	16 bits	16 bits
Internal device	Incremental device (Internal SRAM)	32 bits	W/H/B	W/H/B	W/H	H
	Fixed address device	32 bits	W/H/B	W/H/B	W/H	H
External device	Incremental device	16 bits	W/H	W/H	W/H	H
	Fixed address device	16 bits	H	H	H	H

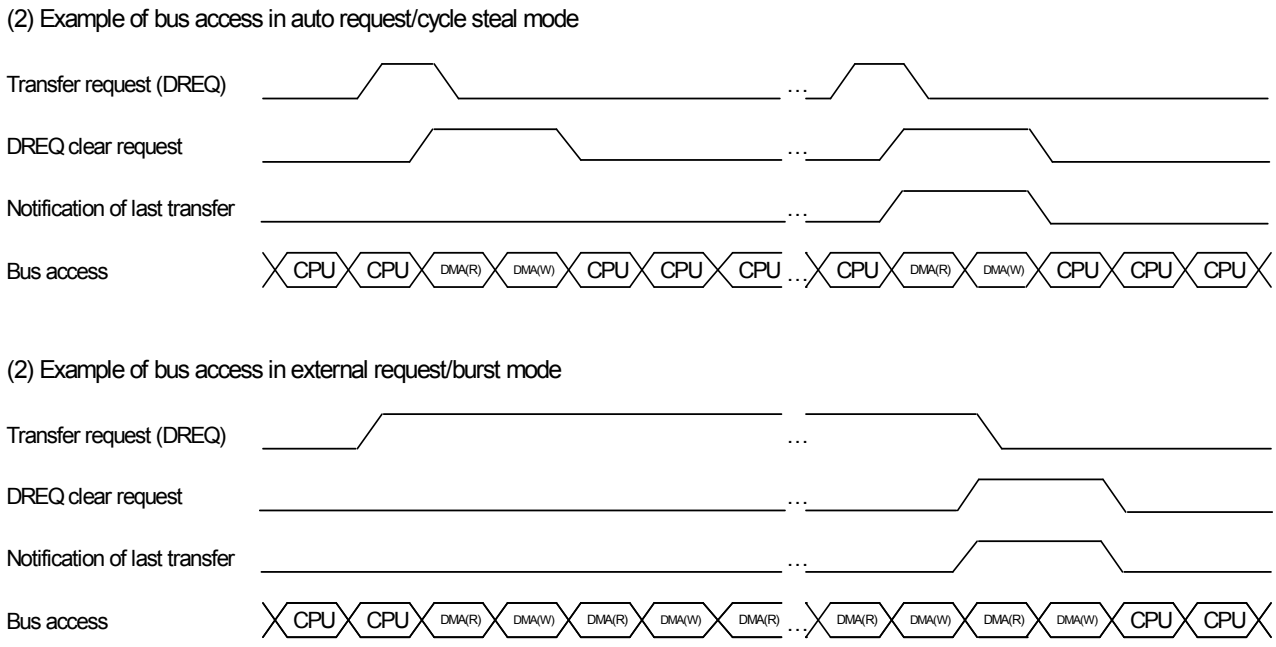
W: Word (32bit) transfer
H: Half-word (16bit) transfer
B: Byte (8bit) transfer

11.4 DMA Transfer Timing

11.4.1 Transfer Start Timing

At least five clocks are required from the occurrence of a transfer request to the start of DMA transfer.

(1) Example of bus access in auto request/cycle steal mode



*1 autoreq is an internal signal that is generated by setting the transfer mode register (DMACTMOD0 to 1).

*2 The addressing mode is a dual mode.

Figure 11-4 Transfer Start Timing

11.4.2 Transfer Timing

Figure 11-5 shows the timing for performing cycle steal transfer from I/O (on the external bus) to the memory by DREQ.

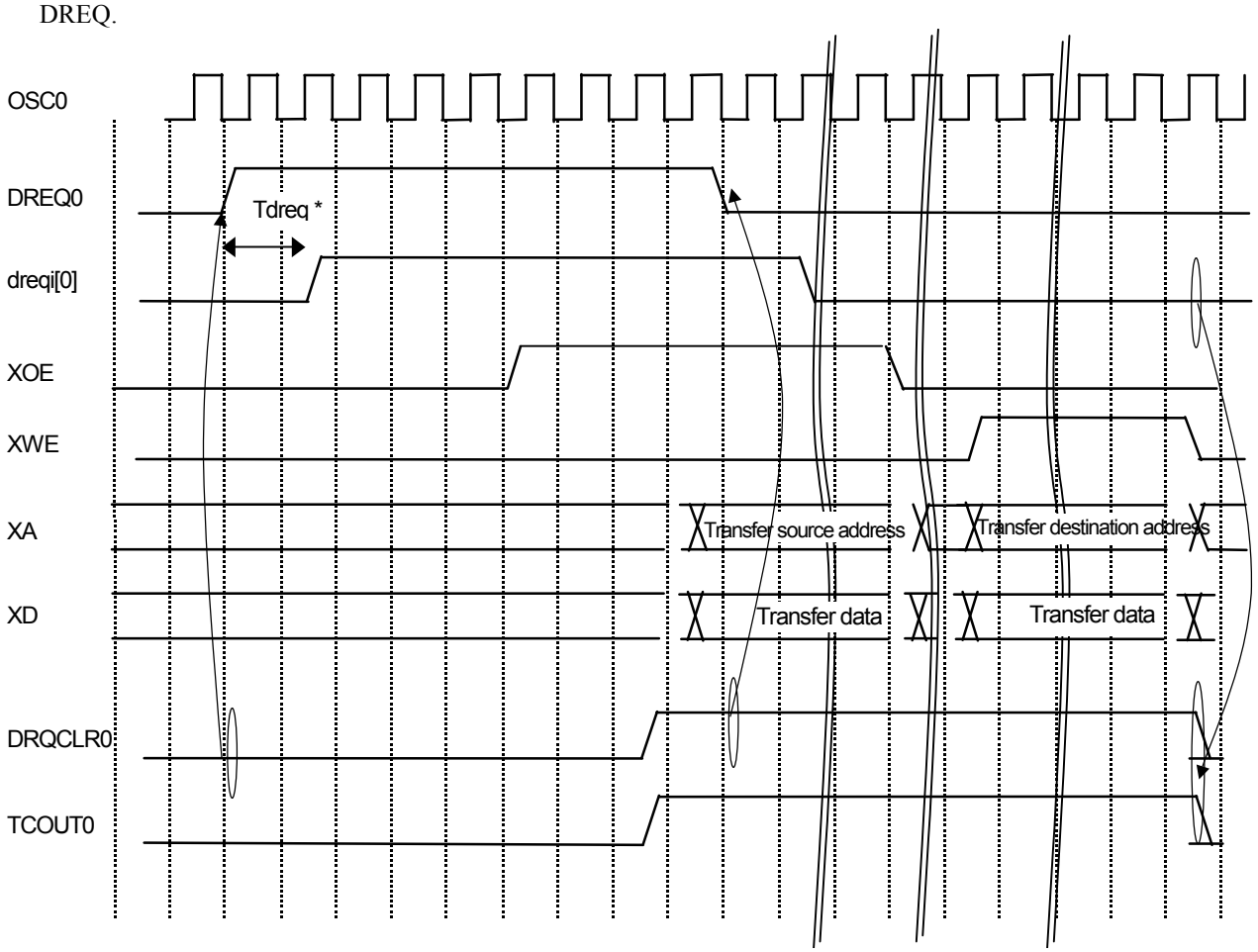


Figure 11-5 Cycle Steal Transfer from I/O (on the External Bus) to the Memory (Word/1 Time of Transfer)

Chapter 12

GPIO

Chapter 12 GPIO

12.1 Overview

This LSI has five channels of internal 16-bit GPIOs and one channel of internal 8-bit GPIO. All GPIO registers only allow word access. Neither half-word nor byte access can be used for the GPIO registers.

Features:

- Input or output can be specified for each bit.
- Enters an input state immediately after a reset.
- GPIOA15–00, GPIOB15–00 and GPIOC15–00 are GPIOs without an interrupt function that can be set to an external bus by the setting of the EXTBUS pin.
- GPIOD15–00 are GPIOs without an interrupt function that can select a secondary function for each bit.
- GPIOE15–12 are dedicated GPIOs that can be used as external interrupts (GPIOD15 is 5V tolerant). A level interrupt/edge interrupt and the interrupt direction can be specified for these GPIOs.
- GPIOE11–00 are dedicated GPIOs without an interrupt function.
- GPIOE06–00 are GPIOs without an interrupt function that can be used when the NAND FLASH mode is selected by the setting of the IDEMODE pin.

12.1.1 Configuration

Figure 12-1 shows the configuration of a GPIO without an interrupt function (main part only).
The figure specifically shows bit 0 in detail, but the identical GPIO configuration applies to all the other bits.

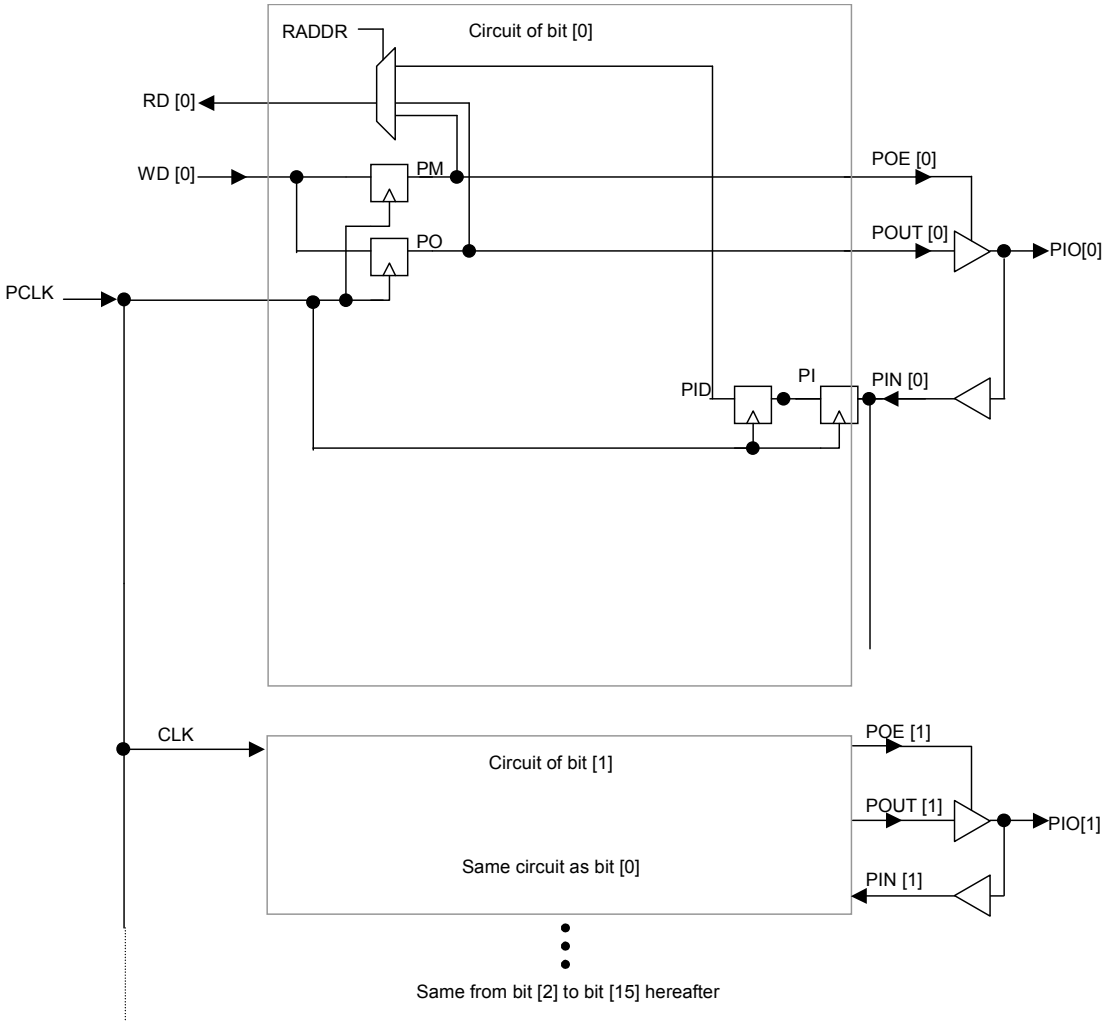


Figure 12-1 Block Diagram of a GPIO without an Interrupt Function

Figure 12-2 shows the configuration of a GPIO with an interrupt function (main part only).
The figure specifically shows bit 0 in detail, but the identical GPIO configuration applies to all the other bits.

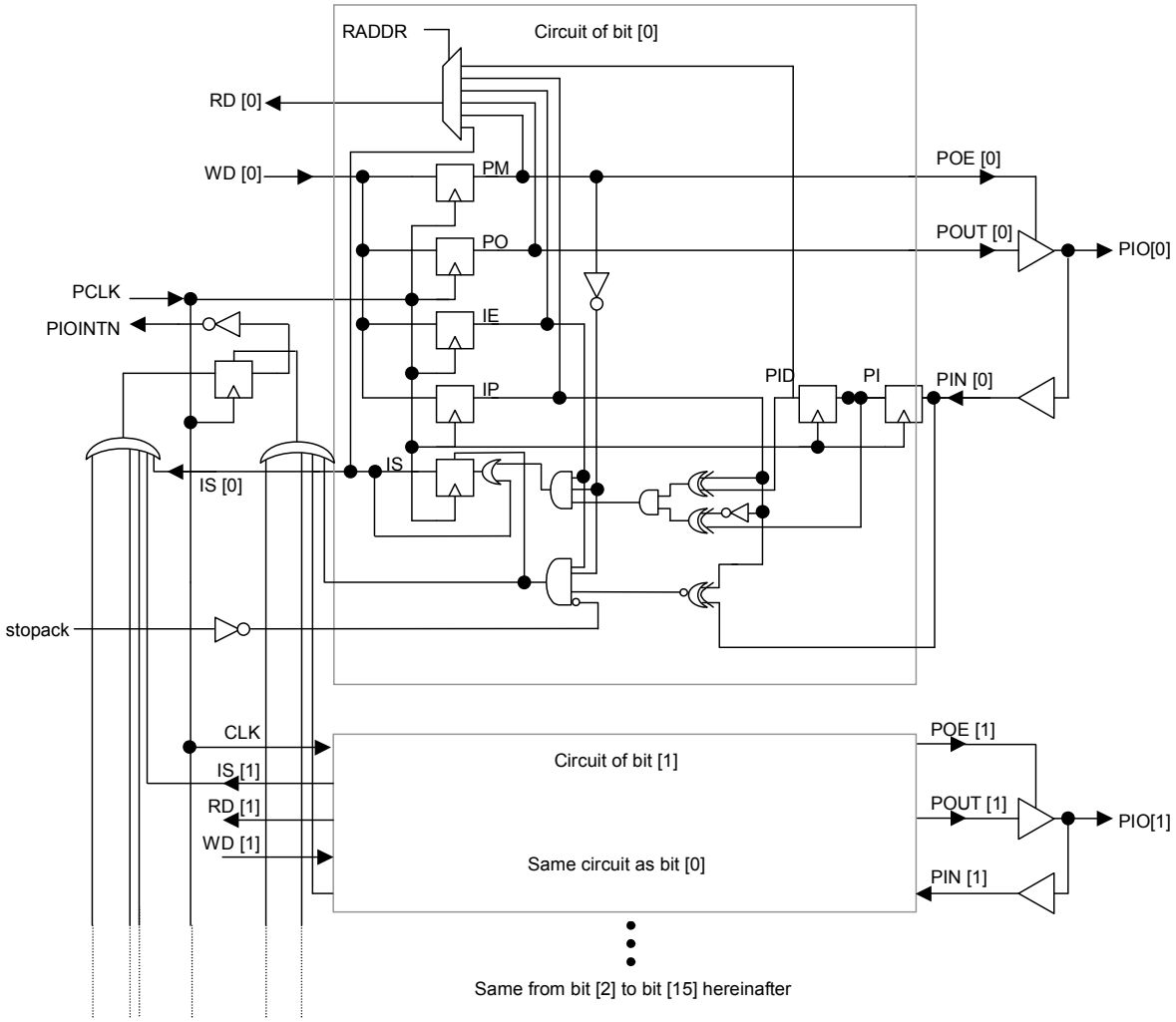


Figure 12- 2 Block Diagram of a GPIO with an Interrupt Function

12.1.2 Pin List

Pin name	I/O	Function	
		Primary function	Secondary function (pin name)
PIOA15	I/O	General purpose port A, bit 15	XA16
PIOA14	I/O	General purpose port A, bit 14	XA15
PIOA13	I/O	General purpose port A, bit 13	XA14
PIOA12	I/O	General purpose port A, bit 12	XA13
PIOA11	I/O	General purpose port A, bit 11	XA12
PIOA10	I/O	General purpose port A, bit 10	XA11
PIOA09	I/O	General purpose port A, bit 09	XA10
PIOA08	I/O	General purpose port A, bit 08	XA09
PIOA07	I/O	General purpose port A, bit 07	XA08
PIOA06	I/O	General purpose port A, bit 06	XA07
PIOA05	I/O	General purpose port A, bit 05	XA06
PIOA04	I/O	General purpose port A, bit 04	XA05
PIOA03	I/O	General purpose port A, bit 03	XA04
PIOA01	I/O	General purpose port A, bit 02	XA03
PIOA01	I/O	General purpose port A, bit 01	XA02
PIOA00	I/O	General purpose port A, bit 00	XA01
PIOB15	I/O	General purpose port B, bit 15	XD15
PIOB14	I/O	General purpose port B, bit 14	XD14
PIOB13	I/O	General purpose port B, bit 13	XD13
PIOB12	I/O	General purpose port B, bit 12	XD12
PIOB11	I/O	General purpose port B, bit 11	XD11
PIOB10	I/O	General purpose port B, bit 10	XD10
PIOB09	I/O	General purpose port B, bit 09	XD09
PIOB08	I/O	General purpose port B, bit 08	XD08
PIOB07	I/O	General purpose port B, bit 07	XD07
PIOB06	I/O	General purpose port B, bit 06	XD06
PIOB05	I/O	General purpose port B, bit 05	XD05
PIOB04	I/O	General purpose port B, bit 04	XD04
PIOB03	I/O	General purpose port B, bit 03	XD03
PIOB02	I/O	General purpose port B, bit 02	XD02
PIOB01	I/O	General purpose port B, bit 01	XD01
PIOB00	I/O	General purpose port B, bit 00	XD00
PIOC15	I/O	General purpose port C, bit 15	XA20
PIOC14	I/O	General purpose port C, bit 14	XA19
PIOC13	I/O	General purpose port C, bit 13	XA18
PIOC12	I/O	General purpose port C, bit 12	XA17
PIOC11	I/O	General purpose port C, bit 11	XOE_N
PIOC10	I/O	General purpose port C, bit 10	XWE_N
PIOC09	I/O	General purpose port C, bit 09	XROMCS_N
PIOC08	I/O	General purpose port C, bit 08	XRAMCS_N
PIOC07	I/O	General purpose port C, bit 07	XIOCS11_N
PIOC06	I/O	General purpose port C, bit 06	XIOCS10_N
PIOC05	I/O	General purpose port C, bit 05	XIOCS01_N
PIOC04	I/O	General purpose port C, bit 04	XIOCS00_N
PIOC03	I/O	General purpose port C, bit 03	XBS1_N
PIOC02	I/O	General purpose port C, bit 02	XBS0_N
PIOC01	I/O	General purpose port C, bit 01	XWAIT1
PIOC00	I/O	General purpose port C, bit 00	XWAIT0
PIOD15	I/O	General purpose port D, bit 15	UP_RXD
PIOD14	I/O	General purpose port D, bit 14	UP_TXD
PIOD13	I/O	General purpose port D, bit 13	SSIOCK1
PIOD12	I/O	General purpose port D, bit 12	SSIORXD1
PIOD11	I/O	General purpose port D, bit 11	SSIOTXD1
PIOD10	I/O	General purpose port D, bit 10	SSIOCK0
PIOD09	I/O	General purpose port D, bit 09	SSIORXD0
PIOD08	I/O	General purpose port D, bit 08	SSIOTXD0
PIOD07	I/O	General purpose port D, bit 07	CKOUTA/CKOUT

PIOD06	I/O	General purpose port D, bit 06	SDA
PIOD05	I/O	General purpose port D, bit 05	WSA
PIOD04	I/O	General purpose port D, bit 04	SCLA/SCL
PIOD03	I/O	General purpose port D, bit 03	CKOUTD/L
PIOD02	I/O	General purpose port D, bit 02	SDD
PIOD01	I/O	General purpose port D, bit 01	WSD
PIOD00	I/O	General purpose port D, bit 00	SCLD/L
PIOE15	I/O	General purpose port E, bit 15	—
PIOE14	I/O	General purpose port E, bit 14	—
PIOE13	I/O	General purpose port E, bit 13	—
PIOE12	I/O	General purpose port E, bit 12	—
PIOE11	I/O	General purpose port E, bit 11	—
PIOE10	I/O	General purpose port E, bit 10	—
PIOE09	I/O	General purpose port E, bit 09	—
PIOE08	I/O	General purpose port E, bit 08	—
PIOE07	I/O	General purpose port E, bit 07	
PIOE06	I/O	General purpose port E, bit 06	
PIOE05	I/O	General purpose port E, bit 05	
PIOE04	I/O	General purpose port E, bit 04	
PIOE03	I/O	General purpose port E, bit 03	
PIOE02	I/O	General purpose port E, bit 02	
PIOE01	I/O	General purpose port E, bit 01	
PIOE00	I/O	General purpose port E, bit 00	
PIOF06	I/O	General purpose port F, bit 06	IDED14
PIOF05	I/O	General purpose port F, bit 05	IDED13
PIOF04	I/O	General purpose port F, bit 04	IDED12
PIOF03	I/O	General purpose port F, bit 03	IDED11
PIOF02	I/O	General purpose port F, bit 02	IDED10
PIOF01	I/O	General purpose port F, bit 01	IDED09
PIOF00	I/O	General purpose port F, bit 00	IDED08

12.1.3 Register List

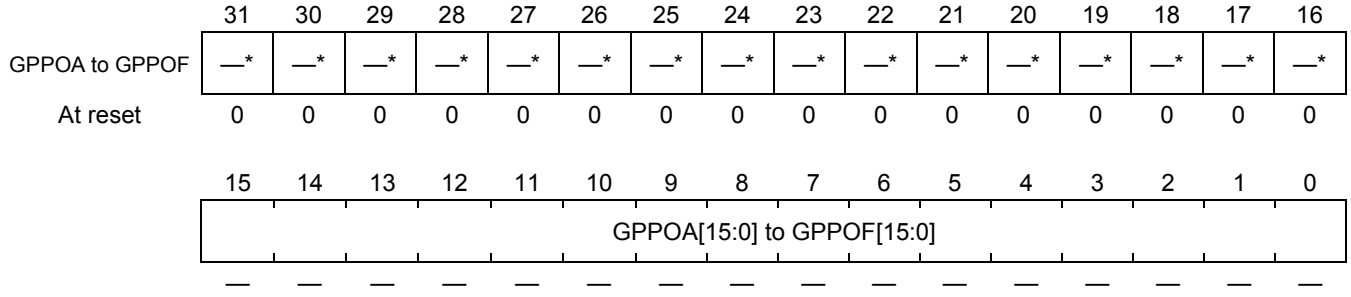
Address [H]	Register name	Symbol	R/W	Initial value [H]
0xB7A0_0000	Port A output register	GPPOA	R/W	Undefined
0xB7A0_0004	Port A input register	GPPIA	R	Depends on the pin status
0xB7A0_0008	Port A mode register	GPPMA	R/W	0x0000_0000
……	<Reserved>	—	—	—
0xB7A0_0020	Port B output register	GPPOB	R/W	Undefined
0xB7A0_0024	Port B input register	GPPIB	R	Depends on the pin status
0xB7A0_0028	Port B mode register	GPPMB	R/W	0x0000_0000
……	<Reserved>	—	—	—
0xB7A0_0040	Port C output register	GPPOC	R/W	Undefined
0xB7A0_0044	Port C input register	GPPIC	R	Depends on the pin status
0xB7A0_0048	Port C mode register	GPPMC	R/W	0x0000_0000
……	<Reserved>	—	—	—
0xB7A0_0060	Port D output register	GPPOD	R/W	Undefined
0xB7A0_0064	Port D input register	GPPID	R	Depends on the pin status
0xB7A0_0068	Port D mode register	GPPMD	R/W	0x0000_0000
……	<Reserved>	—	—	—
0xB7A0_0080	Port E output register	GPPOE	R/W	Undefined
0xB7A0_0084	Port E input register	GPPIE	R	Depends on the pin status
0xB7A0_0088	Port E mode register	GPPME	R/W	0x0000_0000
0xB7A0_008C	Port E interrupt enable register	GPIEE	R/W	0x0000_0000
0xB7A0_0090	Port E interrupt polarity register	GPIPE	R/W	0x0000_0000
0xB7A0_0094	Port E interrupt status register	GPISE	R/W	0x0000_0000
0xB7A0_0098	Port E interrupt mode register	GPIME	R/W	0x0000_0000
……	<Reserved>	—	—	—
0xB7A0_00A0	Port F output register	GPPOF	R/W	Undefined
0xB7A0_00A4	Port F input register	GPPIF	R	Depends on the pin status
0xB7A0_00A8	Port F mode register	GPPMF	R/W	0x0000_0000

Address [H]	Register name	Symbol	R/W	Initial value [H]
0x8000_0008	PIO pin switching register	PIOCTL	R/W	0x0000_0000

12.2 Register Description

12.2.1 Port Output Registers (GPPOA, GPPOB, GPPOC, GPPOD, GPPOE, GPPOF)

The GPPOA, GPPOB, GPPOC, GPPOD, GPPOE and GPPOF registers set the output values of the PIOA, PIOB, PIOC, PIOD, PIOE and PIOF ports, respectively.
For the bits set to "output" in the port mode registers (GPPMA, GPPMB, GPPMC, GPPMD, GPPME, GPPMF), the values set in the Port Output Registers are output to the pins PIOA15–00, PIOB15–00, PIOC15–00, PIOD15–00, PIOE15–00, and PIOF06–00.
The GPPOA, GPPOB, GPPOC, GPPOD, GPPOE and GPPOF registers can be read from/written to by program.
The GPPOF [15:7] bits of the GPPOF register are not used. These bits will always read "0" when read. Write "0" for write.
The value at reset is undefined.



Address: 0xB7A00000 (GPPOA), 0xB7A00020 (GPPOB), 0xB7A00040 (GPPOC), 0xB7A00060 (GPPOD), 0xB7A00080 (GPPOE), 0xB7A000A0 (GPPOF)
Access: R/W
Access size: 32 bits

12.2.2 Port Input Registers (GPPIA, GPPIB, GPPIC, GPPID, GPPIE, GPPIF)

The GPPIA, GPPIB, GPPIC, GPPID, GPPIE and GPPIF registers reflect the input values of the PIOA, PIOB, PIOC, PIOD, PIOE and PIOF ports.
If the pins for these ports are set to output, the output values from those pins are read.
The GPPIA, GPPIB, GPPIC, GPPID, GPPIE and GPPIF registers can only be read by programs.
The GPPIF [15:7] bits of the GPPIF register are not used. These bits will always read "0" when read. Write "0" for write.
The values at reset are the input values of the pins PIOA15–00, PIOB15–00, PIOC15–00, PIOD15–00, PIOE15–00, and PIOF06–00.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPPIA to GPPIF	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPPIA[15:0] to GPPIF[15:0]																
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Address: 0xB7A00004 (GPPIA), 0xB7A00024 (GPPIB), 0xB7A00044 (GPPIC), 0xB7A00064 (GPPID), 0xB7A00084 (GPPIE), 0xB7A000A4 (GPPIF)
Access: R
Access size: 32 bits

[Note]
If a secondary function has been selected for the applicable I/O ports of PIOA [15:0], PIOB [15:0], PIOC [15:0], PIOD [15:0] and PIOF [6:0], the values at a read operation are undefined.

12.2.3 Port Mode Registers (GPPMA, GPPMB, GPPMC, GPPMD, GPPME, GPPMF)

The GPPMA, GPPMB, GPPMC, GPPMD, GPPME and GPPMF registers set the input/output mode of the PIOA, PIOB, PIOC, PIOD, PIOE and PIOF ports for each bit.
The GPPMA, GPPMB, GPPMC, GPPMD, GPPME and GPPMF registers can be read from/written to by programs.
The GPPMF [15:7] bits of the GPPMF register are not used. These bits will always read "0" when read. Write "0" for write.
The value at the time of a reset is 0x0000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPPMA to GPPMF	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPPMA[15:0] to GPPMF[15:0]																
	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Address: 0xB7A00008 (GPPMA), 0xB7A00028 (GPPMB), 0xB7A00048 (GPPMC), 0xB7A00068 (GPPMD), 0xB7A00088 (GPPME), 0xB7A000A8 (GPPMF)
Access: R/W
Access size: 32 bits

[Bit Description]

- **GPPMA[15:0]–GPPME[15:0]** (bits 15 to 0), **GPPMF[7:0]** (bits 7 to 0)
  These bits set the input/output mode for each bit of the PIO ports.

GPPMA[15:0]–GPPME[15:0], GPPMF[7:0]	Description
0	Input
1	Output

12.2.4 Port E Interrupt Enable Register (GPIEE)

The GPIEE register specifies whether to generate an interrupt when any bits of PIOE are set to the input mode and the input values for those bits of PIOE detect the edge or level specified in the GPIME register on the polarity[*1] specified in the GPIPE register.
The bits set to the output mode in the GPPME register are not used as interrupt causes regardless of the value of the GPIEE register.
The GPIEE register can be read from/written to by programs.
The value at the time of a reset is 0x0000.
*1: "Polarity" mentioned in this manual means "a positive or negative going (edge)."

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIEE	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIEE[15:12]				—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7A0008C (GPIEE)
Access: R/W
Access size: 32 bits

[Note]
If the input mode is set, interrupt enable is set, and the interrupt mode is set to level interrupt, clock supply can be resumed by PIOE [15:12] input when clocks stop in the standby mode. Because of a level interrupt, the input signal must maintain the interrupt status until clocks are supplied.
While an interrupt is being generated, merely setting this register to interrupt disable can never cancel the interrupt. To cancel an interrupt, be sure to clear the corresponding bit of the GPISE register.
For more information about resuming clock supply, see Chapter 7, "Power Consumption Control and System Control."

[Bit Description]

- **GPIEE[15:12]** (bits 15 to 12)
  These bits specify interrupt enable or disable for each bit of the PIO ports.

GPIEE[15:12]	Description
0	Interrupt disable
1	Interrupt enable

12.2.5 Port E Interrupt Polarity Register (GPIPE)

The GPIPE register sets a polarity for the bits of PIOE that generate an interrupt.
When any bits of PIOE are set to the input mode and the GPIEE register is set to interrupt enable, an interrupt is generated if the input values for those bits of PIOE detect the edge or level specified in the GPIME register on the polarity[*1] specified in this register.
The GPIPE register can be read from/written to by programs.
The value at the time of a reset is 0x0000.
*1: "Polarity" mentioned in this manual means "a positive or negative going (edge)."

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIPE	—*	—*	—*	—*	—*	—	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIPE [15:12]				—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7A00090 (GPIPE)
Access: R/W
Access size: 32 bits

[Bit Description]

- **GPIPE [15:12]** (bits 15 to 12)

These bits set the polarity for generating interrupts for each bit of the PIO.

GPIPE[15:12]	Description
0	Generates an interrupt at a falling edge or a "L" level of an input signal.
1	Generates an interrupt at a rising edge or a "H" level of an input signal.

12.2.6 Port E Interrupt Status Register (GPISE)

The GPISE register shows which bit has caused the interrupt when an interrupt has occurred in each port.
The GPISE register can be read from/written to by programs, and is cleared to "0" by writing "1" into the applicable bit when in the Edge Detection mode. If "0" is written into the bits, it is ignored. At this point, an interrupt output (exintn) is also canceled.
The value at the time of a reset is 0x0000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPISE	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPISE [15:12]				—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7A00094 (GPISE)
Access: R/W
Access size: 32 bits

[Note]
Regarding the operation when in the Level Detection mode, an interrupt cannot be canceled by this register while it is occurring.
Furthermore, an interrupt cannot be canceled by merely setting an input signal (PIN) to the Interrupt Cancel level. To cancel an interrupt, it is necessary to write "1" into the applicable bit of this register after setting an input signal (PIN) to the Interrupt Cancel level.
If the generation of an interrupt cause and the cancellation of an interrupt from the CPU occur at the same time, the generation of an interrupt takes precedence.

[Bit Description]

- **GPISE[15:12]** (bits 15 to 12)
  These bits indicate which bits have caused an interrupt.

GPISE[15:12]	Description
0	No interrupt occurred.
1	An interrupt occurred.

Each bit will be set to "1" if one of the following conditions is met (example of bit 0):
1. After setting GPPME [15] ="0," GPIPE [15] ="1," GPIEE [15] ="1," and GPIME [15] ="0," a rising edge of PIOE [15] is detected.
2. After setting GPPME [15] ="0," GPIPE [15] ="0," GPIEE [15] ="1," and GPIME [15] ="1," a "L" level of PIOE [15] is detected.

12.2.7 Port E Interrupt Mode Register (GPIME)

The GPIME register specifies the Edge Detection mode or Level Detection mode for generating an interrupt for each port.
The GPIME register can be read from/written to by programs.
The value at the time of a reset is 0x0000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIME	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIME [15:12]				—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7A00098 (GPIME)
Access: R/W
Access size: 32 bits

[Bit Description]

- **GPIME[15:12]** (bits 15 to 12)

These bits specify the Edge Detection or Level Detection mode for generating an interrupt for each bit of the PIO.

GPIME[15:12]	Description
0	Edge Detection mode
1	Level Detection mode

12.2.8 PIO Pin Switching Register (PIOCTL)

The PIO pin switching register (PIOCTL) of the configuration register module selects the primary function or secondary function of PIOD [15:0].
The PIOCTL register can be read from/written to by programs.
Prior to writing into this register, it is necessary to write 0x0000003C in order to remove write protect.
This register can be read as is when reading.
The value at the time of a reset is 0x0000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIOCTL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIOCTL [15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x80000008
Access: R/W
Access size: 32 bits

[Bit Description]

- **PIOCTL[0]** (bit 0)

PIOCTL[0]="0" (primary function)		PIOCTL[0]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD00	Input/Output	SCLD	Input/Output

- **PIOCTL[1]** (bit 1)

PIOCTL[1]="0" (primary function)		PIOCTL[1]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD01	Input/Output	WSD	Input/Output

- **PIOCTL[2]** (bit 2)

PIOCTL[2]="0" (primary function)		PIOCTL[2]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD02	Input/Output	SDD	Output

- **PIOCTL[3]** (bit 3)

PIOCTL[3]="0" (primary function)		PIOCTL[3]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD03	Input/Output	CKOUTD	Output

- **PIOCTL[4]** (bit 4)

PIOCTL[4]="0" (primary function)		PIOCTL[4]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD04	Input/Output	SCLA/SCL	Input/Output

- **PIOCTL[5]** (bit 5)

PIOCTL[5]="0" (primary function)		PIOCTL[5]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD05	Input/Output	WSA	Input/Output

- **PIOCTL[6]** (bit 6)

PIOCTL[6]="0" (primary function)		PIOCTL[6]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD06	Input/Output	SDA	Input

- **PIOCTL[7]** (bit 7)

PIOCTL[7]="0" (primary function)		PIOCTL[7]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD07	Input/Output	CKOUTA/CKOUT	Output

- **PIOCTL[8]** (bit 8)

PIOCTL[8]="0" (primary function)		PIOCTL[8]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD08	Input/Output	SSIOTXD0	Output

- **PIOCTL[9]** (bit 9)

PIOCTL[9]="0" (primary function)		PIOCTL[9]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD09	Input/Output	SSIORXD0	Input

- **PIOCTL[10]** (bit 10)

PIOCTL[10]="0" (primary function)		PIOCTL[10]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD10	Input/Output	SSIOCK0	Input/Output

- **PIOCTL[11]** (bit 11)

PIOCTL[11]="0" (primary function)		PIOCTL[11]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD15	Input/Output	SSIOTXD1	Output

- **PIOCTL[12]** (bit 12)

PIOCTL[12]="0" (primary function)		PIOCTL[12]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD12	Input/Output	SSIORXD1	Input

- **PIOCTL[13]** (bit 13)

PIOCTL[13]="0" (primary function)		PIOCTL[13]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD13	Input/Output	SSIOCK1	Input/Output

- **PIOCTL[14]** (bit 14)

PIOCTL[14]="0" (primary function)		PIOCTL[14]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD14	Input/Output	UP_TXD	Output

- **PIOCTL[15]** (bit 15)

PIOCTL[15]="0" (primary function)		PIOCTL[15]="1" (secondary function)	
Pin name	Input/Output	Pin name	Input/Output
PIOD15	Input/Output	UP_RXD	Input

12.3 Operational Description

Each of ports A to F performs input/output to/from the PIO pins via the port output registers (GPPOA/GPPOB/ GPPOC/ GPPOD/GPPOE/GPPOF) and the port input registers (GPPIA/GPPIB/GPPIC/GPPID/GPPIE/GPPIF). Furthermore, an operation can be set for each bit by setting the input/output mode in the port E mode registers (GPPME), whether or not to generate an interrupt for each bit in the port E interrupt enable register (GPIEE) and the port E interrupt polarity register (GPIPE), and the edge detection/level detection conditions in the port E interrupt mode register (GPIME).

12.3.1 Interrupt Request

As for PIOE15 to PIOE12, while generating an interrupt, an interrupt of a different cause can be generated by the interrupt controller.
The following describes bit 15 of the PIOE.
When bit 15 of the PIOE is set to the input mode in the port mode register (GPPME [15] ="0") and in the case of the Edge Interrupt mode (GPIME [15] ="0"), the corresponding bit (GPISE [15]) of the status register is set to "1" if PIOE [15] is set to "0" when GPIPE [15]="0" OR if PIOE [15] is set to "1" when GPIPE [15]="1." At this point, an interrupt request is generated if interrupt enable (GPIEE [15] ="1") is set.
In the case of the Level Interrupt mode (GPIME [15] = "1"), the corresponding bit (GPISE [15]) of the status register is set to "1" at the same time the interrupt level is set. At this point, an interrupt request is generated if interrupt enable (GPIEE [15] ="1") is set.
In the case of a level mode interrupt, the CPU can return from the state in which the PCLK is stopped.

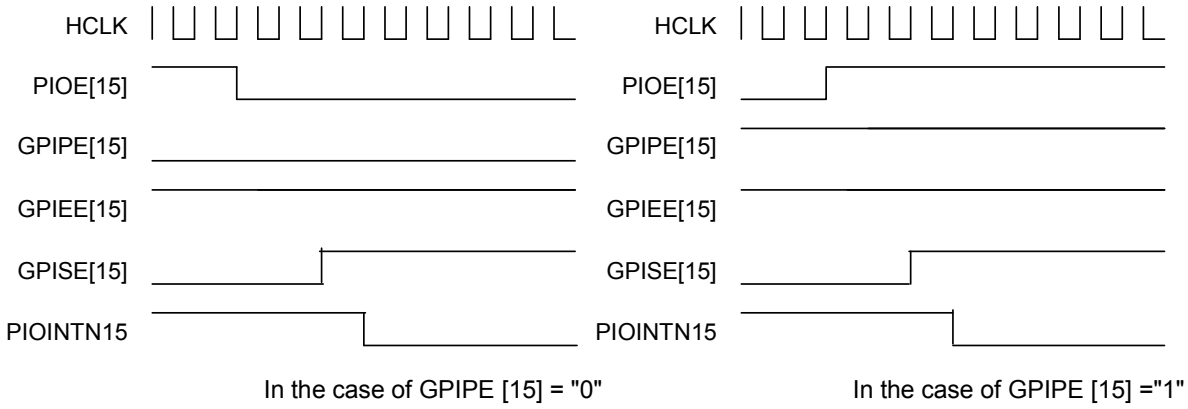


Figure12-2 Interrupt Operation in the Edge Mode

12.3.2 Primary/Secondary Function Control

The PIOA [15:0], PIOB [15:0], PIOC [15:0], PIOD [15:0] and PIOF [6:0] each have a secondary function.
For the PIOA, PIOB and PIOC, a GPIO is selected if the EXTBUS pin is set to "L," and the external memory bus is selected if the EXTBUS pin is set to "H."
For the PIOD, either a GPIO as the primary function or the secondary function can be selected for each pin by the PIOCTL register of the configuration register module.
PIOE[15:0] do not have secondary functions. PIOE[15:12] (four pins) function as external interrupt pins by setting the port E interrupt related registers.
For the PIOF06 to PIOF00, a GPIO is selected if the IDEMODE pin is set to "L," and the IDE controller is selected if the IDEMODE pin is set to "H."

Chapter 13

Watchdog Timer (WDT)

Chapter 13 Watchdog Timer (WDT)

13.1 Overview

This LSI has a built-in 16-bit watchdog timer (WDT) featuring Interval Timer mode and Watchdog Timer mode. The WDT is used to monitor program crash or runaway (running out of control).
In Interval Timer mode, an interrupt is generated when the WDT overflows. In Watchdog Timer mode, an interrupt or forced reset is generated when the WDT overflows. Because 7.5 MHz is input as the CCLK and counting is performed using a clock divided into 1/1, 1/4, 1/16 or 1/64 inside this module, a cycle of 9 ms, 34 ms, 140 ms or 560 ms can be set. The reset pulse width is 16 cycles of the HCLK.

13.1.1 Configuration

Figure 13-1 shows the configuration of the WDT.

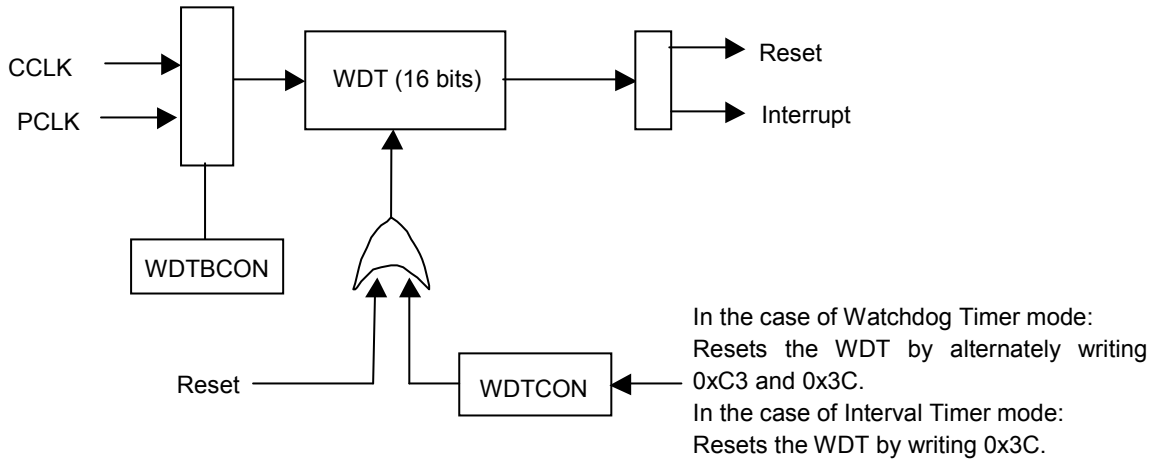


Figure 13-1 Configuration of the WDT

13.1.2 Register List

Address	Name	Abbreviation	R/W	Initial value
0xB7E0_0000	Watchdog timer control register	WDTCON	W	—
0xB7E0_0004	Time base counter control register	WDTBCON	R/W	0x0000_0000
0xB7E0_0014	Status register	WDSTAT	R/W	0x0000_0000

13.2 Register Description

13.2.1 Watchdog Timer Control Register (WDTCON)

The WDTCON register clears the watchdog timer to 0. Only write operation can be performed to the WDTCON register by a program. To operate the watchdog timer after a system reset, write 0x3C into this register. The watchdog timer stops until 0x3C is written into this register. Once the watchdog timer starts, it can be cleared to 0 by alternately writing 0xC3 and 0x3C periodically.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDTCON	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	WDTCON[7:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7E00000
Access: W
Access size: 32 bits

13.2.2 Time Base Counter Control Register (WDTBCON)

The WDTBCON register controls the operation of the watchdog timer. This register can set various functions including Interval Timer mode/Watchdog Timer mode, the generation of an interrupt/reset when the watchdog timer overflows, and the selection of a watchdog timer clock.
Read/Write operation is enabled for the WDTBCON register through program control.
Once 0x5A is written into this register by the write protect function, it becomes possible to write each setting value. This function protects the operation of the watchdog timer from erroneous writing by a program. The value is set to 0x00 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDTBCON	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTBCON	—*	—*	—*	—*	—*	—*	—*	—*	WDHLT	OFINTMODE	—*	ITEN	ITM	—*	WDCLK[1:0]	
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7E00004
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **WDCLK[1:0]** (Bit 1 to 0)
  This bit field selects the operating clock of the watchdog timer.

WDCLK[1:0]		Description
1	0	
0	0	CCLK/ (= 7.5MHz)
0	1	CCLK/4
1	0	CCLK/16
1	1	CCLK/64

- **ITM** (Bit 3)
  This bit selects the mode of the watchdog timer, either Interval Timer mode or Watchdog Timer mode.

ITM	Description
0	Watchdog Timer mode
1	Interval Timer mode

- **ITEN** (Bit 4)
  This bit specifies the stop and start of the watchdog timer in Interval Timer mode (ITM = 1).

ITEN	Description
0	Stops.
1	Starts.

- **OFINTMODE** (Bit 6)
  This bit specifies the operation when the watchdog timer overflows

OFINTMODE	Description
0	Generates an interrupt.
1	Generates a system reset.

- **WDHLT** (Bit 7)
  This bit specifies the stop and start of the watchdog timer in Interval Timer mode/Watchdog Timer mode.

WDHLT	Description
0	Starts.
1	Stops.

13.2.3 Status Register (WDSTAT)

The WDSTAT register is a status register that indicates the generation status of an interrupt or reset by the watchdog timer operation. Only write operation can be performed to the WDTCON register by a program. The value is set to 0x00 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDSTAT	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	IVTIST	WDTIST	—*	—*	—*	RSTSTATUS
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7E00014
Access: R/W
Access size: 32 bits

[Note
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- RSTSTATUS (Bit 0)
  This bit indicates a reset source.
  Once this bit is set to "1", it will not be cleared to "0" until an nRESET signal is input

RSTSTATUS	Description
0	Reset by power-on
1	Reset by the watchdog timer

- **WDTIST** (Bit 4)
  This bit indicates that an interrupt by the watchdog timer has occurred.
  The bit can be cleared to "0" by writing "1" by a program.

[Note]
Be sure to clear this bit every time an interrupt exception request has occurred. If this bit is not cleared, an interrupt exception request will not be cleared and, as a result, an interrupt request continues. Also, since it requires a maximum of 20 HCLKs for the clear operation of this bit, be sure to clear the interrupt controller by allocating at least 20 HCLKs after clearing this bit.

WDTST	Description
0	A watchdog timer interrupt has not occurred.
1	A watchdog timer interrupt has occurred.

- **IVTIST** (Bit 5)
  This bit indicates that an interrupt by the watchdog timer has occurred.
  The bit can be cleared to "0" by writing "1" by a program.

[Note]
Be sure to clear this bit every time an interrupt exception request has occurred. If this bit is not cleared, an interrupt exception request will not be cleared and, as a result, an interrupt request continues. Also, since it requires a maximum of 20 HCLKs for the clear operation of this bit, be sure to clear the interrupt controller by allocating at least 20 HCLKs after clearing this bit.

IVTST	Description
0	An interval timer interrupt has not occurred.
1	An interval timer interrupt has occurred.

13.3 Operational Description

The watchdog timer (WDT) is a function that, by periodically clearing the WDT to "0" by the program, can generate an interrupt or system reset when the WDT perceives that a program is running out of control and, as a result, the WDT overflows. If the timer is not used as a watchdog timer, it can be used as an interval timer.

13.3.1 Operating Mode

The watchdog timer of this LSI has two types of operating modes: Interval Timer mode and Watchdog Timer mode, one of which is selected by setting the ITM bit of the WDTBCON register. Also, by setting the WDCLK[1:0] bit field of the WDTBCON register, the division ratio can be chagned for the watchdog timer.

13.3.2 Interval Timer Mode

Interval Timer mode is set by setting the ITM bit and ITEN bit of the WDTBCON register. Then, counting starts when 0x3C is written into the WDTCON register. After an operation starts, the timer is cleared to "0" by writing 0x3C. If the timer is not cleared and an overflow occurs, an overflow interrupt request is generated. A system reset signal is not generated in Interval Timer mode.

13.3.3 Watchdog Timer Mode

Watchdog Timer mode is set by clearing the ITM bit of the WDTBCON register to "0". Then, counting starts when 0x3C is written into the WDTCON register. The count operation of the watchdog timer is stopping until 0x3C is written into the WDTCON register. After an operation starts, the watchdog timer is cleared to "0" by alternately writing 0xC3 and 0x3C. In this mode, program so as not to generate an overflow. If the timer counter is not reset by a program runaway and therefore it overflows, an interrupt or system reset occurs.

13.3.4 Starting the Watchdog Timer

A count operation starts when 0x3C is written into the WDTCON register after an operating mode and others are set in the WDTBCON register. The count operation of the watchdog timer stops until 0x3C is written. Thereafter, the timer is cleared to "0" by alternately writing 0xC3 and 0x3C when in Watchdog Timer mode, or by writing 0x3C when in Interval Timer mode. If the timer is not cleared to "0" and thus an overflow occurs, a system reset or interrupt signal is generated according to the set value of the OFINTMODE bit of the WDTBCON register. However, only an interrupt signal is generated in Interval Timer mode.
The operation of the watchdog timer continues even after the CPU enters Halt mode. To stop the operation of the watchdog timer before entering Halt mode, set "1" in the ITM bit and "0" in the ITEN bit of the WDTBCON register. By setting these values, the watchdog timer enters Interval Timer mode and is placed in the count stop state. To resume the operation of the timer as the watchdog timer, write "0" into the ITM bit. By writing this value, the timer returns to Watchdog Timer mode from Interval Timer mode and resumes counting. The operation of the watchdog timer also stops in Standby mode.

Chapter 14

Timers

Chapter 14 Timers

14.1 Overview

This LSI has a built-in 1-channel 16-bit system timer in the CPU platform (μPLAT) and a built-in 3-channel 16-bit auto reload timer as a peripheral (APB module).

These timers have the following features:
As the timer clock (CCLK), a 7.5 MHz fixed frequency is input, and frequency dividing is performed inside this module.

- System timer
  - 16-bit timer counter
  - Generation of an interrupt by an overflow
  - Interval mode
  - Clock periods that can be set when CCLK = 7.5 MHz: 2.133 μs to 139.5 ms
- Auto reload timer
  - Full-channel 16-bit timer counter
  - Generation of an interrupt using the Compare function
  - A clock can be set independently for each channel.
  - One Shot or Interval mode can be set for each channel.
  - Clock periods that can be set when CCLK = 7.5 MHz: 0.133 μs to 2.237 ms
  - Clock periods that can be set when CCLK = 2.048 kHz: 0.488 ms and higher (when the CPU clock is in 32 kHz mode)

14.1.1 Configuration

Figure 14-1 shows the configuration of the system timer and Figure 14-2 the configuration of the auto reload timer.

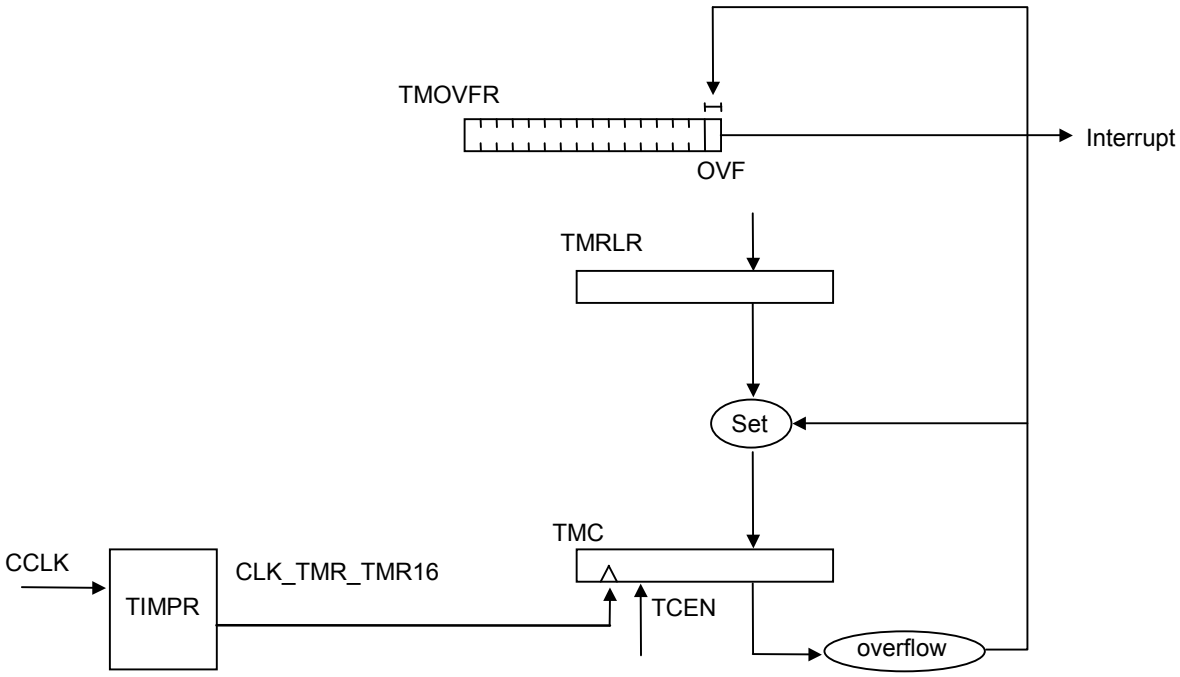


Figure 14-1 System Timer Block Diagram

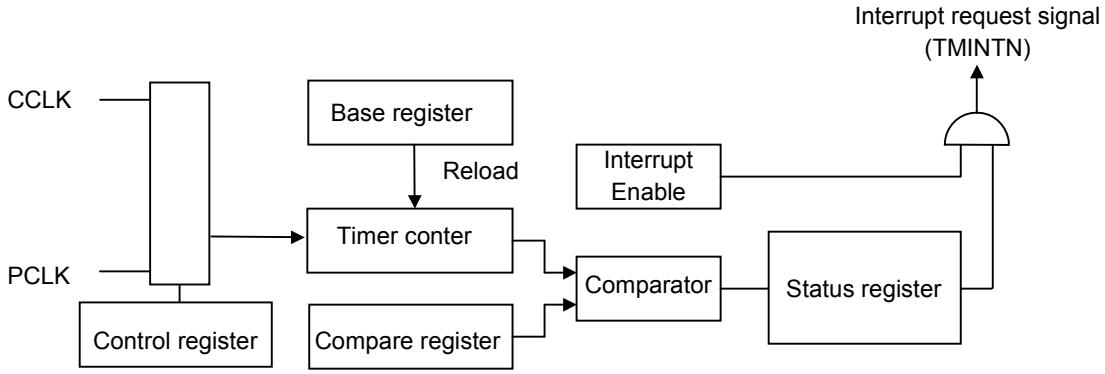


Figure 14-2 Auto Reload Timer Block Diagram

14.1.2 Register List

• μPLAT-System Timer

Address	Register name	Symbol	R/W	Initial value
0xB800_1004	System timer enable register	TMEN	R/W	0x0000_0000
0xB800_1008	System timer reload register	TMRLR	R/W	0x0000_0000
0xB800_1010	System timer overflow register	TMOVFR	R/W	0x0000_0000

• APB Timer

Address	Register name	Symbol	R/W	Initial value
0xB7F0_0000	Timer 0 control register	TIMECNTL0	R/W	0x0000_0000
0xB7F0_0004	Timer 0 base register	TIMEBASE0	R/W	0x0000_0000
0xB7F0_0008	Timer 0 counter register	TIMECNT0	R	0x0000_0000
0xB7F0_000C	Timer 0 compare register	TIMECMP0	R/W	0x0000_FFFF
0xB7F0_0010	Timer 0 status register	TIMESTAT0	R/W	0x0000_0000
0xB7F0_0020	Timer 1 control register	TIMECNTL1	R/W	0x0000_0000
0xB7F0_0024	Timer 1 base register	TIMEBASE1	R/W	0x0000_0000
0xB7F0_0028	Timer 1 counter register	TIMECNT1	R	0x0000_0000
0xB7F0_002C	Timer 1 compare register	TIMECMP1	R/W	0x0000_FFFF
0xB7F0_0030	Timer 1 status register	TIMESTAT1	R/W	0x0000_0000
0xB7F0_0040	Timer 2 control register	TIMECNTL2	R/W	0x0000_0000
0xB7F0_0044	Timer 2 base register	TIMEBASE2	R/W	0x0000_0000
0xB7F0_0048	Timer 2 counter register	TIMECNT2	R	0x0000_0000
0xB7F0_004C	Timer 2 compare register	TIMECMP2	R/W	0x0000_FFFF
0xB7F0_0050	Timer 2status register	TIMESTAT2	R/W	0x0000_0000

14.2 Register Description

14.2.1 System Timer Enable Register (TMEN)

The TMEN register is used to set the start/stop of the count operation of the timer counter (TMC).
Read/Write operation is enabled for the TMEN register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMEN	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	TCEN
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8001004
Access: R/W
Access size: 32 bits

[Note]
When successive write operations are performed to the system timer enable register (TMEN) when the timer clock (CCLK) is slower than the bus clock (HCLK), the data written first may not take effect but the data written last may take effect since the write operations synchronize with the timer clock. To perform successive write operations normally, allocate an interval of n × HCLK + 32 × CCLK (n = 16 × HCLK frequency/CCLK frequency), then perform the next write access. This resolves the above problem. The count clock of the system timer is fixed to 1/16 of the CCLK (7.5 MHz).

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **TCEN** (Bit 0)
  This bit controls the start/stop of the system timer.

TCEN	Description
0	Stops the system timer.
1	Starts the system timer.

14.2.2 System Timer Reload Register (TMRLR)

The TMRLR sets the reload value of the timer counter (TMC). When a reload value is written into this register, the same value is also written into the TMC.
Read/Write operation is enabled for the TMRLR register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMRLR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMRLR[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8001008
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
Do not perform successive write operations to the TMRLR. To write successively, allocate an interval of n × HCLK + 79 × CCLK between write operations.

14.2.3 System Timer Overflow Register (TMOVFR)

The TMOVFR generates a timer interrupt when the timer counter (TMC) generates an overflow, and also sets the OVF bit to "1". At this point, a system timer overflow interrupt is also generated.
Read/Write operation is enabled for the TMOVFR register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMOVFR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	OVF
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8001010
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **OVF** (Bit 0)
  This bit is used as a flag to indicates whether or not an overflow has occurred. This bit is cleared by writing "1". If "0" is written, it is ignored.

OVF	Description
0	An overflow has not occurred.
1	An overflow has occurred.

[Note]
Be sure to clear this bit every time an interrupt exception request has occurred. If this bit is not cleared, an interrupt exception request will not be cleared and, as a result, an interrupt request continues.

14.2.4 Timer Control Registers (TIMECNTL0–2)

The TIMECNTL0–2 registers control the operation of the auto reload timer. They are used to select the operating clock of the auto reload timer, enable/disable interrupts, and set the start/stop of the auto reload timer and the operating mode of the auto reload timer.
Read/Write operation is enabled for the TIMECNTL0–2 registers through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMECNTL0 to 2	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	CLKSEL[2:0]			IE	START	—*	—*	MODE
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7F00000 (CH0), 0xB7F00020 (CH1), 0xB7F00040 (CH2)
Access: R/W
Access size: 32 bits

[Note]
If the clock period of the bus clock (HCLK) is longer than that of the timer clock (CCLK = 7.5 MHz) by at least 8 times and the settings of the CLKSEL are changed before or after starting the auto reload timer, the count value may advance excessively by a maximum of "6" immediately after starting the auto reload timer. When using the auto reload timer under the above condition, the auto reload timer operates normally by changing the settings of the CLKSET before starting the timer by the START bit.
The setup procedure is as follows (in the case of TIMECNTL0 ⇐ 0x0058):
1. Set the CLKSET only (TIMECNTL0 ⇐ 0x0040).
2. Set the same CLKSEL value that has been set in step 1 above, then set a value to START (also to IE and MODE), thereby starting the auto reload timer (TIMECNTL0 ⇐ 0x0058).

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **MODE** (Bit 0)
  This bit specifies the operating mode of the auto reload timer .

MODE	Description
0	Interval mode
1	One Shot mode

- **START** (Bit 3)
  This bit controls the start/stop of the auto reload timer .

START	Description
0	Stops the auto reload timer.
1	Starts the auto reload timer.

- **IE** (Bit 4)
  This bits enables/disables interrupts.

IE	Description
0	Disables interrupts.
1	Enables interrupts.

- **CLKSEL** (Bit 5 to 7)
  This bit field selects the operation clock of the timers.

CLKSEL			Description
7	6	5	
0	0	0	CCLK(= 7.5 MHz or 2.048 kHz)
0	0	1	CCLK/2
0	1	0	CCLK/4
0	1	1	CCLK/8
1	0	0	CCLK/16
1	0	1	CCLK/64
1	1	0	CCLK/128
1	1	1	CCLK/256

14.2.5 Timer Base Registers (TIMEBASE0–2)

The TIMEBASE0–2 registers set the timer value at the time the operation of the timer starts.
Each value set in these registers is also set in the timer counter (TIMECNT0–2) registers at the same time.
Read/Write operation is enabled for the TIMEBASE0–2 registers through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMEBASE0 to 2	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMEBASE[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7F00004 (CH0), 0xB7F00024 (CH1), 0xB7F00044 (CH)
Access: R/W
Access size: 32 bits

14.2.6 Timer Counter Registers (TIMECNT0–2)

The TIMECNT0–2 registers are 16-bit up counters. The current auto reload timer value can be read from each of these registers.
Only Read operation is enabled for the TIMECNT0–2 registers through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMECNT0 to 2	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMECNT[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7F00008 (CH0), 0xB7F00028 (CH1), 0xB7F00048 (CH2)
Access: R
Access size: 32 bits

14.2.7 Timer Compare Registers (TIMECMP0–2)

The set value in each of the TIMECMP–2 registers is compared with the auto reload timer value. If they match, an interrupt is generated, and the set values in the TIMEBASE0–2 registers are loaded into the TIMECNT0–2 registers.
Read/Write operation is enabled for the TIMECNT0–2 registers through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMECMP0 to 2	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMECMP[15:0]															
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Address: 0xB7F0000C (CH0), 0xB7F0002C (CH1), 0xB7F0004C (CH2)
Access: R/W
Access size: 32 bits

14.2.8 Timer Status Registers (TIMESTAT0–2)

The TIMESTAT0–2 registers indicate whether the auto reload timer value and the compared value have matched.
Read/Write operation is enabled for the TIMESTAT0–2 registers through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMESTAT0 to 2	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	STATUS
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7F00010 (CH0), 0xB7F00030 (CH1), 0xB7F00050 (CH2)
Access: R/W
Access size: 32 bits

*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **STATUS** (Bit)
  This bit is used as a flag to indicate whether the auto reload timer value and the compared value have matched. This bit is cleared by writing "1". If "0" is written in this bit, it is ignored.

STATUS	Description
0	No match
1	Match

14.3 Operational Description

14.3.1 System Timer

1. Reload Timer Operation
   (a) Specify a setting value in the timer reload register (TMRLR). At this point, this value is also set in the timer counter (TMC).
   However, after this, do not perform successive write operations to the TMRLR during the period of n x HCLK + 79 x CCLK.
   (b) When the value of the timer enable register (TMEN) is "1", the timer counter (TMC) starts incrementing from the value set in the TMRLR. An interrupt is asserted by the next count clock after the counter value becomes "0xFFFF" and, at the same time, the value of the TMRLR is automatically set in the TMC, and the operation of the timer continues. The count clock of the timer is fixed to 1/16 of the CCLK (7.5 MHz).
2. Timer Interrupt (Overflow)
   (a) An interrupt is generated by an overflow of the TMC, and the OVF bit of the timer overflow register (TMOVFR) is set to "1".
   (b) The OVF bit of the TMOVFR is cleared by writing "1".

14.3.2 Auto Reload Timer

The auto reload timer has three channels, each of which has a unique timer counter. Also, a clock and operating mode can be set for each channel.

(1) One Shot mode
When the counter value matches the compare register, the base register is reloaded and then the opeation stops. At the same time, the START bit is cleared to "0". If an interrupt is enabled at this point, an interrupt request is generated. However, an interrupt request is not canceled automatically. It can be cleared by clearing the STATUS bit.

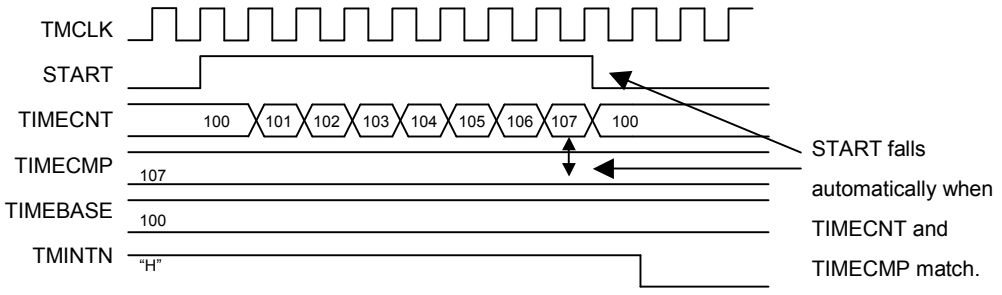


Figure 14-3 One Shot Mode

(2) Interval mode
When the counter value matches the value of the compare register, the base register is reloaded and incrementing is continued. The START bit is not cleared to "0" in this mode. Also, if an interrupt is enabled at this point, an interrupt request is generated. However, an interrupt request is not canceled automatically. It can be cleared by clearing the STATUS bit.

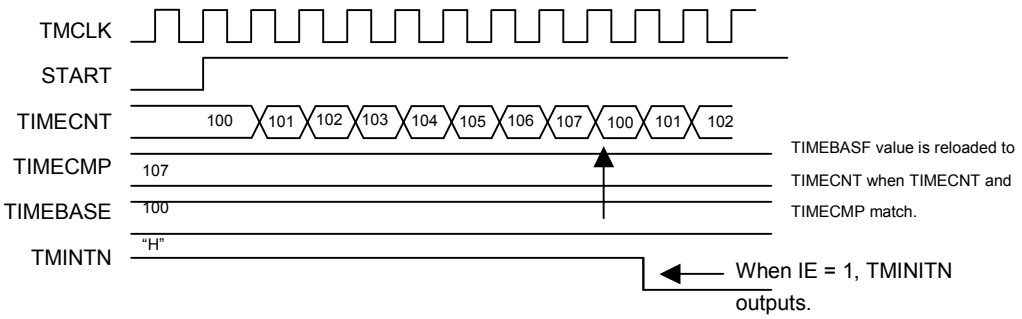


Figure 14-4 Interval Mode

14.3.3 Selecting a Clock and Starting the Application Timer

The clock for the application timer is selected by the CLKSEL[2:0] bit field of the timer control register (TIMECNTRL0-2). By this setting, an individual clock can be set for each of the three channels of the application timer. A clock can be selected from CCLK, CCLK/2, CCLK/4, CCLK/8, CCLK/16, CCLK/64, CCLK/128 and CCLK/256 according to the setting of CLKSEL[2:0].

The application timer is started by the START bit of the timer control register (TIMECNTRL0-2). In the case of One Shot mode, once the counter value matches the value of the compare register, the START bit is automatically cleared and the timer operation stops. In the case of Interval mode, the START bit is not automatically cleared.

[Note]
Do not change the clock settings while the timer is operating.

Chapter 15

PWM

Chapter 15 PWM

15.1 Overview

This LSI has 1-channel of pulse width modulation (PWM) function that can change duty at a predetermined cycle. The PWM output resolution is 16 bits. The CCLK is fixed to a frequency of 7.5 MHz, which is divided inside this module.

- The PWM counter clock is selectable from CCLK, CCLK/4, CCLK/16 and CCLK/32.
- The cycle can be set in 16-bit resolution using the above counter clock as the source.
- The duty can be set in the unit of the above counter clock cycle.

15.1.1 Configuration

Figure 15-1 shows the configuration of the PWM module.

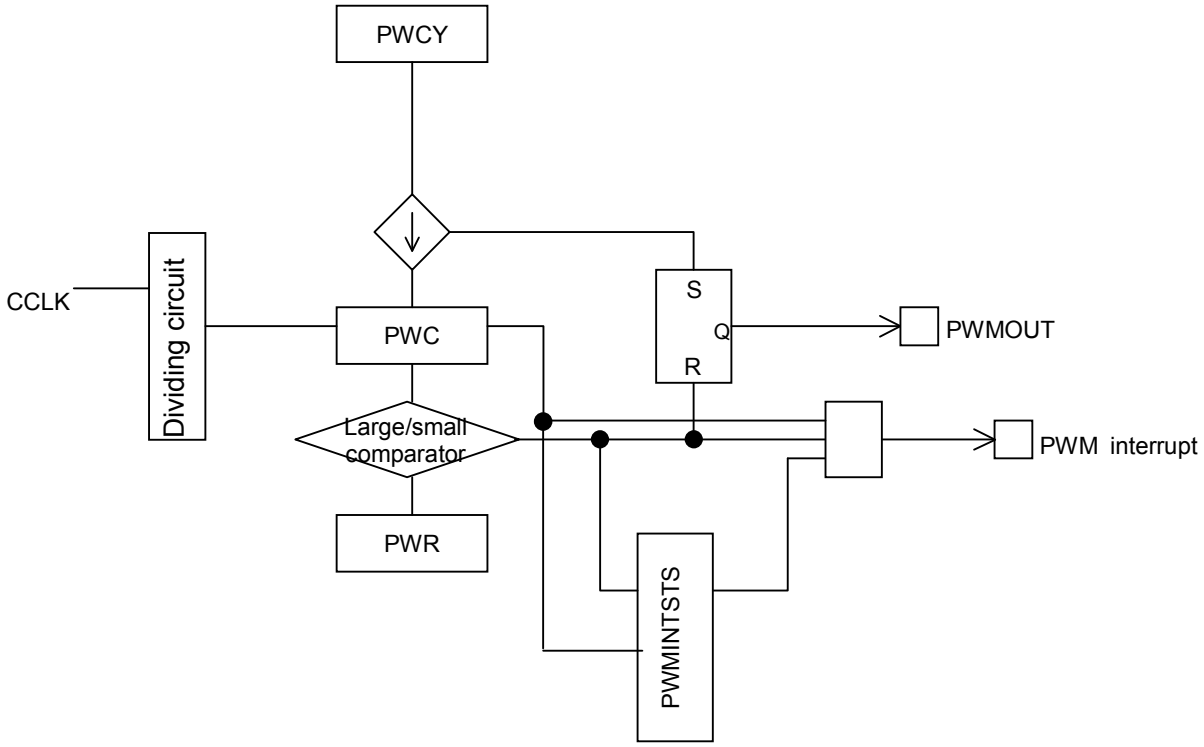


Figure 15-1 Configuration of the PWM Module

15.1.2 Pin List

Pin name	I/O	Description
PWMOUT	O	PWM output

15.1.3 Register List

Address	Register name	Symbol	R/W	Initial value
0xB7D0_0000	PWM register	PWR	R/W	0x0000_0000
0xB7D0_0004	PWM cycle register	PWCY	R/W	0x0000_0000
0xB7D0_0008	PWM counter	PWC	R/W	0x0000_0000
0xB7D0_000C	PWM0 control register	PWCON	R/W	0x0000_0000
……	<Reserved>	—	—	—
0xB7D0_003C	PWM interrupt status register	PWINTSTS	R/W	0x0000_0000

Access to the <Reserved> space is prohibited.

15.2 Register Description

15.2.1 PWM Register (PWR)

The PWR sets the duty value (high period of output).
The setting of the duty value is limited to within the range of the cycle set in the RWCY register.
Duty = ((PWR – PWCY + 1) / (65536 – PWCY)) × 100 [%]
(PWCY: See the PWM cycle register. This is the setting value of the PWM output cycle.)
Read/Write operation is enabled for the PWR register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWR[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D00000
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
Be sure to set a duty value in the PWR register that is smaller than the cycle value set in the PWCY register.

15.2.2 PWM Cycle Register (PWCY)

The PWCY register sets the cycle of PWM.

The formula of the PWM cycle:
T (pwmcyc) = (65536 – PWCY)/((Frequency division ratio set by PWCK) × PWMCLK) [sec]

[Example]
In the case of PWCY = 65535, PWCK = 0x01 (setting of 1/4), CCLK = 7.5 MHz
Tpwmcyc = $(65536 - 65535)/(1/4 \times 7.5 \times 10^6)$ [sec]
= 533.3 [nsec]

Read/Write operation is enabled for the PWCY register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWCY	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWCY[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D00004
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
Be sure to set a cycle in the PWCY register that is larger than the duty value set in the PWR register.

15.2.3 PWM Counter (PWC)

The PWM counter is a 16-bit up counter. When the PWM counter overflows, the value of the PWCY register is loaded into the PWM counter.
Read/Write operation is enabled for the PWM counter through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWC	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWC[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D00008
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.
When a value is written into the PWC register, the same value is also written into the PWCY register.

15.2.4 PWM Control Register (PWCON)

The PWCON register starts/stops the PWM counter, selects a count clock and specifies an interrupt source. Read/Write operation is enabled for the PWCON register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWCON	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWCON	—*	—*	—*	—*	—*	—*	—*	—*	PWCOV	INTIE	—*	—*	—*	PWCK		PW RUN
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D0000C
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.

[Description of Bits]

- **PWRUN** (Bit 0)
  This bit starts/stops the PWM counter register (PWC).

PWRUN	Description
0	Stops the PWC register.
1	Starts the PWC register.

- **PWCK** (Bits 1 - 2)
  This bit field sets the input clock (PWMCLK).

PWCK		Description
bit2	bit1	
0	0	CCLK(=7.5MHz)
0	1	1/4CCLK
1	0	1/16CCLK
1	1	1/32CCLK

- INTIE (Bit 6)
  This bit controls the generation of interrupts.

INTIE	Description
0	Disables interrupts.
1	Enables interrupts.

- **PWCOV** (Bit 7)
  This bit sets the interrupt generation condition of the PWC.

PWCOV	Description
0	When PWC = PWR (fall of PWM output)
1	When PWC overflows (rise of PWM output)

15.2.5 PWM Interrupt Status Register (PWINTSTS)

This register indicates the interrupt status of the PWM output.
It indicates the interrupt status when read, and cancels the interrupt status when written.
Read/Write operation is enabled for the PWMINTSTS register through program control.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWINTSTS	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	INTS	—*	—*	—*	—*	—*	—*	—*	INTCLR
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D0003C
Access: R/W
Access size: 32 bits

[Note]
Writing to bit 8 of this register is ignored.
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read, and always write "0" at write operation.

[Description of Bits]

- **INTCLR** (Bit 0)
  This pin clears an interrupt. The INTS bit is cleared by writing "1" to this bit. If "0" is written to this bit, it is ignored. "0" is always read when read.

- **INTS** (Bit 8)
  This bit is used as a flag to indicate the interrupt status of the channel.

INTS	Description
0	An interrupt has not occurred.
1	An interrupt has occurred.

15.3 Operational Description

15.3.1 Operation of PWM

The counter clock of PWC is set by the PWCK bit of the PWCON register as shown in the table below.

CCLK frequency: 7.5 MHz

PWCK[1:0]	Gear value	PWM counter cycle [μs]
00	1/1	0.133
01	1/4	0.533
10	1/16	2.133
11	1/32	4.267

When the PWM counter overflows, the value of the PWCY register is loaded into the PWM counter. After counting up to 65535, the PWM counter overflows in the next clock, and the value of the PWCY register is loaded into the PWM counter. Therefore, as the cycle of PWM, the value of "PWM counter cycle × (65536 – PWCY)" above is set. The value of the PWCY register can be set within the range between 0 and 65535.

The PWR register controls the high level segment of the PWMOUT pin. It outputs a high level signal until the value of the PWM counter matches the value of the PWR register. The output changes to low level in the next clock.
Since the PWM counter starts incrementing from the value of the PWCY register, the value of the PWR register must be larger than the value of the PWCY register. Therefore, the value of the high level segment is PWM counter cycle × ((PWR – PWCY) + 1).

PWM is started by setting the PWRUN bit to "1". When the PWRUN bit is set to "1", the PWM counter starts incrementing and, at the same time, the output flip-flop is set to "1", and a high level signal is output to the PWMOUT pin. The PWM counter continues incrementing, and when the value matches the value of the PWR register, the output flip-flop is cleared to "0", and a low level signal is output to the PWMOUT pin. If INTIE = 1 and PWCOV = 0 at this point, an interrupt request is notified.
Thereafter, this operation is repeated until the PWRUN bit is cleared to "0", and a duty-controlled waveform is then output from the PWMOUT pin.

[Note]
Depending on the selected count clock of the PWM counter, the PWM output duty (one cycle only) immediately after starting PWM may become short.
If the value of the PWCY register is 0x0000 and the value of the PWR register is 0x0000, the duty output becomes 1/65536. When the value of the PWR register increases, the output duty (high level) increases. If the value of the PWR register is 0xFFFF, the duty output becomes 65536/65536, a duty of 100%. Note that the duty output of 0/65536, i.e., a duty of 0%, is not supported by the PWM function.
After the occurrence of a PWM interrupt, if the next interrupt has occurred before clearing the current interrupt, that next interrupt will be degenerated.

15.3.2 Operation When the Setting of the PWC/PWR/PWCY Register Is Changed During a PWM Operation

If the setting of the PWR register is changed when the PWRUN bit is "1" and the PWM module is in operation, since the PWR value itself that is compared with the value of PWC is so configured that it is updated when an overflow occurs, the PWM module operates according to the value of the PWR before change for the cycle in which the setting is changed, and the PWM operates using the value of the PWR register after change from the next cycle.

Also, if the value of the PWCY register is changed when the PWM module is in operation, the value of the PWCY register is loaded into PWC when an overflow occurs. Therefore, the PWM module operates according to the value of the PWCY before change for the cycle in which the setting is changed, and the PWM operates using the value of the PWCY register after change from the next cycle.

If the value of PWC is changed while the PWM module is in operation, the value of PWC/PWCY is rewritten. Therefore, basically, do not rewrite it.

15.3.3 How to Use the PWM Module

1) Make sure that the PWM module is in the stop state.
2) Set the PWCY/PWR registers. Set the initial value in PWC (generally, it is the same as the value of the PWCY register).
3) Set the PWRUN bit to "1" to enable operation.
4) The PWM module starts operating.
5) To change the cycle/duty of the PWM module, finish writing into the PWCY/PWR registers between one overflow and the next.
6) The PWM module operates using the cycle/duty after change from the next cycle.

15.3.4 Example of Timing

Figure 15-2 shows an example of an output operation, and Figure 15-3 shows an example of PWM output change timing.

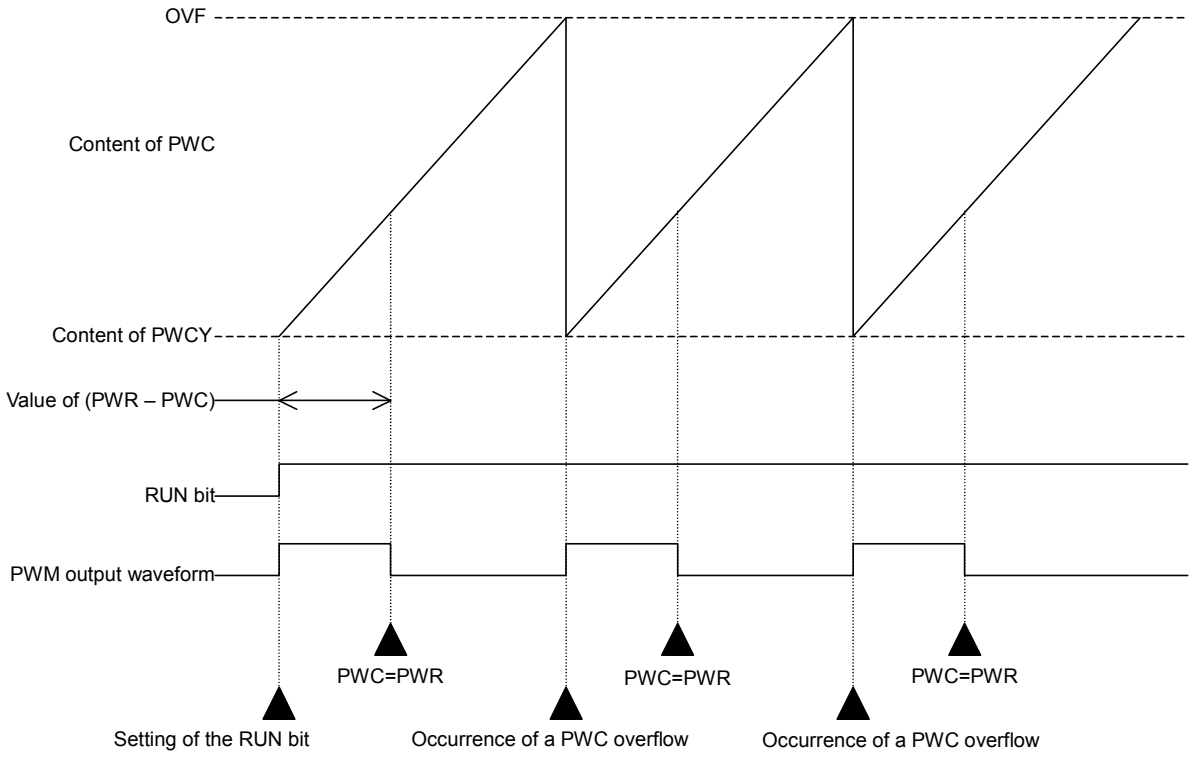
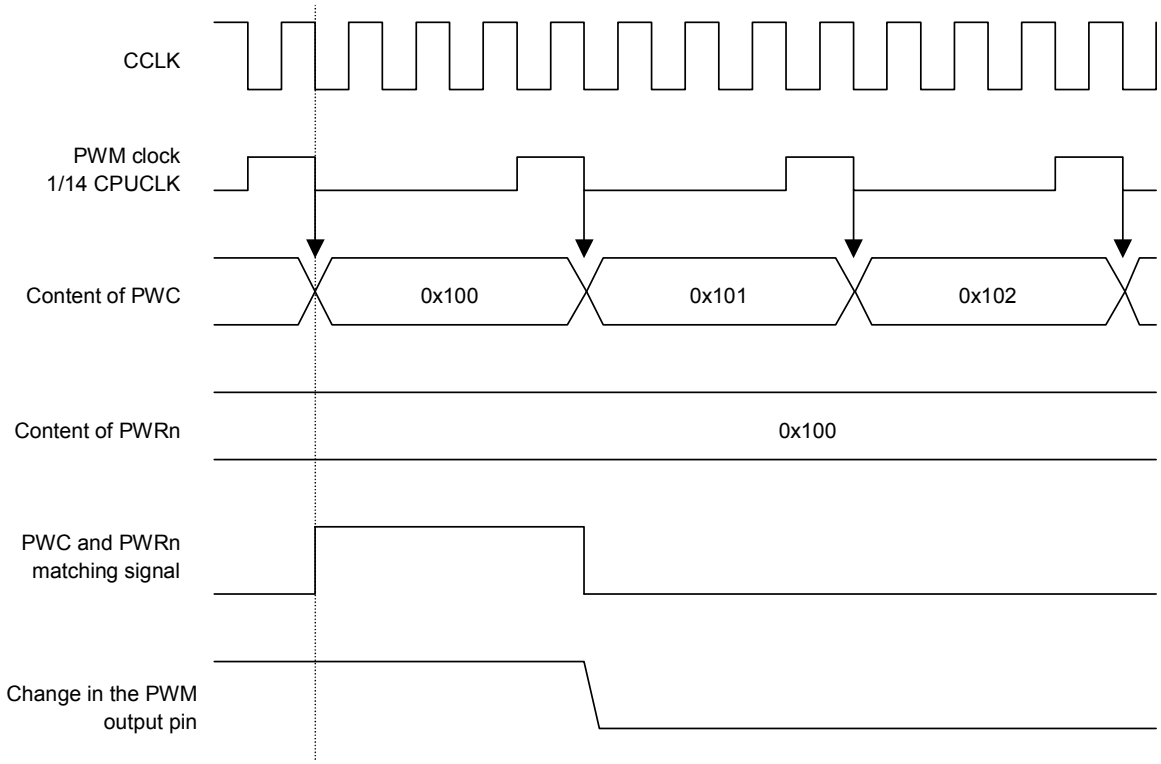


Figure 15-2 Example of Output Operation


CCLK
PWM clock
1/14 CPUCLK
Content of PWC
0x100
0x101
0x102
Content of PWRn
0x100
PWC and PWRn
matching signal
Change in the PWM
output pin

Chapter 16

μPLAT-SIO (UART)

Chapter 16 μPLAT-SIO (UART)

16.1 Overview

The asynchronous serial interface is a serial port that performs synchronization in character units and transfers data. Various parameters are available to set the communication speed by a dedicated baud rate generator using baud rate clocks independent of bus clocks, the data length, addition of the stop bit, and addition of a parity bit. The source clock frequency for generating baud rate clocks is 30 MHz.

Features:

- Full duplex start-stop synchronization method
- Sampling rate = Baud rate x 16 samples
- Character length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or no parity
- Error detection: Parity, framing, overrun
- Loopback function: ON/OFF, parity error, framing error, overrun error forcibly added
- Baud rate generation: Built-in dedicated baud rate generator (8-bit counter), independent of the bus clock
- The internal baud rate clock stops during Halt mode.

16.1.1 Configuration

Figure 16-1 shows the configuration of the asynchronous serial interface.

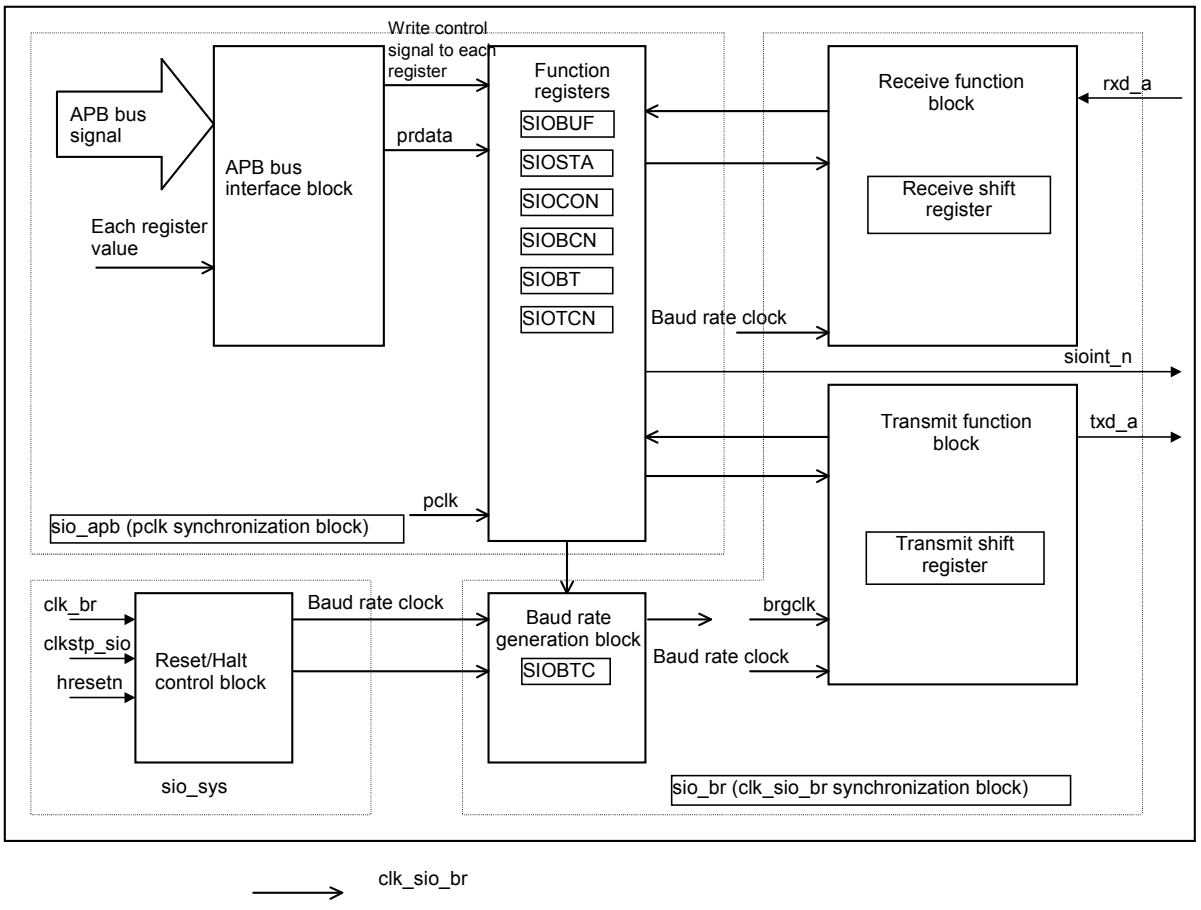


Figure 16-1 Block Configuration

16.1.2 Pin List

Pin name	I/O	Function
STXD	O	SIO transmit data output. Assigned as a secondary function.
SRXD	I	SIO receive data input. Assigned as a secondary function.

16.1.3 Control Register List

Address	Register name	Symbol	R/W	Size	Initial value
0xB800_2000	Transmit/receive buffer register	SIOBUF	R/W	32	0x0000_0000
0xB800_2004	SIO status register	SIOSTA	R/W	32	0x0000_0000
0xB800_2008	SIO control register	SIOCON	R/W	32	0x0000_0000
0xB800_200C	Baud rate control register	SIOBCN	R/W	32	0x0000_0000
0xB800_2010	(Reserved)		—	—	—
0xB800_2014	Baud rate timer register	SIOBT	R/W	32	0x0000_0000
0xB800_2018	SIO test control register	SIOTCN	R/W	32	0x0000_0000

16.2 Control Register Description

16.2.1 Transmit/Receive Buffer Register (SIOBUF)

The SIOBUF register holds transmit/receive data. The SIOBUF register can be read/written by program; it has a dual structure in which it operates differently for read and write operations. The SIOBUF register functions as a receive buffer for a read operation, and as a transmit buffer for a write operation. When a receive operation finishes, the content of the receive shift register is transferred to the receive buffer and, at the same time, a receive interrupt request is generated. The content of the receive buffer is held until the next receive operation finishes. The value is set to 0x0000 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOBUF	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	SIOBUF[7:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8002000
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

16.2.2 SIO Status Register (SIOSTA)

The SIOSTA register holds the transmission and reception ready flags for the asynchronous serial port (SIO) and the flag that indicates the status of an error detected during a receive operation. The SIOSTA register can be read/written by program. The lower three bits are updated by a reception error detected when a receive operation finishes. Once this bit field is set to "1" (reception error detection), it will not be cleared to "0" even if that error is not generated when the next receive operation finishes. Therefore, it is necessary to clear it to "0" by a program. It is also necessary to clear the transmission and reception ready flags to "0" by program. These flags are cleared to "0" when "1" is written into them. The status does not change when "0" is written, however. The value is set to 0x0000 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOSTA	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIOSTA	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	TRIRQ	RVIRQ	—*	PERR	OERR	FERR
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8002004
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **FERR** (Bit 0)
  The FERR bit is used as a flag to indicate whether or not a framing error has occurred. This bit is set to "1" as it is determined that frame synchronization has not been performed correctly if RXD = "0" is detected at a timing when a stop bit is received.

FERR	Description
0	A framing error has not occurred.
1	A framing error has occurred.

- **OERR** (Bit 1)
  The OERR bit is used as a flag to indicate whether or not an overrun error has occurred. When a receive operation finishes, this bit is set to "1" if the data received previously is not read from the SIOBUF register. However, even in this case, newly received data is loaded into the SIOBUF register from the receive shift register. The value of received data is not guaranteed in this situation.

OERR	Description
0	An overrun error has not occurred.
1	An overrun error has occurred.

- **PERR** (Bit 2)
  This bit is used as a flag to indicate whether or not a parity error has occurred. This bit is set to "1" when the parity of received data and the parity attached to data are compared and they do not match.

PERR	Description
0	A parity error has not occurred.
1	A parity error has occurred.

- **RVIRQ** (Bit 4)
  The RVIRQ bit is used as a flag to indicate the reception ready state. The reception ready state is set when the received data in the receive buffer (SIOBUF) is updated, and the RVIRQ bit is set to "1". Because the RVIRQ bit is not automatically cleared to "0" by executing interrupt processing or loading received data, it is necessary to clear it to "0" by program. This bit is cleared to "0" when "1" is written into it. The status does not change when "0" is written, however. Even if any of framing, overrun and parity errors has been detected upon completion of a receive operation, the SIOBUF register is updated with received data, and the RVIRQ bit is set to "1".

RVIRQ	Description
0	The reception ready state has not occurred.
1	The reception ready state has occurred.

- **TRIRQ** (Bit 5)
  The TRIRQ bit is used as a flag to indicate the transmission ready state. The transmission ready state is set when the transmit data loaded into the transmit buffer (SIOBUF) is transferred to the transmit shift register, and the TRIRQ bit is set to "1". Because the TRIRQ bit is not automatically cleared to "0" by executing interrupt processing or loading transmit data, it is necessary to clear it to "0" by program. This bit is cleared to "0" when "1" is written into it. The status does not change when "0" is written, however.

TRIRQ	Description
0	The transmission ready state has not occurred.
1	The transmission ready state has occurred.

16.2.3 SIO Control Register (SIOCON)

The SIOCON register controls the transmit and receive operations of the asynchronous serial port (SIO). The SIOCON register can be read/written by program. The content of the SIOCON register is set to 0x0000 at the time of a system reset, and it is initialized to have the character length of eight bits, the stop bit length to two bits, and no parity.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOCON	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIOCON	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	TSTB	EVN	PEN	LN
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8002008
Access: R/W
Access size: 32 bits

[Note]
Be sure to change the content of the SIOCON register only after a transmit or receive operation is complete. If it is changed before a transmit or receive operation is complete, operation cannot be guaranteed.
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **LN** (Bit 0)
  The LN bit specifies the character length of the transmit/receive operation of the SIO.

LN	Description
0	8-bit character length
1	7-bit character length

- **PEN** (Bit 1)
  The PEN bit specifies whether or not there is a parity bit during transmission/reception. When this bit is set to "1", a parity bit is added and a transmit operation is performed. When receiving, the received parity bit and the parity of data are compared.

PEN	Description
0	No parity bit.
1	Parity bit.

- **EVN** (Bit 2)
  The EVN bit specifies the logic (even/odd) of the parity bit during transmission/reception.

EVN	Description
0	Odd parity
1	Even parity

- **TSTB** (Bit 3)
  The TSTB bit specifies the stop bit length during transmission/reception.

TSTB	Description
0	2 stop bits
1	1 stop bit

16.2.4 Baud Rate Control Register (SIOBCN)

The SIOBCN register controls the start and stop of the count operation of the baud rate timer counter (SIOBTC). The SIOBON register can be read/written by D program. The content of the SIOBCN register is 0x0000 at the time of a system reset, and the count operation of the SIOBTC stops.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOBCN	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *	—*	— *
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	BGRUN	—*	—*	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB800200C
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **BGRUN** (Bit 4)
  The SIOBTC bit specifies the start and stop of the SIOBTC.

BGRUN	Description
0	Stops a count operation.
1	Starts a count operation.

16.2.5 Baud Rate Timer Register (SIOBT)

The SIOBT register holds the value loaded into the baud rate timer counter (SIOBTC). The SIBOT register can be read/written by program. The content of the SIOBT register is loaded into the SIOBTC register when an overflow occurs in the SIOBTC register. The value is set to 0x0000 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOBT	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	SIOBT[7:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8002014
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

16.2.6 SIO Test Control Register (SIOTCN)

The SIOTCN register is provided to simplify the software debugging operation. The SIOTCN register can be read/written by program. The value is set to 0x0000 at the time of a reset. Set a value of 0x0000 for regular operations.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOTCN	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIOTCN	—*	—*	—*	—*	—*	—*	—*	—*	LBTST	—*	—*	—*	—*	—*	WPERR	MFERR
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB8002018
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **MFERR** (Bit 0)
  The MFERR bit specifies the forcible generation of a framing error during loopback testing.

MFERR	Description
0	Does not generate a framing error.
1	Generates a framing error.

- **MPERR** (Bit 1)
  The MPERR bit specifies the forcible generation of a parity error during loopback testing.

MPERR	Description
0	Does not generate a parity error.
1	Generates a parity error.

- **LBTST** (Bit 7)
  The LBTST bit selects the loopback of transmit signal output to the receive function.
  When this bit is set to "1", a transmit signal is output, which is then input as a receive signal.

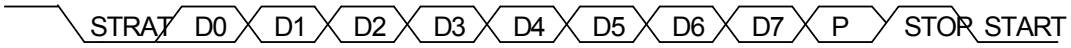
LBTST	Description
0	Loopback is not performed.
1	Loopback is performed.

16.3 Operational Description

Set the SIOCON register according to the character length of data to be transmitted/received, the stop bit length and the presence of a parity bit.

Figure 16-2 shows an example of SIOCON register settings and the format of a transmit/receive frame.

(1) Parity, 8 bits, 1 stop bit (settings of TSTB = "1," PEN = "1," LN = "0")



(2) No parity, 8 bits, 1 stop bit (settings of TSTB = "1," PEN = "0," LN = "0")

(3) Parity, 8 bits, 2 stop bits (settings of TSTB = "0," PEN = "1," LN = "0")

(4) Parity, 7 bits, 1 stop bit (settings of TSTB = "1," PEN = "1," LN = "1")

Figure 16-2 Examples of Frame Format

16.3.1 Data Transmission

A transmit operation is started by writing 8-bit data into the SIOBUF register. While the data length is set to 7 bits, bit 7 (MSB) of data written is ignored. The data written into the SIOBUF register is transferred to the transmit shift register and, at the same time, a transmission ready interrupt request is generated, then the TRIRQ flag of the SIOSTA register is set to "1". At the point where the transmission ready state is set, the SIOBUF register is emptied, so data to be transmitted next can be written into the SIOBUF register. Thereafter, 1-frame data is transmitted in the order of a 1-bit start bit, 7 or 8-bit data, a parity bit and a 1 or 2-bit stop bit, according to the settings of the SIOCON register.

16.3.2 Data Reception

When a start bit is detected, a frame data receive operation is started according to the format of the frame set in the SIOCON register. The received data is transferred from the shift register to the transmit/receive buffer (SIOBUF) register at the timing the stop bit is received. A reception ready interrupt request is generated at the same time, and then the RVIRQ flag of the SIOSTA register is set to "1". Also, if an error has been detected, the PERR, OERR and FERR flags of the SIOSTA register are set to "1". Since the receive shift register is emptied at the point where received data is transferred to the SIOBUF register, it becomes possible to receive the next data.

16.3.3 Baud Rate Clock Generation

The blocks relating the generation of the baud rate are made up of the following registers.
Figure 16-3 shows a block diagram.

- Baud rate timer counter (SIOBTC: 8-bit counter)
- Baud rate timer register (SIOBT)
- Baud rate control register (SIOBCN)

The SIOBTC generates a baud rate clock when a overflow occurs. The content of the baud rate timer register (SIOBT) is loaded into the SIOBTC register at the same time, and the SIOBTC register continues to increment from the value loaded. When a value is written into the SIOBT register, the same value is also written into the SIOBTC register.

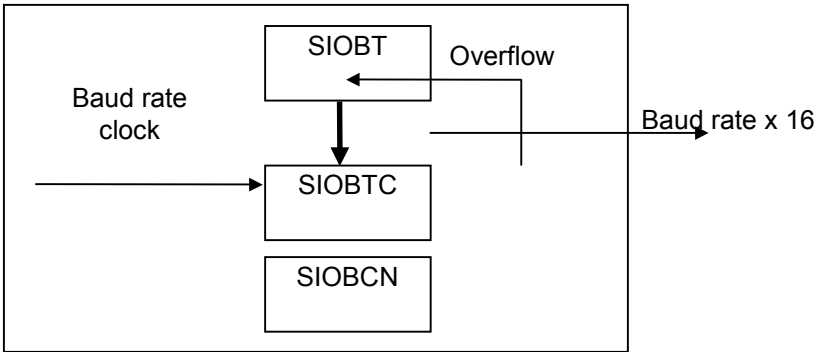


Figure 16- 3 Blocks Related to Baud Rate Generation

The baud rates are determined as follows:

1. Calculate D from the equation below, according to frequency B of the baud rate to be used and frequency $f_{(B)}$ of the baud rate clock. However, the value of D is limited between 0 and 255.

$$D = 256 - f_{(B)}/(16 \times B) \qquad (B = f_{(B)} \times 1/(256 - D) \times 1/16)$$

B: Baud rate
$f_{(B)}$: Counter clock frequency that has been input [Hz]
D: Reload value (0 to 255)

2. Set the calculated value of D in the SIOBT register.
3. Set the BGRUN bit of the SIOBCN to "1". By this, the SIOBTC register starts outputting baud rate clocks. Thereafter, once five baud rate clocks have elapsed, communication is enabled.

Since the input counter clock frequency $f_{(B)}$ is 30 MHz in the ML69(Q)6500, the following baud rates can be set:

f(B) = 30 MHz

Baud rate	Logical clock frequency	Actual clock frequency	Error [%]	Setting value of D
9600	153600	153846.2	-0.160	3D
19200	307200	306122.4	0.351	9E
38400	614400	612244.9	0.351	CF
56000	896000	909090.9	-1.461	DF
57600	921600	909090.9	1.357	DF
115200	1843200	1875000	-1.725	F0
230400	3686400	3750000	-1.725	F8
460800	7372800	7500000	-1.725	FC
921600	14745600	15000000	-1.725	FE

16.3.4 Reception Interrupt

An interrupt is generated when correct received data is set in the SIOBUF register or when a reception error (framing error, overrun error or parity error) has occurred, and the corresponding flag of the SIOSTA register is set to "1". Once asserted, this interrupt signal remains asserted, which is deasserted by clearing the interrupt source flag of the SIOSTA register to "0" by program. If assertion and deassertion are competed, assertion is valid.

16.3.5 Transmission Interrupt

An interrupt is generated when the transmit data stored in the SIOBUF register is set in the transmit shift register, and then the TRIRQ flag of the SIOSTA register is set to "1". Once asserted, this interrupt signal remains asserted, which is deasserted by clearing the TRIRQ flag of the SIOSTA register to "0" by program. If assertion and deassertion are competed, assertion is valid.

16.4 Notes on Use

1. Perform the initial settings of the baud rate in the order of the SIOBT register first and then the SIOBCN register. After setting the SIOBCN register, output is performed at stable baud rate cycles once five baud rate clocks have elapsed.
2. If the settings of the SIOCON register are changed after communication has started, operation cannot be guaranteed.
3. Perform interrupt processing during the transfer of one character.
4. In interrupt processing, perform read and write operations from/to the SIOBUF register only after clearing an interrupt source.
5. Do not write to the SIOBT register continuously. If it is necessary to continuously write to the SIOBT register, allocate five baud rate clocks in between.
6. To access each register from the bus, use word access.

Chapter 17

A/D Converter

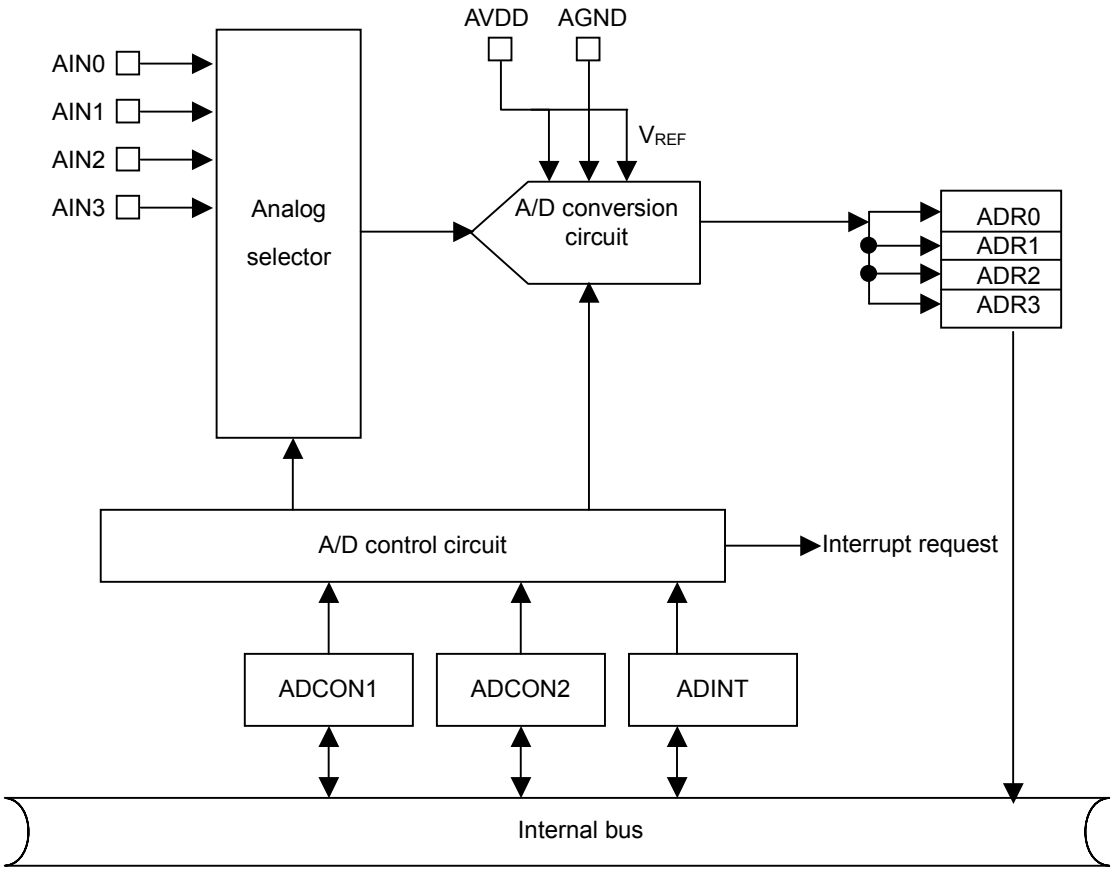
Chapter 17 A/D Converter

17.1 Overview

This LIS has a built-in 4-channel A/D converter for output at 10-bit resolution. As the operating mode of the A/D converter, only the Select mode in which a single selected channel is converted is supported. For conversion from analog quantity to digital quantity, the sequential comparison method with a sample and hold function is used. An interrupt can be generated by ending A/D conversion.

17.1.1 Configuration

Figure 17-1 shows the configuration of the A/D converter.



AIN0 to AIN3: Analog input pins
ADR0 to ADR3: A/D result registers (10 bits)
ADINT: A/D interrupt control register
ADCON1: A/D control register 1
ADCON2: A/D control register 2
AVDD: Analog VDD pin
AGND: Analog GND pin

Figure 17-1 Configuration of the A/D Converter

17.1.2 Pin List

Pin name	I/O	Function
AVDD	VDD	Power supply for the A/D converter
AIN[0]	I	Analog input port 0 for the A/D converter
AIN[1]	I	Analog input port 1 for the A/D converter
AIN[2]	I	Analog input port 2 for the A/D converter
AIN[3]	I	Analog input port 3 for the A/D converter
AGND	GND	Ground for the A/D converter

17.1.3 Control Register List

Address [H]	Register name	Symbol	R/W	Access size	Initial value [H]
……	<Reserved>	—	—	—	—
0xB600_0004	A/D control 1 register	ADCON1	R/W	32	0x0000_0000
0xB600_0008	A/D control 2 register	ADCON2	R/W	32	0x0000_0003
0xB600_000C	A/D interrupt control register	ADINT	R/W	32	0x0000_0000
……	<Reserved>	—	—	—	—
0xB600_0014	A/D result 0 register	ADR0	R/W	32	0x0000_0000
0xB600_0018	A/D result 1 register	ADR1	R/W	32	0x0000_0000
0xB600_001C	A/D result 2 register	ADR2	R/W	32	0x0000_0000
0xB600_0020	A/D result 3 register	ADR3	R/W	32	0x0000_0000

[Note]
No access to the "Reserved" space is allowed.
Do not perform a write operation to ADR0 to ADR3 during A/D conversion. Otherwise, the results of A/D conversion for all of ADR0 to ADR3 cannot be guaranteed.

17.2 Control Register Description

17.2.1 A/D Control 1 Register (ADCON1)

The ADCON1 register sets the operation of the A/D converter in the Select mode. The ADCON1 register can be read from/written to by programs. The value is set to 0x0000 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCON1	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCON1	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	STS	—*	ADSTM[2:0]		
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB6000004
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.
Because the ADCON1 register performs synchronization between the bus clock (HCLK) and the counter clock (CCLK), put an interval of 2HCLK + 2CCLK between one write operation and the next if write operations are performed successively.

[Bit Description]

- **ADSTM[2:0]** (bits 2 to 0)
  These bits specify the channels for A/D conversion.

ADSTM			Channel for use
2	1	0	
0	0	0	ch0
0	0	1	ch1
0	1	0	ch2
0	1	1	ch3
1	0	0	Setting prohibited
1	0	1	Setting prohibited
1	1	0	Setting prohibited
1	1	1	Setting prohibited

  Change channel selection while the conversion of the A/D converter is stopped. Changing channel selection is only valid when STS ="0."

- **STS** (bit 4)
  This bit specifies the start or stop of A/D conversion.

STS	Description
0	Stops A/D conversion.
1	Starts A/D conversion.

When A/D conversion is completed, the STS bit is automatically cleared by hardware.

17.2.2 A/D Control 2 Register (ADCON2)

The ADCON2 register sets the frequency of the operating clock for the A/D converter.
The ADCON2 register can be read from/written to by programs. The value is set to 0x0003 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCON2	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCON2	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	ACKSEL[1:0]	
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Address: 0xB6000008
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.
If the setting of the ACKSEL bits is changed during A/D conversion, operation cannot be guaranteed.

[Bit Description]

- **ACKSEL[1:0]** (bits 1 and 0)
  These bits select the frequency of the operating clock for the A/D converter.

ACKSEL		Channel for use
1	0	
0	0	Reserved
0	1	CCLK/2
1	0	CCLK/4
1	1	CCLK/8

[Note]
ACKSEL and CCLK
The CCLK is fixed to 7.5 MHz, and the conversion time per channel is as follows:

CCLK	ACKSEL		
(MHz)	01	10	11
7.5	6.7 μs	13.3 μs	26.7 μs

17.2.3 A/D Interrupt Control Register (ADINT)

The ADINT register sets interrupts for the A/D converter. The ADINT register can be read from/written to by programs. The value is set to 0x0000 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADINT	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	ADSTIE	ADSNIE	INTST	INTSN
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB600000C
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **INTSN** (bit 0)

This bit cannot be used in this LSI. Always write "0" when writing. When reading this bit, "0" is always read.

- **INTST** (bit 1)

This bit indicates whether A/D conversion has been completed.

INTST	Description
0	A/D conversion has been completed.
1	A/D conversion has not been completed.

It is necessary to clear ("0") this bit by a program. This bit can be cleared by writing "1."

- **ADSNIE** (bit 2)

This bit cannot be used in this LSI. Always write "0" when writing. When reading this bit, "0" is always read.

- **ADSTIE** (bit 3)

This bit enables or disables the generation of an A/D conversion end interrupt request.

ADSTIE	Description
0	Enables the A/D conversion end interrupt.
1	Disables the A/D conversion end interrupt.

17.2.4 A/D Result (0 to 3) Registers (ADR0 to ADR3)

The ADR0 to ADR3 registers store the results of A/D conversions. The ADR0 to ADR3 registers can be read from/written to by programs. The value is set to 0x0000 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADRn	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	DTn[9:0]									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(n = 0 to 3)
Address: 0xB6000014 to 0xB6000020
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.
If a write operation is performed to the A/D result register during A/D conversion, all the results of A/D conversions for the ADR0 to ADR3 registers cannot be guaranteed.

[Bit Description]

- **DTn** (bits 9 to 0)
  These bits store the result of A/D conversion.

17.3 Operation of the A/D Converter

As the operating mode of the A/D converter, only the Select mode in which a single selected channel is converted is supported. The Select mode performs A/D conversion of one channel selected from CH0 to CH3 only once.

Example of setting

Where the operating conditions are as follows:
- Operating clock: 1/8 CCLK
- Interrupt: None
- Selected channel: Channel 2

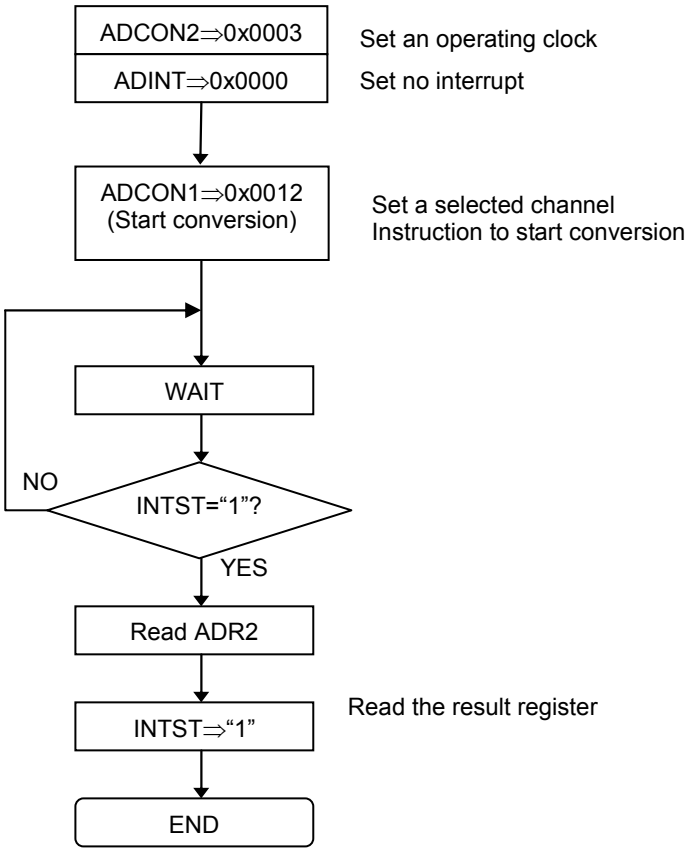


Figure 17-2 Example of Select Mode Setting

Example of a pin circuit

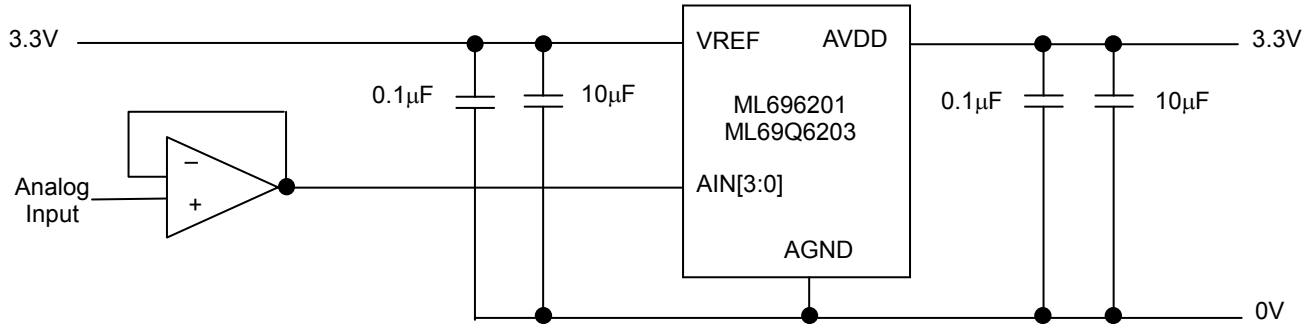


Figure 17-3 Example of A/D Converter Connection Circuit

Chapter 18

Synchronous SIO

Chapter 18 Synchronous SIO

18.1 Overview

The synchronous SIO is an 8-bit clock synchronous serial I/O port. This synchronous SIO has two channels, CH0 and CH1. For CH0 and CH1, a communication clock can be selected and used from a total of seven types of clock sources, including six types of clocks that divide a 15 MHz frequency and are independent of the bus clock, and one overflow clock from the timer. The maximum transfer rate is 15 Mbps.

Features:

- The maximum transfer rate is 15 Mbps.
- For CH0 and CH1, a clock master can be selected from seven types of clock sources.
  - The following six types of clocks are generated by dividing 15 MHz frequency by 1/1, 1/2, 1/4, 1/16, 1/32 and 1/128: 15 MHz, 7.5 MHz, 3.75 MHz, 937.5 kHz, 468.75 kHz and 117.18 kHz, respectively.
  - The seventh type of clock is the timer overflow clock (up to 1.875 MHz (tmout [2:1]) can also be selected. (CH0 uses the overflow clock of TIMER0 and CH1 the overflow clock of TIMER1.)
- Clock polarity can be selected.
- Either LSB first or MSB first can be selected.
- The Master or Slave mode can be selected for clock selection.
- Transmission/reception completion interrupt and empty interrupt can be generated.
- Enables an automatic data transfer in the Master mode.
- The transmit/receive buffer has a dual structure.

18.1.1 Configuration

Figure 18-1 shows the configuration of the synchronous serial interface (showing only for one channel).

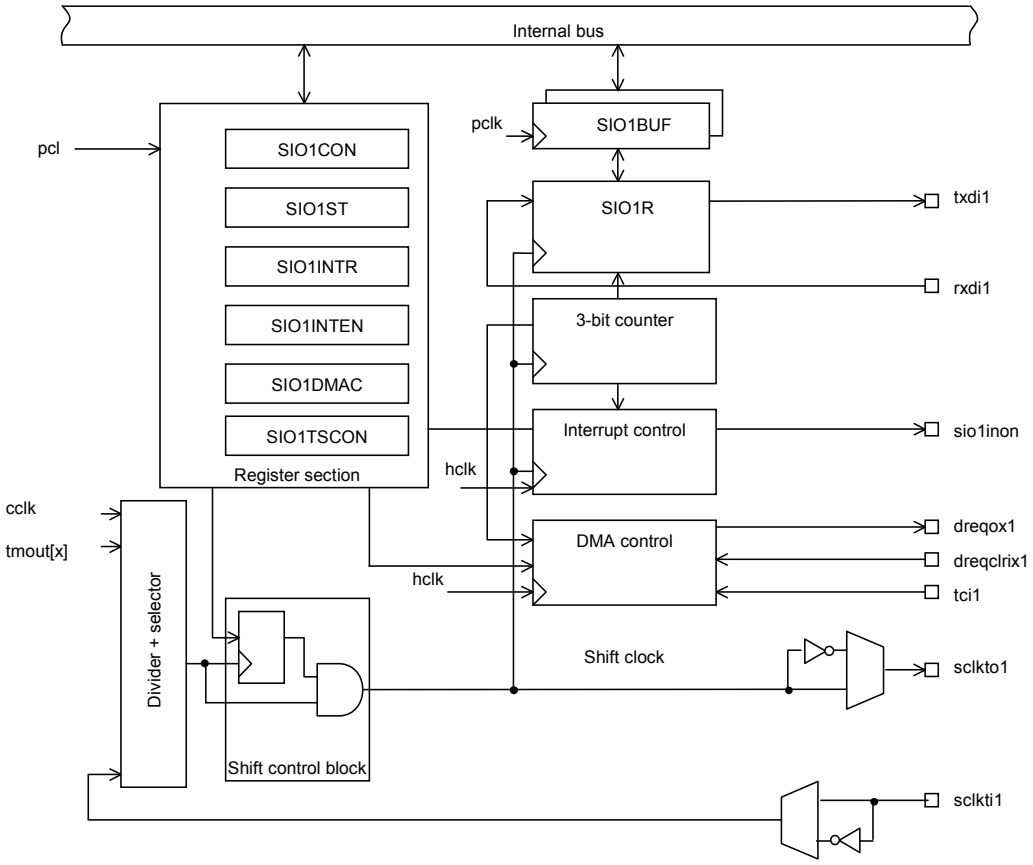


Figure 18-1 Block Configuration

18.1.2 Pin List

Pin name	I/O	Function
SSIOCK0	I/O	Input/output of SSIO communication clock 0. Assigned as a secondary function.
SSIORXD0	I	Input of SSIO receive data 0. Assigned as a secondary function.
SSIOTXD0	O	Output of SSIO transmit data 0. Assigned as a secondary function.
SSIOCK1	I/O	Input/output of SSIO communication clock 1. Assigned as a secondary function.
SSIORXD1	I	Output of SSIO receive data 1. Assigned as a secondary function.
SSIOTXD1	O	Output of SSIO transmit data 1. Assigned as a secondary function.

18.1.3 Control Register List

Address [H]	Register name	Symbol	R/W	Size	Initial value [H]
0xB7B1_0000	SSIO transmit/receive buffer register 0	SSIOBUF0	R/W	32	Undefined
0xB7B1_0004	SSIO transmit/receive status register 0	SSIOSTA0	R/W	32	0x0000_0000
0xB7B1_0008	SSIO transmit/receive control register 0	SSIOCON0	R/W	32	0x0000_0000
0xB7B1_000C	SSIO interrupt request register 0	SSIOINT0	R/W	32	0x0000_0002
0xB7B1_0010	SSIO interrupt enable register 0	SSIOINTEN0	R/W	32	0x0000_0000
0xB7B1_0014	SSIO automatic data transfer register 0	SSIODMAC0	R/W	32	0x0000_0000
0xB7B1_0018	SSIO test control register 0	SSIOTSCON0	R/W	32	0x0000_0000
……	<Reserved>	—	—	—	—
0xB7B1_0020	SSIO transmit/receive buffer register 1	SSIOBUF1	R/W	32	Undefined
0xB7B1_0024	SSIO transmit/receive status register 1	SSIOSTA1	R/W	32	0x0000_0000
0xB7B1_0028	SSIO transmit/receive control register 1	SSIOCON1	R/W	32	0x0000_0000
0xB7B1_002C	SSIO interrupt request register 1	SSIOINT1	R/W	32	0x0000_0002
0xB7B1_0030	SSIO interrupt enable register 1	SSIOINTEN1	R/W	32	0x0000_0000
0xB7B1_0034	SSIO automatic data transfer register 1	SSIODMAC1	R/W	32	0x0000_0000
0xB7B1_0038	SSIO test control register 1	SSIOTSCON1	R/W	32	0x0000_0000

18.2 Control Register Description

18.2.1 SSIO Transmit/Receive Buffer Registers (SSIOBUF1, SSIOBUF0)

The SSIOBUFx register holds transmit/receive data.
The SSIOBUFx register can be read from/written to by programs; it has a dual structure in which it operates differently for read and write operations.
The SSIOBUFx register functions as a receive buffer for a read operation, and as a transmit buffer for a write operation. The values at reset are undefined.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSIOBUFx	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	SIOBUF[7:0]							
	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—

Address: 0xB7B10000 (CH0), 0xB7B10020 (CH1)
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

18.2.2 SSIO Transmit/Receive Shift Registers (SSIOR1, SSIOR0)

The SSIORx register is an 8-bit shift register that performs a shift operation during transmission/reception. During transmission, the lower 8 bits of the data written into the SSIOBUFx register are transferred to the SSIORx register, starting a transmission operation. During reception, the data of 8 bits loaded into the SSIORx register is transferred to the lower 8 bits of the SSIOBUFx register, terminating a reception operation.
Note that since the SSIO transmit/receive shift registers cannot perform read and write operations by programs, they have not been mapped on addresses.

18.2.3 SSIO Transmit/Receive Status Registers (SSIOSTA1, SSIOSTA0)

The SSIOSTAx regiter is a 32-bit read/write register that indicates the operating status of the synchronous serial port (SSIO).
The value at the time of a reset is 0x0000_0000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSIOSTAx	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	SxSFCT2	SxSFCT1	SxSFCT0	—*	—*	—*	—*	BUSYx
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B10004 (CH0), 0xB7B10024 (CH1)
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **BUSYx** (bit 0)
  This bit indicates that data is being transmitted/received. Once the transmission/reception of data starts, this bit is automatically set to "1" and then reset to "0" when data transfer is complete. When this bit is set to "1," it indicates that data is being transferred. "1" cannot be written into this bit. Although "0" can be written into this bit, do not write "0" except when some problem has occurred. Also, when writing "0" to this bit, be sure to first execute initialization with the SxRST bit of the SSIO transmit/receive control register, then write "0".

BUSYx	Description
0	READY
1	BUSY

- **SxSFCT0 to SxSFCT2** (bits 5 to 7)
  These bits are used to read the count value of the 3-bit shift counter during data transmission/reception. They are set to "000" when there is no data transmission/reception in progress. The value is incremented by one each time transmit/receive data is shifted by one bit, and returns to "000" when data transmission/reception is complete. These bits are read only, and no writing can be performed to these bits.

SxSFCT2	SxSFCT1	SxSFCT0	Description
0	0	0	When data transmission has been completed, or data has not been transmitted.
0	0	1	1-bit transmission complete
0	1	0	2-bit transmission complete
0	1	1	3-bit transmission complete
1	0	0	4-bit transmission complete
1	0	1	5-bit transmission complete
1	1	0	6-bit transmission complete
1	1	1	7-bit transmission complete

18.2.4 SSIO Transmit/Receive Control Registers (SSIOCON1, SSIOCON0)

The SSIOCONx register is a 32-bit read/write register that controls transmission/reception operations. Write into this register only after a transmission/reception operation is complete. If a change is made in the SSIOCONx register before a transmission/reception operation is complete, the current and subsequent transmission/reception operations cannot be performed normally.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSIOCONx	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	SxRST	SCKxINV	SLMSBx	SFTSLVx	—*	SxSFTCK2	SxSFTCK1	SxSFTCK0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B10008 (CH0), 0xB7B10028 (CH1)
Access: R/W
Access size: 32 bits

[Note]
When changing the content of the SSIOCONx register, be sure to do it only after transmission/reception is complete. If the content is changed before transmission/reception is complete, operation cannot be guaranteed.
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- SxSFTCK0 to SxSFTCK2 (bits 0 to 2)
  These bits specify the synchronous clock when in the Master mode. These bits are invalid when in the Slave mode.

SxSFTCK2	SxSFTCK1	SxSFTCK0	Description
0	0	0	Selects 15 MHz as a synchronous clock
0	0	1	Selects 7.5 MHz as a synchronous clock
0	1	0	Selects 3.75 MHz as a synchronous clock
0	1	1	Selects 937.5 kHz as a synchronous clock
1	0	0	Selects 468.75 kHz as a synchronous clock
1	0	1	Selects 117.18 kHz as a synchronous clock
1	1	0	Setting prohibited
1	1	1	Selects TMOUTx (up to 1.875 MHz) as a synchronous clock

- **SFTSLVx** (bit 4)
  This bit specifies the Master or Slave mode. A synchronous clock is output from SSIOCK0 and SSIOCK1 in the Master mode, and an external clock is input to SSIOCK0 and SSIOCK1 in the Slave mode.
  An automatic data transfer can be used only in the Master mode.

SFTSLVx	Description
0	Master mode
1	Slave mode

- **SLMSBx** (bit 5)
  This bit specifies LSB first or MSB first for transmit/receive data transfer.

SLMSBx	Description
0	LSB first
1	MSB first

- **SCKxINV** (bit 6)
  This bit specifies whether to keep the polarity of the shift clock normal or whether to reverse it.

SCLxINV	Description
0	Clock polarity: Normal (when there is no transmission/reception in progress, the clock stops at "H" level.)
1	Clock polarity: Reverse (when there is no transmission/reception in progress, the clock stops at "L" level.)

- **SxRST** (bit 7)
  This bit can be used to reset and initialize a data transfer if data transfer stops for some reason. Do not manipulate this bit during data transfer. If data transfer stops, the data being transferred cannot be guaranteed.

SxRST	Description
0	Serial reset: OFF
1	Serial reset: ON

18.2.5 SSIO Interrupt Request Registers (SSIOINTR1, SSIOINTR0)

The SSIOINTx register is a 32-bit read/write register that indicates the operating status of the channel of the clock synchronous serial port. The transmit buffer empty interrupt generation bit is read only.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSIOINTRx	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	BUFEMPx	TRCENDx
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Address: 0xB7B1000C (CH0), 0xB7B1002C (CH1)
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **TRCENDx** (bit 0)
  This bit is set to "1" when a transmission/reception completion interrupt occurs. Clear this bit to "0" by writing "1" after data transfer is complete. This bit cannot be cleared by writing "0." This bit is at a "0" state when in the automatic data transfer mode.

TRCENDx	Description
0	A transmission/reception completion interrupt has not been generated.
1	A transmission/reception completion interrupt has been generated.

- **BUFEMPx** (bit 1)
  This bit is set to "1" when the transmit buffer becomes empty. When data is written into the transmit buffer, this bit is cleared to "0." This register is read only, and cannot be cleared to "0" via software control.

BUFENDx	Description
0	A transmit buffer empty interrupt has not been generated.
1	A transmit buffer empty interrupt has been generated.

18.2.6 SSIO Interrupt Enable Registers (SSIOINTEN1, SSIOINTEN0)

The SSIOINTENx register is a 32-bit read/write register that control transmission/reception operations. Interrupt signals connected to the interrupt controller of μPLAT from the SSIO can be masked for each interrupt cause.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSIOINTENx	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—	—	—*	—*	—*	—*	BUFEMPIEx	TRCENDIEx
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B10010 (CH0), 0xB7B10030 (CH1)
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.
If an interrupt is generated during an automatic data transfer, it leads to a malfunction. Therefore, set this register to "00h" when the automatic data transfer is employed.

[Bit Description]

- **TRCENDIEx** (bit 0)
  This bit enables or masks a transmission/reception completion interrupt. If an interrupt is generated during an automatic data transfer, it may cause a wrong operation. Therefore, set this bit to "0" when the automatic data transfer is employed.

TRCENDIEx	Description
0	Masks an interrupt request to the interrupt controller for a transmission/reception completion interrupt.
1	Enables an interrupt request to the interrupt controller for a transmission/reception completion interrupt.

- **BUFEMPIEx** (bit 1)

This bit enables or masks a transmit buffer empty interrupt. If an interrupt is generated during an automatic data transfer, it may cause a wrong operation. Therefore, set this bit to "0" when the automatic data transfer is used.

BUFENDIEx	Description
0	Masks an interrupt request to the interrupt controller for a transmission buffer empty interrupt.
1	Enables an interrupt request to the interrupt controller for a transmission buffer empty interrupt.

18.2.7 SSIO Automatic Data Transfer Registers (SSIODMAC1, SSIODMAC0)

The SSIODMACx register is a 32-bit read/write register that allows the use of the automatic data transfer control circuit during transmission/reception. If "0" is written into the SxDMAMOD bit, operation cannot be guaranteed. Therefore, do not write "0." An automatic data transfer is only available when in the Master mode.

If transfer direction is switched from transmission to reception or vice versa during data transfer, operation cannot be guaranteed. Be sure to switch them after the completion of the data transfer.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSIODMACx	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	TRCxSW	—*	—*	—*	SxDMA MOD
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B10014 (CH0), 0xB7B10034 (CH1)
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **SxDMAMOD** (bit 0)
  This bit enables or disables an automatic data transfer. Writing "0" into this bit is prohibited. When an automatic data transfer is complete, this bit is automatically reset to "0."

SxDMAMOD	Description
0	Disables the automatic data transfer mode.
1	Enables the automatic data transfer mode.

- **TRCxSW** (bit 4)
  This bit selects a transmission or reception operation during an automatic data transfer.

TRCxSW	Description
0	Transmission operation
1	Reception operation

18.2.8 SSIO Test Control Registers (SSIOTSCON1, SSIOTSCON0)

The SSIOTSCONx register is a 32-bit read/write register that facilitates internal tests. To change the content of the SSIOTSCON register, do it only after a transmission/reception operation is complete. If a change is made in the SSIOTSCON register before a transmission/reception operation is complete, the current and subsequent transmission/reception operations cannot be performed normally.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSIOTSCONx	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At rest	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	LBTSTx	—*	—*	—*	—*	—*	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B10018 (CH0), 0xB7B10038 (CH1)
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **LBTSTx** (bit 7)

This bit selects the normal mode or test mode. Setting this bit to "1" selects the test mode. When this bit is set to "1," a transmit signal is output from the transmit data output pin, which is then input as a receive signal.

LBTSTx	Description
0	Normal mode
1	Test mode (loopback mode)

18.3 Operational Description

18.3.1 Transmission Operation

Writing transmit data into the transmit buffer register (SSIOBUFx) triggers the start of a transmission operation, setting the BUSY flag to "1." After writing, the transmit data is set up in the transmit/receive shift register (SSIORx). At the same time, a transmit buffer empty interrupt is generated, and the next transmit data can then be written.

Subsequently, the transmit data is output from the SSIOTXDx pin in LSB first or MSB first in sync with a rising edge of the shift clock when the clock polarity* is normal mode, or in sync with a falling edge of the shift clock when the clock polarity is reverse mode. (*: "Polarity" mentioned in this manual means "a positive or negative going (edge).")

Thereafter, transmit data is output in sync with the shift clock according to the specification of the SSIO transmit/receive control register (SSIOCONx), and then the transmission of one frame is terminated. At this point, if the next transmit data has not been written into the transmit buffer register, the BUSY flag is cleared to "0," and a transmission/reception completion interrupt is generated at the same time, completing the transmission operation.

The transmit buffer empty interrupt flag is automatically cleared when data has been written into the SSIOBUFx register.

If data has been written into the SSIOBUFx during a transmission operation, the next data is automatically set up in the SSIORx upon completion of the current transmission operation, and data transfer is executed in succession.

If data has been written into the SSIOBUFx more than once during the transmission operation of one frame, it is updated with the latest data written.

If the next data has been written into the transmit buffer when finishing the transmission of data for one frame, no transmission/reception completion interrupt signal will be generated.

Clear the transmission/reception completion flag by software.

When in the Slave mode, write the next data into the SSIOBUFx after completing the current transmission operation, and then transfer the data. If data has been written into the SSIOBUFx during transfer by a buffer empty interrupt, that data will not be transferred.

A reception operation is performed concurrently with a transmission operation. Therefore, data is input from the SSIORXDx pin in sync with the clock pin, and a value is set in the SSIO transmit/receive buffer register. If this data is not required, ignore it.

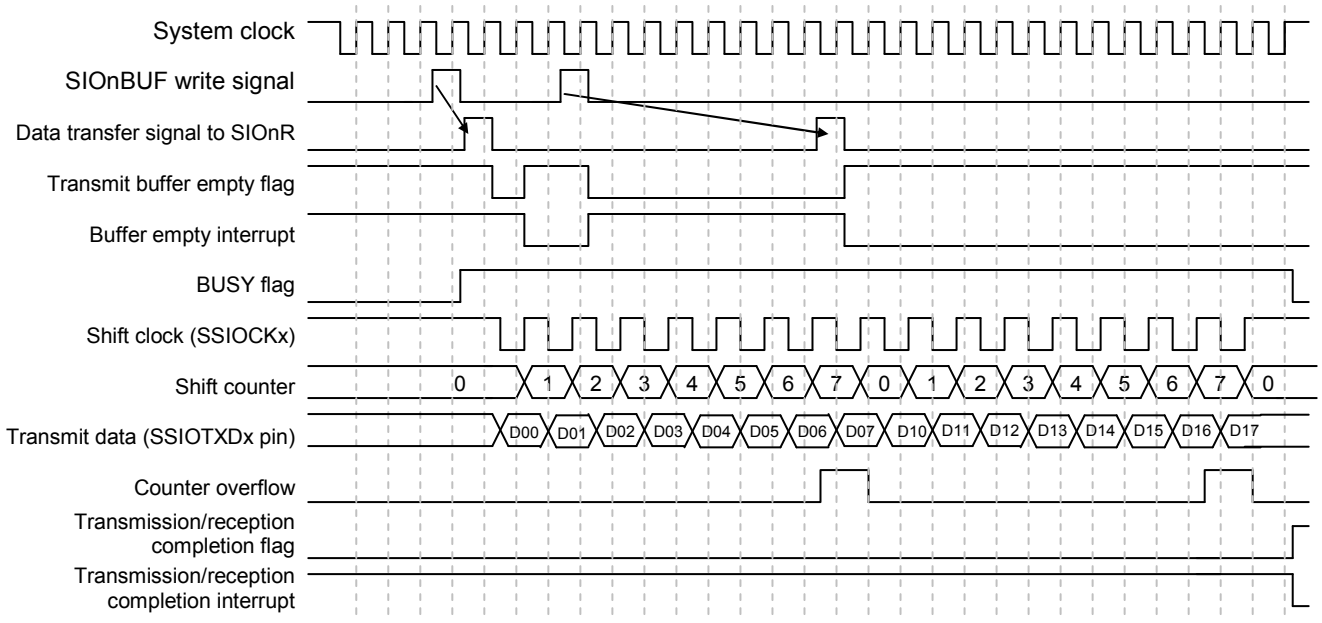


Figure 18-2 Transmission Timing Diagram (Master Mode, Clock Polarity = Normal, LSB First)

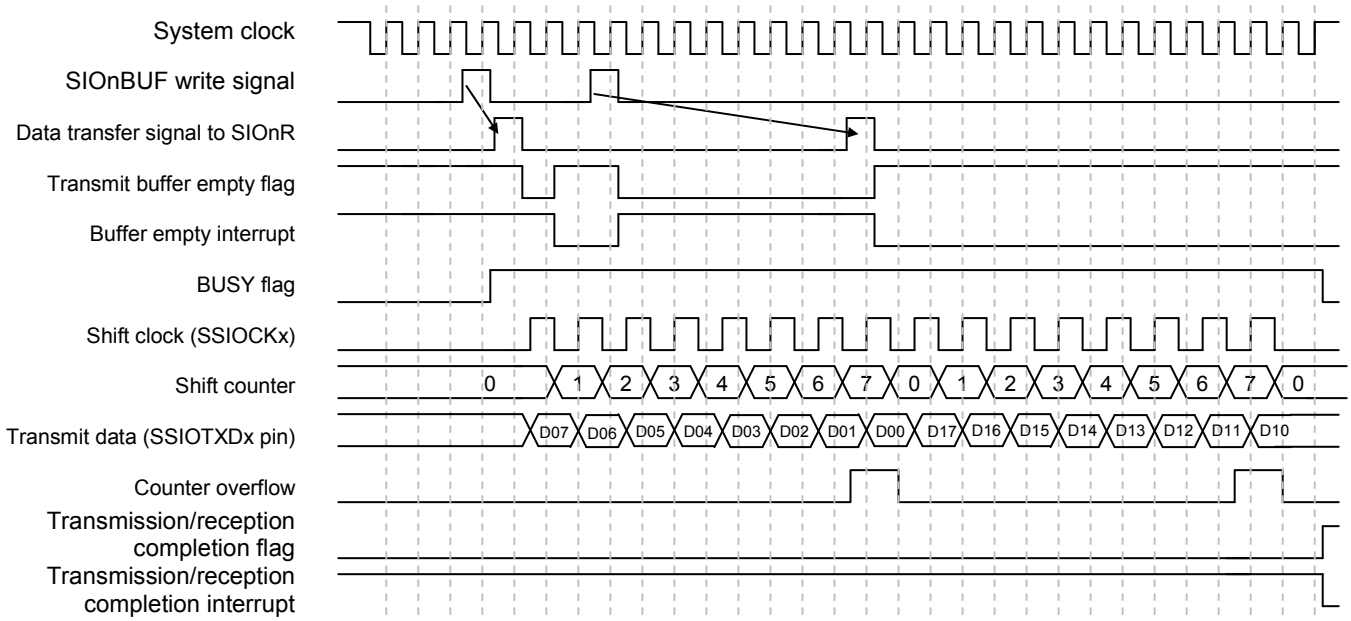


Figure 18-3 Transmission Timing Diagram (Master Mode, Clock Polarity = Reverse, MSB First)

18.3.2 Reception Operation

Writing data into the transmit buffer register (SSIOBUFx) triggers the start of a reception operation, setting the BUSY flag to "1."
This LSI loads the data that is input from the SSIORXDx pin at a rising edge of the shift clock, and shifts the data in the transmit/receive shift register (SSIORx). Thereafter, data reception is continued in accordance with the specification of the SSIO transmit/receive control register (SSIOCONx). After the reception of one frame is complete, the content of the SSIORx is transferred to the SSIOBUFx, the BUSY flag is cleared to "0," generating a transmission/reception completion interrupt.
Once the reception operation is completed, data can be read from the transmit/receive buffer register (SSIOBUFx).
Write dummy data into the transmit buffer register when starting a reception operation. Note that the dummy data is output from the SSIOTXDx pin of the transmit pins upon a reception operation.
Clear the transmission/reception completion flag by software.
When in the Slave mode, perform a reception operation by writing the next dummy data into the SSIOBUFx upon completion of a reception operation. Even if dummy data is written into the SSIOBUFx during transfer by a buffer empty interrupt, the data of the next frame will not be received.

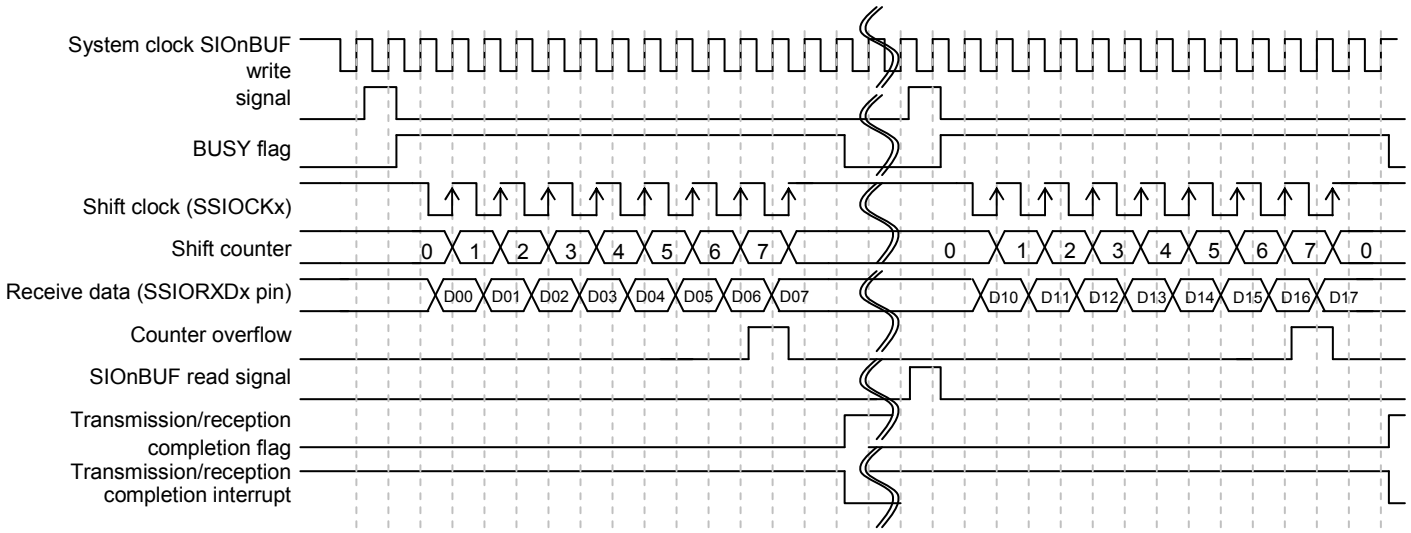


Figure 18-4 Reception Timing Diagram (Master Mode, Clock Polarity = Normal, LSB First)

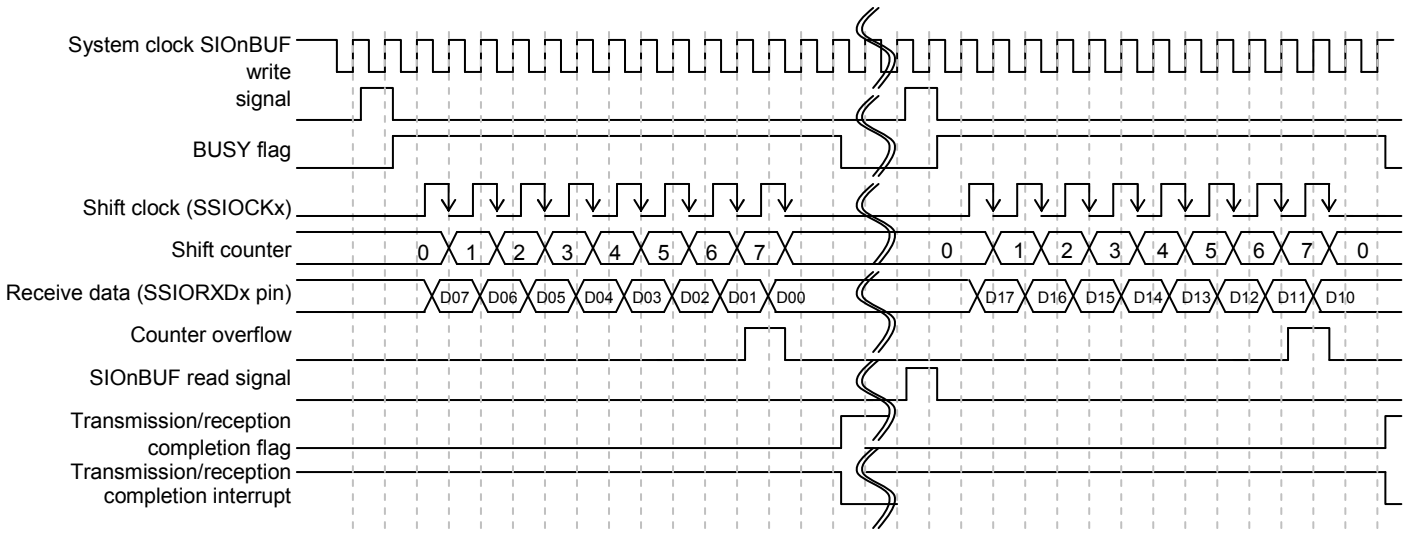


Figure 18-5 Reception Timing Diagram (Master Mode, Clock Polarity = Reverse, MSB First)

18.3.3 Automatic Data Transfer Function

This SIO has an automatic data transfer function. for the automatic data transfer, refer to Chapter 11. The automatic data transfer function can be used when only in the Master mode. The automatic data transfer function is supported only for CH0 and CH1 of synchronous SIO. Do not use CH2 in the automatic data mode. The external sagest signal on channels 2 and 3 of the automatic data transfer control section are connected to CH0 and CH1 of the synchronous SIO.
The following describes how to transmit/receive data via an automatic data transfer.

18.3.3.1 Transmission Operation (An Example of Transferring CH0 of Synchronous SIO on Channel 2 of Automatic Data Transfer)

① Make necessary settings for the automatic data transfer control section.
② After making the setting for the SSIO transmit/receive control register (SSIOCON0) (fixed to the Master mode), specify the DMA transfer mode by setting bit 4 of the SSIO automatic data transfer register (SIODMAC0) to "0" (transmission operation) and bit 0 to "1."
③ The external request signal to the automatic data transfer control section is asserted, and a data transfer request is made to the automatic data transfer control section.
When the automatic data transfer control section accepts the transfer request, an external request clear signal is output ("1") from the automatic data transfer control section, transmit data is transferred to the transmit buffer register (SSIOBUF0), and then data is written. At this time, the external request signal is deasserted.
④ Then, transmit data is loaded into the shift register (SSIOR0) and its shifting operation starts. At this time, the external request signal is set to "0."
⑤ Once the transmit data has been loaded into the shift register, the transmit buffer becomes empty, the external request signal is asserted again, and the transfer request for the next transmit data is made to the automatic data transfer control section.
⑥ If the next data has been written into the transmit buffer register via the automatic data transfer after finishing the transmission of one frame of data, the transmission operation is performed continuously. Also, if a transfer request has not been accepted by the automatic data transfer control section and the transmission data has not been transferred to the transmit buffer register, the shift clock stops until the data is transferred.
⑦ Steps ④ to ⑥ are repeated until the final transfer start signal is output from the automatic data transfer control section.
⑧ After the final transfer start signal has been output, the transmit buffer register is empty. After transmitting the last 8-bit data, the automatic data transfer mode is automatically cleared, terminating the automatic data transfer.

Note: Make the transmission/reception completion flag fixed to low.

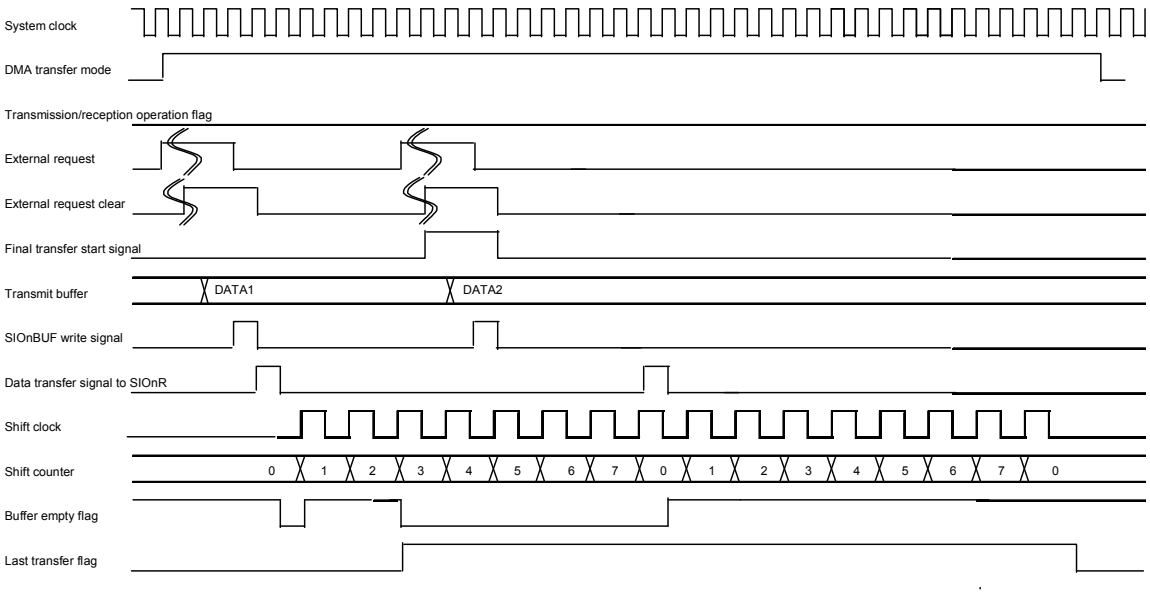


Figure 18-6 Automatic Data Transfer Timing Diagram (Transmission Operation; Transfer Count: 2)

18.3.3.2 Reception Operation (An Example of Transferring CH0 of Synchronous SIO on Channel 2 of Automatic Data Transfer)

① Make necessary settings for the automatic data transfer control section.
② After making the setting for the SSIO transmit/receive control register (SSIOCON0) (fixed to the Master mode), specify the automatic data transfer mode by setting bit 4 of the SSIO automatic data transfer register (SIODMAC0) to "1" (receive operation) and bit 0 to "1."
③ After starting the automatic data transfer mode, the shift clock operates to make the received data shifted in the transmit/receive shift register (SSIOR0).
④ After the receive data of one frame has been written into the receive buffer register, the external request signal is asserted, and a transfer request is made to the automatic data transfer control section. The shift clock stops until an automatic data transfer request is accepted.
⑤ Once the automatic data transfer control section accepts the transfer request, the external request clear signal is output ("1") from the automatic data transfer control section, and the receive data is read from the receive buffer register. The shift clock that that had been stopped resumes operation to start receiving data for the next frame. At this time, the external request signal is deasserted.
⑥ Steps ④ to ⑤ are repeated until the final transfer start signal is output from the automatic data transfer control section.
⑦ When the receive data is read from the receive buffer register after the final transfer start signal is output, the automatic data transfer mode is automatically cleared, terminating the automatic data transfer.

Note: Make the transmission/reception completion flag fixed to low.

[Precautions in Using the Automatic Data Transfer Function]

1. The burst mode and the cyclic mode of bus requesting method are available for DMA transfer. Since wrong operations can occur due to the timing of the transmit/receive buffer read/write operations in the burst mode, it is recommended to use only the cyclic mode. The operations cannot be guaranteed if the burst mode is used.
2. The auto request mode and the external request mode are available for automatic data transfer request. Since wrong operations can occur due to the timing of the transmit/receive buffer read/write operations in the auto request mode, use only the external request mode. The operations cannot be guaranteed if the auto request mode is used.

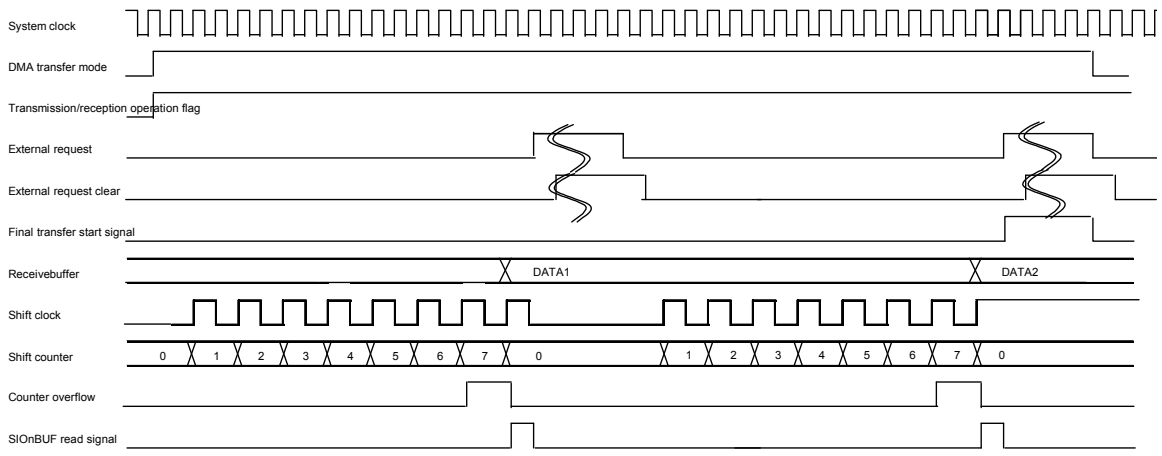


Figure 18-7 Automatic Data Transfer Timing Diagram (Reception Operation; Transfer Count: 2)

18.3.4 Master Mode/Slave Mode

There are two transmission/reception modes: Master mode and Slave mode.
Bit 4 (SFTSLVx) of the SSIO transmit/receive control register (SSIOCONx) is used to set the mode.

If the Master mode is set:
The clock selected by bits 0, 1 and 2 (SFTCLK0, SFTCLK1, SFTCLK2) of the SSIO transmit/receive control register (SSIOCONx) is used as the shift clock, and transmit/receive data is shifted and then output from the SSIOCKx pin to the outside as a synchronous clock.

If the Slave mode is set:
An external clock is input from the SSIOCKx pin, and transmit/receive data is shifted.

Notes:
1. Perform an automatic data transfer only in the Master mode.
2. When in the Slave mode, input an external clock after writing data into the transmit buffer.

18.3.5 Clock Polarity

A clock polarity can be selected for this SSIO.

The shift register SIOxR shifts out data at a rising edge of a shift clock.

The clock polarity is set by bit 6 (SCKxIXV) of the SSIO transmit/receive control register (SSIOCONx).

• When in the Master mode:
The clock (SSIOCKx) to be output to the outside is output with the selected polarity.
When SCKxIXV = "0" (normal), the internal shift clock is output as the master clock.
When not transmitting/receiving, the clock stops at "H" level.
When SCKxIXV = "1" (reverse), the clock generated by reversing the polarity of the internal shift clock is output.
When not transmitting/receiving, the clock stops at "L" level.

• When in the Slave mode:
The clock (SSIOCKx) to be input from the outside is set to the selected polarity and used as a shift clock.
Be sure to select the polarity for transmitting/receiving at a rising edge of the shift clock.
When SCKxIXV = "0" (normal), the slave clock is used as a shift clock as is.
When SCKxIXV = "1" (reverse), the clock generated by reversing the polarity of the slave clock is used as a shift clock.

18.3.6 Timer Overflow Baud Rate Formula

The following shows a formula for calculation when the overflow from timers 1 and 2 is used as a baud rate.

$$B = \frac{f}{\left(2^{16} - D\right)} \times \frac{1}{2}$$

B : Baud rate (bps)
f : Frequency (Hz)
D : Timer reloaded value (0 to 65535)

If the overflow clock of the TIMER is selected, the clock of TIMER0 can be used for CH0 and the clock of TIMER1 can be used for CH1.

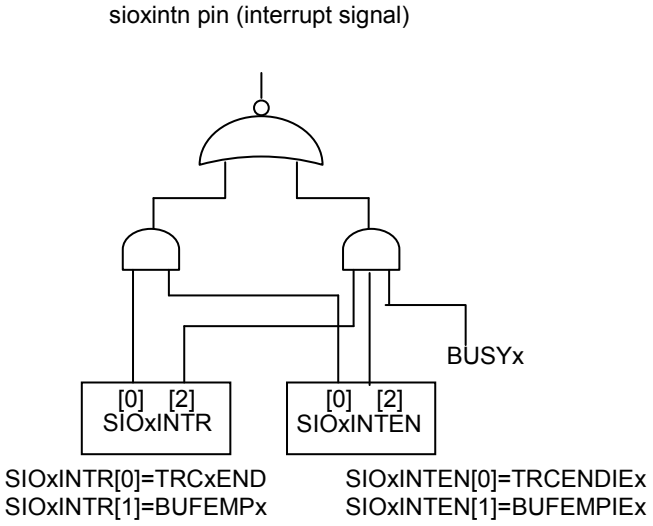
18.3.7 About Interrupts

A single interrupt signal is provided, and it is shared by a transmit buffer empty interrupt and a transmission/reception completion interrupt. These interrupts occur under the following conditions:

1. A transmit buffer empty interrupt has occurred (BUFEMPx="1"), a transmit buffer empty interrupt request is enable (BUFEMPIEx="1"), and during transmission/reception (BUSYx="1")
2. A transmission/reception completion interrupt has occurred (TRCxEND="1"), and a transmission/reception completion interrupt request is enabled (TRCENDIEx="1")

Canceling Interrupts by Software

Regarding item 1, mask a transmit buffer empty interrupt request and cancel the interrupt. If any other operation is performed, normal operation cannot be guaranteed.
Regarding item 2, an interrupt can be cancelled by clearing either TRCxEND or TRCENDIEx to "0."



Chapter 19

I2C

Chapter 19 I2C

19.1 Overview

The I2C module can be used as an extended external interface with EEPROM, PLL, LCD driver and others. The I2C module contains a 1-channel I2C bus interface that complies with Phillips I2C Bus standard. Since the I2C module only operates as a master device, it cannot transfer data as a slave device. Moreover, since the I2C module does not function as a multi-master bus (arbitration disabled, clock synchronization recognition disabled), no other master device can be connected to it.

Features:

- Communication mode: Master transmitter/master receiver
- Communication speed: 100 kbps (Standard mode)/400 kbps (Fast mode)
- Addressing format: 7-bit/10-bit
- Data buffer: 1 byte (1 stage)
- Data transfer sequence: MSB first

19.1.1 Configuration

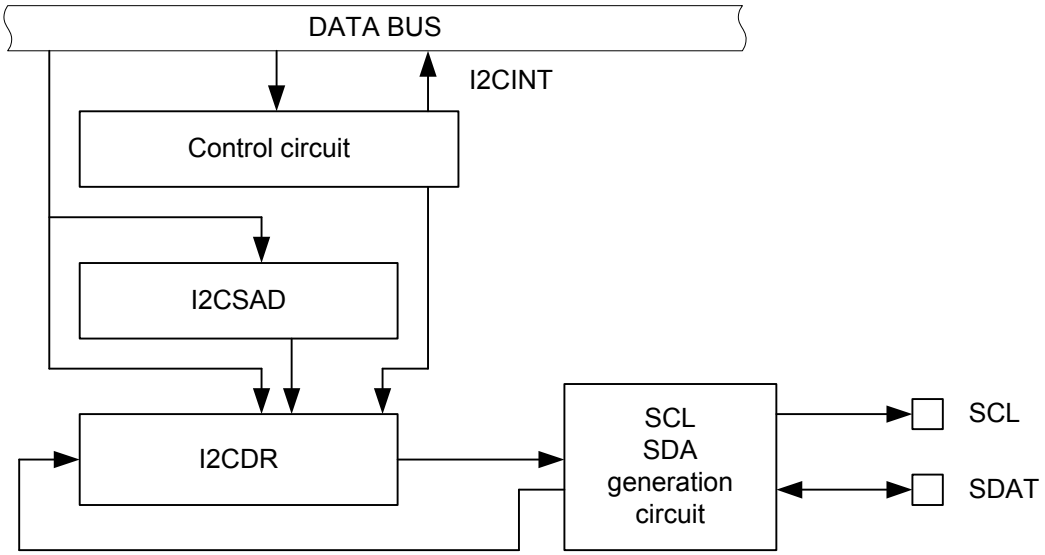


Figure 19-1 Block Configuration

19.1.2 Pin List

Pin Name	I/O	Function
SCL	O	Clock output pin for I2C data transfer
SDAT	I/O	Data receive/transmit I/O pin for I2C data transfer

19.1.3 Control Register List

Address	Register name	Symbol	R/W	Initial value
0xB7B0_0000	I2C bus control register	I2CCON	R/W	0x0000_0000
0xB7B0_0004	I2C bus slave address mode setting register	I2CSAD	R/W	0x0000_0000
0xB7B0_0008	I2C bus transfer speed setting register	I2CCLR	R/W	0x0000_0000
0xB7B0_000C	I2C bus status register	I2CSR	R	0x0000_0000
0xB7B0_0010	I2C bus interrupt request register	I2CIR	R/W	0x0000_0000
0xB7B0_0014	I2C bus interrupt mask register	I2CIMR	R/W	0x0000_0001
0xB7B0_0018	I2C bus transmit/receive data setting register	I2CDR	R/W	0x0000_0000

19.2 Control Register Description

19.2.1 I2C Bus Control Register (I2CCON)

The I2CCON register is an I2C control register that controls the transmission and reception of the I2C bus. Each valid bit of the I2CCON register is set to "0" at the time of a system reset

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2CCON	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	START	RESTR	STCM	I2COC	I2CEN
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B00000
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **I2CEN** (Bit 0)
  The I2CEN bit sets a restart request. When changing the transfer direction or slave address in the middle of transmitting/receiving successive data, the RESTART condition can be transmitted and the transmission/reception of data can be changed by setting this I2CEN bit to "1". In this case, it is also necessary to set the I2COC bit to "0" in order to perform the transmission/reception of one byte, and to set the I2COC bit to "1" in order to continuously perform the transmission/reception of data. Note that the I2C bus slave address mode setting register (I2CSAD) must be set prior to setting the I2CEN bit.

I2CEN	Description
0	Stops the RESTART condition.
1	Starts the RESTART condition.

- **I2COC** (Bit 1)
  This bit indicates the presence or absence of transmitting the RESTART condition. Set this bit to "1" to continuously occupy the I2C bus.
  When the I2COC bit is "1", the following operation is performed:
  In the case of data transmission out of the I2CDR register, an interrupt is generated after receiving an acknowledgment, and then the I2C module enters a state of waiting for the setting of the control register without executing the STOP condition.
  In the case of reception of data transmitted by the slave (not including the output of a slave address), an interrupt is generated after transmitting an acknowledgment, and then the I2C module enters a state of waiting for the setting of the control register without transmitting the STOP condition. When in this waiting state, the SCL pin holds its "0" state. It is possible to carry out continuous data transmission/reception by writing "1" again in the I2COC bit and in the STCM bit.
  Further, by setting the I2COC bit to "0" and the STCM bit to "1" when an interrupt is generated after continuous data transmission/reception, data is transmitted/received in the following manner:
  - In the case of transmission, data is transmitted, and the STOP condition is transmitted automatically after receiving an acknowledgment to terminate the communication.
  - In the case of reception, data is received, and the STOP condition is transmitted automatically after transmitting a non-acknowledgment to terminate the communication.
  An interrupt is generated at the end of communication. Because the I2COC bit is not automatically reset, it is necessary to clear it by software.

I2COC	Description
0	Releases the I2C bus after 1-byte communication.
1	Occupies the I2C bus after communication.

- **STCM** (Bit 2)
  The STCM bit starts/stops communication. By setting the STCM bit to "1", the transmission/reception of 1-byte data (not including the output of a slave address) is performed. The transmission/reception of 1-byte data includes the transmission/reception of data as well as its acknowledgment. Upon completion of communication, an interrupt is generated, and the STCM bit is automatically reset. Also, in cases where the I2COC bit has been set to "1", it is possible to transmit the STOP condition and terminate the communication by setting the I2COC bit to "0" and the STCM bit to "1" after data transmission/reception. Also in this case, an interrupt is generated upon completion of communication.

STCM	Description
0	Stops communication
1	Starts communication

- **RESTR** (Bit 3)
  The RESTR bit sets restarting. This bit indicates the presence or absence of the RESTART condition for the next data transfer after the current data transfer. Setting this bit to "0" indicates no restarting, and setting this bit to "1" indicates restarting.

RESTR	Description
0	No restarting
1	Restarting

- **START** (Bit 4)
  The START bit sets the start byte. This bit indicates the presence or absence of a start byte in the I2C control circuit. Furthermore, for the purpose of controlling the start byte, set the start byte pattern ("0000_0001b") defined in the I2C specification in the I2C bus slave address mode setting register (I2CSAD) in advance.

START	Description
0	There is no start byte.
1	There is a start byte.

19.2.2 I2C Bus Slave Address Mode Setting Register (I2CSAD)

The I2CSAD register is an I2C control register that sets the slave address of the transmission/reception counterpart device as well as transmission and reception modes. Each valid bit of the I2CSAD register is set to "0" at the time of a system reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2CSAD	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	SAD6	SAD5	SAD4	SAD3	SAD2	SAD1	SAD0	I2CRW
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B00004
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
Be sure to set the I2CSAD register prior to setting the I2CCON register.
The address of the second byte following a 10-bit address or a general call address is transmitted, with the value written in the I2CDR register taken as that address.

[Description of Bits]

- **I2CRW** (Bit 0)
  The I2CRW bit sets the write/read method to/from the communication counterpart.

I2CRW	Description
0	Data transmission mode
1	Data reception mode

- **SAD0 to SAD7** (Bit 1 to 7)
  The SAD0 to SAD7 bits set the slave address of the communication counterpart side.

19.2.3 I2C Bus Transfer Speed Setting Register (I2CCLR)

The I2CCLR register is an I2C control register that sets the communication speed (mode) for transmission/reception. Each valid bit of the ICCLR register is set to "0" at the time of a system reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2CCLR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	I2CMD
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B00008
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **I2CMD** (Bit 0)
  The I2CMD bit sets the transfer mode. Set the I2CCLR register prior to setting the I2CCON register.

I2CMD	Description
0	Selects Standard mode (100 kHz).
1	Selects Fast mode (400 kHz).

19.2.4 I2C Bus Status Register (I2CSR)

The I2CSR register is an I2C control register that controls the status of the I2C bus. Each valid bit of the I2CSR register is set to "0" at the time of a system reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2CSR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	I2C AAK	I2C DAK
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B0000C
Access: R
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read.

[Description of Bits]

- **I2CDAK** (Bit 0)
  The I2CDAK bit is set to "1" if an acknowledgment is not received in the transmit/receive operation of slave data. The I2CDAK bit continues to hold "1" until the I2C bus interrupt request register (I2CIR) is cleared.

I2CDAK	Description
0	Normal reception of an ACK
1	ACK reception error

- **I2CAAK** (Bit 1)
  The I2CAAK bit is set to "1" if an acknowledgment is not received in the transmit/receive operation of slave address. The I2CAAK bit continues to hold "1" until the I2C bus interrupt request register (I2CIR) is cleared.

I2CAAK	Description
0	Normal reception of an ACK
1	ACK reception error

19.2.5 I2C Bus Interrupt Request Register (I2CIR)

The I2CIR register is an I2C control register that indicates an I2C bus interrupt request.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2CIR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	I2CIR
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7B00010
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **I2CIR** (Bit 0)
  The I2CIR bit is set to "1" when a transmit/receive operation is completed or an ACK reception is abnormal, indicting that there is an interrupt request. An interrupt request is cleared by writing "1" to this I2CIR bit.

I2CIR	Description
0	There is no interrupt request.
1	There is an interrupt request.

19.2.6 I2C Bus Interrupt Mask Register (I2CIMR)

The I2CIMR register is an I2C control register that sets the mask for an interrupt source corresponding to each bit. Each valid bit of the I2CIMR register is set to "0" at the time of a system reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2CIMR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	I2CMF
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Address: 0xB7B00014
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
Set the I2CIMR register prior to setting the I2CCON register.

[Description of Bits]

- **I2CMF** (Bit 0)
  Setting "1" in the I2CMF bit masks an interrupt after the completion of communication. Mask the back stage of the I2CIR register.

I2CMF	Description
0	Generates an interrupt at the end of communication.
1	Masks an interrupt at the end of communication.

19.2.7 I2C Bus Transmit/Receive Data Setting Register (I2CDR)

The I2CDR register is an I2C control register that sets or stores transmit/receive data. Each valid bit of the I2CDR register is set to "0" at the time of a system reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2CDR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	I2CD7	I2CD6	I2CD5	I2CD4	I2CD3	I2CD2	I2CD1	I2CD0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

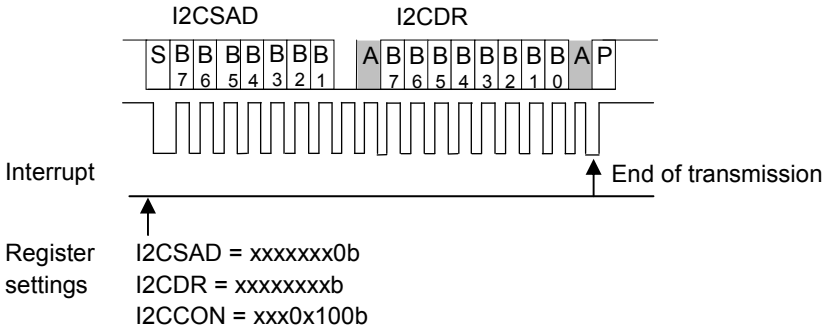
Address: 0xB7B00018
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
Be sure to set the I2CDR register prior to setting the I2CCON register.
Be sure to read the above I2CDR register upon completion of byte-data transfer after setting the I2CCON register. Transmit data cannot be read immediately after it is set.

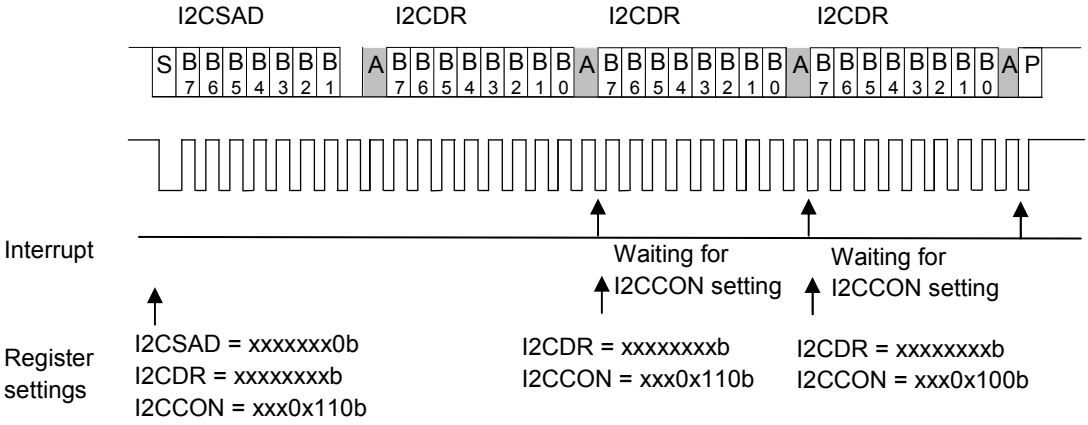
[Description of Bits]
- **I2CD0 to I2CD7** (Bit 0 to 7)
  During transmission, the value of the I2CDR register is output from the MSB side, and during reception, receive data shifts from the LSB side to the MSB side and is then stored. To read received data, be sure to read the I2CDR register before transmitting/receiving the next data.

19.3 Operational Description

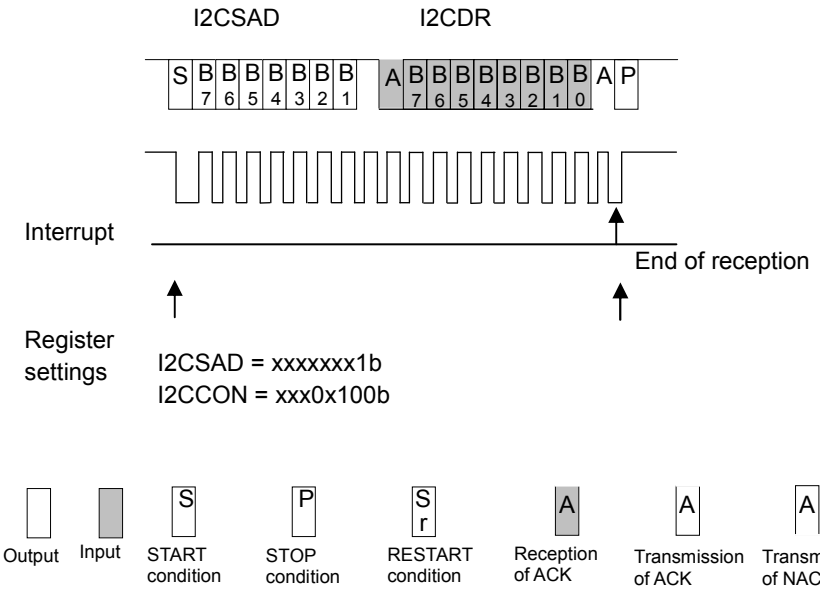
19.3.1 When Transmitting 1-Byte Data



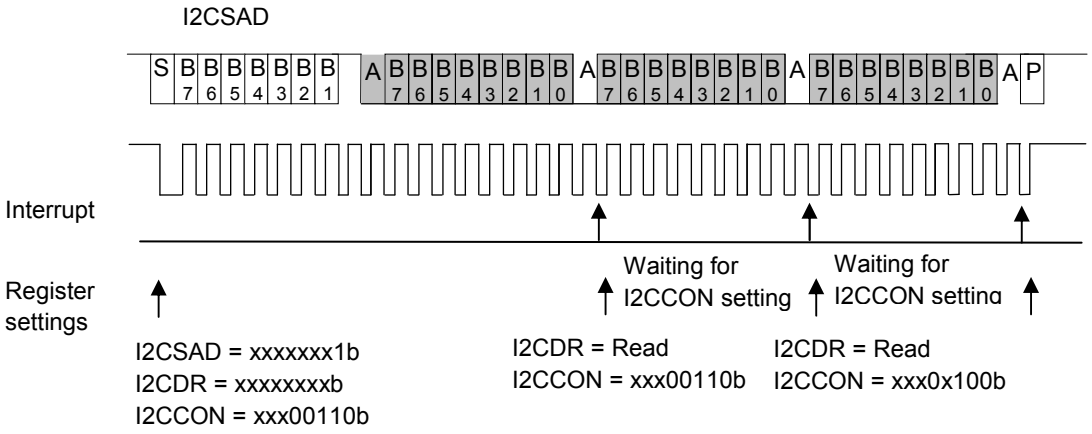
19.3.2 When Transmitting Data of Two or More Bytes



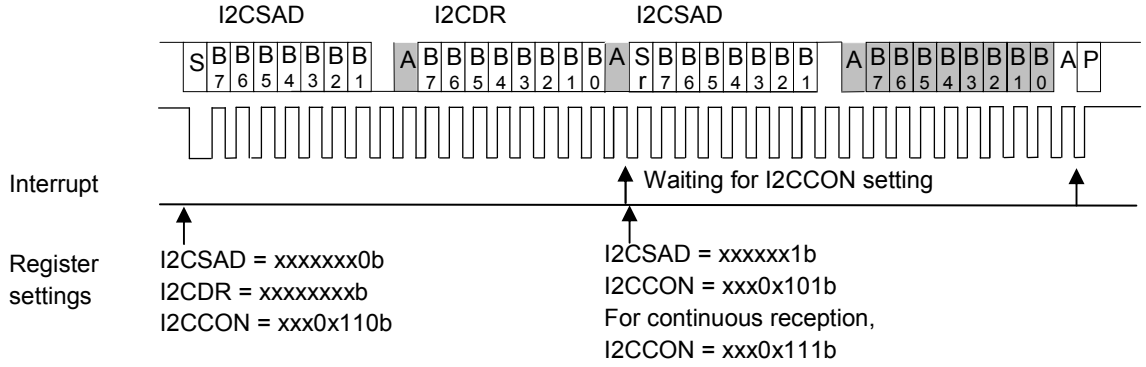
19.3.3 When Receiving 1-Byte Data



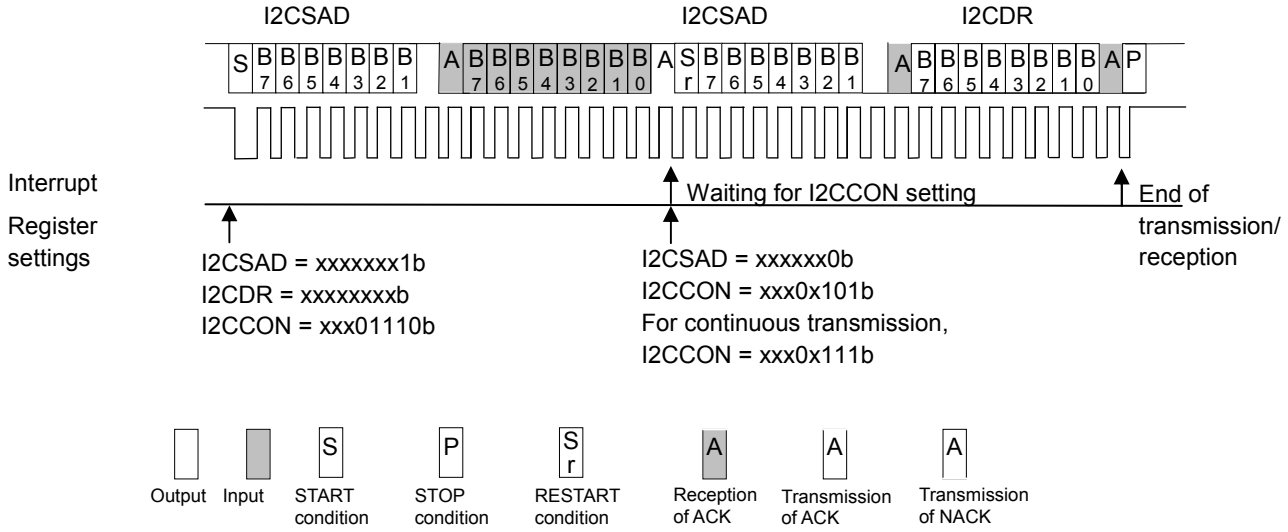
19.3.4 When Receiving Data of Two or More Bytes



19.3.5 When Continuously Receiving 1-Byte Data After Transmitting 1-Byte Data



19.3.6 When Continuously Transmitting 1-Byte Data After Receiving 1-Byte Data



Chapter 20

RTC

Chapter 20 RTC

20.1 Overview

The RTC module counts a 32.768 kHz clock, and compares a 15-bit counter that generates one second with a 32-bit counter that counts using a 32.768 kHz clock as a source. If they match, it generates an interrupt. Since an interrupt is output by a 32 kHz clock, it is generated even when the PCLK is being stopped.

Features:

- 32-bit counter of 1-second clock
- 32-bit comparison interrupt function

20.1.1 Configuration

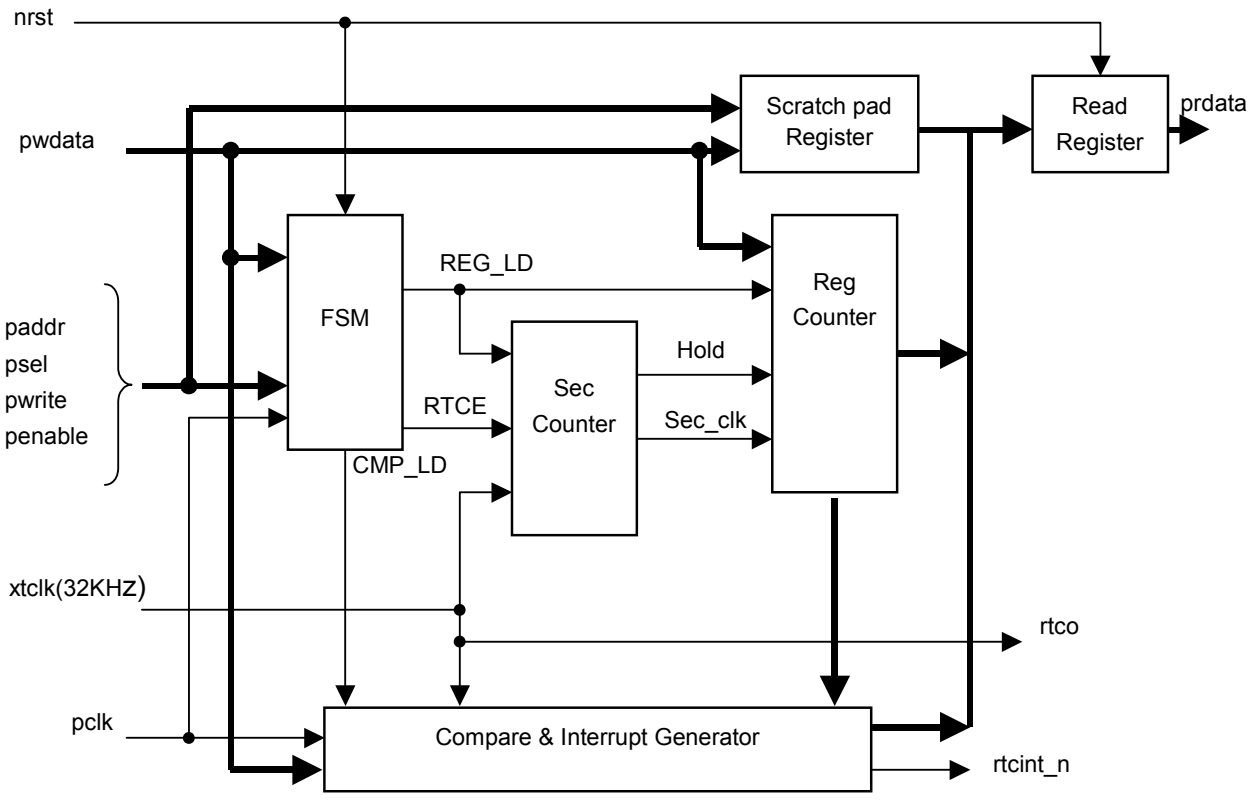


Figure 20-1 Block Configuration

20.1.2 Pin List

Pin name	I/O	Function
OSC32K0	I	32.768 kHz oscillation input pin
OSC32K1B	O	32.768 kHz oscillation input pin
RTC_TESTMODE	I	SCAN mode pin in the RTC module. Since this pin has been pulled down, keep it open. This is used as an interface in the VDDRTC pin (1.5V).
32K_TESTMODE	I	Test mode pin of the 32 kHz oscillation circuit. Because this pin has been pulled down, keep it open. This is used as an interface for the VDDRTC pin (1.5V).
VDDRTC	Power supply	Backup power supply pin of the RTC module
GNDRTC	Power supply	Backup ground pin of the RTC module

20.1.3 Control Register List

Address	Register name	Symbol	R/W	Initial value
0xB7C0_0000	RTC register	RTCREG	R/W	Undefined
0xB7C0_0004	RTC control register	RTCCON	R/W	Undefined
0xB7C0_0008	RTC compare register	RTCCMP	R/W	Undefined
0xB7C0_000C	RTC scratch pad register	RTCSCRP	R/W	Undefined
0xB7C0_0010	RTC status register	RTCST	R/W	Undefined

20.2 Control Register Description

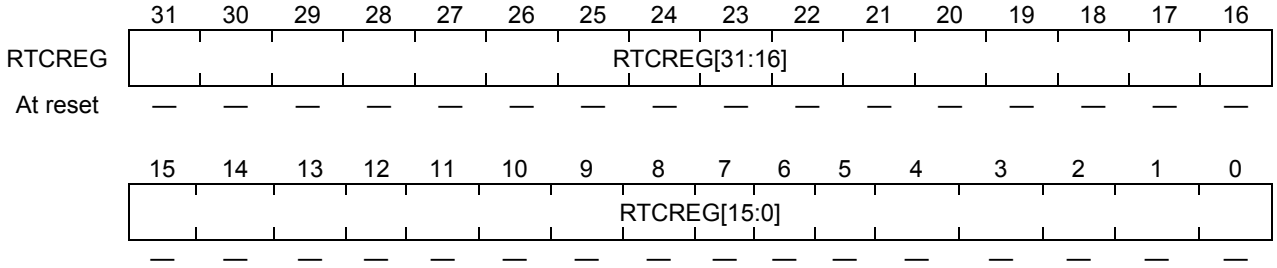
When writing to all the registers except the RTC scratch pad register, it is necessary to write required data after writing two data items in succession—0x0000_003C and then 0x0000_005A—to the same address. By writing these two data items of 0x0000_003C and 0x0000_005A, the write enable state is set, and then the next write data is written. The write enable state is not set if any other data is written. Also, the write enable state may be cancelled if a different address is accessed. Therefore, be sure to complete write operations in succession.

The RTC scratch pad register does not have a write protect function, and therefore regular read and write operations can be performed. When reading, data can be read via regular access.

In this LSI, access is masked by the power down mask register of the configuration module when the power is turned on. After setting the RTCEN bit of the power down mask register to "1", use the scratch pad register to determine whether or not access can be made. Then, perform processing for each register of the RTC.

20.2.1 RTC Register (RTCREG)

The RTCREG register is a read/write enable 32-bit second counter.



Address: 0xB7C00000
Access: R/W
Access size: 32 bits

[Note]
When 128 kHz or less is selected as the APB BUS clock, undefined values may be read while the register is incremented. Two reads should be done when 32 kHz is selected as the CPUCLK, and the matched read value should be used. The read interval should be within 50 ms.

[Description of Bits]

- **RTCREG[31:0]** (Bit 0 to 31)
  The RTCREG[31:0] bit field makes up a 32-bit second counter, which is a read/write enable register. If the RTCE bit of the RTCCON register is "1", this counter is incremented at every second. If a value is set in this register, the 15-bit counter that counts using a 32.768 kHz clock as a source is also cleared at the same time. The value of the 32-bit second counter will not be read while a count operation is in progress even if the 32-bit second counter is being incremented.

20.2.2 RTC Control Register (RTCCON)

The RTCCON register controls the operation of the RTC counter.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTCCON	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	RTCE
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-

Address: 0xB7C00004
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

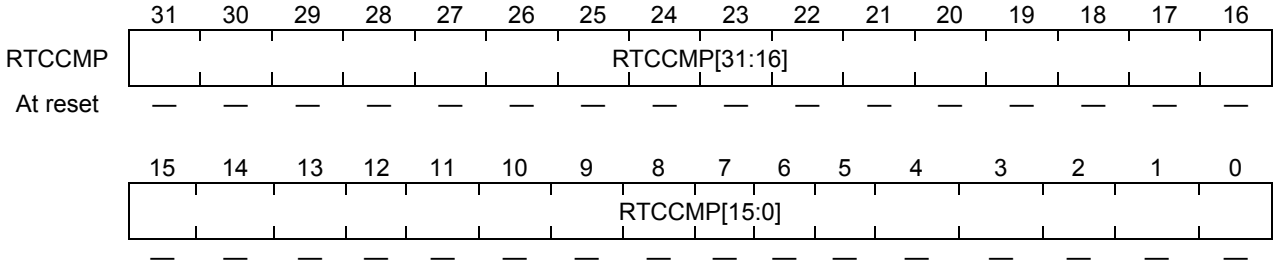
[Description of Bits]

- **RTCE** (Bit 0)
  The RTCE bit is the RTC count enable bit (RTCE). When this bit is set to "0", the count operation of the RTCREG stops; when this bit is set to "1", the count operation of the RTCREG starts. The initial value is undefined.

RTCE	Description
0	Stops a counter operation as the RTC.
1	Starts a counter operation as the RTC.

20.2.3 RTC Compare Register (RTCCMP)

The RTCCMP register generates an interrupt if the value of this register and the value of the RTC counter match. If an interrupt request is enabled in the RTC interrupt enable register, an interrupt request to the interrupt controller is generated.



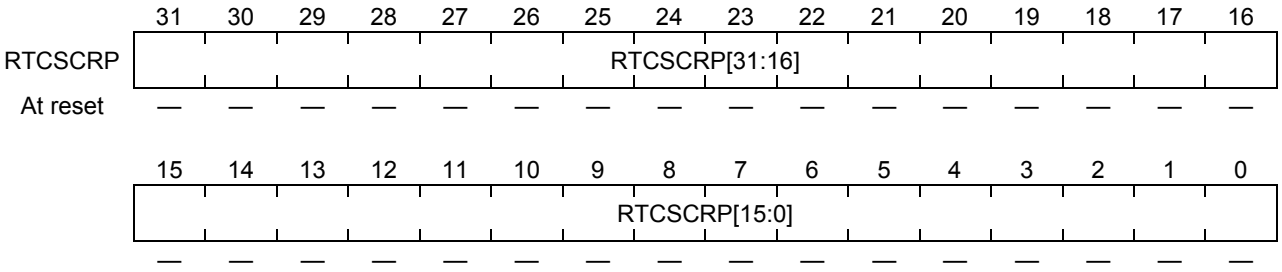
Address: 0xB7C00008
Access: R/W
Access size: 32bits

[Description of Bits]

- **RTCCMP[31:0]** (Bit 0 to 31)
  The RTCCMP[31:0] bit field makes up a read/write enable 32-bit register. It generates an interrupt if the value set in this register and the value of the RTCREG register match.

20.2.4 RTC Scratch Pad Register (RTCSCRP)

The RTCSCRP register is a read/write enable 32-bit register. This register is used to check whether or not access to the RTC can be made.



Address: 0xB7C0000C
Access: R/W
Access size: 32 bits

[Description of Bits]

- **RTCSCRP[31:0]** (Bit 0 to 31)
  The RTCSCRP[31:0] bit field makes up a read/write enable 32-bit register.
  Access to the RTC is not allowed when the power is turned on or when recovering from backup mode. Access is enabled by setting the power down mask register of the configuration register module. Because it takes time to enable access to the RTC after setting the power down mask register of the configuration register module, this register can be used to determine whether or not access to the RTC has become enabled by reading/writing an arbitrary value from/to this register.

20.2.5 RTC Status Register (RTCST)

The RTCST register indicates the interrupt request status.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTCST	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	INT
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-

Address: 0xB7C00010
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **INT** (Bit 0)
  The INT bit is a matching interrupt request bit. If the RTCREG and RTCCMP register values match, the INTI bit is set to "1" and, at the same time, the RTCST register makes an interrupt signal active. This bit is cleared by writing "1". This bit does not change when "0" is written. Since an interrupt is output by a 32 kHz clock, an interrupt is generated even when the PCLK is being stopped.
  However, no interrupt is generated while the RTCEN bit of the PDMASK register of the configuration register module is in the "1" state. Since the status of this bit remains "0" even if the RTCREG and RTCCMP register values match in that period, no interrupt is generated even by setting the RTCEN bit of the PDMASK register to "0" after the match.

INT	Description
0	There is no interrupt request.
1	There is an interrupt request.

20.3 Operational Description

20.3.1 RTC Function

This RTC module generates one second from 32.768 kHz by a 15-bit counter. It operates as a 32-bit counter using one second as a clock.

In addition, the RTC module has a 32-bit compare register. It generates an interrupt if the value of the 32-bit compare register match the value of the 32-bit counter register. An interrupt can be cleared by controlling the RTC status register. Since the RTC module cannot set an interrupt mask, it is always set in the interrupt enable state, and the RTC module notifies the interrupt status to the interrupt controller. If interrupt processing is not required, disable interrupts via the interrupt controller.

Moreover, because the values of all the registers of the RTC modules are undefined when the power is turned on, there is a possibility that an interrupt might have been generated. Therefore, be sure to set each register of the RTC module after the power is turned on. Here, "turning on the power" does not include the case of recovering from backup mode.

20.3.2 Write Protect

When writing to all the registers except the RTC scratch pad register, it is necessary to write required data after writing two data items in succession—0x0000_003C and then 0x0000_005A—to the same address. By writing these two data items of 0x0000_003C and 0x0000_005A, the write enable state is set, and then the next write data is written. The write enable state is not set if any other data is written. Also, the write enable state may be cancelled if a different address is accessed. Therefore, be sure to complete write operations in succession.

The RTC scratch pad register does not have a write protect function, and therefore regular read and write operations can be performed. When reading, data can be read via regular access.

In this LSI, access is masked by the power down mask register of the configuration module when the power is turned on. After setting the RTCEN bit of the power down mask register to "1", use the scratch pad register to determine whether or not access can be made. Then, perform processing for each register of the RTC.

20.3.3 Access Restrictions on the RTC

Access to the RTC is not allowed when the power is turned on or when recovering from backup mode. Access is enabled after setting the RTCEN bit in the PDMASK register of the configuration register module to "1" and then waiting for the designated time. The RTC scratch pad register can be used to check whether or not access to the RTC is currently enabled. Read/write an arbitrary value from/to the RTC scratch pad register. Conduct a test to read/write the inverted value of that arbitrary value several times. If the values obtained match the expected value, it can be determined that access to the RTC is enabled.

20.3.4 Note on Successive Accesses to Registers

When the CPU writes to or reads from registers in the RTC module or registers in the other modules connected to the APB buses successively after writing to registers (other than RTCSCRP) in the RTC module, RTCREG register values may be changed to unintended values.
Insert a dummy cycle by the NOP instruction immediately after writing to registers (other than RTCSCRP) in the RTC module when requiring successive accesses as described above. The required dummy cycle period is one clock cycle of PCLK.

Chapter 21

Configuration Registers

Chapter 21 Configuration Registers

21.1 Overview

The configuration registers are used to set power down and clock stop, and during Test mode. It is necessary to write 0x0000003C to the six configuration registers described in this chapter in order to remove write protect, prior to writing to them. When reading, they can be read as they are. Although unused bits exist as registers, they have no function.

Features:

- Configurations such as power down and clock stop can be set.
- The primary and secondary functions of the GPIO can be switched.
- Each test mode can be controlled.

21.1.1 Configuration

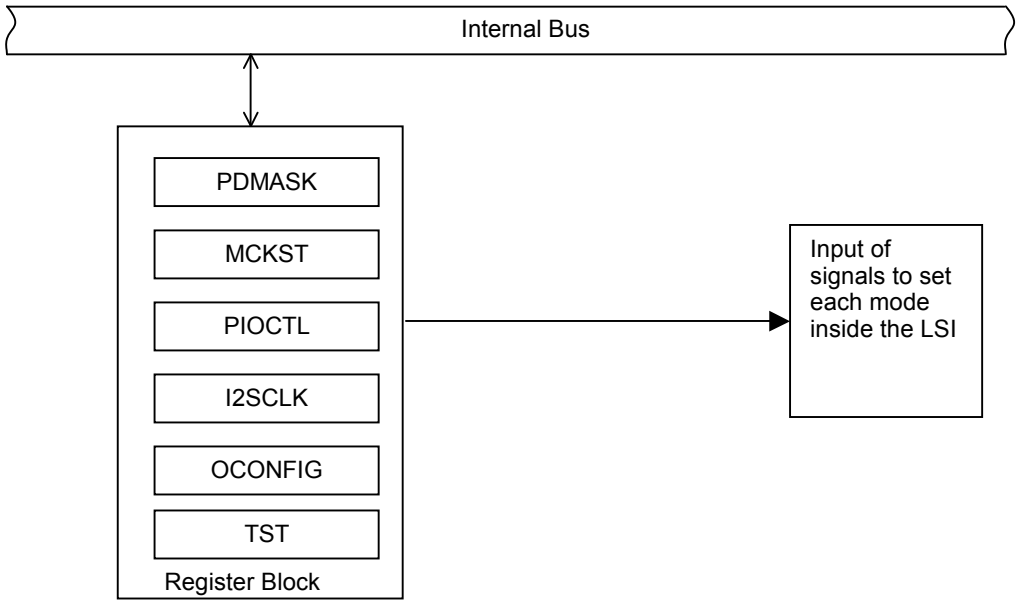


Figure 2-1 Block Configuration

21.1.2 Pin List

Signals are not output to the outside of the LSI.

21.1.3 Configuration Register List

Address	Register name	Symbol	Size	R/W	Initial value
0x8000_0000	Power down mask register	PDMASK	32	R/W	0x0000_0000
0x8000_0004	Module clock stop register	MCKST	32	R/W	0x0000_0000
0x8000_0008	PIO pin switching register	PIOCTL	32	R/W	0x0000_0000
0x8000_000C	I2S control register	I2SCNTL	32	R/W	0x0000_0000
0x8000_0010	OTHER configuration register	OCONFIG	32	R/W	0x0000_0000
0x8000_0014	Test register	TST	32	R/W	0x0000_0000

21.2 Register Description

When writing to all the configuration registers, it is necessary to write 0x0000_003C first, then write required data. By writing 0x0000_003C, each configuration register is set in the write enable state, and then the next write data is written. Writing any other data will not place the configuration registers in the write enable state. When reading, data can be read by regular access.

21.2.1 Power Down Mask Register (PDMASK)

Before only the RTC is moved to backup mode, this register controls to disable access to RTC. Without performing this control, turning off the power supplies other than the power supply to RTC may damage the contents of the registers inside the RTC.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDMASK	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	RTC EN
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x80000000
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
When writing into this register, write "0x3C" first, and immediately after that, write an applicable value to prevent erroneous writing.

[Description of Bits]

- **RTCEN** (Bit 0)
  The RTCEN bit is used as a control register to protect the values of the registers in RTC before moving to backup mode that shuts down the power supplies other than the power supply to RTC. The initial value is 0, disabling access to RTC.
  Set this bit to "1" to generate the 32 kHz clock.

RTCEN	Description
0	Power supplies other than to RTC can be turned off (access to RTC is disabled).
1	Power supplies other than RTC cannot be turned off (access to RTC is enabled).

21.2.2 Module Clock Stop Register (MCKST)

The MCKST register controls clock supply to the direct memory access controller (DMAC), DRAM controller (DRAMC), I2S transmitter module, I2S receiver module, NAND Flash controller and NAND Flash buffer memory. This register also controls clock output to the XSYSCLK pin.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCKST	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCKST	—*	—*	—*	—*	—*	—*	—*	—*	—*	XSYS CLKEN	CKST NFB	CKST NFC	CKST I2SR	CKST I2ST	CKST DRAM	CKST DMAC
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x80000004
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
When writing into this register, write "0x3C" first, and immediately after that, write an applicable value to prevent erroneous writing.

[Description of Bits]

- **CKSTDMAC** (Bit 0)
  The CKSTDMAC bit controls clock supply to the DMAC.

CKSTDMAC	Description
0	Starts clock supply to the DMAC
1	Stops clock supply to the DMAC.

- **CKSTDRAM** (Bit 1)
  The CKSTDRAM bit controls clock supply to the DRAM controller (DRAMC).

CKSTDRAM	Description
0	Starts clock supply to the DRAMC.
1	Stops clock supply to the DRAMC.

- **CKSTI2ST** (Bit 3)
  The CKSTI2ST bit controls clock supply to the I2S transmitter module.

CKSTI2ST	Description
0	Starts clock supply to the I2S transmitter module.
1	Stops clock supply to the I2S transmitter module.

- **CKSTI2SR** (Bit 3)
  The CKSTI2SR bit controls clock supply to the I2S receiver module.

CKSTI2SR	Description
0	Starts clock supply to the I2S receiver module.
1	Stops clock supply to the I2S receiver module.

- **CKSTNFC** (Bit 4)
  The CKSTNFC bit controls clock supply to the NAND Flash controller.

CKSTNFC	Description
0	Starts clock supply to the NAND Flash controller.
1	Stops clock supply to the NAND Flash controller.

- **CKSTNFB** (Bit 5)
  The CKSTNFB bit controls clock supply to NAND Flash buffer memory.

CKSTNFB	Description
0	Starts clock supply to NAND Flash buffer memory.
1	Stops clock supply to NAND Flash buffer memory.

- **XSYSCLKE**N
  The XSYSCLKEN bit controls AHB clock output to the XSYSCLK pin. This bit is set to "0" at the time of a reset, and thus the output disable state is set. By writing "1" to this bit, AHB clock output to the XSYSCLK pin is enabled. The XSYSCLK pin is fixed to "H" while AHB clock output is disabled.

XSYSCLKEN	Description
0	Stops clock output to the XSYSCLK pin.
1	Starts clock output to the XSYSCLK pin.

21.2.3 PIO Pin Switching Register (PIOCTL)

The PIO pin switching register (PIOCTL) of the configuration register module selects the primary or secondary function assigned to PIOCTL [15:0]. The PIOCTL register can be read from or written to by program. The value is set to 0x0000 at the time of a reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIOCTL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIOCTL [15:0]																
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x80000008
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
When writing into this register, write "0x3C" first, and immediately after that, write an applicable value to prevent erroneous writing.

[Description of Bits]

- **PIOCTL[0]** (Bit 0)

PIOCTL[0] = "0" (primary function)		PIOCTL[0] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD00	In/Out	SCLD	In/Out

- **PIOCTL[1]** (Bit 1)

PIOCTL[1] = "0" (primary function)		PIOCTL[1] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD01	In/Out	WSD	In/Out

- **PIOCTL[2]** (Bit 2)

PIOCTL[2] = "0" (primary function)		PIOCTL[2] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD02	In/Out	SDD	Out

- **PIOCTL[3]** (Bit 3)

PIOCTL[3] = "0" (primary function)		PIOCTL[3] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD03	In/Out	CKOUTD	Out

- **PIOCTL[4]** (Bit 4)

PIOCTL[4] = "0" (primary function)		PIOCTL[4] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD04	In/Out	SCLA/SCL	In/Out

- **PIOCTL[5]** (Bit 5)

PIOCTL[5] = "0" (primary function)		PIOCTL[5] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD05	In/Out	WSA	In/Out

- **PIOCTL[6]** (Bit 6)

PIOCTL[6] = "0" (primary function)		PIOCTL[6] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD06	In/Out	SDA	In

- **PIOCTL[7]** (Bit 7)

PIOCTL[7] = "0" (primary function)		PIOCTL[7] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD07	In/Out	CKOUTA/CKOUT	Out

- **PIOCTL[8]** (Bit 8)

PIOCTL[8] = "0" (primary function)		PIOCTL8] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD08	In/Out	SSIOTXD0	Out

- **PIOCTL[9]** (Bit 9)

PIOCTL[9] = "0" (primary function)		PIOCTL[9] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD09	In/Out	SSIORXD0	In

- **PIOCTL[10]** (Bit 10)

PIOCTL[10] = "0" (primary function)		PIOCTL[10] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD10	In/Out	SSIOCK0	In/Out

- **PIOCTL[11]** (Bit 11)

PIOCTL[11] = "0" (primary function)		PIOCTL[11] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD11	In/Out	SSIOTXD1	Out

- **PIOCTL[12]** (Bit 12)

PIOCTL[12] = "0" (primary function)		PIOCTL[12] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD12	In/Out	SSIORXD1	In

- **PIOCTL[13]** (Bit 13)

PIOCTL[13] = "0" (primary function)		PIOCTL[13] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD13	In/Out	SSIOCK1	In/Out

- **PIOCTL[14]** (Bit 14)

PIOCTL[14] = "0" (primary function)		PIOCTL[14] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD14	In/Out	UP_TXD	Out

- **PIOCTL[15]** (Bit 15)

PIOCTL[15] = "0" (primary function)		PIOCTL[15] = "1" (secondary function)	
Function	Input/Output	Function	Input/Output
PIOD15	In/Out	UP_RXD	In

21.2.4 I2S Control Register (I2SCNTL)

The I2SCNTL register sets the I2S setup mode.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SCLK	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	I2SCNTCLR	I2SOSCST	I2SPLLEN	I2SOUT	I2SCLKSEL[1:0]		I2SCKO[1:0]		I2SSCLC[1:0]	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8000000C
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read at Read processing and the bit is ignored at Write processing.
When writing into this register, write "0x3C" first, and immediately after that, write an applicable value to prevent erroneous writing.

[Description of Bits]

- **I2SSCLC[1:0]** (Bit 0,1)
  Out of the SCLA and SLCD pins of the I2S receiver/transmitter module, the I2SSCLC[1:0] bit field enables only the SCLA pin of the I2S receiver module, and outputs OR of the SCLA pin of the I2S receiver module and the SCLD pin of the I2S transmitter module. Fix the SCLD pin of the I2S transmitter module to "L."

I2SSCLC [1]	I2SSCLC [0]	Description
0	*	Outputs the receive clock from the SCLA pin, and the transmit clock from the SCLD pin.
1	0	Outputs the SCLA output of the I2S receiver module from the SCLA pin. The SCLD pin outputs a level fixed to "L".
1	1	Outputs the SCLD output of the I2S transmitter module from the SCLA pin. The SCLD pin outputs a level fixed to "L".

- **I2SCKO[1:0]** (Bit 2, 3)
  The I2SCKO[1:0] bit field sets whether to output the CLKOUT output of the I2S receiver/transmitter module only from the CLKOUTA pin and have the CLKOUTD pin output a level fixed to "L", or whether to output a CLKOUT signal from the CLKOUTA pin for reception and from the CLKOUTD pin for transmission.

I2SCKO[1]	I2SCKO[0]	Description
0	*	Outputs a CLKOUT signal from the CLKOUTA pin for reception, and from the CLKOUTD pin for transmission.
1	0	Outputs CLKOUT output of the receiver module from the CLKOUTA pin. The CLKOUTD pin outputs a level fixed to "L".
1	1	Outputs CLKOUT output of the transmitter module from the CLKOUTA pin. The CLKOUTD pin outputs a level fixed to "L".

- **I2SCLKSEL[1:0]** (Bit 4, 5)
  The I2SCLKSEL[1:0] bit field sets the system clock frequency of the I2S receiver/transmitter module.

I2SCLKSEL[1]	I2SCLKSEL [0]	Description
0	0	The system clock of I2S is 11.2896 MHz.
0	1	The system clock of I2S is 12.288 MHz.
1	0	The system clock of I2S is 11.2896 MHz.
1	1	The system clock of I2S is 8.192 MHz.

  Set the sampling frequency for I2S transmission according to the FSO1 and FSO0 bits of the I2S transmitter module. Set the sampling frequency for I2S reception according to the FSI1 and FSI0 bits of the I2S receiver module.

- **I2SOUT** (Bit 6)
  The I2SOUT bit is set when the input/output signal pin is used. Since the I2S input/output signal pin is shared by the GPIO pin, set the PIO pin switching register as well as this bit when using the I2S signal pin.

I2SOUT	Description
0	The I2S input/output signals are not input or output.
1	The I2S input/output signals are output to the external pins in order to control the CODEC module external to the LSI.

- **I2SPLLEN** (Bit 7)
  The I2SPLLEN bit controls to enable/disable the PLL for the I2S receiver/transmitter module. As the initial value, the PLL is disabled. If 12.288 MHz or 8.192 MHz is used as the system clock of the I2S receiver/transmitter module, it is necessary to enable the PLL using this bit. If 11.2896 MHz is used as the system clock of the I2S receiver/transmitter module, the PLL may remain disabled. Enable the PLL first, wait for the PLL stability wait time (1 ms: preliminary value), then set the system clock of the I2S receiver/transmitter module to 12.288 MHz or 8.192 MHz.

I2SPLLEN	Description
0	Disables the PLL.
1	Enables the PLL.

- **I2SOSCST** (Bit 8)
  The I2SOSCST bit controls the enable/disable state of 11.2896 MHz oscillation for the I2S receiver/transmitter module. As the initial value, the oscillation is enabled. If the I2S receiver/transmitter module is used, it is necessary to enable oscillation using this bit.
  Enable oscillation first, and then wait for the oscillation stability wait time (10 ms: tentative time). If the PLL is going to be used, enable the PLL. If the PLL is not used, set the system clock of the I2S receiver/transmitter module to 11.2896 MHz.

I2SOSCST	Description
0	Starts oscillation.
1	Stops oscillation.

- **I2SCNTCLR** (Bit 9)
  The I2SCNTCLR bit is used for the stop control of the PLL clock frequency dividing counter when the PLL in the clock generator for the I2S receiver/transmitter module is used. Since the initial value is "0," the PLL clock frequency dividing counter is running. If "1" is set, the counter is cleared. This bit is used to synchronize PLL output during testing, and usually set the value to "0."

I2SCNTCLR	Description
0	Operates the counter.
1	Clears the counter.

21.2.5 OTHER Configuration Register (OCONFIG)

The OCONFIG register sets other configurations such as the software reset of Flash ROM.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OCONFIG	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	FRRST
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x80000010
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
When writing into this register, write "0x3C" first, and immediately after that, write an applicable value to prevent erroneous writing.

[Description of Bits]

- **FRRST** (Bit 0)
  The FRRST bit is a software reset bit of Flash ROM. This bit outputs a reset level when "1" and a non-reset level when "0" to the FMRST_N pin (this bit and the FMRST_N output are reversed).

FRRST	Description
0	Releases a software reset to Flash ROM.
1	Performs a software reset to Flash ROM.

21.2.6 Test Register (TST)

The TST register is used for testing only. Do not access this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCTST	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	USB TST	WDT TST	—*	—*	—*	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x80000014
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
When writing into this register, write "0x3C" first, and immediately after that, write an applicable value to prevent erroneous writing.

[Description o Bits]

- **WDTTST** (Bit 5)
  The WDTTST bit is used to test the WDT. It is possible to bypass part of the WDT counter.
  Always set this bit to "0".

- **USBTST** (Bit 6)
  The USBTST bit is used to test the USB. It is possible to bypass part of the USB counter.
  Always set this bit to "0."

Chapter 22

I2S Transmission

Chapter 22 I2S Transmission

22.1 Overview

The I2S transmission module outputs audio or voice data. The data to be transmitted is written to FIFO via the APB bus. The written data in FIFO is transmitted according to the serial audio interface format. Since the FIFO memory is shared by transmit and receive operations, operation of the I2S receive module must be stopped at transmission.

Features:

- Sampling frequency: (8/11.025/12/16/22.05/24/32/44.1/48 kHz)
- System clock: Supports 256Fs
- Channel data length: 16 bits
- Operation mode: Master only
- Serial data format: with/without 1-bit delay specifiable, MSB first only, standard/inverted output of word select signal
- FIFO size: 9,216 bytes (shared with the I2S receive module)

22.1.1 Configuration

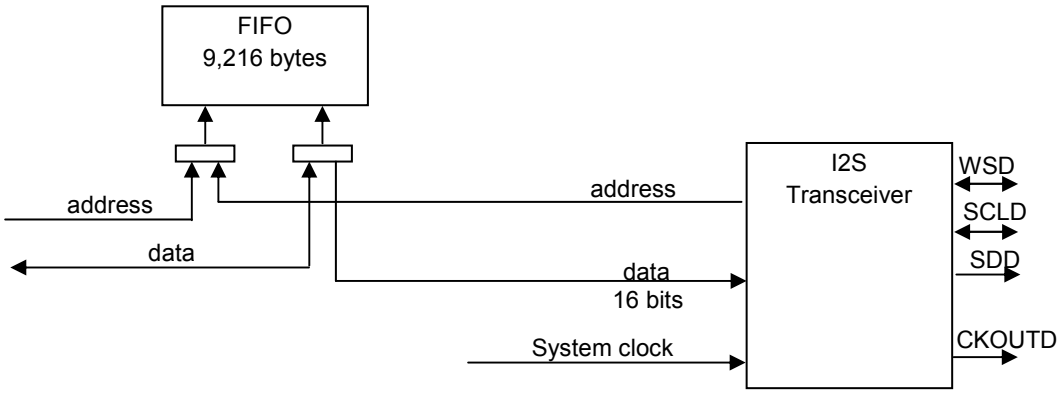


Figure 22-1 Block Configuration

22.1.2 List of Pins

Pin Name	I/O	Function
CKOUTD	O	I2S transmit system clock
SDD	O	I2S transmit data
WSD	O	I2S transmit channel selection
SCLD	O	I2S transmit transfer clock

22.1.3 List of Control Registers

Address	Register name	Symbol	R/W	Initial value [H]
0x8200_0000	I2S transceiver FIFO register	I2SFIFOO	R/W	Undefined
0x8200_0004	I2S transceiver control register 0	I2SCONO0	R/W	0x0000_0000
0x8200_0008	I2S transceiver control register 1	I2SCONO1	R/W	0x0000_0000
0x8200_000C	I2S transceiver Almost Full threshold value setting register	I2SAFRO	R/W	0x0000_0900
0x8200_0010	I2S transceiver Almost Empty threshold value setting register	I2SAERO	R/W	0x0000_0000
0x8200_0014	I2S transceiver interrupt mask register	I2SIMRO	R/W	0x0000_000F
0x8200_0018	I2S transceiver interrupt status register	I2SISTO	R/W	0x0000_0000
0x8200_001C	I2S transceiver FIFO write address register	I2SWADRO	R/W	0x0000_0000
0x8200_0020	I2S transceiver FIFO read address register	I2SRADRO	R/W	0x0000_0000
0x8200_0024	I2S transceiver FIFO occupied data size register	I2SDNOO	R	0x0000_0000

22.2 Control Registers

22.2.1 I2S Transceiver FIFO Register (I2SFIFOO)

This register is used for writing data to FIFO or reading data from FIFO.
When data is written to this register, the data is written to the address that is indicated by the FIFO write address register (I2SWADRO). In this case, the value in the FIFO write address register (I2SWADRO) is automatically incremented. Data of the LEFT channel and data of the RIGHT channel can be written simultaneously. The data that is written to I2SFIFOO LOW[15:0] is output first and the data that is written to I2SFIFOO HIGH[15:0] is output subsequently. When data is written to this register while FIFO is Full, the data becomes invalid and the value in the FIFO address register (I2SWADRO) is not incremented automatically.
For serial data output, an even number of data items combining the LEFT channel and the RIGHT channel are transmitted.
By reading data from this register while the I2S transmit module is inactive, the FIFO data that is indicated by the FIFO write address register (I2SWADRO) can be read. In this case, the value in the FIFO write address register (I2SWADRO) is also incremented. When data is read from this register while the I2S transmit module is active, an undefined value is read and the value in the FIFO write address register (SAIWADRO) is incremented.
When this register is reset, an undefined value is set.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SFIFOO	I2SFIFOO_HIGH[15:0]															
At reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2SFIFOO_LOW[15:0]															
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Address: 0x8200_0000
Access: R/W
Access size: 32 bits

[Note]
Only 32-bit accessing is enabled and half-word/byte accessing is prohibited.

[Description of Bits]

- **I2SFIFOO_LOW[15:0]** (Bit 0 to 15)
  FIFO register used for writing transmit data. Data can also be read from the register.
  Data written to I2SFIFOO_LOW[15:0] is output prior to I2SFIFOO_HIGH[15:0]. So, normally, write data on the LEFT channel into this register.

- **I2SFIFOO_HIGH[15:0]** (Bit 16 to 31)
  FIFO register used for writing transmit data. Data can also be read from the register.
  Data written to I2SFIFOO_ HIGH [15:0] is output after I2SFIFOO_LOW[15:0]. So, normally, write data on the RIGHT channel into this register.

22.2.2 I2S Transceiver Control Register 0 (I2SCONO0)

This register is used for setting I2S.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SCONO0	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	FSO1	FSO0	RUNO	DLYO	WSLO	MSBO	—*	SFSO
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8200_0004
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
If the contents of bits other than bit 5 (RUNO) are changed while I2S is operating, the operation is not guaranteed.

[Description of Bits]

- **SFSO** (Bit 0)
  The SFSO bit specifies the system clock frequency. Since this LSI supports 256Fs only, this bit must be fixed to "0".

SFSO	Description
0	256Fs
1	Setting not allowed

- **MSBO** (Bit 2)
  The MSBO bit specifies the format of serial output data. Since this LSI supports MSB first only, this bit must be fixed to "0".

SFSO	Description
0	MSB first
1	Setting not allowed

- **WSLO** (Bit 3)
  The WSLO bit specifies the format of the word select signal of the LEFT channel. For the format of the word select signal of the RIGHT channel, the inverted value of that format of the word select signal of the LEFT channel which is set in this bit is set.

WSLO	Description
0	LEFT channel word select signal = "L" level
1	LEFT channel word select signal = "H" level

- **DLYO** (Bit 4)
  The DLYO bit specifies one-clock delay/no-delay in serial output data.

DLYO	Description
0	No serial output data delay
1	Delay in serial output data

- **RUNO** (Bit 5)
  The RUNO bit specifies the start and stop of serial output operation.
  When this bit is cleared, output of the word select signal pin (WSD) is fixed to the state that was set when the transmit clock was stopped and output of the transmit clock pin (SCLD) and output of the transmit serial data pin (SDD) are both set to the "L" level.
  When this bit is set to "1", transmission of serial data starts. After this bit is set, initially the "L" level data of the LEFT and RIGHT channels is transmitted and then the data that has been stored in FIFO is transmitted. At reset, the "L" level data of the LEFT and RIGHT channels is transmitted once more.
  When this bit is cleared during data transmission, an even number of data items combining the LEFT channel and the RIGHT channel are transmitted before the operation stops. When this bit is set to "1" while FIFO is empty, the word select signal pin (WSD) and the transmit clock pin (SCLD) output data in the same way as for normal transmit operation, but the transmit serial data pin (SDD) always outputs "L" level data.

RUNO	Description
0	Stops serial output operation
1	Starts serial output operation

- **FSO1, FSO0** (Bit 6, 7)
  These bits specify the sampling frequency level.
  The I2S system clock is specifid with the I2SCLKSEL[1:0] bit of the configuration register 0.

FSO1	FSO0	Description
0	0	The sampling frequency is 1/256 of the system clock: 32/44.1/48 kHz
0	1	The sampling frequency is (1/256)/2 of the system clock: 16/22.05/24 kHz
1	0	The sampling frequency is (1/256)/4 of the system clock: 8/11.025/12 kHz
1	1	Setting not allowed

22.2.3 I2S Transceiver Control Register 1 (I2SCONO1)

This register is used for setting I2S.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SCONO1	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	MSTO	CLKO	CLRO
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8200_0008
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **CLRO** (Bit 0)
  When the CLRO bit is cleared, the values of the FIFO write address register (I2SWADRI) and FIFO read address register (I2SRADRI) are cleared. In the initial state, this bit is cleared. When writing data in or reading data from the FIFO register (I2SFIFOI), set this bit to "1". To simply clear the FIFO write address register and the read address register, clear this bit and set it to "1" again. When the CLRO bit is set to 0, the value of the I2SISTO register changes as follows according to the I2SDNOO register.

  1. When the CLRO bit is set to 0 while the value of the I2SDNOO register is not 0x000, the almost full/full interrupt request remains unchanged and the almost empty/empty interrupt request bit is set.
  2. When the CLRO bit is set to 0 while the value of the I2SDNOO register is 0x000, the almost empty/empty interrupt request is set if the value of the SAIRADRO or the SAIWADRO register has already been changed. In other cases, no interrupt request bit is set.
  3. EMPTY interrupt will not be generated by setting this bit to "1" after resetting this module with the RESET signal and then setting this bit to "0".

CLRO	Description
0	Resets the I2S transmit address pointer
1	Enables operation of the I2S transmit address pointer

- **CLKO** (Bit 1)
  The CLKO bit specifies with/without system clock output from the system clock pin (CKOUTD). When this bit is cleared, the system clock pin (CKOUTD) output is set to the "L" level.

CLKO	Description
0	Without system clock output (CKOUTD pin is fixed to the "L" level)
1	With system clock output

- **MSTO** (Bit 2)
  The MSTO bit is used to determine whether I2S is used in master mode or slave mode. Since this LSI does not support the slave mode, fix this bit to "0".

MSTO	Description
0	Master mode
1	Setting not allowed

22.2.4 I2S Transceiver Almost Full Threshold Value Setting Register (I2SAFRO)

This register sets a threshold value of the number of FIFO data items for which Almost Full interrupt is generated.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SAFRO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SAFRO	—*	—*	—*	—*	AFO 11	AFO 10	AFO 09	AFO 08	AFO 07	AFO 06	AFO 05	AFO 04	—*	—*	—*	—*
At reset	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0

Address: 0x8200_000C
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **AFO04 to AFO11** (Bit 4 to 11)
  Set a threshold value of the number of FIFO data items for which Almost Full interrupt is generated.
  A value greater than 0x900, which is the upper limit of the number of FIFO data items, can be set; however, if so set, an Almost Full interrupt request will not be generated.
  When an address is written into the FIFO read address register (I2SRADRO) or the FIFO write address register (I2SWADRO), or when the address value is incremented or data is written to this register, the number of FIFO data items is compared with the Almost Full threshold value.

22.2.5 I2S Transceiver Almost Empty Threshold Value Setting Register (I2SAERO)

This register sets a threshold value of the number of FIFO data items for which Almost Empty interrupt is generated.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SAERO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	AEO 11	AEO 10	AEO 09	AEO 08	AEO 07	AEO 06	AEO 05	AEO 04	—*	—*	—*	—*
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8200_0010
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **AEO04 to AEO11** (Bit 4 to 11)
  Set a threshold value of the number of FIFO data items for which Almost Empty interrupt is generated.
  A value greater than 0x900, which is the upper limit of the number of FIFO data items (number of words), can be set; however, if so set, an Almost Empty interrupt request will not be generated.
  When an address is written into the FIFO read address register (I2SRADRO) or the FIFO write address register (I2SWADRO), or when the address value is incremented or data is written to this register, the number of FIFO data items is compared with the Almost Empty threshold value.

22.2.6 I2S Transceiver Interrupt Mask Register (I2SIMRO)

This register is used for enabling or masking interrupts at Full, Almost Full, Empty, or Almost Empty of FIFO.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SIMRO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SIMRO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	AEMO	EMO	AFMO	FMO
At reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Address: 0x8200_0014
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **FMO** (Bit 0)
  Masks/enables interrupt at FIFO Full.

FMO	Description
0	Enables interrupt at FIFO Full
1	Masks interrupt at FIFO Full

- **AFMO** (Bit 1)
  Masks/enables interrupt at FIFO Almost Full.

AFMO	Description
0	Enables interrupt at FIFO Almost Full
1	Masks interrupt at FIFO Almost Full

- **EMO** (Bit 2)
  Masks/enables interrupt at FIFO Empty.

EMO	Description
0	Enables interrupt at FIFO Empty
1	Masks interrupt at FIFO Empty

- **AEMO** (Bit 3)
  Masks/enables interrupt at FIFO Almost Empty.

AEMO	Description
0	Enables interrupt at FIFO Almost Empty
1	Masks interrupt at FIFO Almost Empty

22.2.7 I2S Transceiver Interrupt Status Register (I2SISTO)

This register indicates the generation status of FIFO interrupts of Full, Almost Full, Empty, and Almost Empty. By writing "1" in this register, the interrupt can be cleared. Writing "0" is invalid. This register always indicates an interrupt status independently of the value of the interrupt mask register.
When the CLRO bit is reset to 0 after the reset signal is released and then the CLRO bit of the 12S transceiver control register 1 is set to 1, the interrupt request bitwill be set and it may occur that the value of the I2SISTO register differs from 0x00. Mask the interrupt when setting the CLRO bit to 0, if necessary. Restart the operation by writing data in FIFO after clearing all the interrupts through software.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SISTO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	AEIO	EIO	AFIO	FIO
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8200_0018
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **FIO** (Bit 0)
  This bit indicates the interrupt status at FIFO Full.

FIO	Description
0	Without interrupt request at FIFO Full
1	With interrupt request at FIFO Full

- **AFIO** (Bit 1)
  This bit indicates the interrupt status at FIFO Almost Full.

AFIO	Description
0	Without interrupt request at FIFO Almost Full
1	With interrupt request at FIFO Almost Full

- **EIO** (Bit 2)
  This bit indicates the interrupt status at FIFO Empty

EIO	Description
0	Without interrupt request at FIFO Empty
1	With interrupt request at FIFO Empty

- **AEIO** (Bit 3)
  This bit indicates the interrupt status at FIFO Almost Empty

AEIO	Description
0	Without interrupt request at FIFO Almost Empty
1	With interrupt request at FIFO Almost Empty

22.2.8 I2S Transceiver FIFO Write Address Register (I2SWADRO)

This register specifies the FIFO address at which the CPU writes data into FIFO.
When the CPU writes data in FIFO, this register value is automatically incremented. When the address (register value) exceeds 0x900 as a result of the value increment, the address (register value) becomes 0x000. Use this register when changing the data that has already been written to FIFO.
The address that is read from the CPU can be specified in this register only when I2S output is stopped. When I2S reads the address from FIFO, the value set in this register is incremented automatically.
When a value greater than 0x900 (upper limit of the FIFO address) is written to this register, the data write operation becomes invalid.
When this register is reset or the CLRO bit of the I2S transceiver control register 1 (I2SCONO1) is cleared, the value of this register becomes 0x000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SWADRO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SWADRO	—*	—*	—*	—*	WAO11	WAO10	WAO09	WAO08	WAO07	WAO06	WAO05	WAO04	WAO03	WAO02	WAO01	WAO00
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8200_001C
Access: R/W
Acess size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]
- **WAO00 to WAO11** (Bit 0 to 11)
  These bits specify the FIFO address at which the CPU writes data into FIFO.

22.2.9 I2S Transceiver FIFO Read Address Register (2SRADRO)

This register specifies the address at which I2S reads data from FIFO.
When I2S reads data from FIFO, the value set in this register is incremented automatically. When the address exceeds 0x900 as a result of the value increment, the address (register value) becomes 0x000. To change the address for I2S to read from FIFO, use this register. The address that is indicated by the FIFO write address register (I2SWADRO) cannot be exceeded.
When a value greater than 0x900 (upper limit of the FIFO address) is written to this register, the data write operation becomes invalid.
When this register is reset or the CLRO bit of I2S transceiver control register 1 (I2SCONO1) is cleared, the value of this register becomes 0x000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SRADRO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SRADRO	—*	—*	—*	—*	RAO 11	RAO 10	RAO 09	RAO 08	RAO 07	RAO 06	RAO 05	RAO 04	RAO 03	RAO 02	RAO 01	RAO 00
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8200_0020
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]
- **RAO00 to RAO11** (Bit 0 to 11)
  These bits specify the address at which I2S reads data from FIFO.

22.2.10 I2S Transceiver FIFO Occupied Data Size Register (I2SDNOO)

This register is a read only register that indicates the number of data items stored in FIFO. Data write operation is disabled. When this register is reset or the CLRO bit of I2S transceiver control register 1 (I2SCONO1) is cleared, value 0x000 is set in this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SDNOO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SDNOO	—*	—*	—*	—*	DNO 11	DNO 10	DNO 09	DNO 08	DNO 07	DNO 06	DNO 05	DNO 04	DNO 03	DNO 02	DNO 01	DNO 00
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8200_0024
Access: R
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]
- **DNO00 to DNO11** (Bit 0 to 11)
  These bits indicate the difference between the FIFO write pointer and the FIFO Read pointer.

[I2SDNOO Operation]
The operation of the I2SDNOO register varies according to the condition, as shown below.

Condition 1: When I2SWADRO > I2SRADRO,
I2SDNOO = I2SWADRO - I2SRADRO
Condition 2: When I2SWADRO < I2SRADRO,
I2SDNOO = 0x900 – (I2SRADRO – I2SWADRO)
Condition 3: When the values of I2SWADRO and I2SRADRO are equal
(1) At reset or if CLRO (bit 0 of the I2SCONO1 register) is set to "0",
I2SDNOO = 0x000 (Empty status)
(2) If the CPU writes FIFO data, making two register values equal,
I2SDNOO = 0x900 (Full status)
(3) If the CPU rewrites the value of I2SWADRO or I2SRADRO, making the values of the two registers equal or if bit 1 of the I2SCONO1 register is set to "0",
I2SDNOO = 0x000 (Empty status)
(4) If I2S reads data into FIFO by transmitting serial data, making the values of the two registers equal,
I2SDNOO = 0x000 (Empty status)

In (2), (3), and (4) of condition 3, an interrupt in that state such as Full, Almost Full, Empty, Almost Empty is generated. However, an interrupt is not generated in (1) only.

22.3 Operational Description

22.3.1 FIFO Memory

Figure 1 shows the relationship among the write address register, read address register, and the occupied data size register.

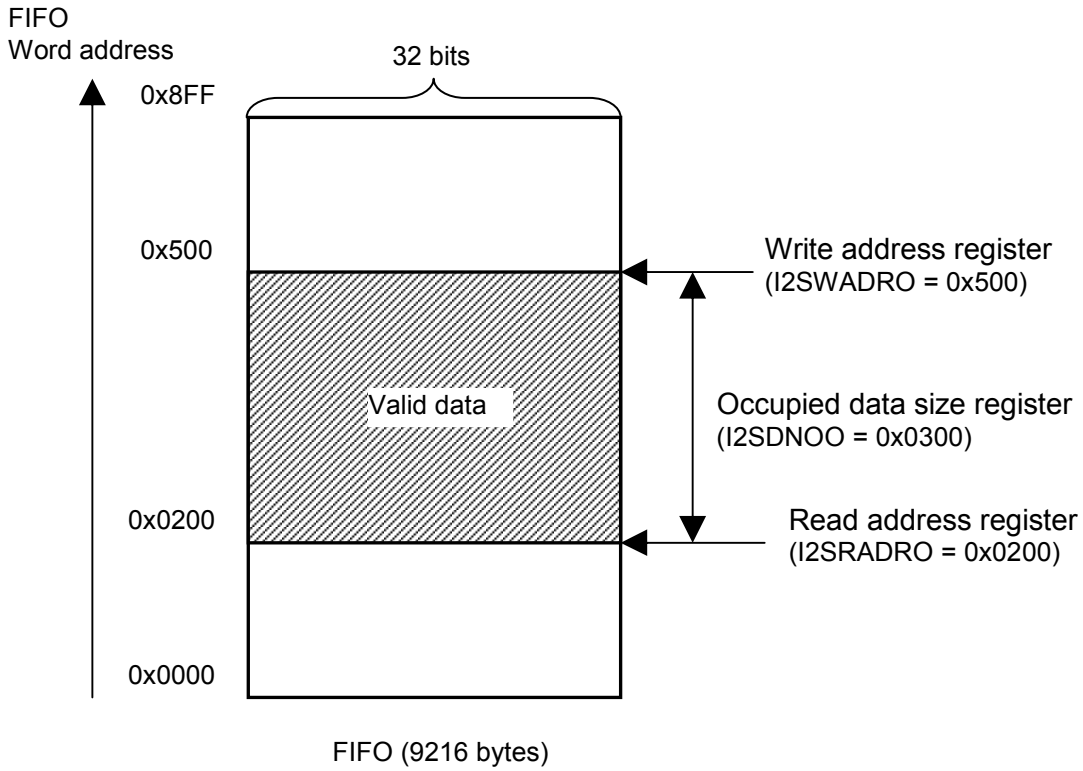


Figure 22-2 Configuration and Address Map of FIFO

22.3.2 Single-Port SRAM Access Contention between I2S and APB bus

Because FIFO uses single-port SRAM, FIFO data write timing from the I2S module and FIFO data write timing from the APB bus may collide.
In this case, priority is given to the access from I2S and the access from the APB bus is set to a wait state. The wait duration is either 1 clk or 2 clk, depending on the collision timing.
Therefore, when DMA transfer is performed for all the 9.2 Kbytes with 48 kHz sampling rate (one time of RAM access at every 20.8 μs) and APB clock at 30 MHz (when there is no wait, 2304 words × 2 clk = 153.6 μs), a maximum of eight collisions occur and the penalty will be 16 clk (0.5 μs).

22.3.3 Serial Data Format

The format of serial data is shown below.

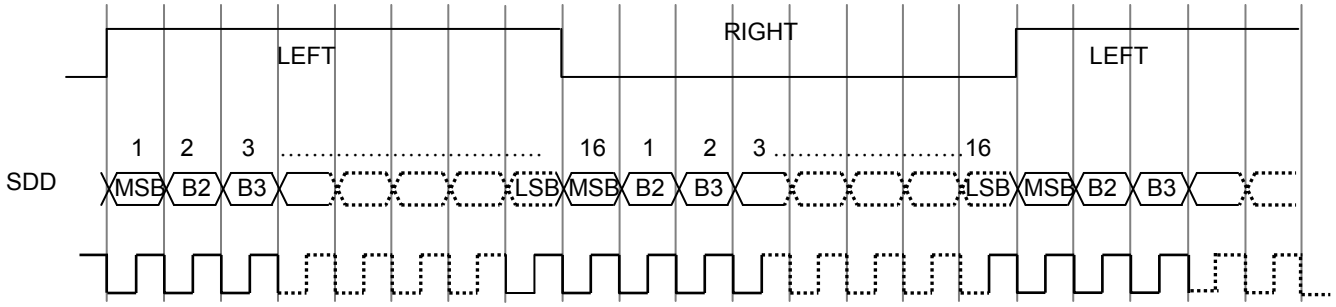


Figure 22-3 Normal Format (MSB first, LEFT channel word select signal = "H" level, no delay)

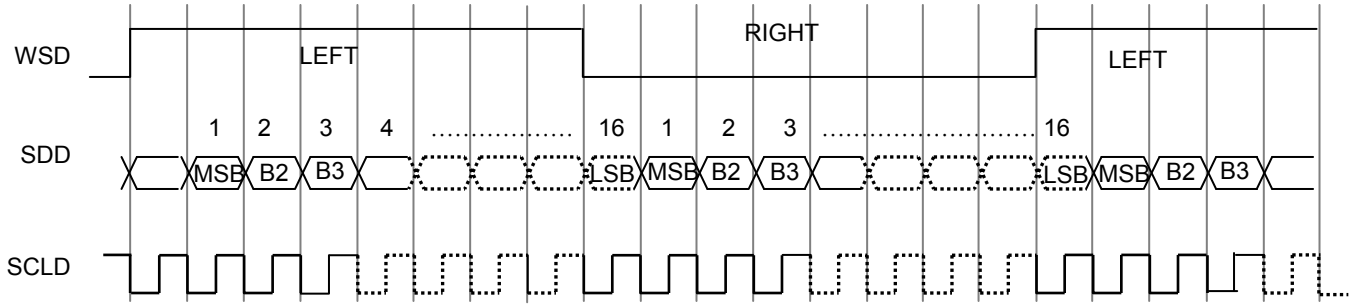


Figure 22-4 Normal Format (MSB first, LEFT channel word select signal = "H" level, with delay)

22.3.4 Transmit/Receive Timing Diagrams

Figures 22-5 to 22-8 show the output waveforms when serial data transmission starts, stops, and restarts under the standard condition (channel select signal is Low at output of left channel data). They are the waveforms that are generated when 1) written data in FIFO has been output part of the way, 2) output of that data is stopped halfway, 3) transmission is restarted, and 4) transmission is continued until the FIFO becomes empty.

At the start of transmission under the standard condition (channel select signal is Low at output of left channel data), output of effective transmission data starts when, after the RUNO bit is set to "1" after being reset and after the channel select signal is output for two cycles, the first timing of left channel data output occurs.

At the start of transmission under the channel select signal inversion condition (channel select signal is High at left channel data output), output of effective transmission data starts when, after the RUNO bit is set to "1" and the channel select signal is output for three cycles, the first timing of left channel data output occurs.

To stop operation temporarily, set the RUNO bit of the I2SCONO0 register to 0.

Also when transmission is restarted by setting the RUNO bit of the I2SCONO0 register to "1", output starts from the left channel. Transmission of effective data starts when, after the RUNO bit is set to "1" and the channel select signal is changed, data is output from the left channel after 0.5 or 1.5 cycles of the channel select signal. However, even if output is delayed by 1.5 cycles, silence data is inserted before the effective data.

To output the next data continuously after output of the entire data, make the I2SAERO register setting as required, check the almost empty flag, and write data to FIFO as soon as the flag is set.

As normal operation, it is assumed that a half of the capacity of FIFO is used each time when data is written to FIFO continuously. If the empty flag is set, an end of data transmission is assumed unless data is output continuously. Therefore, when outputting data after that, set the RUNO bit of the I2SCONO0 register to "0", set the CLRO bit of the I2SCONO1 register to "0" to reset the I2S pointer after the entire transmit data is output, enable the I2S pointer by setting the CLRO bit to "1", write data to FIFO again, and then output data by setting the RUNO bit to "1". In this case, since an interrupt flag may be set in the I2SISTO register, start outputting data after clearing the flag if processing the interrupt request is not required.

Output timing of the subsequent effective data is the same as the timing after resetting described before.

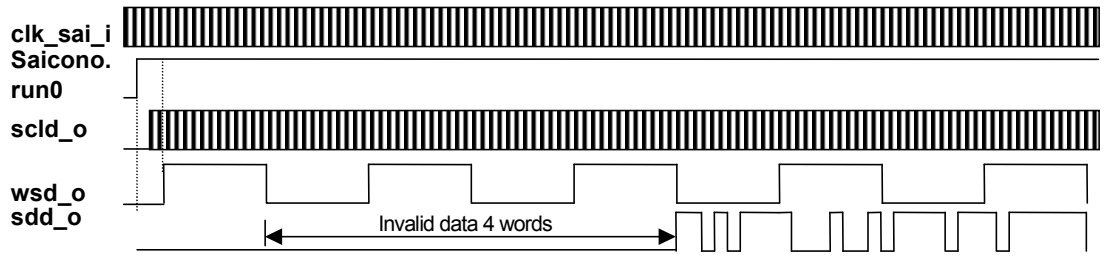


Figure 22-5 Waveform of transmit clock, channel select signal, and transmit data signal at the start of transmission (under the standard condition)

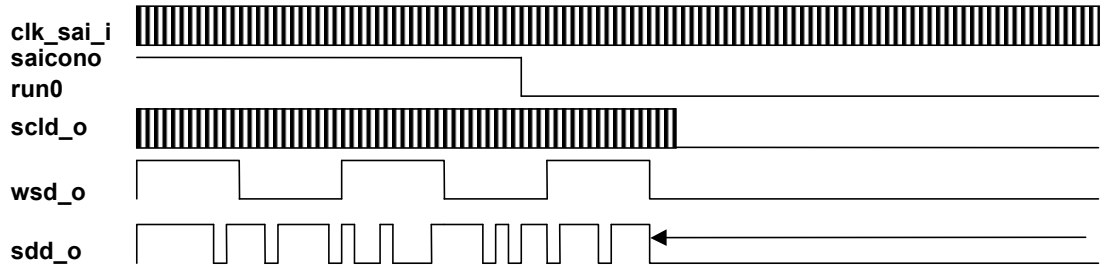


Figure 22-6 Waveform of transmit clock, channel select signal, and transmit data signal at the stopping of transmission (under the standard condition)

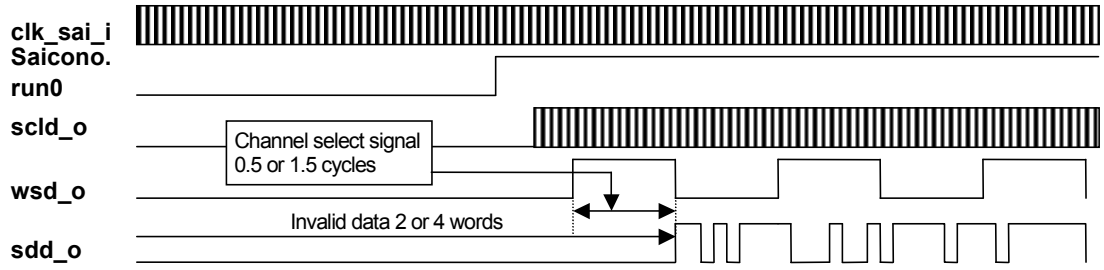


Figure 22-7 Waveform of transmit clock, channel select signal, and transmit data signal at the restart of transmission (under the standard condition)

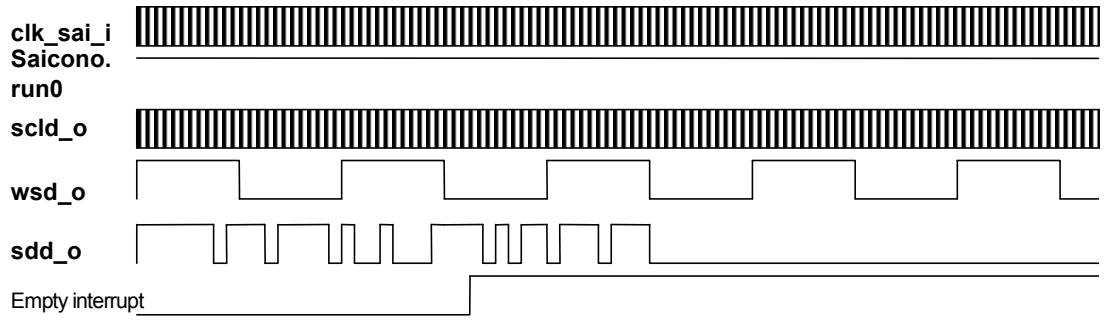


Figure 22-8 Waveform of transmit clock, channel select signal, and transmit data signal at exhaustion of transmit data (under the standard condition)

Chapter 23

I2S Reception

Chapter 23 I2S Reception

23.1 Overview

Voice or audio data is input serially from the external ADC. The received data is written to FIFO according to the serial audio interface format. The written data in FIFO can be read from the CPU.
Since FIFO memory is shared by transmit and receive operations, operation of the 12S receive module must be stopped at transmission.

Features:

- Sampling frequency: (8/11.025/12/16/22.05/24/32/44.1/48 kHz)
- System clock: Supports 256Fs
- Channel data length: 16 bits
- Operation mode: Master only
- Serial data format: with/without 1-bit delay specifiable, MSB first only, standard/inverted output of word select signal
- FIFO size: 9,216 bytes (shared with the I2S transmit module)

23.1.1 Configuration

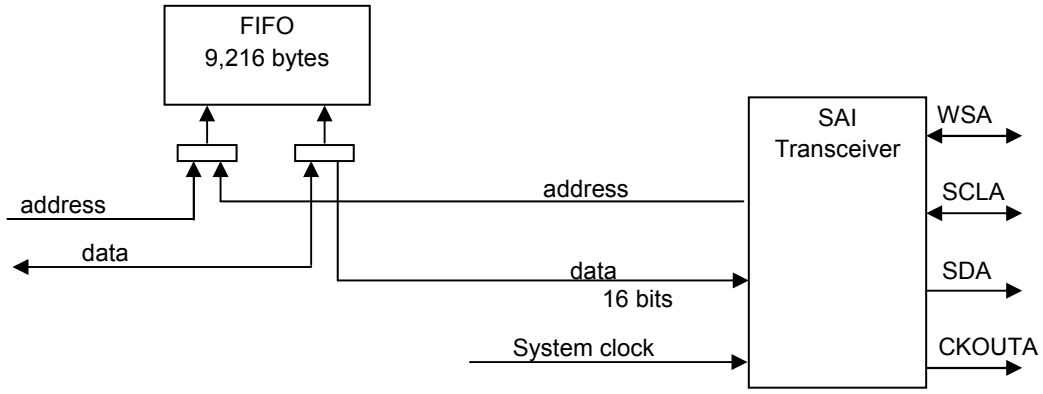


Figure 23-1 Block Configuration

23.1.2 Pin List

Pin name	I/O	Function
CKOUTA/CLKOUT	O	I2S receive system clock
SDA	I	I2S receive data
WSA	O	I2S receive channel selection
SCLA/SCL	O	I2S receive transfer clock

23.1.3 Control Register List

Address	Register name	Symbol	R/W	Initial value [H]
0x8400_0000	I2S receiver FIFO register	I2SFIFOI	R	0x0000_0000
0x8400_0004	I2S receiver control register 0	I2SCONI0	R/W	0x0000_0000
0x8400_0008	I2S receiver control register 1	I2SCONI1	R/W	0x0000_0000
0x8400_000C	I2S receiver Almost Full threshold value setting register	I2SAFRI	R/W	0x0000_0900
0x8400_0010	I2S receiver Almost Empty threshold value setting register	I2SAERI	R/W	0x0000_0000
0x8400_0014	I2S receiver interrupt mask register	I2SIMRI	R/W	0x0000_000F
0x8400_0018	I2S receiver interrupt status register	I2SISTI	R/W	0x0000_0000
0x8400_001C	I2S receiver FIFO write address register	I2SWADRI	R/W	0x0000_0000
0x8400_0020	I2S receiver FIFO read address register	I2SRADRI	R/W	0x0000_0000
0x8400_0024	I2S receiver FIFO occupied data size register	I2SDNOI	R	0x0000_0000

23.2 Control Registers

23.2.1 I2S Receiver FIFO Register (I2SFIFOI)

This register is used for reading data from FIFO. Data write operation to this register is disabled.
The FIFO data of the address indicated by the FIFO read address register (12SRADRI) can be read by reading data from this register. In this case, the counter value of the FIFO read address register is incremented automatically.
Since data transferred earlier is stored in I2SFIFO_LOW[15:0], normally the stored data becomes LEFT data.
Since data transferred later is stored in I2SFIFO_HIGH[15:0], normally the stored data becomes RIGHT data.
When this register is reset, value 0x00000000 is set.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SFIFOI	I2SFIFOI_HIGH[15:0]															
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2SFIFOI_LOW[15:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_0000
Access: R
Access size: 32 bits

[Description of Bits]

- **I2SFIFOI_LOW[15:0]** (Bit 0 to 15)
  FIFO register that is used for reading received data.
  Since data transferred earlier is stored in I2SFIFII_LOW[15:0], normally the stored data becomes LEFT data.

- **I2SFIFOI_HIGH[15:0]** (Bit 16 to 31)
  FIFO register that is used for reading received data.
  Since data transferred following 12SFIFOI_LOW[15:0] is stored in I2SFIFOI_HIGH[15:0], normally the stored data becomes RIGHT data.

23.2.2 I2S Receiver Control Register 0 (I2SCONI0)

This register is used for setting I2S.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SCONI0	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	FSI1	FSI0	RUNI	DLYI	WSLI	MSBI	—*	SFSI
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_0004
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.
If the contents of bits other than bit 5 (RUNI) are changed while I2S is operating, the operation cannot be guaranteed.

[Description of Bits]

- **SFSI** (Bit 0)
  The SFSI bit specifies the system clock frequency. Since this LSI supports 256Fs only, this bit must be fixed to "0".

SFSI	Description
0	256Fs
1	Setting not allowed

- **MSBI** (Bit 2)
  The MSBIbit specifies the format of serial receive data. Since this LSI supports MSB first only, this bit must be fixed to "0".

SFSI	Description
0	MSB first
1	Setting not allowed

- **WSLI** (Bit 3)
  The WSL1 bit specifies the format of the word select signal of the LEFT channel. For the format of the word select signal of the RIGHT channel, the inverted value of that format of the word select signal of the LEFT channel which is set in this bit is set.

WSLI	Description
0	LEFT channel word select signal = "L" level
1	LEFT channel word select signal = "H" level

- **DLYI** (Bit 4)
  The DLYI bit specifies one-clock delay/no-delay in serial receive data.

DLYI	Description
0	No serial receive data delay
1	Delay in serial receive data

- **RUNI** (Bit 5)
  The RUNI bit specifies the start and the stop of serial receive operation.
  When this bit is cleared, the output of the word select signal pin (WSA) and receive clock pin (SCLA) is set to the "L" level after reception of data from the Right channel and the operation stops. When this bit is set to "1", reception of serial data starts. When this bit is cleared during reception of serial data, operation also stops after data reception from the Right channel is terminated. When this bit is set to "1" while FIFO is Full, data is output from the word select signal pin (WSA) and receive clock pin (SCLA) normally, but serial data cannot be received.

RUNI	Description
0	Stops serial receive operation
1	Starts serial receive operation

- **FSI1, FSI0** (Bit 6,7)
  Bits FSI0 and FSI1 specify the sampling frequency level.
  The system clock of I2S is specified with the I2SCLKSEL[1:0] bit of configuration register 0.

FSI1	FSI0	Description
0	0	The sampling frequency is 1/256 of the system clock: 32/44.1/48 kHz
0	1	The sampling frequency is (1/256)/2 of the system clock: 16/22.05/24 kHz
1	0	The sampling frequency is (1/256)/4 of the system clock: 8/11.025/12 kHz
1	1	Setting not allowed

23.2.3 I2S Receiver Control Register 1 (I2SCONI1)

This register is used for setting I2S.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SCONI1	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—	—*	MSTI	CLKI	CLRI
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_0008
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **CLRI** (Bit 0)
  When the CLRI bit is cleared, the values of the FIFO write address register (I2SWADRI) and FIFO read address register (I2SRADRI) are cleared. In the initial status, this bit is cleared. To read data from the FIFO register (I2SFIFOI), set this bit to "1". To simply clear the FIFO write address register and the FIFO read address register, clear this bit and set it again.
  When the value of the I2SDNOI register is 0, the FIFO EMPTY interrupt is not generated even if this pointer is reset by setting this bit to 0.

CLRI	Description
0	Resets the I2S receive address pointer
1	Enables operation of the I2S receive address pointer

- **CLKI** (Bit 1)
  The CLKI bit specifies with/without system clock output from the system clock pin (CKOUTA). When this bit is cleared, the system clock pin (CKOUTA) output is set to the "L" level.

CLKI	Description
0	Without system clock output
1	With system clock output

- **MSTI** (Bit 2)
  The MSTI bit is used to determine whether I2S is used in master mode or slave mode. Since this LSI does not support the slave mode, fix this bit to "0".

MSTI	Description
0	Master mode
1	Setting not allowed

23.2.4 I2S Receiver Almost Full Threshold Value Setting Register (I2SAFRI)

This register sets a threshold value of the number of FIFO data items for which Almost Full interrupt is generated.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SAFRI	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SAFRI	—*	—*	—*	—*	AFI 11	AFI 10	AFI 09	AFI 08	AFI 07	AFI 06	AFI 05	AFI 04	—*	—*	—*	—*
At reset	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0

Address: 0x8400_000C
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **AFI04 to AFI11** (Bit 4 to 11)
  Set a threshold value of the number of FIFO data items for which Almost Full interrupt is generated.
  A value greater than 0x900, which is the upper limit of the number of FIFO data items (number of words), can be set; however, if so set, an Almost Full interrupt request will not be generated.
  When an address is written into the FIFO read address register (I2SRADRI) or the FIFO write address register (I2SWADRI), or when the address value is incremented or data is written to this register, the number of FIFO data items is compared with the Almost Full threshold value.

23.2.5 I2S Receiver Almost Empty Threshold Value Setting Register (I2SAERI)

This register sets a threshold value of the number of FIFO data items for which Almost Empty interrupt is generated.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SAERI	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SAERI	—*	—*	—*	—*	AEI 11	AEI 10	AEI 09	AEI 08	AEI 07	AEI 06	AEI 05	AEI 04	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_0010
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **AEI04 to AEI11** (Bit 4 to 11)
  Set a threshold value of the number of FIFO data items for which Almost Empty interrupt is generated.
  A value greater than 0x900, which is the upper limit of the number of FIFO data items (number of words), can be set; however, if so set, an Almost Empty interrupt request will not be generated.
  When an address is written into the FIFO read address register (I2SRADRI) or the FIFO write address register (I2SWADRI), or when the address value is incremented or data is written to this register, the number of FIFO data items is compared with the Almost Empty threshold value.

23.2.6 I2S Receiver Interrupt Mask Register (I2SIMRI)

This register is used for enabling or masking interrupts at Full, Almost Full, Empty, or Almost Empty of FIFO.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SIMRO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SIMRO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	AEMI	EMI	AFMI	FMI
At reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Address: 0x8400_0014
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **FMI** (Bit 0)
  Masks/enables interrupt at FIFO Full

FMI	Description
0	Enables interrupt at FIFO Full
1	Masks interrupt at FIFO Full

- **AFMI** (Bit 1)
  Masks/enables interrupt at FIFO Almost Full

AFMI	Description
0	Enables interrupt at FIFO Almost Full
1	Masks interrupt at FIFO Almost Full

- **EMO** (Bit 2)
  Masks/enables interrupt at FIFO Empty

EMO	Description
0	Enables interrupt at FIFO Empty
1	Masks interrupt at FIFO Empty

- **AEMO** (Bit 3)
  Masks/enables interrupt at FIFO Almost Empty

AEMO	Description
0	Enables interrupt at FIFO Almost Empty
1	Masks interrupt at FIFO Almost Empty

23.2.7 I2S Receiver Interrupt Status Register (I2SISTI)

This register indicates the generation status of FIFO interrupts of Full, Almost Full, Empty, and Almost Empty. By writing "1" in this register, the interrupt can be cleared. Writing "0" is invalid. This register always indicates an interrupt status independently of the value of the interrupt mask register.
When the CLRO bit is reset to 0 after the reset signal is released and then the CLRO bit of the 12S receiver control register 1 is set to 1, an interrupt request will be set and it may occur that the value of the I2SISTO register differs from 0x00. Mask the interrupt when setting the CLRO bit to 0, if necessary. Restart the operation by writing data in FIFO after clearing all the interrupts through software.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SISTI	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	AEII	EII	AFII	FII
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_0018
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **FII** (Bit 0)
  This bit indicates the interrupt status at FIFO Full.

FII	Description
0	Without interrupt request at FIFO Full
1	With interrupt request at FIFO Full

- **AFII** (Bit 1)
  This bit indicates the interrupt status at FIFO Almost Full.

AFII	Description
0	Without interrupt request at FIFO Almost Full
1	With interrupt request at FIFO Almost Full

- **EII** (Bit 2)
  This bit indicates the interrupt status at FIFO Empty.

EII	Description
0	Without interrupt request at FIFO Empty
1	With interrupt request at FIFO Empty

- **AEII** (Bit 3)
  This bit indicates the interrupt status at FIFO Almost Empty.

AEII	Description
0	Without interrupt request at FIFO Almost Empty
1	With interrupt request at FIFO Almost Empty

23.2.8 I2S Receiver FIFO Write Address Register (I2SWADRI)

This register specifies the FIFO address at which 12S writes receive data into FIFO.
When I2S writes data into FIFO, the counter value of this register is incremented. When the address (register value) exceeds 0x900 as a result of the counter increment, the address (register value) is set to 0x000. Use this register if it is required to change the FIFO address to which I2S writes receive data.
When a value greater than 0x900 (upper limit of the FIFO address) is written in this register, the data write operation becomes invalid.
When this register is reset or the CLRI bit of the I2S receiver control register 1 (I2SCONI1) is cleared, the value of this register becomes 0x000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SWADRI	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	WAI 11	WAI 10	WAI 09	WAI 08	WAI 07	WAI 06	WAI 05	WAI 04	WAI 03	WAI 02	WAI 01	WAI 00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_001C
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]
- **WAI00 to WAI11** (Bit 0 to 11)
  These bits specify the FIFO address at which I2S writes receive data into FIFO.

23.2.9 I2S Receiver FIFO Read Address Register (I2SRADRI)

This register specifies the address at which data is read from the FIFO register (I2SFIFOI).
When data is read from the FIFO register (I2SFIFOI), the counter value that is set in this register is automatically incremented. When the address (register value) exceeds 0x900 as a result of the counter increment, the address (register value) is set to 0x000. The address value cannot exceed the address that is indicated by the write address register (I2SWADRI).
When a value greater than 0x900 (upper limit of the FIFO address) is written to this register, the data write operation becomes invalid.
When this register is reset or the CLRI bit of 12S receiver control register 1 (I2SCONI1) is cleared, the value of this register becomes 0x000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SWADRI	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SWADRI	—*	—*	—*	—*	RAI 11	RAI 10	RAI 09	RAI 08	RAI 07	RAI 06	RAI 05	RAI 04	RAI 03	RAI 02	RAI 01	RAI 00
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_0020
Access: R/W
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]
- **RAI00 to RAI11** (Bit 0 to 11)
  These bits specify the address at which data is read from the FIFO register (I2SFIFOI).

23.2.10 I2S Receiver FIFO Occupied Data Size Register (I2SDNOI)

This register is a read only register that indicates the number of data items stored in FIFO. Data write operation is disabled. When this register is reset or the CLRI bit of I2S receiver control register 1 (I2SCONI1) is cleared, the value of this register becomes 0x000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SWADRO	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	DNI 11	DNI 10	DNI 09	DNI 08	DNI 07	DNI 06	DNI 05	DNI 04	DNI 03	DNI 02	DNI 01	DNI 00
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_0024
Access: R
Access size: 32 bits

[Note]
*: The bit is reserved for future extension. In this LSI, "0" is read when any of reserved bits is read and any write operation to such a bit is ignored.

[Description of Bits]

- **DNI00 to DNI11** (Bit 0 to 11)
  These bits indicate the difference between the FIFO write pointer and the FIFO Read pointer.

[I2SDNOI Operation]
The operation of the I2SDNOI register varies according to the condition, as shown below.

Condition 1: When I2SWADRI > I2SRADRI,
I2SDNOI = I2SWADRI – I2SRADRO
Condition 2: When I2SWADRI < I2SRADRI,
I2SDNOI = 0x900 – (I2SRADRI – I2SWADRI)
Condition 3: When the values of I2SWADRI and I2SRADRI are equal

(1) If this register is reset or CLRI (bit 0 of the I2SCONI1 register) is set to "0",
I2SDNOI = 0x000 (Empty status)
(2) If the CPU reads FIFO data, making two register values equal,
I2SDNOI = 0x000 (Empty status)
(3) If the CPU rewrites the value of I2SWADRI or I2SRADRI, making the values of the two registers equal or if bit 1 of the I2SCONO1 register is set to "0",
I2SDNOI = 0x000 (Empty status)
(4) If I2S writes data into FIFO by receiving serial data, making the values of the two registers equal,
I2SDNOI = 0x000 (Full status)

In (2), (3), and (4) of condition 3, an interrupt in that state such as Full, Almost Full, Empty, Almost Empty occurs. However, an interrupt is not generated in (1) only.

23.3 Operational Description

23.3.1 FIFO Memory

Since the FIFO memory is shared by the I2S transmit module and the I2S receive module, transmit operation and receive operation cannot be performed concurrently. Therefore, if the RUNO bit of I2S transceiver control register of the I2S transmit module and the RUNI bit of I2S receiver control register 0 of the I2S receive module are set to the operation start level concurrently, the operation cannot be guaranteed. Be sure to set either one or the other to the operation start level. Thus, at transmission, the RUNO bit of I2S transceiver control register 0 of the I2S transmit module must be set to the operation stop level.

Figure 1 shows the relationship among the write address register, read address register, and the occupied data size register.

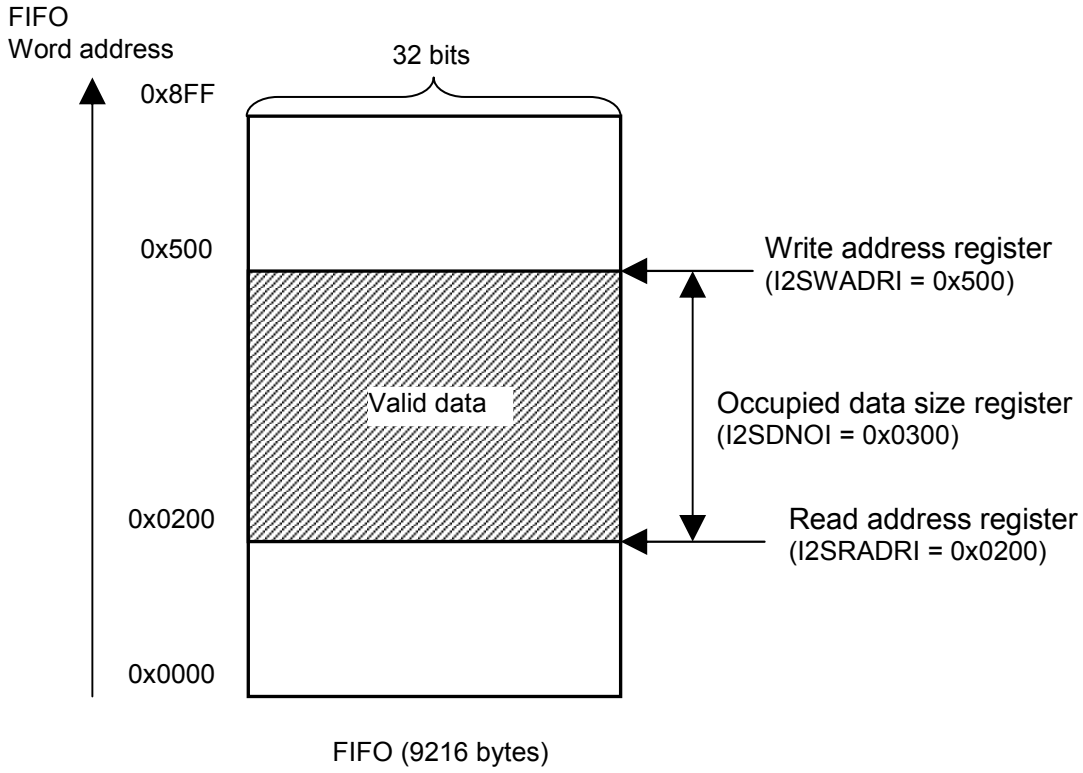


Figure 23- 2 Configuration and address map of FIFO

23.3.2 Single-port SRAM Access Contention between I2S and APB bus

Because FIFO uses single-port SRAM, FIFO data write timing from the I2S module and FIFO data write timing from the APB bus may collide.
In this case, priority is given to the access from I2S and the access from the APB bus is set to a wait state. The wait duration is either 1 clk or 2 clk, depending on the collision timing.
Therefore, when DMA transfer is performed for all the 9.2 Kbytes with 48 kHz sampling rate (one time of RAM access at every 20.8 μs) and APB clock at 30 MHz (when there is no wait, 2304 words × 2 clk = 153.6 μs), a maximum of eight collisions occur and the penalty will be 16 clk (0.5 μs).

23.3.3 Serial Data Format

The format of serial data is shown below.

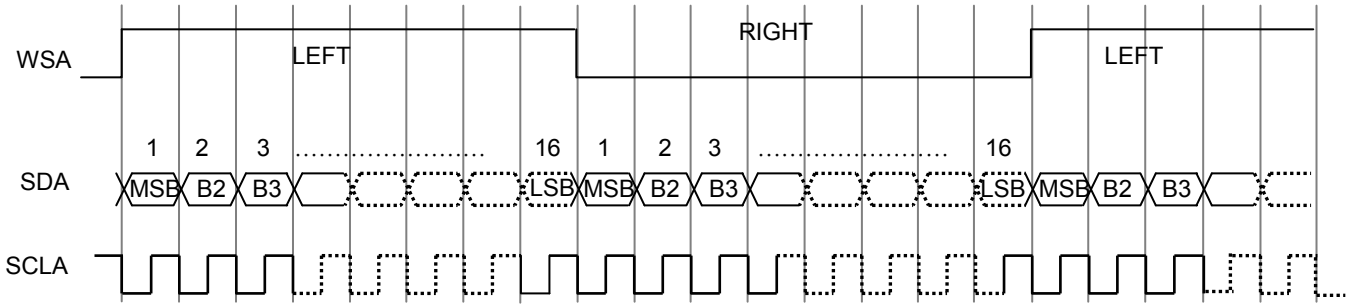


Figure 23-3 Normal Format (MSB first, LEFT channel word select signal = "H" level, no delay)

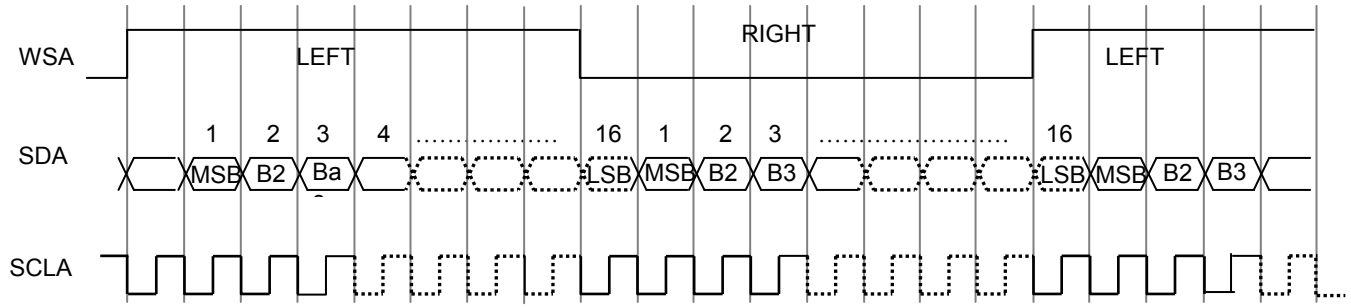


Figure 23-4 Normal Format (MSB first, LEFT channel word select signal = "H" level, with delay)

Chapter 24

NAND Flash Controller

Chapter 24 NAND Flash Controller

24.1 Overview

The NAND Flash controller is a module for controlling data transfer between Smart Media (or NAND Flash memory) and a NAND Flash buffer.

Features

- Smart Media Standard 2000 compliant (at 512 bytes/sector access)
- Support for Smart Media of 8 to 128 Mbytes
- Built-in ECC circuit (for 512 bytes/sector, 2048 bytes/sector data transfer)
- Automatic 512 bytes/sector data transfer function
- Automatic 2048 bytes/sector data transfer function
- Maximum operating frequency of 30 MHz

24.1.1 Configuration

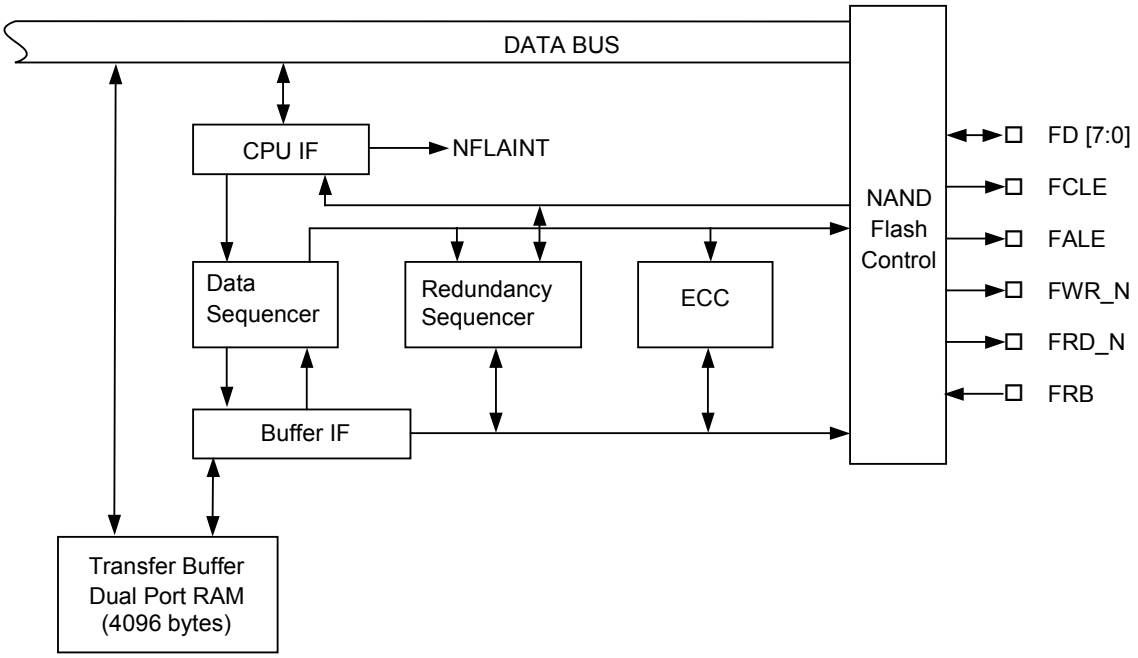


Figure 24-1 Block Configuration

24.1.2 Pin List

Pin name	I/O	Function
FD7–FD0	I/O	Data input-output
FRD_N	O	Read enable
FWR_N	O	Write enable
FRB	I	Ready/busy (1: ready, 0: busy)
FALE	O	Address latch enable
FCLE	O	Command latch enable

24.1.3 Control Register List

Address [H]	Register name	Symbol	R/W	Access [bits]	Initial value [H]
0x8600_0000	Media bank register	MBANK	R/W	32	0x0000_0000
0x8600_0004	Media sequencer control register	MSCTRL	R/W	32	0x0000_0000
0x8600_0008	Media sequencer wait register	MSWAIT	R/W	32	0x0000_0000
0x8600_000C	Media sequencer status register	MSSTS	R	32	0x0000_001E
0x8600_0010	Media sequencer interrupt enable register	MINTENBL	R/W	32	0x0000_0000
0x8600_0014	Media sequencer error status register	MSERR	R	32	0x0000_0000
0x8600_0018	Media command register	MMCMD	W	32	—
0x8600_001C	Media address register	MMADR	W	32	—
0x8600_0020	Media select register	MMSEL	R/W	32	0x0000_0000
0x8600_0024	Media data read control register	MMRDCTL	R/W	32	0x0000_0000
0x8600_0028	Media section option register	MOPTION	R/W	32	0x0000_0000
0x8600_002C	Media section read data storage register	MMRDDATA	R	32	0x0000_0000
0x8600_0030	ECC line parity register 1	ECCLP1	R	32	0xFFFF_FFFF
0x8600_0034	ECC column parity register 1	ECCCP1	R	32	0x003F_003F
0x8600_0038	ECC error pointer register 1	ECCERR1	R	32	0x0000_0000
……	<Reserved>	—	—		—
0x8600_0040	Redundancy part reserve data register 1	HREV1	R/W	32	0x0000_0000
0x8600_0044	Redundancy part data/block status & block address register 1	HSTAD1	R/W	32	0x0000_0000
0x8600_0048	Redundancy part ECC2-High register & redundancy part ECC2-Low/block address 2 register	HECC2	R/W	32	0x0000_0000
0x8600_004C	Redundancy part ECC1-High/block address 2 register & redundancy part ECC1-Low register	HECC1	R/W	32	0x0000_0000
0x8600_0050	ECC line parity register 3	ECCLP3	R	32	0xFFFF_FFFF
0x8600_0054	ECC column parity register 3	ECCCP3	R	32	0x003F_003F
0x8600_0058	ECC error pointer register 3	ECCERR3	R	32	0x0000_0000
……	<Reserved>	—	—		—
0x8600_0060	Redundancy part reserve data register 3	HREV3	R/W	32	0x0000_0000

Address [H]	Register name	Symbol	R/W	Access [bits]	Initial value [H]
0x8600_0064	Redundancy part data/block status & block address register 3	HSTAD3	R/W	32	0x0000_0000
0x8600_0068	Redundancy part ECC4-High register & redundancy part ECC4-Low/block address 4 register	HECC4	R/W	32	0x0000_0000
0x8600_006C	Redundancy part ECC3-High/block address 4 register & redundancy part ECC3-Low register	HECC3	R/W	32	0x0000_0000
0x8600_0070	ECC line parity register 5	ECCLP5	R	32	0xFFFF_FFFF
0x8600_0074	ECC column parity register 5	ECCCP5	R	32	0x003F_003F
0x8600_0078	ECC error pointer register 5	ECCERR5	R	32	0x0000_0000
……	<Reserved>	—	—		—
0x8600_0080	Redundancy part reserve data register 5	HREV5	R/W	32	0x0000_0000
0x8600_0084	Redundancy part data/block status & block address register 5	HSTAD5	R/W	32	0x0000_0000
0x8600_0088	Redundancy part ECC6-High register & redundancy part ECC6-Low/block address 6 register	HECC6	R/W	32	0x0000_0000
0x8600_008C	Redundancy part ECC5-High/block address 6 register & redundancy part ECC5-Low register	HECC5	R/W	32	0x0000_0000
0x8600_0090	ECC line parity register 7	ECCLP7	R	32	0xFFFF_FFFF
0x8600_0094	ECC column parity register 7	ECCCP7	R	32	0x003F_003F
0x8600_0098	ECC error pointer register 7	ECCERR7	R	32	0x0000_0000
……	<Reserved>	—	—		—
0x8600_00A0	Redundancy part reserve data register 7	HREV7	R/W	32	0x0000_0000
0x8600_00A4	Redundancy part data/block status & block address register 7	HSTAD7	R/W	32	0x0000_0000
0x8600_00A8	Redundancy part ECC8-High register & redundancy part ECC8-Low/block address 8 register	HECC8	R/W	32	0x0000_0000
0x8600_00AC	Redundancy part ECC7-High/block address 8 register & redundancy part ECC7-Low register	HECC7	R/W	32	0x0000_0000
0x8600_0100 to 0x8600_01FC	Media data register	MMDATA	R/W	32	Undefined
0x8800_0000 to 0x8800_0FFC	NAND FLASH Buffer memory	—	R/W	8/16/ 32	Undefined

24.2 Control Register Description

24.2.1 Media Bank Register (MBANK)

The media bank register (MBANK) specifies the bank to be used for data transfer between a medium and buffer RAM. A memory bank can thus be specified using the media bank register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBANK	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBANK	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	MBANK[2:0]		
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8600_0000
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.
Access to the CPU bank set in this register is prohibited during data transfer (while the sequencer is in operation).

[Bit Description]

- **MBANK[2:0]** (bits 0 to 2)
  These bits specify the bank of buffer memory to be transmitted/received.

  - When accessing in units of 512 bytes/sector

MBANK[2]	MBANK[1]	MBANK[0]	Description
0	0	0	Uses bank 0 for data transfer between a medium and buffer RAM.
0	0	1	Uses bank 0 for data transfer between a medium and buffer RAM.
0	1	0	Uses bank 0 for data transfer between a medium and buffer RAM.
0	1	1	Uses bank 0 for data transfer between a medium and buffer RAM.
1	0	0	Uses bank 0 for data transfer between a medium and buffer RAM.
1	0	1	Uses bank 0 for data transfer between a medium and buffer RAM.
1	1	0	Uses bank 0 for data transfer between a medium and buffer RAM.
1	1	1	Uses bank 0 for data transfer between a medium and buffer RAM.

Correspondence between bank numbers and RAM addresses

Bank number	RAM address
Bank 7	0x8800_0380 to 0x8800_03FF
Bank 6	0x8800_0300 to 0x8800_037F

Bank 5	0x8800_0280 to 0x8800_02FF
Bank 4	0x8800_0200 to 0x8800_027F
Bank 3	0x8800_0180 to 0x8800_01FF
Bank 2	0x8800_0100 to 0x8800_017F
Bank 1	0x8800_0080 to 0x8800_00FF
Bank 0	0x8800_0000 to 0x8800_007F

- When accessing in units of 2048 bytes/sector

MBANK[2]	MBANK[1]	MBANK[0]	Description
0	0	0	Uses bank 0 for data transfer between a medium and buffer RAM.
0	0	1	Cannot be set.
0	1	0	Cannot be set.
0	1	1	Cannot be set.
1	0	0	Uses bank 1 for data transfer between a medium and buffer RAM.
1	0	1	Cannot be set.
1	1	0	Cannot be set.
1	1	1	Cannot be set.

Correspondence between bank numbers and RAM addresses

Bank number	RAM address
Bank 1	0x8800_0200 to 0x8800_03FF
Bank 0	0x8800_0000 to 0x8800_01FF

24.2.2 Media Sequencer Control Register (MSCTRL)

The media sequencer control register (MSCTRL) specifies the operation of the media sequencer.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSCTRL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	PARITY	ECC	HEAD[1:0]		DLEN[1:0]		DIR	START
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8600_0004
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **START** (bit 0)
  This bit controls the start of data transfer. This bit will always read "0" when read.

START	Description
0	Does not operate the media sequencer.
1	Starts data transfer (read/write).

- **DIR** (bit 1)
  This bit sets the data transfer direction.

DIR	Description
0	Read operation
1	Write operation

- **DLEN[1:0]** (bits 2 and 3)
  These bits specify the number of data bytes transferred by this controller between media and buffer RAM.

DLEN[1]	DLEN[0]	Description
0	0	512 bytes This setting cannot be used for 2048 bytes/sector data transfer (setting prohibited).
0	1	2048 bytes This setting cannot be used for 512 bytes/sector data transfer (setting prohibited).
1	0	0 bytes (no transfer) This is used only for access to the redundancy part during 512 bytes/sector data transfer. This setting cannot be used for 2048 bytes/sector data transfer (setting prohibited).
1	1	0 bytes (no transfer) This is used only for access to the redundancy part during 2048 bytes/sector data transfer. This setting cannot be used for 512 bytes/sector data transfer (setting prohibited).

- **HEAD[1:0]** (bits 4 and 5)
  These bits set whether to read/write the redundancy part.
  Operation cannot be guaranteed if ECC is enabled when accessing only the redundancy part. Be sure to disable ECC first and then access the redundancy part.

HEAD[1]	HEAD[0]	Description
0	0	Reads/writes the redundancy part. (In accordance with Smart Media™ or 2048 bytes/sector physical format)(*1)
0	1	Reads/writes the redundancy part. (No format)
1	x	Does not read/write the redundancy part.

*1: A fixed value of FFFFFFFFh is written into a reserved area and a fixed value of 00010b is written into the first five bits of a block address area in accordance with the redundancy part format of the Smart Media during a write operation. Furthermore, the same value as block address area 1 is written into block address area 2. No control is performed during a read operation.

- **ECC** (bit 6)
  This bit sets ECC to enable or disable.

ECC	Description
0	Enables ECC (transmits ECC when writing, and checks ECC when reading.)
1	Disables ECC.

- **PARITY** (bit 7)
  This bit sets the parity to enable or disable.
  If the redundancy part is in the Smart Media format, the parity bit of a block address is controlled.

PARITY	Description
0	Enables the parity (transmits the parity when writing, and checks the parity when reading.)
1	Disables the parity.

24.2.3 Media Sequencer Wait Register (MSWAIT)

The media sequencer wait register (MSWAIT) controls the wait function of the media sequencer.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSWAIT	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	DISFWT	WAITW[1:0]		WAITR[1:0]	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8400_0008
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **WAITR[1:0]** (bits 0 and 1)
  These bits control the wait function when the sequencer reads data from NAND Flash memory.

WAITR[1]	WAITR[0]	Description
0	0	No wait
0	1	Sets the number of waits to 1.
1	x	Sets the number of waits to 2.

- **WAITW[1:0]** (bits 2 and 3)
  These bits control the wait function when the sequencer writes data into NAND Flash memory.

WAITW[1]	WAITW[0]	Description
0	0	No wait
0	1	Sets the number of waits to 1.
1	x	Sets the number of waits to 2.

- **DISFWT** (bit 4)
  This is a disable Flash ready wait bit. This bit disables the function that, when the sequencer is started, waits for the ready state of NAND Flash memory.

DISFWT	Description
0	Flash ready wait function enabled
1	Flash ready wait function disabled

24.2.4 Media Sequencer Status Register (MSSTS)

The media sequencer status register (MSSTS) shows the status of the media sequencer.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSSTS	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	MRDY	PRTYOK	ECCOK	MSRDY	MSCMP
	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

Address: 0x8600_000C
Access: R
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **MSCMP** (bit 0)
  This bit indicates that the sequencer operation has been completed.
  This bit is cleared by writing "1". Be aware that this bit is not cleared by writing "0".

MSCMP	Description
0	Either the sequencer operation has not been completed, or the sequencer is not operating.
1	The sequencer operation has been completed.

- **MSRDY** (bit 1)
  This bit indicates the ready/busy state of the media sequencer.

MSRDY	Description
0	Busy
1	Ready

- **ECCOK** (bit 2)
  This bit indicates that an ECC error was not detected. The result of ECC operation mentioned here is based on the Smart Media specificationss, and, because of the ECC specifications, an error might be corrected, wrongly detected, or cannot be detected.

ECCOK	Description
0	Error
1	Normal end

- **PRTYOK** (bit 3)
  This bit indicates that a parity error was not detected. This is the result of a parity check in the block address area when the redundancy part was read according to the Smart Media format.
  This bit is cleared when accessed, and is set to "0" if an error occurs at the end of access.

PRTYOK	Description
0	Error
1	Normal end

- **MRDY** (bit 4)
  This bit indicates the ready/busy state of connected NAND Flash memory.

MRDY	Description
0	Busy
1	Ready

24.2.5 Media Sequencer Interrupt Enable Register (MINTENBL)

The media sequencer interrupt enable register (MINTENBL) enables or masks the interrupts of the media sequencer.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MINTENBL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	INTENBL
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8600_0010
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **INTENBL** (bit 0)
  This bit enables or masks the interrupts of the media sequencer.

INTENBL	Description
0	Masks interrupts.
1	Enables interrupts.

24.2.6 Media Sequencer Error Status Register (MSERR)

The media sequencer error status register (MSERR) indicates the error information of the media sequencer.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSERR	—*	—*	PAR8 ERR	PAR7 ERR	ECC8 COR	ECC8 UNC	ECC7 COR	ECC7 UNC	—*	—*	PAR6 ERR	PAR5 ERR	ECC6 COR	ECC6 UNC	ECC5 COR	ECC5 UNC
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	PAR4 ERR	PAR3 ERR	ECC4 COR	ECC4 UNC	ECC3 COR	ECC3 UNC	—*	—*	PAR2 ERR	PAR1 ERR	ECC2 COR	ECC2 UNC	ECC1 COR	ECC1 UNC
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8600_0014
Access: R
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **ECCxUNC** (bits 0, 2, 8, 10, 16, 18, 24, 26)
  These bits indicate that an error of two or more bits has occurred in data area x.
  The result of ECC operation mentioned here is based on the Smart Media specifications, and, because of the ECC specifications, an error might be corrected, wrongly detected, or cannot be detected.
  When accessing Smart Media of 512 bytes/sector, data for the bits corresponding to 'x=1, 2' applies.

ECCxUNC	Description
0	An error of two or more bits has not occurred in data area x.
1	An error of two or more bits has occurred in data area x.

- **ECCxCOR** (bits 1, 3, 9, 11, 17, 19, 25, 27)
  These bits indicate that a correctable 1-bit error has occurred in data area x.
  The result of ECC operation mentioned here is based on the Smart Media specifications, and, because of the ECC specifications, an error might be corrected, wrongly detected, or cannot be detected.
  When accessing Smart Media of 512 bytes/sector, data for the bits corresponding to 'x=1, 2' applies.

ECCxCOR	Description
0	A correctable 1-bit error has not occurred in data area x.
1	A correctable 1-bit error has occurred in data area x.

- **PARxERR** (bits 4, 5, 12, 13, 20, 21, 28, 29)
  These bits indicate that a parity error has occurred in block address area x.
  When accessing Smart Media of 512 bytes/sector, data for the bits corresponding to 'x=1, 2' applies.

PARxERR	Description
0	A parity error has not occurred in block address area x.
1	A parity error has occurred in block address area x.

24.2.7 Media Command Register (MMCMD)

The media command register (MMCMD) writes data to NAND Flash memory as a command.
This register sets CLE="1" and ALE="0" at the start of an operation, and holds CLE="1" after the operation is completed.
"00" is read from this register if a read operation is performed in write only mode.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMCMD	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	MMCMD[7:0]							
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Address: 0x8600_0018
Access: W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **MMCMD[7:0]** (bits 0 to 7)
  These bits specify to execute a write operation of data into NAND Flash memory as a command.

24.2.8 Media Address Register (MMADR)

The media address register (MMADR) writes data to NAND Flash memory as an address.
This register sets CLE="0" and ALE="1" at the start of an operation, and holds CLE="1" after the operation is completed.
"00" is read from this register if a read operation is performed in write only mode.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMADR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	MMADR[7:0]							
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Address: 0x8600_001C
Access: W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **MMADR[7:0]** (bits 0 to 7)
  These bits specify to execute a write operation of data into NAND Flash memory as an address.

24.2.9 Media Select Register (MMSEL)

The media select register (MMSEL) controls the CLE and ALE signals of NAND Flash memory.
The value of each bit is output to the CLE and ALE pins as is.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMSEL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	ALE	CLE
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8600_0020
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **CLE** (bit 0)
  This bit controls the signal of the CLE pin. It is set to "0" at a reset and when starting the sequencer.

CLE	Description
0	Sets the CLE pin to "0".
1	Sets the CLE pin to "1".

- **ALE** (bit 1)
  This bit controls the signal of the ALE pin. It is set to "0" at a reset and when starting the sequencer.

ALE	Description
0	Sets the ALE pin to "0".
1	Sets the ALE pin to "1".

24.2.10 Media Data Read Control Register (MMRDCTL)

The media data read control register (MMRDCTL) suppresses that operation of reading data from NAND Flash memory which is controlled by the MMDATA register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMRDCTL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	MDR DIS	—*	—*	MDWAIT[1:0]	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8600_0024
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **MDWAIT[1:0]** (bits 0 and 1)
  These bits set the number of waits when reading the MMDATA register. A value between 1 and 4 can be set as the number of waits.

MDWAIT[1]	MDWAIT[0]	Description
0	0	Sets the number of waits to "1".
0	1	Sets the number of waits to "2".
1	0	Sets the number of waits to "3".
1	1	Sets the number of waits to "4".

- **MDRDIS** (bit 4)
  This bit suppresses that operation of reading data from a medium which is controlled by the MMDATA register.

MDRDIS	Description
0	Can directly read data from a medium by the MMDATA register.
1	Suppresses the direct data reading from a medium done by the MMDATA register (no read cycle to a medium will be generated).

24.2.11 Media Section Option Register (MOPTION)

The media section option register (MOPTION) controls each function of the media section. Do not access this register as it is solely used for debugging.
Also, do not change the values of this register while the sequencer is in operation.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MOPTION	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	PRTYMD	MDSTART
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8600_0028
Access: R/W
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **MDSTART** (bit 0)
  This bit specifies execution of reading from a medium. When "1" is written into this bit, a read cycle is executed. The read data is stored in the MMRDDATA register. This bit will always read "0" when read.

MDSTART	Description
0	Does not operate the media sequencer.
1	Executes reading from a medium.

- **PRTYMD** (bit 1)
  This bit changes the parity generation and detection method for the logical addresses of the redundancy part.

PRTYMD	Description
0	Includes a fixed value of 00010b in parity generation and detection.
1	Does not include a fixed value of 00010b in parity generation and detection.

24.2.12 Media Section Read Data Storage Register (MMRDDATA)

The media section read data storage register (MMRDDATA) stores data read from a medium by using the function of the MOPTION register. Do not access this register as it is solely used for debugging.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMRDDATA	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	MMRDDATA[7:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x8600_002C
Access: R
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **MMRDDATA[7:0]** (bits 0 to 7)
  These bits store data that has been read from a medium by writing "1" to the MDSTART bit of the MOPTION register. The stored data is held until "1" is written into the MDSSTART bit. Do not access these bits during a regular operation.

24.2.13 ECC Line Parity Register x (ECCLPx) (x = 1, 3, 5, 7)

The ECC line parity register x (ECCLPx) holds the line parity of ECCx/ECCx+1. This register stores the ECC data generated from the data written or read by the sequencer.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCLPx	LPx+1[15:0]															
At reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LPx[15:0]															
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Address: ECCLP1:0x8600_0030, ECCLP3:0x8600_0050, ECCLP5:0x8600_0070, ECCLP7:0x8600_0090
Access: R
Access size: 32 bits

[Bit Description]

- **LPx[15:0]** (bits 0 to 15)
  These bits hold the line parity of ECCx.
  This register stores the ECC data generated from the data written or read by the sequencer.

- **LPx+1[15:0]** (bits 16 to 31)
  These bits hold the line parity of ECCx+1.
  This register stores the ECC data generated from the data written or read by the sequencer.

24.2.14 ECC Column Parity Register x (ECCCPx) (x = 1, 3, 5, 7)

The ECC column parity register x (ECCLPx) holds the column parity of ECCx/ECCx+1. This register stores the ECC data generated from the data written or read by the sequencer.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCCPx	CPx+1[15:0]															
At reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPx[15:0]															
	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Address: ECCCP1:0x8600_0034, ECCCP3:0x8600_0054, ECCCP5:0x8600_0074, ECCCP7:0x8600_0094
Access: R
Access size: 32 bits

[Bit Description]

- **CPx[15:0]** (bits 0 to 15)
  These bits hold the column parity of ECCx.
  This register stores the ECC data generated from the data written or read by the sequencer.

- **CPx+1[15:0]** (bits 16 to 31)
  These bits hold the column parity of ECCx+1.
  This register stores the ECC data generated from the data written or read by the sequencer.

24.2.15 ECC Error Pointer Register x (ECCERRx) (x = 1, 3, 5, 7)

The ECC error pointer register (ECCERRx) holds the pointer of ECCx/ECCx+1 error detection result.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCERRx	—*	—*	—*	—*	—*	BITPx+1[7:0]			BYTEPx+1[7:0]							
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	BITPx[7:0]			BYTEPx[7:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: ECCERR1:0x8600_0038, ECCERR3:0x8600_0058, ECCERR5:0x8600_0078, ECCERR7:0x8600_0098
Access: R
Access size: 32 bits

[Note]
*: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.

[Bit Description]

- **BYTEPx[7:0]** (bits 0 to 7)
  Byte pointer of ECCx. These bits indicate the byte address of an error detected.

- **BITPx[10:8]** (bits 8 to 10)
  Bit pointer of ECCx. These bits indicate the bit address of an error detected.

- **BYTEPx+1[7:0]** (bits 16 to 23)
  Byte pointer of ECCx. These bits indicate the byte address of an error detected.

- **BITPx+1[10:8]** (bits 14 to 26)
  Bit pointer of ECCx. These bits indicate the bit address of an error detected.

24.2.16 Redundancy Part Reserve Data Register x (HREVx) (x = 1, 3, 5, 7)

When Smart Media is used, only the HREV1 register is enabled. In this case, when the sequencer reads the redundancy part, the values of the bytes 512, 513, 514 and 515 of the redundancy part are written into this register. In addition, when the sequencer writes the redundancy part, this register sets the bytes 512, 513, 514 and 515 of the redundancy part are set in this register. In the Smart Media format, this is equivalent to reserved data (the function select information area in the redundancy part of the CIS area).

When a medium of 2048 bytes/sector is used, the bytes 2048, 2049, 2050 and 2051 are written into the HREV1, the bytes 2064, 2065, 2066 and 2067 are written into the HREV3, the bytes 2080, 2081, 2082 and 2083 are written into the HREV5, and the bytes 2096, 2097, 2098 and 2099 are written into the HREV7. The data in these registers is set as reserved data.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HREVx	REVx+1D4[7:0]								REVx+1D3[7:0]							
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REVxD2[7:0]								REVxD1[7:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: HREV1:0x8600_0040, HREV3:0x8600_0060, HREV5:0x8600_0080, HREV7:0x8600_00A0
Access: R/W
Access size: 32 bits

[Bit Description]

- **REVxD1[7:0]** (bits 0 to 7)
  These bits make up a data register for redundancy part reserve 1 of packet x (normally FFh).

- **REVxD2[7:0]** (bits 8 to 15)
  These bits make up a data register for redundancy part reserve 2 of packet x (normally FFh).

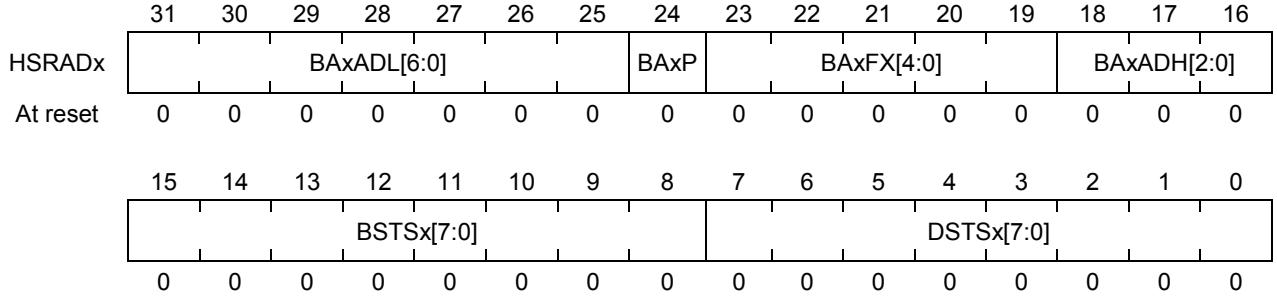
- **REVx+1D3[7:0]** (bits 16 to 23)
  These bits make up a data register for redundancy part reserve 3 of packet x+1 (normally FFh).

- **REVx+1D4[7:0]** (bits 24 to 31)
  These bits make up a data register for redundancy part reserve 4 of packet x+1 (normally FFh).

24.2.17 Redundancy Part Data/Block Status & Block Address Register x (HSTADx) (x = 1, 3, 5, 7)

When Smart Media is used, only the HSTAD1 register is enabled. In this case, the values of the bytes 516 and 517 of the redundancy part are written into the data status/block status (bits 0 to 15) of the redundancy part when the sequencer reads the redundancy part. Also, when the sequencer writes the redundancy part, the bytes 516 and 517 byte of the redundancy part are set in this register. In the Smart Media format, they are equivalent to the block status and data status (in the redundancy part of the CIS area, the data will be set as an invalid data flag (byte 516) and a fixed value of FFh (byte 517)).
On the other hand, the values of the bytes 518 and 519 of the redundancy part are written into the block address (bits 16 to 31) when the sequencer reads the redundancy part. Also, when the sequencer writes the redundancy part, the bytes 518 and 519 of the redundancy part are set in this register. In the Smart Media format, they are equivalent to block address 1. In write, a parity generated from the block address (BA1ADH/BA1ADL) is written into the BA1P bit. In read, the PARITY bit of the MSSTS register is set to "1" if the parity data generated from the block address (BA1ADH/BA1ADL) is different from the value of the BA1P bit. (In the redundancy part of the CIA area, the data will be a fixed value of 0000h.)

When a medium of 2048 bytes/sector is used, 1) for the data status/block status (bits 0 to 15) of the redundancy part, the values of the bytes 2052 and 2053 of the redundancy part apply to the HSTAD1 register, the values of the bytes 2068 and 2069 of the redundancy part to the HSTAD3 register, the values of the bytes 2084 and 2085 of the redundancy part to the HSTAD5 register, and the values of the bytes 2100 and 2101 of the redundancy part to the HSTAD7 register.
Likewise, 2) for the block address (bits 16 to 31), the values of the bytes 2054 and 2055 of the redundancy part apply to the HSTAD3 register, the values of the bytes 2070 and 2071 of the redundancy part to the HSTAD3 register, the values of the bytes 2086 and 2087 of the redundancy part to the HSTAD5 register, and the values of the bytes 2102 and 2103 of the redundancy part to the HSTAD7 register.



Address: HSTAD1:0x8600_0044, HSTAD3:0x8600_0064, HSTAD5:0x8600_0084, HSTAD7:0x8600_00A4
Access: R/W
Access size: 32 bits

[Bit Description]

- **DSTSx[7:0]** (bits 0 to 7)
  These bits indicate that the data of Data Area-x and Data Area-x+1 on the page read is not normal. Normally they are FFh. 00h is set if abnormal data has been written. If four or more bits are "0," it is assumed to be the same as 00h.
  As for the CIS area, these bits indicate the validity of CIS data read. If the data is valid, FFh is set; if invalid, 00h is set. If four or more bits are "0," it is assumed to be the same as 00h.

- **BSTSx[7:0]** (bits 8 to 15)
  These bits indicate whether the blocks read are normal or faulty for Data Area x and x+1. Normally they are FFh (normal blocks). In the case of a faulty block, 00h (subsequent faulty block) is set. If two more bits are "0," it is judged as a faulty block. All the data in these data areas has the same value within the same block. As for the CIS area, it is fixed to FFh.

- **BAxADH[2:0]** (bits 16 to 18)
  These bits set the address on the "High" side of block address x in the Smart Media format. They are equivalent to BlockAddress[9:7]. They are fixed to "0" in the redundancy part of the CIA area.

- **BAxFX[4:0]** (bits 19 to 23)
  These bits set block address x in the Smart Media format. Set a fixed value of "00010b" in these bits. They are fixed to "0" in the redundancy part of the CIA area.

- **BAxP** (bit 24)
  This bit is a parity bit (even) of block address x in the Smart Media format. In write, a parity generated from the block address (BAxADH/BAxADL) is written into this bit. In read, if the parity bit generated from the block address (BAxADH/BAxADL) is different from the value of the BAxP bit, the PARITY bit of the MSSTS register is set to "1".
  It is fixed to "0" in the redundancy part of the CIA area.

- **BAxADL[6:0]** (bits 25 to 31)
  These bits set the address on the "Low" side of block address x in the Smart Media format. They are equivalent to BlockAddress[6:0]. They are fixed to "0" in the redundancy part of the CIA area.

24.2.18 Redundancy Part ECCx+1-High Register & Redundancy Part ECCx+1-Low/Block Address x+1 Register (HECCx+1) (x = 1, 3, 5, 7)

When Smart Media is used, only the HECC2 register is enabled. In this case, the values of the bytes 520 and 521 of the redundancy part are written into the ECC2-High register section (bits 0 to 15) of the redundancy part when the sequencer reads the redundancy part. In addition, when the sequencer writes the redundancy part with ECC disabled, the bytes 520 and 521of the redundancy part are set in this register (if ECC is enabled, generated ECC data will be written). They are equivalent to upper two bytes of ECC code of Data Area-2 in the Smart Media format. If the sequencer has executed a data write operation to Flash memory with ECC enabled, generated ECC data is stored in the ECC Line Parity Register 2 and ECC Column Parity Register 2, and thus this register is not affected.
On the other hand, the values of the bytes 522 and 523 of the redundancy part are written into the ECC2-Low/block address register section (bits 16 to 31) of the redundancy part when the sequencer reads the redundancy part. In addition, when the sequencer writes the redundancy part, the bytes 522 and 523 of the redundancy part are set in this register. They are equivalent to lower one byte of ECC code of Data Area-2 and upper one byte of block address 2 in the Smart Media format. When the sequencer writes the redundancy part in the Smart Media format, the value of block address 2 is the same as that of block address 1.

When a medium of 2048 bytes/sector is used, 1) for the ECC2-High register section (bits 0 to 15) of the redundancy part, the values of the bytes 2056 and 2057 of the redundancy part apply to the HECC2 register, the values of the bytes 2072 and 2073 of the redundancy part to the HECC4 register, the values of the bytes 2088 and 2089 of the redundancy part to the HECC6 register, and the values of the bytes 2104 and 2105 of the redundancy part to the HECC2 register.
Likewise, 2) for the ECC2-Low/block address register section (bits 16 to 31), the values of the bytes 2058 and 2059 of the redundancy part apply to the HECC2 register, the values of the bytes 2074 and 2075 of the redundancy part to the HECC4 register, the values of the bytes 2090 and 2091 of the redundancy part to the HECC6 register, and the values of the bytes 2106 and 2107 of the redundancy part to the HECC2 register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HECCx+1	BAx+1FX[4:0]					BAx+1ADH[2:0]			ECCx+1L[7:0]							
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCx+1M[7:0]								ECCx+1H[7:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: HECC2:0x8600_0048, HECC4:0x8600_0068, HECC6:0x8600_0088, HECC8:0x8600_00A8
Access: R/W
Access size: 32 bits

[Bit Description]

- **ECCx+1H[7:0]** (bits 0 to 7)
  In the case of the Smart Media format, these bits make up upper one byte of ECC code of Data Area-2.
  In the case of 2048 bytes/sector access, these bits make up upper one byte of ECC code of Data Area-x+1.

- **ECCx+1M[7:0]** (bits 8 to 15)
  In the case of the Smart Media format, these bits make up middle one byte of ECC code of Data Area-2.
  In the case of 2048 bytes/sector access, these bits make up middle one byte of ECC code of Data Area-x+1.

- **ECCx+1L[7:0]** (bits 16 to 23)
  In the case of the Smart Media format, these bits make up lower one byte of ECC code of Data Area-2.
  In the case of 2048 bytes/sector access, these bits make up lower one byte of ECC code of Data Area-x+1.

- **BAx+1ADH[2:0]** (bits 24 to 26)
  These bits set the address on the "High" side of block address 2 in the Smart Media format. They are equivalent to BlockAddress[9:7].
  In the case of 2048 bytes/sector access, these bits set the address on the "High" side of block address x+1. They are equivalent to BlockAddress[9:7].

- **BAx+1FX[4:0]** (bits 27 to 31)
  These bits set block address 2 in the Smart Media format. Set a fixed value of "00010b" in these bits.
  In the case of 2048 bytes/sector access, these bits set block address x+1.

24.2.19 Redundancy Part ECCx-High/Block Address x+1 Register & Redundancy Part ECCx-Low Register (HECCx)

When Smart Media is used, only the HECC1 register is enabled. In this case, the values of the bytes 524 and 525 of the redundancy part are written into the ECC1-High/block address 2 section (bits 0 to 15) of the redundancy part when the sequencer reads the redundancy part. In addition, when the sequencer writes the redundancy part, the bytes 524 and 525 of the redundancy part are set in this register. They are equivalent to lower one byte of block address 1 and the most significant one byte of ECC code of Data Area-1 in the Smart Media format. When the sequencer writes the redundancy part in the Smart Media format, the value of block address 2 is the same as that of block address 1.
On the other hand, the values of the bytes 526 and 527 of the redundancy part are written into the ECC1-Low register section (bits 16 to 31) of the redundancy part when the sequencer reads the redundancy part. In addition, when the sequencer writes the redundancy part, the bytes 526 and 527 of the redundancy part are set in this register. They are equivalent to lower two bytes of ECC code of Data Area-1 in the Smart Media format.

When a medium of 2048 bytes/sector is used, 1) for the ECC1-High/block address 2 section (bits 0 to 15) of the redundancy part, the values of the bytes 2060 and 2061 of the redundancy part apply to the HECC1 register, the values of the bytes 2076 and 2077 of the redundancy part to the HECC3 register, the values of the bytes 2092 and 2093 of the redundancy part to the HECC5 register, and the values of the bytes 2108 and 2109 of the redundancy part to the HECC7 register.
Likewise, 2) for the ECC1-Low register section (bits 16 to 31), the values of the bytes 2062 and 2063 of the redundancy part apply to the HECC1 register, the values of the bytes 2078 and 2079 of the redundancy part to the HECC3 register, the values of the bytes 2094 and 2095 of the redundancy part to the HECC5 register, and the values of the bytes 2110 and 2111 of the redundancy part to the HECC7 register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HECC2	ECC1L[7:0]								ECC1M[7:0]							
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECC1H[7:0]								BA2ADL[6:0]							BA2P
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: HECC1:0x8600_004C, HECC3:0x8600_006C, HECC5:0x8600_008C, HECC7:0x8600_00AC
Access: R/W
Access size: 32 bits

[Bit Description]

- **BAx+1P** (bit 0)
  This bit is a parity bit (even) of block address 2 in the Smart Media format. In write, a parity generated from the block address (BA2ADH/BA2ADL) is written into this bit. In read, if the parity bit generated from the block address (BA2ADH/BA2ADL) is different from the value of the BA2P bit, the parity bit of the MSSTS register is set to "1".
  It is fixed to "0" in the redundancy part of the CIS area.
  In the case of 2048 bytes/sector access, this bit is a parity bit (even) of block address x+1. In write, a parity generated from the block address (Bax+1ADH/Bax+1ADL) is written into this bit. In read, if the parity bit generated from the block address (Bax+1ADH/Bax+1ADL) is different from the value of the Bax+1P bit, the parity bit of the MSSTS register is set to "1".
  It is fixed to "0" in the redundancy part of the CIS area.

- **BAx+1ADL[6:0]** (bits 1 to 7)
  These bits set the address on the "Low" side of block address 2 in the Smart Media format. They are equivalent to BlockAddress[6:0]. They are fixed to "0" in the redundancy part of the CIS area.
  In the case of 2048 bytes/sector access, these bits set the address on the "Low"side of block address x+1. They are equivalent to BlockAddress[6:0].

- **ECCxH[7:0]** (bits 8 to 15)
  In the case of the Smart Media format, these bits make up upper one byte of ECC code of Data Area-1.
  In the case of 2048 bytes/sector access, these bits make up upper one byte of ECC code of Data Area-x.

- **ECCxM[7:0]** (bits 16 to 23)
  In the case of the Smart Media format, these bits make up middle one byte of ECC code of Data Area-1.
  In the case of 2048 bytes/sector access, these bits make up middle one byte of ECC code of Data Area-x.

- **ECCxL[7:0]** (bits 24 to 31)
  In the case of the Smart Media format, these bits make up lower one byte of ECC code of Data Area-1.
  In the case of 2048 bytes/sector access, these bits make up lower one byte of ECC code of Data Area-x.

24.2.20 Media Data Register (MMDATA)

The media data register (MMDATA) writes data into NAND Flash memory or reads data from NAND Flash memory.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMDATA	—*	—*	—*	—	—*	—*	—*	—*	—*	—*	—*	—	—	—*	—*	—*
At reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	MMDATA[7:0]							
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Address: 0x8600_0100 to 0x8600_01FC
Access: R/W
Access size: 32 bits

[Note]
*—: Indicates a reserved bit for future expansion. In this LSI, "0" is read when any of reserved bits is read, and any write operation to reserved bits is ignored.
CLE and ALE are cleared after the controller recognizes a write or read signal from the CPU. Therefore, because CLE/ALE cannot be cleared in time for a read signal to NAND Flash memory during a read operation, it is necessary to set CLE/ALE to "0" output using the MMSEL register in advance before reading the MMDATA register.
*: The access size of only 32 bits is allowed. Half-word access and byte access are prohibited.

[Bit Description]

- **MMDATA[7:0]** (bits 0 to 7)
  These bits write data into NAND Flash memory during a write operation.
  These bits read data from NAND Flash memory during a read operation. Both CLE and ALE are set to "0" when starting an operation. However, CLE and ALE are cleared after the controller recognizes a write or read signal from the CPU. Therefore, because CLE/ALE cannot be cleared in time for a read signal to NAND Flash memory during a read operation, it is necessary to set CLE/ALE to "0" output using the MMSEL register in advance before reading the MMDATA register. When the MMRDCTL register is used to suppress a data read into NAND Flash memory, "00" is read and a read operation to NAND Flash memory is not performed.

24.2.21 NAND Flash Buffer Memory

This is 4-Kbyte buffer memory for NAND Flash memory access.
When an automatic transmission mode is used by the NAND Flash Controller, the contents that are stored in this memory in advance are written to the NAND Flash memory. When an automatic reception mode is used, the data that has been read from the NAND Flash memory is stored in this memory.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
At reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
At reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Address: 0x8800_0000 to 0x8800_0FFC
Access: R/W
Access size: 8/16/32 bits

24.3 Operational Description

24.3.1 Buffer Memory

This LSI includes SRAMs as Buffer memory for the NAND Flash controller. The SRAMs, which consist of dual port memory, can be read from/written to from both the APB interface and NAND Flash controller.

- Built-in 4096-byte dual port memory
- Can be accessed from the APB
- Can be accessed from the NAND Flash controller

24.3.1.1 Configuration

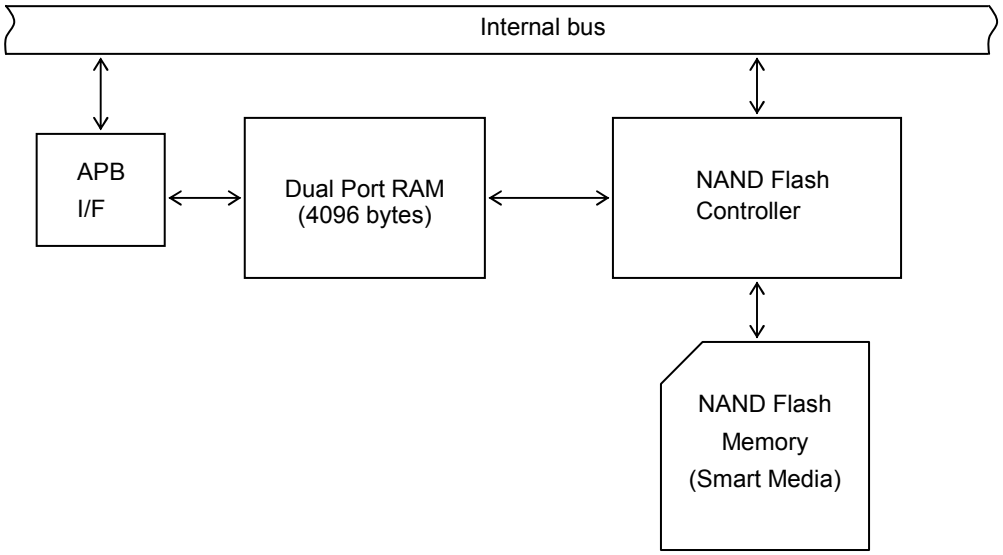


Figure 24-2 Block Configuration

24.3.1.2 Access from APB

A byte of data, a half word of data, or a word of data can be usually read/written from the APB. Addresses are assigned to a range of 0x8800_0000 to 0x8800_0FFC.

24.3.1.3 Access from the NAND Flash Controller

The contents stored in the Buffer memory in advance are written in the NAND Flash memory by the NAND Flash controller when auto-transmit mode is used. Data read from the NAND Flash memory is stored into the Buffer memory by the NAND Flash controller when auto-receive mode is used.

24.3.2 Wait Function

NAND flash memory control signal of the sequencer can be switched by bits [3:0] in the MSWAIT register. The use of this register is separated according to the circuit's operating frequency and the NAND flash memory specifications.

24.3.2.1 Wait Function in Sequencer Write

Regarding the control signal timing when writing into NAND Flash memory, wait cycles can be inserted as shown below based on the WAITW[1:0] setting of the MSWAIT register.

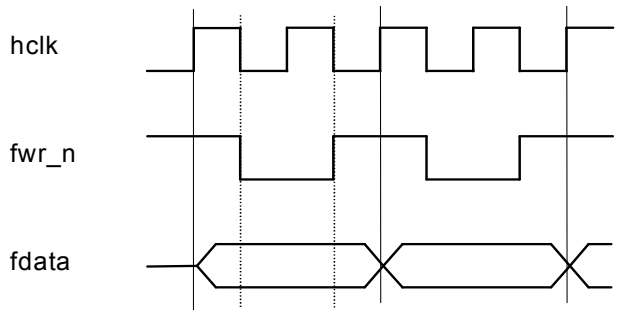


Figure 24-3 No Write Wait (MSWAIT Register: WAITW[1:0]= 00b)

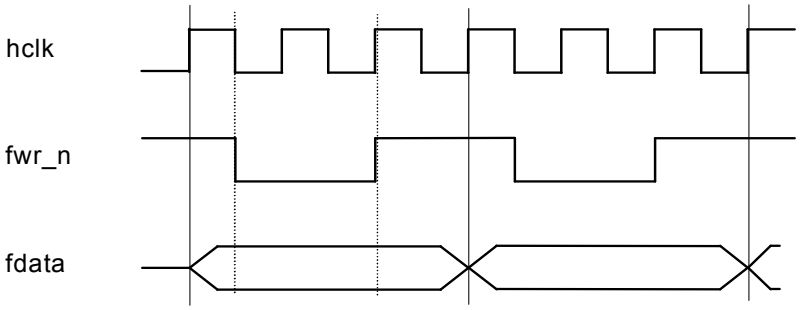


Figure 24-4 Write Wait 1 (MSWAIT Register: WAITW[1:0]= 01b)

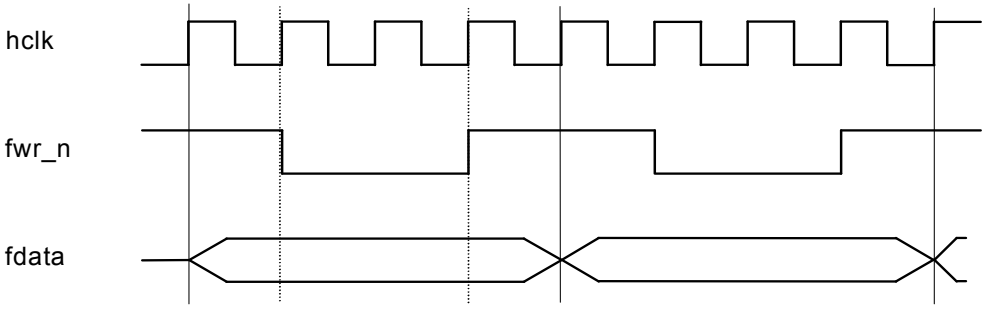


Figure 24-5 Write Wait 2 (MSWAIT Register: WAITW[1:0]= 1Xb)

24.3.2.2 Wait Function in Sequencer Read

Regarding the control signal timing when reading from NAND Flash memory, wait cycles can be inserted as shown below based on the WAITW[1:0] setting of the MSWAIT register. In reading, data is loaded on the rising edge of frd_n.

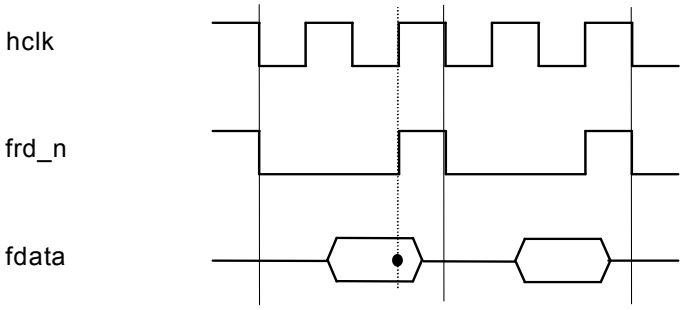


Figure 24-6 No Read Wait (MSWAIT Register: WAITR[1:0]= 00b)

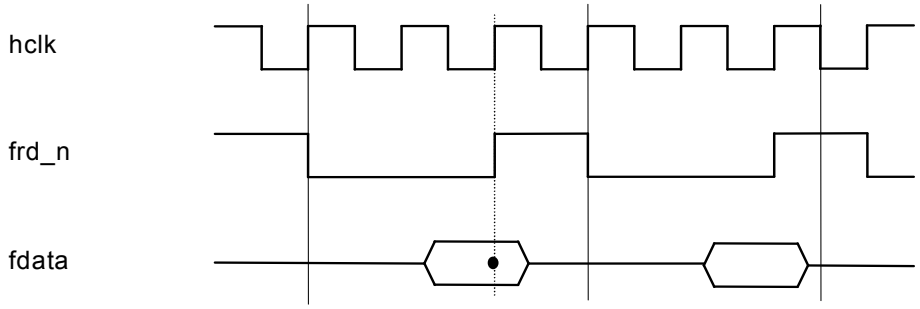


Figure 24-7 Read Wait 1 (MSWAIT Register: WAITR[1:0]= 01b)

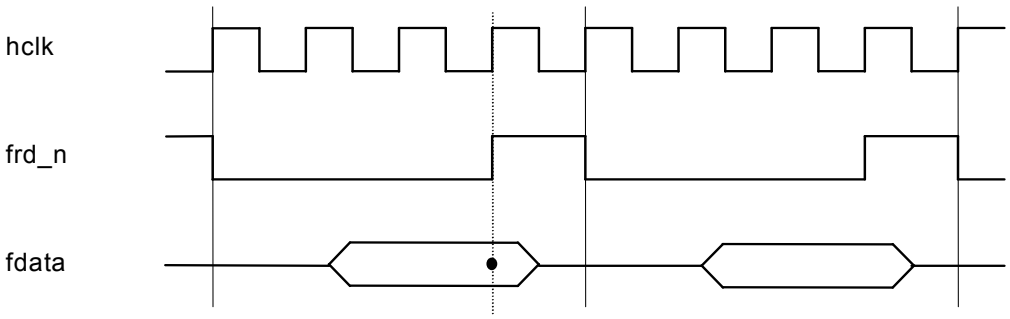


Figure 24-8 Read Wait 2 (MSWAIT Register: WAITR[1:0]= 1Xb)

24.3.3 Access to the MMCMD/MMADR/MMDATA Registers

Write/read to the NAND flash memory is performed if the MMCMD, MMADR, or MMDATA register is written to or read from.
The timing of access to these registers is described below.

24.3.3.1 Access to the MMCMD Register

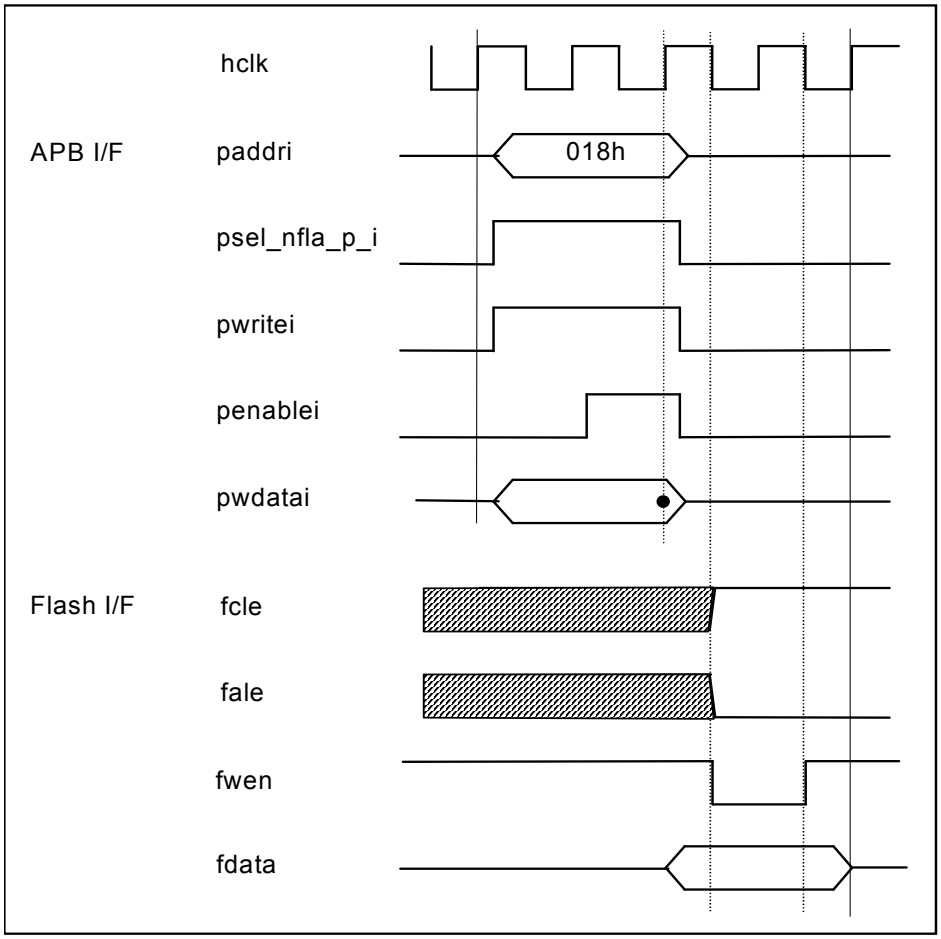


Figure 24-9 Write to the MMCMD Register (No Write Wait)

When the CPU writes data to MMCMD register, the CPU sets CLE to "1" and ALE to "0", and then writes data to the flash memory as a command. Holds CLE to "1" at the end of the operation.
MMCMD register is write only. If read, it will read 00h.
If the WAITW bit in the MMWAIT register has been set to "1", the wait function will affect the MMCMD register also.

24.3.3.2 Access to the MMADR Register

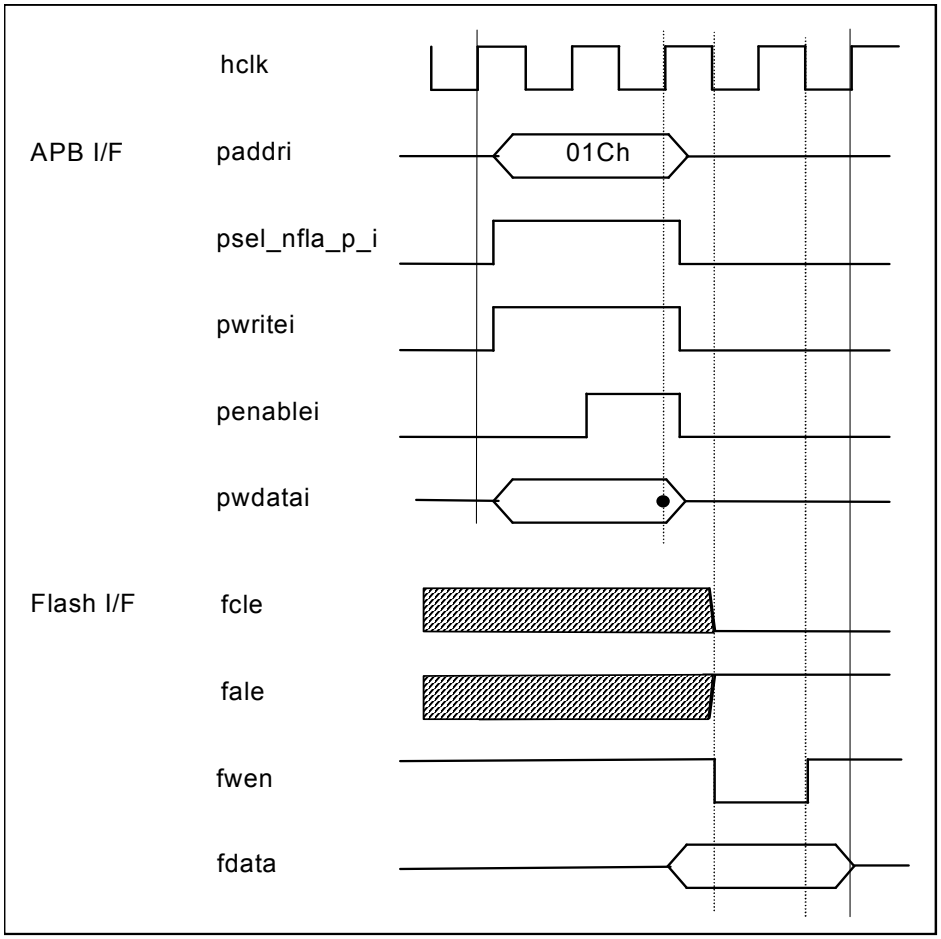


Figure 24-10 Write to the MMADR Register (No Write Wait)

When the CPU writes data to the MMADR register, the CPU sets CLE to "0" and ALE to "1", and then writes data to the NAND flash memory as an address. Holds ALE to "1" at the end of the operation.
The MMADR register is write only. If read, it will read 00h.
If the WAITW bit in the MMWAIT register has been set to "1", the wait function will affect the MMADR register also.

24.3.3.3 Access to the MMDATA Register

- For Write

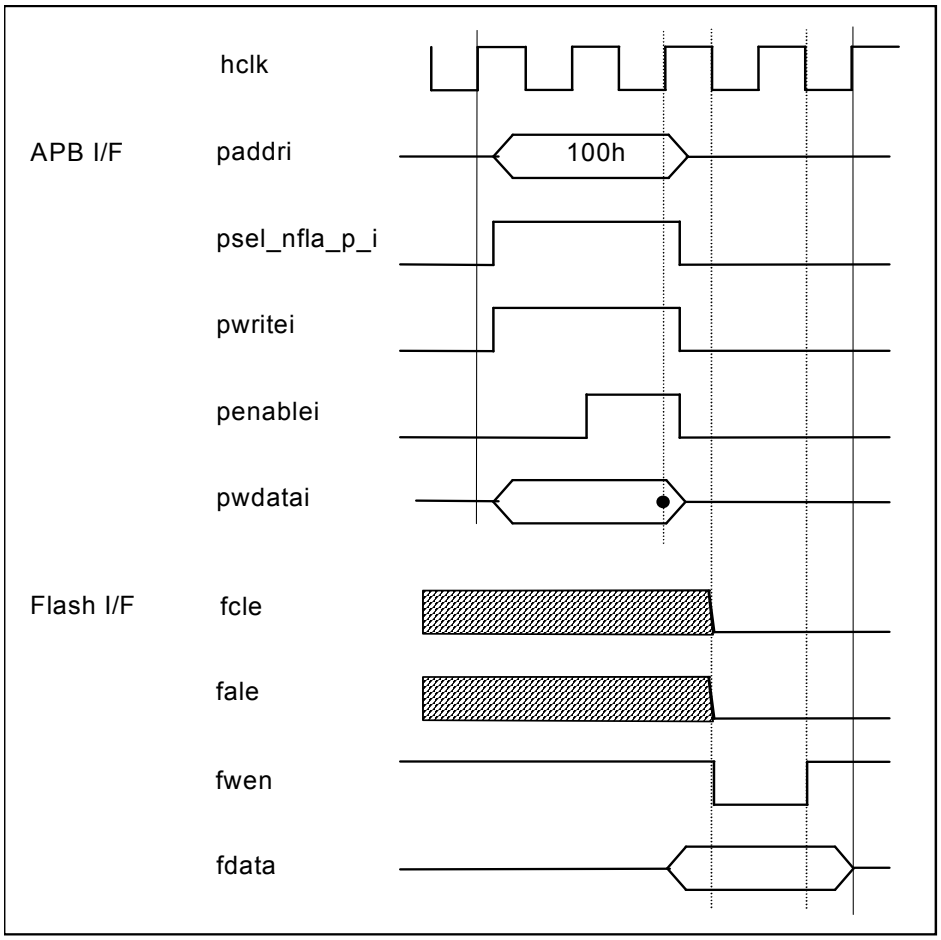


Figure 24-11 Write to the MMDATA Register (No Write Wait)

When the CPU writes data to the MMDATA register, the CPU sets CLE to "0" and ALE to "0", and then writes data to the NAND flash memory.
If the WAITW bit in the MMWAIT register has been set to "1", the wait function will affect the MMDATA register also.

- For Read
  - When there is no external wait

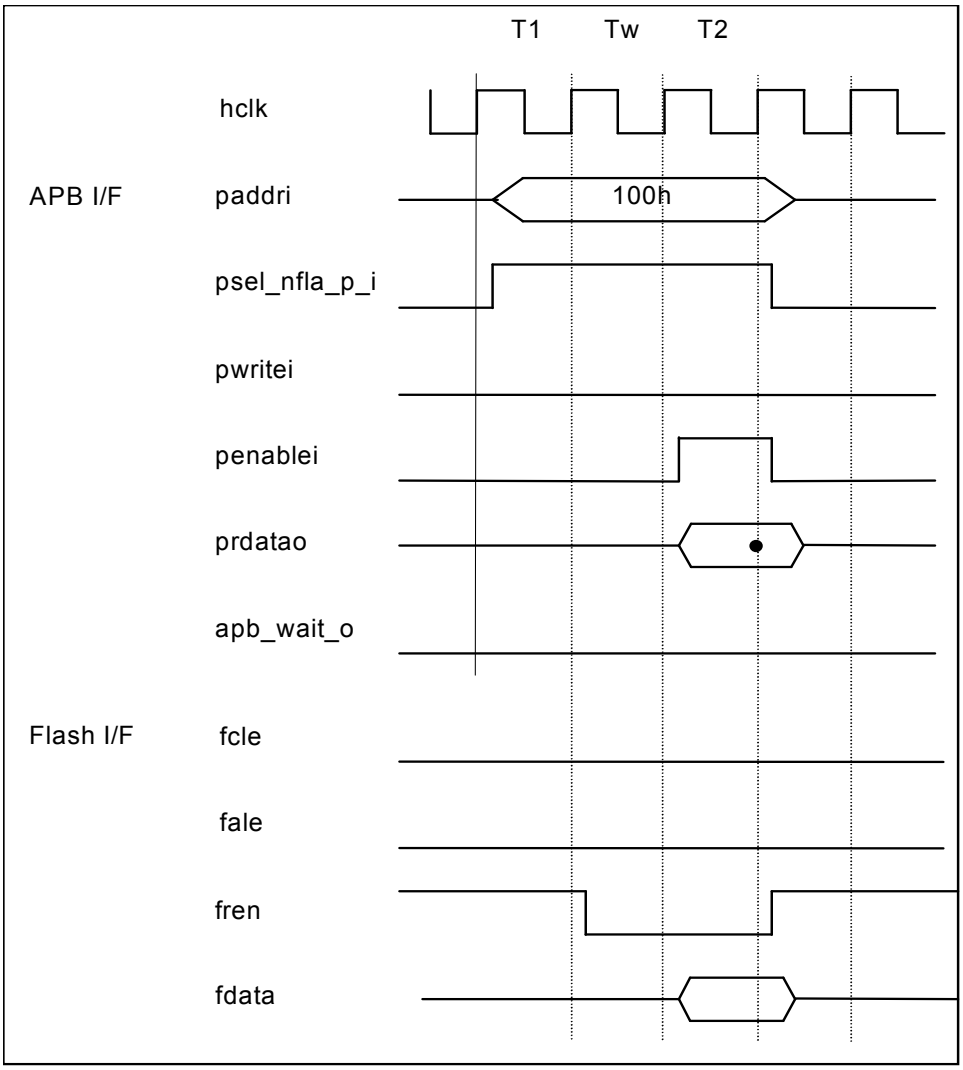


Figure 24-12 Read from the MMDATA Register (No External Wait)

When the CPU reads data in MMDATA register, the CPU sets CLE to "0" and ALE to "0", and then reads data from the NAND flash memory.
Since CLE/ALE will be cleared after the controller has recognized the read signal of CPU, clearing CLE/ALE is not in time for the read signal to the flash memory. Therefore, it is necessary to hold CLE and ALE low using the MMSEL register prior to reading the MMDATA register, according to the timing specification of the flash memory.
1 cycle wait is automatically added for the APB I/F when reading the MMDATA register. The cycle shown by Tw in the above figure is the wait added in APB bridge.

- When there is an external wait

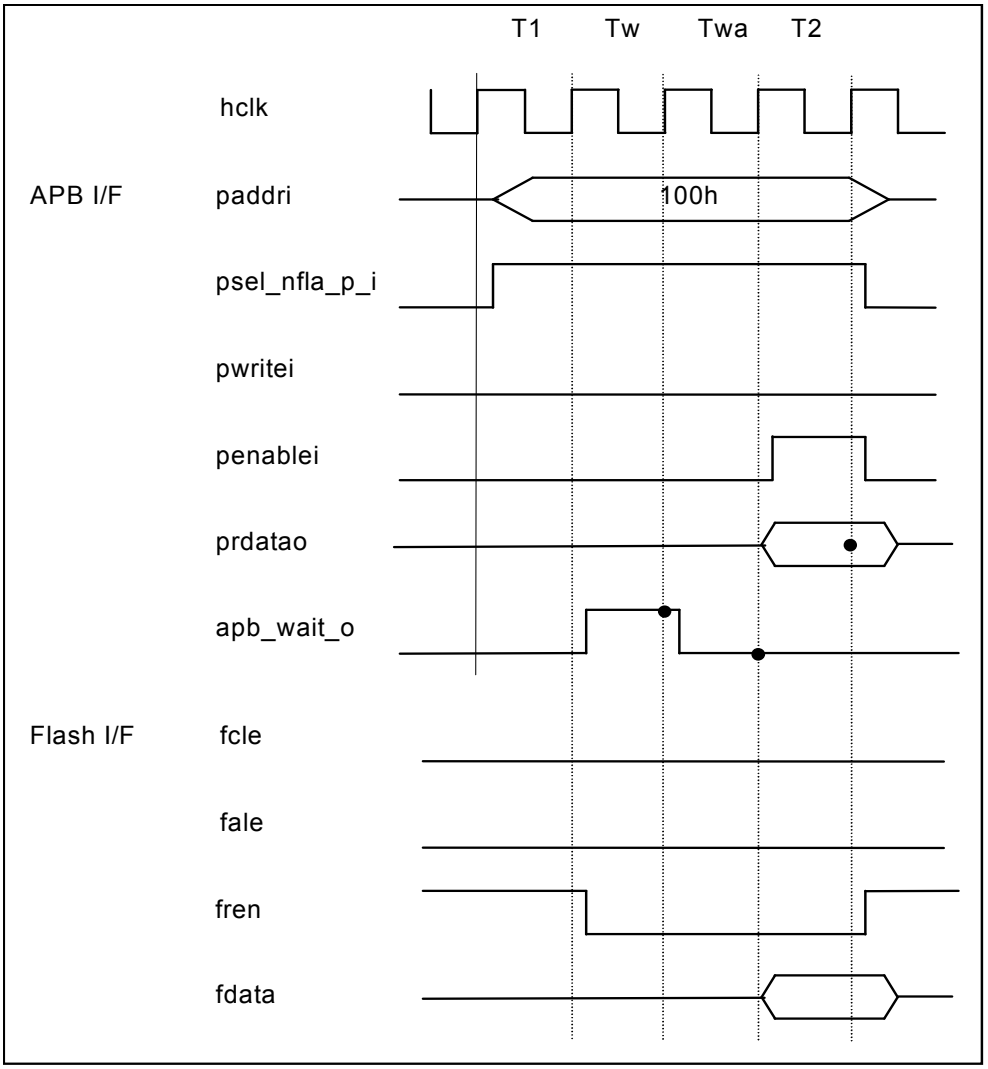
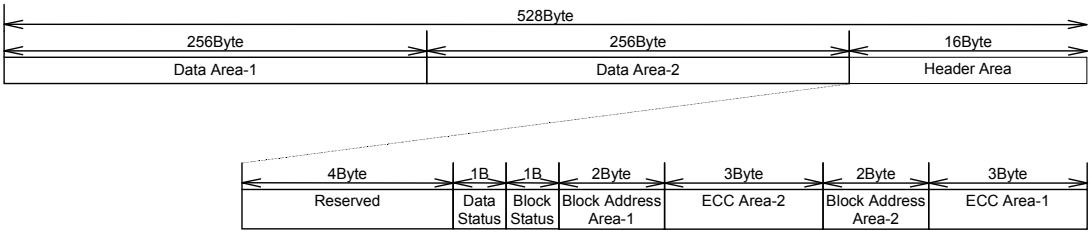


Figure 24-13 Read from the MMDATA Register (with External Wait)

It is possible to add wait to the read cycle of MMDATA register by MDWAIT bit in MMRDCTL register. The above figure is a sequence of case in which the MDWAIT bit is set to 01b and 1 cycle wait has been added.
As mentioned earlier, 1 cycle wait in the APB bridge is automatically added (shown by Tw in the above figure) for read of MMDATA register. Wait of 1 cycle will be further added (shown by Twa in the above figure) if MDWAIT bit in MMRDCTL is set to 001b.

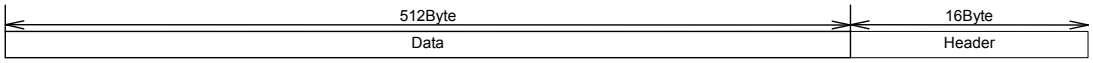
The apb_wait_o signal will be asserted for the cycle count set in MDWAIT bit and for the APB bridge from the media control section. The APB bridge samples the apb_wait_o signal at the rising edge of the clock, and adds the wait cycle.

24.3.4 Smart Media™ Physical Format (supported by 512-byte automatic access mode)



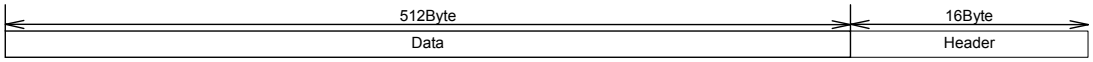
Various data transfer modes and the purpose of use are shown below.

- Data transfer mode 1 (MSCTRL register: HEAD = 00b, DLEN = 00b)
  Normal data transfer in Smart Media (read/write)



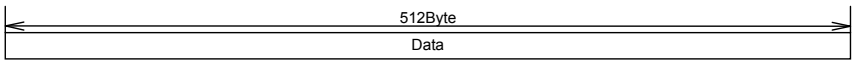
  - Data transfer memory: 512-byte transfer
  - Redundancy part processing: Performs read/write in accordance with the Smart Media format.

- Data transfer mode 2 (MSCTRL register: HEAD = 01b, DLEN = 00b)
  Normal data transfer (read/write)



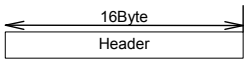
  - Data transfer memory: 512-byte transfer
  - Redundancy part processing: Performs read/write without recognizing the format.

- Data transfer mode 3 (MSCTRL register: HEAD = 1Xb, DLEN = 00b)
  Data read for debugging



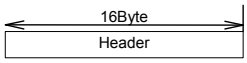
  - Data transfer memory: 512-byte transfer
  - Redundancy part processing: None

- Data transfer mode 4 (MSCTRL register: HEAD = 00b, DLEN = 11b)
  Normal redundancy part read in Smart Media



  - Data transfer memory: 0-byte transfer
  - Redundancy part processing: Performs read in accordance with the Smart Media format.

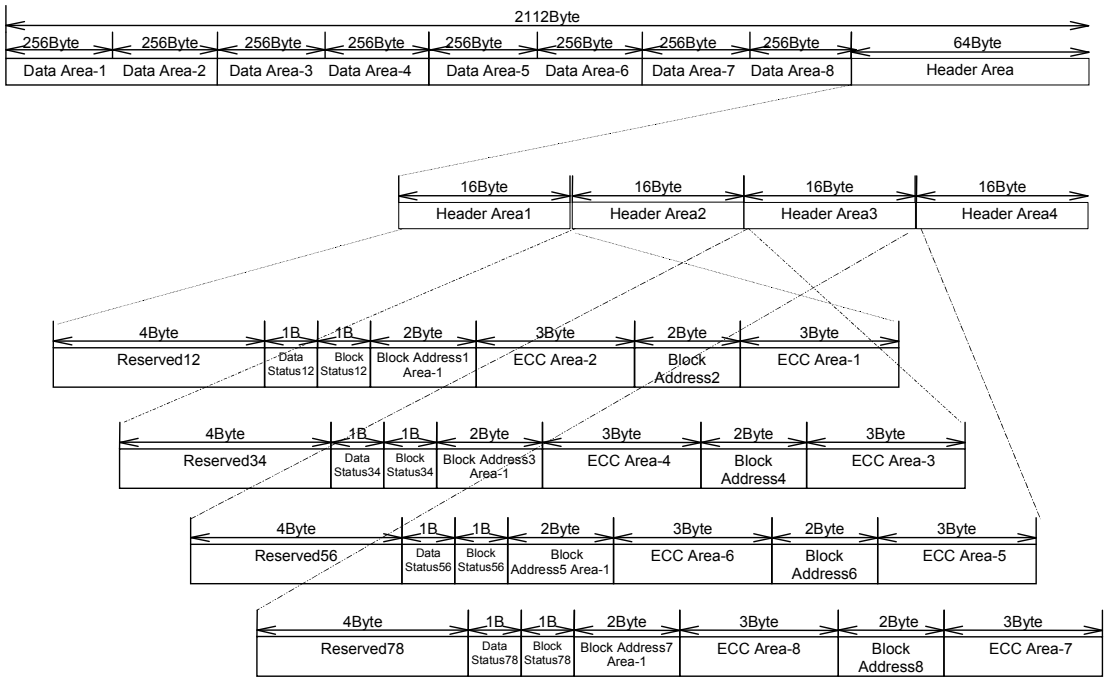
- Data transfer mode 5 (MSCTRL register: HEAD = 01b, DLEN = 11b)
  Normal redundancy part read



  - Data transfer memory: 0-byte transfer
  - Redundancy part processing: Performs read without recognizing the format.

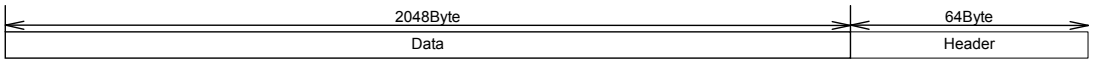
24.3.5 Physical Format for 2048 Bytes/Sector Media

2048 bytes/sector media are not Smart Media™ standard compliant. This LSI defines and operates the format as described below for the redundancy part when 2048 bytes/sector media are used in a mode with a redundancy part.



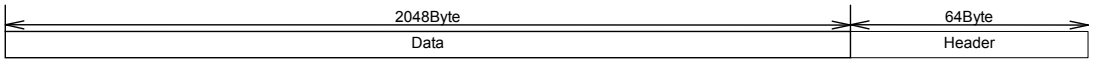
The following describes the various data transfer modes and their purposes.

- Data transfer mode 1 (MSCTRL register: HEAD = 00b, DLEN = 01b)
  Normal redundancy part read in Smart Media



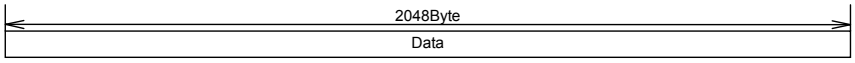
  - Data transfer memory: 2048-byte transfer
  - Redundancy part processing: Performs read in accordance with the format described above.

- Data transfer mode 2 (MSCTRL register: HEAD = 01b, DLEN = 01b)
  Normal data transfer (read/write)



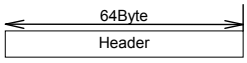
  - Data transfer memory: 2048-byte transfer
  - Redundancy part processing: Performs read/write without recognizing the format.

- Data transfer mode 3 (MSCTRL register: HEAD = 1Xb, DLEN = 01b)
  Data read for debugging



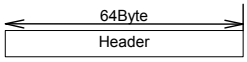
  - Data transfer memory: 2048-byte transfer
  - Redundancy part processing: None

- Data transfer mode 4 (MSCTRL register: HEAD = 00b, DLEN = 10b)
  Normal redundancy part read in the Smart Media format



  - Data transfer memory: 0-byte transfer
  - Redundancy part processing: Performs read in accordance with the format described above.

- Data transfer mode 5 (MSCTRL register: HEAD = 01b, DLEN = 10b)
  Normal redundancy part read



  - Data transfer memory: 0-byte transfer
  - Redundancy part processing: Performs read/write without recognizing the format.

24.3.6 On ECC Function

The ECC-related registers are enabled only if they are accessed with the ECC function enabled. The values of the ECC-related registers become invalid after they are accessed with ECC disabled.

Chapter 25

ATA IP (IDE Host Controller)

Chapter 25 ATA IP (IDE Host Controller)

25.1 Overview

This IP controller is used for the IDE/ATA interface. The IP controller can transfer data at high speed between the ATA/ATAPI device and the AHB interface in multiword DMA or Ultra DMA mode.
This controller, in accordance with the ATA-6 standard, can provide an interface of up to Ultra DMA mode 4 (66 Mbytes/s).

Features

- Supports PIO mode, multiword DMA mode, and Ultra DMA66 mode.
- PIO mode: Supports up to mode 4.
- Multiword DMA mode: Supports up to mode 2.
- Ultra DMA mode: Supports up to mode 4.
- Scatter-gather DMA function
- Set to the primary channel (secondary channel cannot be used)
- Only device 0 (master) can be used (Device 1 (slave) cannot be used).
- AHB master and slave interface
- Built-in data transfer FIFO: 8 x 32-bit (32 bytes)

25.1.1 Configuration

Figure 25-1 shows the configuration of the IDE host controller.

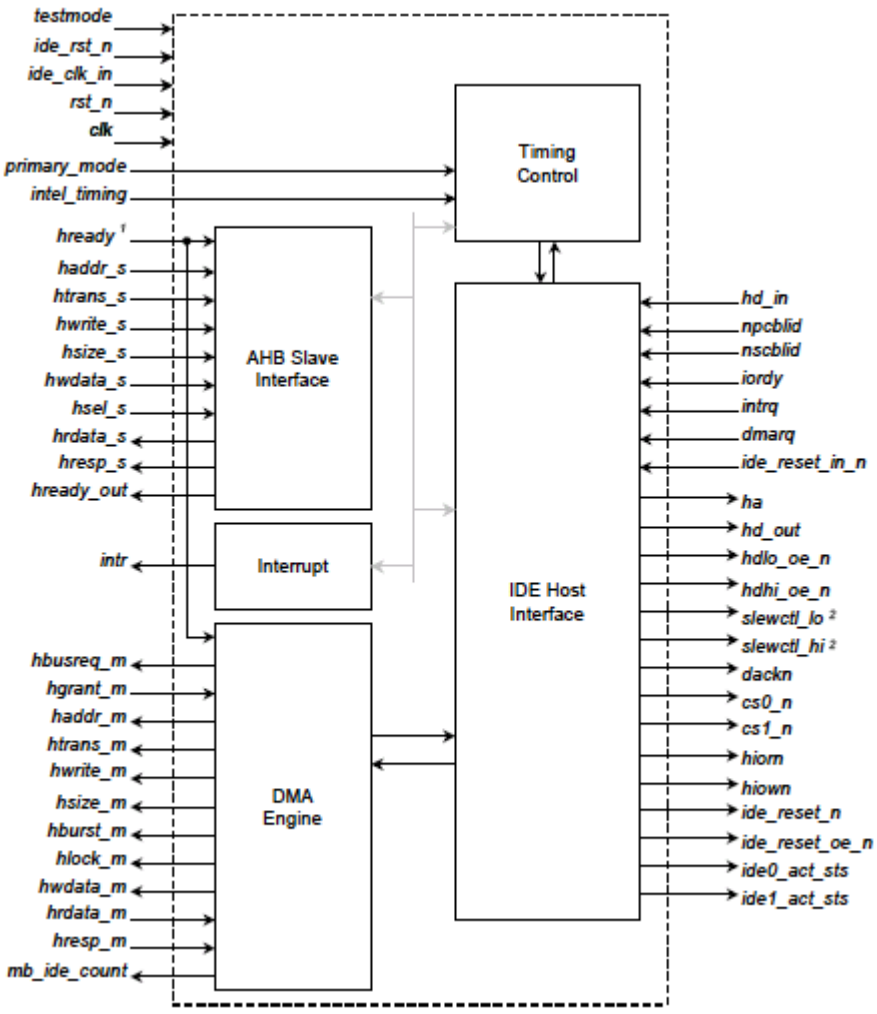


Figure 25-1 IDE Controller Block Diagram

(Note)
1: Used by both the AHB master interface and the AHB slave interface.

Outline of Each Module

(1) AHB slave interface
When operating as an AHB slave interface, this controller decodes the access by the AHB master interface as the access to the internal registers or the access to the registers of the primary device in the IDE interface. (The primary channel is mapped to addresses 1F0h-1F7h and 3F6h.)

(2) Timing control
The controller determines the timing of each signal (address, data, and control) of the IDE interface. The timing is set by the configuration register according to the data transfer mode.

(3) DMA engine
In Multiword DMA and UltraDMA transfer modes, this controller is used as the AHB master interface and performs DMA transfer between the internal FIFO and the external memory. Equipped with the scatter-gather DMA function, this controller can perform DMA transfer automatically to the areas that are scattered in the memory, by referencing the PRD (Physical Region Descriptor) that is arranged by the user in the system memory.

(4) Interrupt
Interrupt signals are asserted in association with the interrupt signal from the ATA/ATAPI device that is connected to the IDE interface or at termination of DMA transfer.

(5) IDE interface
The IDE interface controls all the accesses performed in PIO, Multiword DMA, and UltraDMA modes.

25.1.2 List of Pins

The initial values indicate the initial statuses after resetting by the rst_n signal or the ide_rst_n signal. For the output values after resetting, the same values are retained until the circuit status is changed by the software.

IDE interface pins

LSI pin name	Module internal pin name	I/O	width	Initial value	Function
IDEA[2:0]	ha[2:0]	O	3	0b000	Address
IDED[15:0]	hd_in[15:0](*1)	I	16	–	Read data bus
	hd_out[15:0](*1)	O	16	0x0000	Write data bus
	(hdlo_oen)(*1)	O	1	1	Data [7:0] output enable (Low active)(*2)
	(hdhi_oen)(*1)	O	1	1	Data [15:8] output enable (Low active)
IDERE_N	hiorn	O	1	1	I/O read (DIOR#/HDMARDY#/HSTROBE)(*3)
IDEWR_N	hiown	O	1	1	I/O write (DIOW#/STOP)(*3)
IDERDY	iordy	I	1	–	I/O ready (IORDY/DDMARDY#/DSTROBE)(*3)
IDERST_N	ide_reset_n(*1)	O	1	0	Reset (RESET#)
	(ide_reset_oe_n(*1)	O	1	1	Reset I/O cell output enable (Low active)
	(ide_reset_in_n)(*1)	I	1	–	Feedback of ide_reset_n signal from I/O cell
IDECS1_N	cs1_n	O	1	1	Chip select 1
IDECS0_N	cs0_n	O	1	1	Chip select 0
IDEDREQ	dmarq	I	1	–	DMA request (DMARQ)
IDEDACK_N	dackn	O	1	1	DMA acknowledge (DMACK#)
IDEIRQ	intrq	I	1	–	Interrupt request (INTRQ)
IDENPCBLID	Npcblid(*4)	I	1	–	Primary channel cable ID detection
	(hready)(*1)	I	1	–	AHB transfer complete (shared by master I/F and slave I/F)
	(ide_clk)(*1)	I	1	–	IDE core clock
	(ide_rst_n)(*1)	I	1	–	Power-on reset (synchronized with ide_clk)

(*1) Module internal signal pin
(*2) At access to an 8-bit register (other than the Data register), only hdlo_oen for low-order bytes is asserted.
(*3) The function and polarity of the signal vary according to the transfer mode.
(*4) The npcblid signal is used for distinguishing a cable type for a primary channel. This signal is necessary to distinguish the cable type (40-core or 80-core). If a cable type is restricted by the system (if checking is not necessary), the use of npcblid signal may be excluded.

25.1.3 List of Registers

The offset value notation is used to describe the addresses of the registers. An absolute address is calculated as the "base address + offset address" of each module. The base address value of the IDE controller is 0x7B90_0000.

- Register address value of the IDE controller: 0x7B90_0000 + offset address (address value given in this chapter)

25.1.3.1 DMA Control Registers

Address	Name	Abbreviation	Access size	R/W	Initial value
0x00-0x01(*1)	Primary IDE channel DMA control	BMICP	8/16(*3)	R/W	0x0000
0x02-0x03(*1)	Primary IDE channel DMA status	BMISP	8/16(*3)	R/W	0x0000
0x04-0x07(*1)	Primary IDE channel DMA descriptor pointer	BMIDTPP	8/16/32	R/W	0x0000_0000
0x08-0x09(*2)	Secondary IDE channel DMA control	BMICS	8/16(*4)	R/W	0x0000
0x0A-0x0B(*2)	Secondary IDE channel DMA status	BMISS	8/16(*4)	R/W	0x0000
0x0C-0x0F(*2)	Secondary IDE channel DMA descriptor pointer	BMIDTPS	8/16/32	R/W	0x0000_0000
0x10-0x17	Reserved register (*5)	—	—	—	—
0x18-0x1B	DMA controller feature configuration register	DMAFEAT	8/16/32	R/W	0x0000_0000
0x1C-0x1F	IDE clock control register	CLKCTL	8/16/32	R/W	0x0000_0000
0x20-0x23	Reserved register (*5)	—	—	—	—

(*1) Used Since the controller is set to the primary channel. In this LSI, the controller is fixed to the primary channel in the specification.
(*2) Used when the controller is set to the secondary channel. This register cannot be used in this LSI.
(*3) Access in 32-bit mode is possible by combining the BMICP register and BMISP register.
(*4) Access in 32-bit mode is possible by combining the BMICS register and the BMISS register. This register cannot be used in this LSI.
(*5) Read/Write access to reserved registers is prohibited. When the registers are accessed, the operation is not guaranteed.

25.1.3.2 Configuration Registers

Address	Name	Abbreviation	Access size	R/W	Initial value
0x40-0x41	Primary channel IDE timing control	IDETIMP	8/16(*1)	R/W	0x0000_0000
0x42-0x43(*3)	Secondary channel IDE timing control	IDETIMS	8/16(*1)	R/W	0x0000_0000
0x44	Slave IDE timing Enable	SIDETIM	8	R/W	0x00
0x47	IDE cable ID status	IDESTAT	8	R	0b0000_00xx (*4)
0x48-0x49	UltraDMA control	UDMACTL	8/16(*2)	R/W	0x0000
0x4A-0x4B	UltraDMA timing control	UDMATIM	8/16(*2)	R/W	0x0000
0x50-0x53	IDE timing override control	TIMORIDE	32	R/W	0x0000_0000
0x54-0x57	8-bit register access strobe width	REGSTB	32	R/W	0x0000_0000
0x58-0x5B	8-bit register access recovery time	REGRCVR	32	R/W	0x0000_0000
0x5C-0x5F	Data register access strobe width	DATSTB	32	R/W	0x0000_0000
0x60-0x63	Data register access recovery time	DATRCVR	32	R/W	0x0000_0000
0x64-0x67	DMA access strobe width	DMASTB	32	R/W	0x0000_0000
0x68-0x6B	DMA access recovery time	DMARCVR	32	R/W	0x0000_0000
0x6C-0x6F	UltraDMA access strobe width	UDMASTB	32	R/W	0x0000_0000
0x70-0x73	UltraDMA ready-to-pause time	UDMATRP	32	R/W	0x0000_0000
0x74-0x77	UltraDMA t_{ENV} timing parameter	UDMATENV	32	R/W	0x0000_0000
0x78-0x83	Reserved				
0x84-0x87	Reset pin control	PINCTL	8/16/32	R/W	0x0000_000x (*5)
0x88-0x8B	IORDY timer	IORDYTMR	8/16/32	R/W	0x0000_0000
0x8C-0xFF	Reserved				

(*1) Access in 32-bit mode is possible by combining the IDETIMP register and the IDETIMS register.
(*2) Access in 32-bit mode is possible by combining the UDMACTL register and the UDMATIM register.
(*3) Used when the controller is set to the secondary channel. This register cannot be used in this LSI.
(*4) The values of the npcblid and nscblid signals are reflected in the initial value of the IDESTAT register. Either of the two bits becomes valid according to whether the controller is set to the primary or secondary channel. Refer to the specification of the IDESTAT register for details.
(*5) The initial values of bits 31-1 of the PINCTL register are 0. The initial value of bit 0 is determined by the value of the reset signal of the IDE device.

25.1.3.3 Primary/Secondary Channel Windows

Address	Name	Abbreviation	R/W	Initial value
0x1F0-0x1F7	Primary IDE command block	—	R/W	—
0x3F6	Primary IDE control block	—	R/W	—
0x170-0x177	Secondary IDE command block	—	R/W	—
0x376	Secondary IDE control block	—	R/W	—

The primary and secondary channel windows map ATA registers as shown below. Refer to the ATA Standard for information of each register. Registers other than the Data register are hereafter called 8-bit task file registers.
(Note) In this LSI, the use of secondary channels is prohibited.

Address (Primary)	Address (Secondary)	CS1#	CS0#	DA2	DA1	DA0	Register name	
							Read	Write
3F6h	376h	L	H	H	H	L	Alternate	Status Device Control
1F0h	170h	H	L	L	L	L	Data register (16-bit width)	
1F1h	171h	H	L	L	L	H	Error	Features
1F2h	172h	H	L	L	H	L	Sector Count (For ATAPI, Interrupt Reason) (* For ATAPI, Read Only register)	
1F3h	173h	H	L	L	H	H	Sector Number (Not used by ATAPI)	
1F4h	174h	H	L	H	L	L	Cylinder Low (For ATAPI, Byte Count LSB)	
1F5h	175h	H	L	H	L	H	Cylinder High (For ATAPI, Byte Count MSB)	
1F6h	176h	H	L	H	H	L	Device/Head	
1F7h	177h	H	L	H	H	H	Status	Command

See the table below for the CPU access conditions for primary/secondary channel windows. The sizes of registers other than the Data registers are 8 bits and any access other than 8-bit access is prohibited.
A 32-bit access to Data registers generates two 16-bit accesses.

AHB address	HSIZE	Read	Write
0x1FO (Secondary : 0x170)	010 (32-bit access)	Read from hrdata_s[31:0] →Converted to IDE 16-bit access × 2 times (the Error/Feature register (0x1F1) is NOT accessed)	Write to hwdata_s[31:0] →Converted to IDE 16-bit access × 2 times (the Error/Feature register(0x1F1) is NOT accessed)
	001 (16-bit access)	Read from hrdata_s[15:0] (the Error/Feature register (0x1F1) is NOT accessed)	Write to hwdata_s[15:0] (the Error/Feature register(0x1F1) is NOT accessed)
	000 (8-bit access)	Read from hrdata_s[7:0] (under special circumstances)	Write to hwdata_s[7:0] (under special circumstances)
0x1F1 (or 0x1F2-0x1F7,0x3F6) (Secondary: 0x171-0x177 ,0x376)	010 (32-bit access)	Prohibited	Prohibited
	001 (16-bit access)	Prohibited	Prohibited
	000 (8-bit access)	Read from hrdata_s[7:0]	Write to hwdata_s[7:0]

(Note) In this LSI, the use of a secondary channel is prohibited.

25.2 Registers

25.2.1 DMA Controller Registers

25.2.1.1 Primary/Secondary IDE channel DMA Control (BMICP/BMICS*1)

The BMICP and the BMICS*1 registers are used for starting DMA transfer. The BMICP register is used for the primary channel and the BMICS*1 register is used for the secondary channel. Since the controller is set to the primary channel, the BMICS register is disabled so that, at read operation, "0" is read and the register is ignored at write operation. Read/write operation is enabled for the BMICP register by program control.
At reset, the registers are set to 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMICP/ BMICS*1	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	DMA DIR	—*	—*	DMAS TART
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x00(BMICP)/0x08(BMICS)*1
Access: R/W
Access size: 8/16 bits (32-bit access is possible by combining the BMISP register or the BMISS register)

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.
*1: The BMICS register is used for the secondary channel. This register cannot be used in this LSI.

[Bit Description]

- **DMASTART** (DMA start/stop control) (Bit 0)
  The DMA controller starts when "1" is written to the bit. When "0" is written, DMA data transfer terminates. DMA transfer cannot be restarted. At completion of DMA transfer, this bit remains set to "1".
  If "0" is written to this bit during execution of DMA transfer (the IDEACT bit of the BMISP/BMISS register is set to "1") and before completion of data transfer by the device (INTRSTAT bit of BMISP/BMISS register is set to "0"), the DMA transaction is interrupted and the data transferred from the device may not be written to the memory. This disables interruption/restart of DMA transfer. Make sure that "0" is written to this bit after completion of data transfer. Note that the control method varies depending on whether a single PRD mode or a multi PRD mode is used. Refer to the description of DMA operation in 25.3.2.2 for details.

(Note) Before the start of DMA transfer controlled by this bit, PRD (Physical Region Descriptor) must be stored in the memory and the PRD start address must be set in the BMIDTPP/BMIDTPS register.

DMASTART	Description
0	End of DMA transfer
1	Start of DMA transfer

- **DMADIR** (DMA data transfer direction) (Bit 3)
  Sets a direction of DMA transfer in this bit. When this bit is changed during DMA transfer, the operation is not guaranteed. This setting of this bit can be changed concurrently with the setting of DMASTART bit.

DMADIR	Description
0	DMA read (Write to the device)
1	DMA write (Read from the device)

25.2.1.2 Primary/Secondary IDE Channel DMA Status (BMISP/BMISS*1)

The BMISP and BMISS*1 registers indicate an IDE device status and a DMA transfer status. The registers also indicate an IORDY timeout status at PIO transfer. The BMISP register is used for the primary channel and the BMISS*1 register is used for the secondary channel. Since this controller is set to the primary channel, the BMISS register is disabled so that, at read operation, "0" is read and at write operation, the register is ignored. Read/write operation is enabled for the BMISP register by program control.
At reset, the registers are set to 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMISP/ BMISS*1	—*	—*	—*	—*	—*	—*	—*	HER RINT	SIMP LEX	DMA EN1	DMA EN0	EOTI NT	IORD YINT	INTR STAT	DMAE RROR	IDEA CT
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x02(BMISP)/0x0A(BMISS)*1
Access: R/W
Access size: 8/16 bits (32-bit access is possible by combining with the BMICP register and BMICS register)

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.
*1: The BMISS register is used for the secondary channel. The register cannot be used in this LSI.

[Bit Description]

- **IDEACT** (IDE active) (Bit 0)
  When "1" is written to the DMASTART bit of the BMICP/BMICS register, "1" is set in this bit. This bit is cleared to "0" at termination of the last DMA transfer (transfer for the last PRD). This bit is cleared to "0" also when "0" is written to the DMASTART bit of the BMICSP/BMICS register and DMA transfer is interrupted.

IDEACT	Description
0	End of DMA transfer
1	Executing DMA transfer

The following table shows the status of the controller at termination of DMA data by combination of the EOTINT bit, INTRSTAT bit, and IDEACT bit.

EOTINT	INTRSTAT	IDEACT	Description
0	0	1	DMA transfer is under execution. Interrupt by the IDE device has not occurred.
0	1	0	This indicates the status where the IDE device generated an interrupt, the entire PRD has been used for DMA transfer, and data transfer has been completed. This termination status is set when EOTINTEN bit of the DMAFEAT register is "0" and the size of the physical memory area indicated by PRD and the transfer size of the IDE device are equal.
0	1	1	The IDE device generates an interrupt. Data transfer has not completed to the end of the physical memory area. In this termination status, the size of the physical memory area indicated by PRD is larger than the transfer size of the IDE device.
0	0	0	When the DMAERROR bit is set to "1", data transfer has a problem. When the DMAERROR bit is set to "0", the size specified by PRD is smaller than the IDE transfer size.
1	1	0	In this status, the IDE device generated an interrupt, the entire PRD has been used for DMA transfer, and data transfer has been completed. This termination status is set when the EOTINTEN bit is "1" and the size of the physical memory area indicated by PRD and the remaining transfer size of the IDE device are equal.
1	0	0	When the DMAERROR bit is set to "1", data transfer has a problem. When the size specified by PRD is smaller than the IDE transfer size, the DMAERROR bit is set to "0". This termination status is set when the EOTINTEN bit of the DMAFEAT register is "1" and the size of the physical memory area indicated by PRD is smaller than the remaining transfer size of the IDE device.

Refer to "6.2 Bus Master Function" also for the relationship between the EOTINT bit and DMA transfer.

- **DMAERROR** (DMA error) (Bit 1)
  When an error occurs during data transfer, this bit is set to "1".
  The status can be cleared to "0" by writing "1" to this bit.

(Note) This bit is set to "1" only when the internal buffer overflows or underflows. This status does not normally occur.

DMAERROR	Description
0	Normal
1	DMA error

- **INTRSTAT** (IDE interrupt status) (Bit 2)
  This bit indicates an interrupt from the IDE device. The bit is set to "1" at the rising edge of the IDE device interrupt signal.
  The status can be cleared to "0" by writing "1" to this bit.
  The bit is set to "1" at termination of entire data transfer after the start of DMA transfer.

(Note) Since this bit is set to "1" at the rising edge of the IDE device interrupt signal, it does not necessarily indicate the current status of the IDE interrupt signal.

INTRSTAT	Description
0	Interrupt Clear
1	Interrupt Assert

- **IORDYINT** (PIO IORDY timeout) (Bit 3)
  This bit is set to "1" when timeout occurred in the IORDY timer during PIO data transfer.
  The status can be cleared to "0" by writing "1" to this bit.
  Refer to the section of the IORDYTMR register for details.

(Note) "0" is always read at read operation, when the IORDYTMR register is "0".

IORDYINT	Description
0	Normal
1	IORDY timer timeout

- **EOTINT** (End of PRD Table Interrupt) (Bit 4)
  This bit is set to "1" when the entire data transfer for the current PRD entry is completed and the EOT of the PRD entry is "1".
  The status can be cleared to "0" by writing "1" to this bit.

(Note) This bit is set to "1" only when the EOTINTEN bit of the DMAFEAT register is "1".

EOTINT	Description
0	Normal
1	Completion of data transfer of PRD where EOT = 1

- **DMAEN0** (Device0 DMA enable) (Bit 5)
  This bit indicates whether device 0 (master) can execute DMA transfer.

(Note) This bit is set and used by software, however, it does not have any effect on the controller operation.

DMAEN0	Description
0	Device 0 (master) cannot perform DMA transfer.
1	Device 0 (master) can perform DMA transfer.

- **DMAEN1** (Device1 DMA enable) (Bit 6)
  This bit indicates whether device 1 (slave) can execute DMA transfer.

(Note) This bit is set and used by software, however, it does not have any effect on the controller operation. However, device 1 (slave) cannot be used for this LSI.

DMAEN1	Description
0	Device 1 (slave) cannot perform DMA transfer.
1	Device 1 (slave) can perform DMA transfer.

- **SIMPLEX** (Simplex DMA mode) (Bit 7)
  This bit is fixed to "0".

SIMPLEX	Description
0	The primary and secondary channels can be used concurrently.
1	Only one channel can be used each time.

- **HERRINT** (End of PRD Table Interrupt) (Bit 4)
  This bit indicates that an AHB error response receive interrupt has been generated.
  This bit is set to "1" when an error response has been received from the AHB slave while this controller is operating (during data transfer) as the AHB master.
  The status can be cleared to "0" by writing "1" to this bit.

HERRINT	Description
0	Normal
1	Error response has been received.

25.2.1.3 Primary/Secondary IDE Channel DMA Descriptor Pointer (BMIDTPP/BMIDTPS*1)

The BMIDTPP and BMIDTPS*1 registers indicate the first address of the DMA descriptor table. The BMIDTPP register is used for the primary channel and the BMIDTPS*1 is used for the secondary channel. Since the controller is set to the primary channel, the BMIDTPS register is disabled so that, at read operation, "0" is read and the register is ignored at write operation. Read/write operation is enabled for the BMIDTPP and BMIDTPS*1 registers through program control.
At reset, the registers are set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMIDTPP/ BMIDTPS	BMIDTP[31]	BMIDTP[30]	BMIDTP[29]	BMIDTP[28]	BMIDTP[27]	BMIDTP[26]	BMIDTP[25]	BMIDTP[24]	BMIDTP[23]	BMIDTP[22]	BMIDTP[21]	BMIDTP[20]	BMIDTP[19]	BMIDTP[18]	BMIDTP[17]	BMIDTP[16]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BMIDTP[15]	BMIDTP[14]	BMIDTP[13]	BMIDTP[12]	BMIDTP[11]	BMIDTP[10]	BMIDTP[9]	BMIDTP[8]	BMIDTP[7]	BMIDTP[6]	BMIDTP[5]	BMIDTP[4]	BMIDTP[3]	BMIDTP[2]	—*	—*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x04(BMIDTPP)/0x0C(BMIDTPS)
Access: R/W
Access size: 8/16/32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.
*1: The BMIDTPS register is used for the secondary channel. This register cannot be used in this LSI.

[Bit Description]

- **BMIDTP[31:2]** (Descriptor table base address pointer) (Bits 31-2)
  The first address of the DMA descriptor table (PRD(Physical Region Descriptor)) is set in this register. This register must be set before the DMASTART bit of BMICS/BMICP register is set (before the DMA start).

(Note) Since the descriptor table must apply a double-word (4-byte) alignment format, the bits 1-0 are always set to "0". The descriptor table must not cross a 64K-byte boundary.

25.2.1.4 DMA Controller Feature Configuration Register (DMAFEAT)

The DMAFEAT register is used for setting a DMA controller control mode.
Read/write operation is enabled for the DMAFEAT register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAFEAT	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	EOTINTEN	MULTPRDT
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x18
Access: R/W
Access size: 8/16/32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **MULTPRDT** (Multiple PRD table mode enable) (Bit 0)
  When this bit is "0", the DMA controller operates in the mode assuming that the size of the physical memory area indicated by PRD and the data transfer size of the IDE device are equal. This is called a single PRD mode.
  When this bit is "1", the DMA controller operates in the mode assuming that the size of the physical memory indicated by PRD is smaller than the data transfer size of the IDE device. This is called a multi PRD mode. In this mode, DMA is assumed to be activated as many times as the data transfer size of the IDE device.
  The control mode of the internal DMA controller can be changed by setting this bit. Set the bit according to the executed DMA mode (single mode or multi PRD mode).

(Note) Behaviors of the DMA controller are described below as the reference information.
The DMA controller starts when this bit is "0" and the DMASTART bit of the BMICP/BMICS register is changed from "0" to "1". The internal circuit of the DMA controller is reset automatically when the DMASTART bit is changed from "1" to "0".
The DMA controller starts when this bit is set to "1" and "1" is written to the DMASTART bit of the BMICP/BMICS register (regardless of the previous value). The internal circuit of the DMA controller is automatically reset when the DMASTART bit is changed from "0" to "1". The DMA controller is not reset when "1" is overwritten while the DMASTART bit is "1".

MULTPRDT	Description
0	Single PRD mode
1	Multi PRD mode

For multi PRD mode, refer to "25.3.2 Bus Master Function" also.

- **EOTINTEN** (End of PRD table interrupt enable) (Bit 1)
  When this bit is set to "0", the EOTINT bit of the BMISP/BMISS register is disabled and does not function. When this bit is set to "1", the EOTINT bit of the BMISP/BMISS register is disabled. In DMA transfer, when the EOT bit of PRD is set to "1" and the applicable data transfer is completed, the EOTINT bit is set to "1". When the EOTINT bit is set to "1", a CPU interrupt signal (intr_n) is asserted.

EOTINTEN	Description
0	The EOTINT bit of the BMISP/BMISS register is disabled.
1	The EOTINT bit of the BMISP/BMISS register is enabled.

  Refer to the "25.3.2 Bus Master Function" also for the relationship between the EOTINT bit and DMA transfer.

25.2.1.5 IDE Clock Control Register (CLKCTL)

The CLKCTL register is used for stopping an IDE interface clock.
Read/write operation is enabled for the CLKCTL register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLKCTL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	ICLK DIS
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x1C
Access: R/W
Access size: 8/16/32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **ICLKDIS** (IDE Clock Disable) (Bit 0)
  This bit enables/disables the IDE interface clock (ide_clk signal). When "1" is set in this bit, the IDE interface clock inside of the controller stops. The status is retained until "0" is set in this bit.

(Note) Write operation to the configuration register (address 0x40-0xFF) is disabled while this bit is set to "1" and the IDE interface clock is disabled. Invalid data is read at read operation. Write/read operation is enabled to the DMA controller register (address 0x00-0x1F). The IDE interface is stopped. Change the value of this bit only at the idle status where processing such as DMA transfer is completed.

ICLKDIS	Description
0	The IDE interface clock is enabled.
1	The IDE interface clock is disabled.

25.2.2 Configuration Registers

25.2.2.1 Primary/Secondary Channel IDE Timing Control (IDETIMP/IDETIMS*1)

The IDETIMP and IDETIMS*1 registers are used for setting the timing of the IDE interface. The IDETIMP register is used for the primary channel and the IDETIMS*1 register is used for the secondary channel. Since the controller is set to the primary channel, the IDETIMS register is disabled so that, at read operation, "0" is read and the register is ignored at write operation. Read/write operation is enabled for the IDETIMP and IDETIMS*1 registers through program control.
At reset, the registers are set to 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDETIMS*1	IDEEN	SLVTIMEN	RDYSMPL[1]	RDYSMPL[0]	—*	—*	RDYRCVRY[1]	RDYRCVRY[0]	DMAFTIM1	PREPOST1	RDYSEN1	PIOFTIM1	DMAFTIM0	PREPOST0	RDYSEN0	PIOFTIM0
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDETIMP	IDEEN	SLVTIMEN	RDYSMPL[1]	RDYSMPL[0]	—*	—*	RDYRCVRY[1]	RDYRCVRY[0]	DMAFTIM1	PREPOST1	RDYSEN1	PIOFTIM1	DMAFTIM0	PREPOST0	RDYSEN0	PIOFTIM0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x40 (IDETIMP) /0x42 (IDETIMS)
Access: R/W
Access size: 8/16 bits (32-bit access mode is enabled by combining the IDETIMP register and the IDETIMS register)

[Note]
The IDETIMP register is assigned to address 0x40.
The IDETIMS register is assigned to address 0x42.
*1: The IDETIMS register is used for the secondary channel. This register cannot be used in this LSI.
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **PIOFTIM0** (Device0 PIO fast timing enable) (Bit 0)
  This bit controls PIO data access timing of device 0 (master). This bit is related to the setting of the DMAFTIM0 bit and the TIMORIDE register. Refer to "25.3.4 Timing Setting" for details.

PIOFTIM0	Description
0	PIO data access to device 0 (master) is performed at timing of PIO mode 0
1	When the DMAFTIM0 bit is "0": The PIO data access to device 0 (master) is performed according to the timing parameter that is set in RDYSMPL and RDYRCVRY fields. When the DMAFTIM0 bit is "1": The PIO data access to device 0 (master) is performed at the timing of PIO mode 0.

- **RDYSEN0** (Device0 IORDY sampling point enable) (Bit 1)
  This bit is used for setting the handling of the IORDY signal at PIO access of device 0 (master).

RDYSEN0	Description
0	The IORDY signal of device 0 (master) is ignored.
1	The IORDY signal is used at PIO access to device 0 (master). The sampling point of the IORDY signal is specified by the RDYSMPL bit.

- **PREPOST0** (Device0 PIO prefetch and Postwrite enable) (Bit 2)
  This bit is not used. Always write "0" to this bit. Operation cannot be guaranteed if "1" is written to this bit.

PREPOST0	Description
0	Normal
1	Setting prohibited. Operation cannot be guaranteed if set to 1.

- **DMAFTIM0** (Device0 DMA fast timing enable) (Bit 3)
  This bit controls timing at Multiword DMA transfer of device 0 (master). This bit is related to the setting of the PIOFTIM0 bit and the TIMORIDE register. Refer to "25.3.4 Timing Setting" for details.

DMAFTIM0	Description
0	Multiword DMA transfer to device 0 (master) is performed at mode 0 timing.
1	Multiword DMA transfer to device 0 (master) is performed according to the timing parameter set in the RDYSMPL and RDYRCVRY fields.

- **PIOFTIM1** (Device1 PIO fast timing enable) (Bit 4)
  This bit controls PIO data access timing of device 1 (slave). However, device 1 (slave) cannot be used for this LSI. This bit is related to the setting of the DMAFTIM1 bit and the TIMORIDE register. Refer to "25.3.4 Timing Setting" for details.

PIOFTIM1	Description
0	PIO data access to device 1 (slave) is performed at timing of PIO mode 0.
1	When the DMAFTIM1 bit is "0": The PIO data access to device 1 (slave) is performed according to the timing parameter that is set in RDYSMPL and RDYRCVRY fields. When the DMAFTIM1 bit is "1": The PIO data access to device 1 (slave) is performed at the timing of PIO mode 0.

- **RDYSEN1** (Device1 IORDY sampling point enable) (Bit 5)
  This bit sets the handling of the IORDY signal at PIO access of device 1 (slave). However, device 1 (slave) cannot be used for this LSI.

RDYSEN1	Description
0	The IORDY signal of device 1 (slave) is ignored.
1	The IORDY signal is used at PIO access to device 1 (slave). The sampling point of the IORDY signal is specified by the RDYSMPL bit.

- **PREPOST1** (Device1 PIO prefetch and postwrite enable) (Bit 6)
  This bit controls prefetch/postwrite at PIO data access of device 1 (slave).

PREPOST1	Description
0	Normal
1	Setting prohibited. Operation cannot be guaranteed if set to 1.

- **DMAFTIM1** (Device1 DMA fast timing enable) (Bit 7)
  This bit controls timing at Multiword DMA transfer of device 1 (slave). However, device 1 (slave) cannot be used for this LSI. This bit is related to the setting of the PIOFTIM1 bit and the TIMORIDE register. Refer to "25.3.4 Timing Setting" for details.

DMAFTIM1	Description
0	Multiword DMA transfer to device 1 (slave) is performed at mode 0 timing.
1	Multiword DMA transfer to device 1 (slave) is performed according to the timing parameter set in the RDYSMPL and RDYRCVRY fields.

- **RDYRCVRY[1:0]** (IORDY recovery time) (Bits 9-8)
  These bits set the minimum time allowed from Assert of the IORDY signal to Assert of the DIOW#/DIOR# signal of the next cycle.

(Note) The time set in these bits is valid when 66 MHz is used for the IDE interface clock. When using a clock whose speed is less than 66 MHz, the setting time will get longer according to the frequency of the clock. It is possible to strictly set the timing by validating the TIMORIDE register and setting the clock individually using the timing register group.

RDYRCVRY[1:0]		Description
1	0	
0	0	120 ns
0	1	100 ns
1	0	75 ns
1	1	50 ns

- **RDYSMPL[1:0]** (IORDY sampling point) (Bits 13-12)
  These bits set a time from Assert of the DIOW#/DIOR# signal to the first sampling point of the IORDY signal.

(Note) The time set in these bits is valid when 66 MHz is used for the IDE interface clock. When using a clock whose speed is less than 66 MHz, the setting time will get longer according to the frequency of the clock. It is possible to strictly set the timing by validating the TIMORIDE register and setting the clock individually using the timing register group.

RDYSMPL[1:0]		Description
1	0	
0	0	120 ns
0	1	100 ns
1	0	80 ns
1	1	70 ns

- **SLVTIMEN** (Slave IDE timing enable) (Bit 14)
  This bit controls the access timing setting of device 1 (slave). However, device 1 (slave) cannot be used for this LSI.

SLVTIMEN	Description
0	Access timing to device 0 (master) and device 1 (slave) is set in the RDYSMPL and the RDYRCVRY fields.
1	The RDYSMPL and RDYRCVRY fields are valid only for access timing of device 0 (master). Timing of device 1 (slave) is set by the SIDETIM register.

- **IDEEN** (IDE decode enable) (Bit 15)
  This bit controls the decoding function of access to registers (command and control block) of the IDE device.

IDEEN	Description
0	PIO access to the registers of IDE device is not decoded. Access to the address space of the IDE device does not trigger the access to the IDE interface.
1	PIO access to registers of the IDE device is decoded.

25.2.2.2 Slave IDE Timing Enable (SIDETIM)

The SIDETIM register sets timing of the IDE interface for device 1 (slave). This register can be set for each of the primary and secondary channels. This register is disabled unless the SLVTIMEN bit of the IDETIM register is set to "1". (Register read/write operation is enabled, however, the function is disabled.) However, device 1 (slave) cannot be used for this LSI.
Read/write operation is enabled for the SIDETIME register by program control.
At reset, the register is set to 0x00.

	7	6	5	4	3	2	1	0
SIDETIM	RDYSMPS1 [1]	RDYSMPS1 [0]	RDYRCYS1 [1]	RDYRCYS1 [0]	RDYSMPP1 [1]	RDYSMPP1 [0]	RDYRCYP1 [1]	RDYRCYP1 [0]
	0	0	0	0	0	0	0	0

Address: 0x44
Access: R/W
Access size: 8 bits

[Bit Description]

- **RDYRCYP1[1:0]** (Primary device1 IORDY recovery time) (Bits 1-0)
  These bits are used for setting the minimum time allowed from Assert of the IORDY signal for device 1 (slave) of the primary channel to Assert of the DIOW#/DIOR# signal of the next cycle. However, device 1 (slave) cannot be used for this LSI.

(Note) The time set in these bits is valid when 66 MHz is used for the IDE interface clock. When using a clock whose speed is less than 66 MHz, the setting time will get longer according to the frequency of the clock. It is possible to strictly set the timing by validating the TIMORIDE register and setting the clock individually using the timing register group.

RDYRCYP1[1:0]		Description
1	0	
0	0	120 ns
0	1	100 ns
1	0	75 ns
1	1	50 ns

- **RDYSMPP1[1:0]** (Primary device1 IORDY sampling time) (Bits 3-2)
  These bits are used for setting the time allowed from Assert of the DIOW#/DIOR# signal of device 1 (slave) of the primary channel to the first sampling point of the IORDY signal. However, device 1 (slave) cannot be used for this LSI.

(Note) The time set in these bits is valid when 66 MHz is used for the IDE interface clock. When using a clock whose speed is less than 66 MHz, the setting time will get longer according to the frequency of the clock. It is possible to strictly set the timing by validating the TIMORIDE register and setting the clock individually using the timing register group.

RDYSMPP1[1:0]		Description
1	0	
0	0	120 ns
0	1	100 ns
1	0	80 ns
1	1	70 ns

- **RDYRCYS1[1:0]** (Secondary device1 IORDY recovery time) (Bits 5-4)
  These bits are used for setting the minimum time allowed from Assert of the IORDY signal for device 1 (slave) of the secondary channel to Assert of the DIOW#/DIOR# signal of the next cycle.

(Note) Since the controller is set to the primary channel in this LSI, these bits are disabled.

(Note) The time set in these bits is valid when 66 MHz is used for the IDE interface clock. When using a clock whose speed is less than 66 MHz, the setting time will get longer according to the frequency of the clock. It is possible to strictly set the timing by validating the TIMORIDE register and setting the clock individually using the timing register group.

RDYRCYS1[1:0]		Description
1	0	
0	0	120 ns
0	1	100 ns
1	0	75 ns
1	1	50 ns

- **RDYSMPS1[1:0]** (Secondary device1 IORDY sampling time) (Bits 7-6)
  These bits are used for setting the minimum time allowed from assert of the DIOW#/DIOR# signal for device 1 (slave) of the secondary channel to the first sampling point of the IORDY signal.

(Note) Since the controller is set to the primary channel in this LSI, these bits are disabled.

(Note) The time set in these bits is valid when 66 MHz is used for the IDE interface clock. When using a clock whose speed is less than 66 MHz, the setting time will get longer according to the frequency of the clock. It is possible to strictly set the timing by validating the TIMORIDE register and setting the clock individually using the timing register group.

RDYSMPS1[1:0]		Description
1	0	
0	0	120 ns
0	1	100 ns
1	0	80 ns
1	1	70 ns

25.2.2.3 IDE Cable ID Status (IDESTAT)

The IDESTAT register indicates cable ID pin values.
The IDESTAT register can be read through program control.
At reset, the register is set to 0b000000xx.

	7	6	5	4	3	2	1	0
IDESTAT	—*	—*	—*	—*	—*	—*	CABLEIDP	CABLEIDS
At reset	0	0	0	0	0	0	X	X

Address: 0x47
Access: R
Access size: 8 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **CABLEIDS** (Secondary Cable ID) (Bit 0)
  This bit indicates the cable ID status of the secondary channel. Since the controller is set to the primary channel, the bit is always set to "0".

CABLEIDS	Description
0	80-core cable
1	40-core cable

- **CABLEIDP** (Primary Cable ID) (Bit 1)
  This bit indicates the cable ID status of the primary channel. When the controller is set to the secondary channel, the bit is always set to "0".

CABLEIDS	Description
0	80-core cable
1	40-core cable

(Note) The npcblid and nscblid signals can be read through this register. When the signals are not used, fix the setting to "0". The npcblid signal is connected to the IDENPCBLID pin in this LSI. The nscblid signal is not connected to the outside of this LSI.

25.2.2.4 UltraDMA Control (UDMACTL)

The UDMACTL register validates UltraDMA transfer.
Read/write operation is enabled for the UDMACTL register by program control.
At reset, the register is set to 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDMACTL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	UDMAS1	UDMAS0	UDMAP1	UDMAP0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x48
Access: R/W
Access size: 8/16 bits (32-bit access is enabled by combining with the UDMATIM register)

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **UDMAP0** (Primary device0 UDMA enable) (Bit 0)
  This bit validates UltraDMA transfer of device 0 (master) of the primary channel. When this bit is set to "0", Multiword DMA transfer is used.

UDMAP0	Description
0	Primary device 0 (master): Multiword DMA
1	Primary device 0 (master): UltraDMA

- **UDMAP1** (Primary device1 UDMA enable) (Bit 1)
  This bit validates UltraDMA transfer of device 1 (slave) of the primary channel. When this bit is set to "0", Multiword DMA transfer is used. However, device 1 (slave) cannot be used for this LSI.

UDMAP1	Description
0	Primary device 1 (slave):Multiword DMA
1	Primary device 1 (slave):UltraDMA

- **UDMAS0** (Secondary device0 UDMA enable) (Bit 2)
  These bits cannot be used in this LSI. Always write "0" to this bit. Since the controller is fixed to the primary channel, this bit will read "0" when read and any write operation to this bit will be ignored.
  This bit validates UltraDMA transfer of device 0 (master) of the secondary channel. When this bit is set to "0", Multiword DMA transfer is used.

(Note) Since the controller is set to the primary channel in this LSI, this bit is disabled.

UDMAP0	Description
0	Secondary device 0 (master):Multiword DMA
1	Secondary device 0 (master):UltraDMA

- **UDMAS1** (Secondary device1 UDMA enable) (Bit 3)
  These bits cannot be used in this LSI. Always write "0" to this bit. Since the controller is fixed to the primary channel, this bit will read "0" when read and any write operation to this bit will be ignored.
  This bit validates UltraDMA transfer of device 1 (slave) of the secondary channel. When this bit is set to "0", Multiword DMA transfer is used.

(Note) Since the controller is set to the primary channel in this LSI, this bit is disabled.

UDMAP0	Description
0	Secondary device 1 (slave): Multiword DMA
1	Secondary device 1 (slave): UltraDMA

25.2.2.5 UltraDMA Timing Control (UDMATIM)

The UDMATIM register sets an UltraDMA transfer mode. This register is ignored in PIO transfer mode and Multiword transfer mode. The setting of this register determines the minimum time of the data write strobe cycle time (CT) and Ready to Pause time (RP).
Read/write operation is enabled for the UDMATIM register by program control.
At reset, the register is set to 0x0000.
Since the controller is fixed to the primary channel in this LSI, the controller cannot be set to the secondary channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDMATIM	—*	TCYCS1[2]	TCYCS1[1]	TCYCS1[0]	—*	TCYCS0[2]	TCYCS0[1]	TCYCS0[0]	—*	TCYCP1[2]	TCYCP1[1]	TCYCP1[0]	—*	TCYCP0[2]	TCYCP0[1]	TCYCP0[0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x4A
Access: R/W
Access size: 8/16 bits (32-bit access is enabled by combining with the UDMACTL register)

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **TCYCP0[2:0]** (Primary device0 cycle time) (Bits 2-0)
  These bits set an UltraDMA transfer mode for device 0 (master) of the primary channel.

TCYCP0[2:0]			Description
[2]	[1]	[0]	
0	0	0	Primary device 0 (master): UltraDMA mode 0(UltraATA/33 or higher)
0	0	1	Primary device 0 (master): UltraDMA mode 1(UltraATA/33 or higher)
0	1	0	Primary device 0 (master): UltraDMA mode 2(UltraATA/33 or higher)
0	1	1	Primary device 0 (master): UltraDMA mode 3(UltraATA/33 or higher)
1	0	0	Primary device 0 (master): UltraDMA mode 4(UltraATA/66 or higher only)
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

- **TCYCP1[2:0]** (Primary device1 cycle time) (Bits 6-4)
  These bits set an UltraDMA transfer mode for device 1 (slave) of the primary channel. However, device 1 (slave) cannot be used for this LSI.

TCYCP1[2:0]			Description
[2]	[1]	[0]	
0	0	0	Primary device 1 (slave): UltraDMA mode 0 (UltraATA/33 or higher)
0	0	1	Primary device 1 (slave): UltraDMA mode 1 (UltraATA/33 or higher)
0	1	0	Primary device 1 (slave): UltraDMA mode 2 (UltraATA/33 or higher)
0	1	1	Primary device 1 (slave): UltraDMA mode 3 (UltraATA/33 or higher)
1	0	0	Primary device 1 (slave): UltraDMA mode 4 (UltraATA/66 or higher only)
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

- **TCYCS0[2:0]** (Secondary device0 cycle time) (Bits 10-8)
  These bits cannot be used in this LSI. Fix them to "0". Since the controller is fixed to the primary channel, these bits will read "0" when read and any write operation to these bits will be ignored.
  These bits set an UltraDMA transfer mode for device 0 (master) of the secondary channel.

TCYCP0[2:0]			Description
[2]	[1]	[0]	
0	0	0	Secondary device 0 (master): UltraDMA mode 0 (UltraATA/33 or higher)
0	0	1	Secondary device 0 (master): UltraDMA mode 1 (UltraATA/33 or higher)
0	1	0	Secondary device 0 (master): UltraDMA mode 2 (UltraATA/33 or higher)
0	1	1	Secondary device 0 (master): UltraDMA mode 3 (UltraATA/33 or higher)
1	0	0	Secondary device 0 (master): UltraDMA mode 4 (UltraATA/66 or higher only)
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

- **TCYCS1[2:0]** (Secondary device1 cycle time) (Bits 14-12)
  These bits cannot be used in this LSI. Fix them to "0". Since the controller is fixed to the primary channel, these bits will read "0" when read and any write operation to these bits will be ignored.
  These bits set an UltraDMA transfer mode for device 1 (slave) of the secondary channel.

TCYCP1[2:0]			Description
[2]	[1]	[0]	
0	0	0	Secondary device 1 (slave): UltraDMA mode 0(UltraATA/33 or higher)
0	0	1	Secondary device 1 (slave): UltraDMA mode 1 (UltraATA/33 or higher)
0	1	0	Secondary device 1 (slave): UltraDMA mode 2 (UltraATA/33 or higher)
0	1	1	Secondary device 1 (slave): UltraDMA mode 3 (UltraATA/33 or higher)
1	0	0	Secondary device 1 (slave): UltraDMA mode 4 (UltraATA/66 or higher only)
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

25.2.2.6 IDE Timing Override Control (TIMORIDE)

The TIMORIDE register enables a timing register group to determine the timing of the IDE interface.
Read/write operation is enabled for the TIMORIDE register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMORIDE	—*	—*	—*	—*	—*	—*	—*	—*	—	—*	—*	—*	—*	—*	—	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	TIMORIDE
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x50
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **TIMORIDE** (Override default IDE timing controls) (Bit 0)
  When this bit is set to "1", the timing of the IDE interface is determined by the setting of the timing register group.
  The timing register group includes the registers listed below. When this bit is set to "0", the setting of each timing register is disabled (Read/write processing is enabled for the register).
  REGSTB (8-bit register access strobe width)
  REGRCVR (8-bit register access recovery time)
  DATSTB (Data register access strobe width)
  DATRCVR (Data register access recovery time)
  DMASTB (DMA access strobe width)
  DMARCVR (DMA access recovery time)
  UDMASTB (UltraDMA access strobe width)
  UDMATRP (UltraDMA ready-to-pause time)
  UDMATENV (UltraDMA t_{ENV} timing parameter)
  See "25.3.4 Timing Setting" and description of each register for details.

TIMORIDE	Description
0	The timing of the IDE interface is determined by the parameter in the controller.
1	The timing of the IDE interface is determined by the timing register group.

25.2.2.7 8-Bit Register Access Strobe Width (REGSTB)

The REGSTB register is used for setting a strobe width at 8-bit register access. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the REGSTB register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REGSTB	—*	—*	—*	—*	—*	—*	—*	—*	—*	REGSTB[6]	REGSTB[5]	REGSTB[4]	REGSTB[3]	REGSTB[2]	REGSTB[1]	REGSTB[0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	REGSTB1[6]	REGSTB1[5]	REGSTB1[4]	REGSTB1[3]	REGSTB1[2]	REGSTB1[1]	REGSTB1[0]	—*	REGSTB0[6]	REGSTB0[5]	REGSTB0[4]	REGSTB0[3]	REGSTB0[2]	REGSTB0[1]	REGSTB0[0]
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x54
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **REGSTB0[6:0]** (8-bit register access master strobe width) (Bits 6-0)
  These bits set the assert time of the DIOW#/DIOR# signal at access to the 8-bit task file register of device 0 (master). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register and the PIOFTIM0 bit of the IDETIM register are set to "1".

- **REGSTB1[6:0]** (8-bit register access slave strobe width) (Bits 14-8)
  These bits set the assert time of the DIOW#/DIOR# signal at access to the 8-bit task file register of device 1 (slave). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register and the PIOFTIM1 bit of the IDETIM register are set to "1".

- **REGSTB[6:0]** (8-bit register access default strobe width) (Bits 22-16)
  These bits are unused. At write operation, write REGSTB[6:0] = 000000b only and do not write any other value.

25.2.2.8 8-Bit Register Access Recovery Time (REGRCVR)

The REGRCVR register is used for setting a recovery time at 8-bit register access. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the REGRCVR register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REGRCVR	—*	—*	—*	—*	—*	—*	—*	—*	—*	REGRCVR[6]	REGRCVR[5]	REGRCVR[4]	REGRCVR[3]	REGRCVR[2]	REGRCVR[1]	REGRCVR[0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	REGRCVR1[6]	REGRCVR1[5]	REGRCVR1[4]	REGRCVR1[3]	REGRCVR1[2]	REGRCVR1[1]	REGRCVR1[0]	—*	REGRCVR0[6]	REGRCVR0[5]	REGRCVR0[4]	REGRCVR0[3]	REGRCVR0[2]	REGRCVR0[1]	REGRCVR0[0]
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x58
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **REGRCVR0[6:0]** (8-bit register access master recovery time) (Bits 6-0)
  These bits set the recovery time (time from negate to assert) of the DIOW#/DIOR# signal at access to the 8-bit task file register of device 0 (master). As the recovery time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register and the PIOFTIM0 bit of the IDETIM register are set to "1".

- **REGRCVR1[6:0]** (8-bit register access slave recovery time) (Bits 14-8)
  These bits set the recovery time (time from negate to assert) of the DIOW#/DIOR# signal at access to the 8-bit task file register of device 1 (slave). As the recovery time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register and the PIOFTIM0 bit of the IDETIM register are set to "1".

- **REGRCVR[6:0]** (8-bit register access default recovery time) (Bits 22-16)
  These bits are unused. At write operation, write REGRCVR[6:0] = 000000b only and do not write any other value.

25.2.2.9 Data Register Access Strobe Width (DATSTB)

The DATSTB register is used for setting a strobe width at Data register access. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the DATSTB register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATSTB	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATSTB	—*	DATSTB1 [6]	DATSTB1 [5]	DATSTB1 [4]	DATSTB1 [3]	DATSTB1 [2]	DATSTB1 [1]	DATSTB1 [0]	—*	DATSTB0 [6]	DATSTB0 [5]	DATSTB0 [4]	DATSTB0 [3]	DATSTB0 [2]	DATSTB0 [1]	DATSTB0 [0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x5C
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **DATSTB0[6:0]**(Master Data register access strobe width) (Bits 6-0)
  These bits set the assert time of the DIOW#/DIOR# signal at access to the Data register of device 0 (master). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

- **DATSTB1[6:0]**(Slave Data register access strobe width) (Bits 14-8)
  These bits set the assert time of the DIOW#/DIOR# signal at access to the Data register of device 1 (slave). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

25.2.2.10 Data Register Access Recovery Time (DATRCVR)

The DATRCVR register is used for setting the recovery time at access to the Data register. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the DATRCVR register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATRCVR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATRCVR	—*	DATRCVR1[6]	DATRCVR1[5]	DATRCVR1[4]	DATRCVR1[3]	DATRCVR1[2]	DATRCVR1[1]	DATRCVR1[0]	—*	DATRCVR0[6]	DATRCVR0[5]	DATRCVR0[4]	DATRCVR0[3]	DATRCVR0[2]	DATRCVR0[1]	DATRCVR0[0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x60
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **DATRCVR0[6:0]** (Master Data register access recovery time) (Bits 6-0)
  These bits set the recovery time of the DIOW#/DIOR# signal at access to the Data register of device 0 (master). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

- **DATRCVR1[6:0]** (Slave Data register access recovery time) (Bits 14-8)
  These bits set the recovery time of the DIOW#/DIOR# signal at access to the Data register of device 1 (slave). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

25.2.2.11 DMA Access Strove Width (DMASTB)

The DMASTB register is used for setting a strobe width at Multiword DMA transfer. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the DMASTB register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASTB	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—*	DMASTB1[6]	DMASTB1[5]	DMASTB1[4]	DMASTB1[3]	DMASTB1[2]	DMASTB1[1]	DMASTB1[0]	—*	DMASTB0[6]	DMASTB0[5]	DMASTB0[4]	DMASTB0[3]	DMASTB0[2]	DMASTB0[1]	DMASTB0[0]
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x64
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **DMASTB0[6:0]** (Master DMA access strobe width) (Bits 6-0)
  These bits set the assert time of the DIOW#/DIOR# signal at Multiword DMA transfer of device 0 (master). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

- **DMASTB1[6:0]** (Slave DMA access strobe width) (Bits 14-8)
  These bits set the assert time of the DIOW#/DIOR# signal at Multiword DMA transfer of device 1 (slave). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

25.2.2.12 DMA Access Recovery Time (DMARCVR)

The DMARCVR register is used for setting the recovery time at Multiword DMA transfer. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the DMARCVR register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMARCVR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMARCVR	—*	DMA RCVR 1[6]	DMA RCVR 1[5]	DMA RCVR 1[4]	DMA RCVR 1[3]	DMA RCVR 1[2]	DMA RCVR 1[1]	DMA RCVR 1[0]	—*	DMA RCVR 0[6]	DMA RCVR 0[5]	DMA RCVR 0[4]	DMA RCVR 0[3]	DMA RCVR 0[2]	DMA RCVR 0[1]	DMA RCVR 0[0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x68
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **DMARCVR0[6:0]** (Master DMA access recovery time) (Bits 6-0)
  These bits set the recovery time of the DIOW#/DIOR# signal at Multiword DMA transfer of device 0 (master). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

- **DMARCVR1[6:0]** (Slave DMA access recovery time) (Bits 14-8)
  These bits set the recovery time of the DIOW#/DIOR# signal at Multiword DMA transfer of device 1 (slave). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

25.2.2.13 UltraDMA Access Strobe Width (UDMASTB)

The UDMASTB register is used for setting the strobe width at UltraDMA Data Out transfer. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the UDMASTB register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMASTB	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDMASTB	—*	—*	—*	—*	UDMASTB1[3]	UDMASTB1[2]	UDMASTB1[1]	UDMASTB1[0]	—*	—*	—*	—*	UDMASTB0[3]	UDMASTB0[2]	UDMASTB0[1]	UDMASTB0[0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x6C
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **UDMASTB0[3:0]** (Master Ultra DMA access strobe width) (Bits 3-0)
  These bits set the assert time of the DIOR# signal (strobe signal at Write access to the device) at UltraDMA Data Out transfer of device 0 (master). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

- **UDMASTB1[3:0]** (Slave Ultra DMA access strobe width) (Bits 11-8)
  These bits set the assert time of the DIOR# signal (strobe signal at Write access to the device) at UltraDMA Data Out transfer of device 1 (slave). As the assert time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

25.2.2.14 UltraDMA ready-to-pause Time (UDMATRP)

The UDMATRP register is used for setting the ready-to-pause time at UltraDMA transfer. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the UDMATRP register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMATRP	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDMATRP	—*	—*	—*	UDMATRP1[4]	UDMATRP1[3]	UDMATRP1[2]	UDMATRP1[1]	UDMATRP1[0]	—*	—*	—*	UDMATRP0[4]	UDMATRP0[3]	UDMATRP0[2]	UDMATRP0[1]	UDMATRP0[0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x70
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **UDMATRP0[4:0]**(Master UltraDMA ready-to-pause time) (Bits 4-0)
  These bits set the time between the HREADY signal and the STOP signal at UltraDMA transfer of device 0 (master). As the time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

- **UDMATRP1[4:0]**(Slave UltraDMA ready-to-pause time) (Bits 12-8)
  These bits set the time between the HREADY signal and the STOP signal at UltraDMA transfer of device 1 (slave). As the time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

25.2.2.15 UltraDMA t_{ENV} Timing Parameter (UDMATENV)

The UDMATENV register is used for setting the timing parameter (tENV) at UltraDMA transfer. This register is valid only when the TIMORIDE bit of the TIMORIDE register is set to "1".
Read/write operation is enabled for the UDMATENV register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMATENV	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDMATENV	—*	—*	—*	—*	—*	—*	—*	—*	UDMTENV1[3]	UDMTENV1[2]	UDMTENV1[1]	UDMTENV1[0]	UDMTENV0[3]	UDMTENV0[2]	UDMTENV0[1]	UDMTENV0[0]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x74
Access: R/W
Access size: 32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **UDMTENV0[3:0]** (t_{ENV} timing parameter for master device) (Bits 3-0)
  These bits set the time of timing parameter t_{ENV} at UltraDMA transfer of device 0 (master). As the time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

- **UDMTENV1[3:0]** (t_{ENV} timing parameter for slave device) (Bits 7-4)
  These bits set the time of timing parameter t_{ENV} at UltraDMA transfer of device 1 (slave). As the time, a value of clock cycle count – 1 is set. Value "0" indicates one clock cycle. However, device 1 (slave) cannot be used for this LSI.

(Note): These bits are enabled only when the TIMORIDE bit of the TIMORIDE register is set to "1".

25.2.2.16 Reset Pin Control (PINCTL)

The PINCTL register controls a reset signal to the device in the IDE interface.
Read/write operation is enabled for the PINCTL register by program control.
At reset, bits 31-1 are set to "0" and bit 0 is set to the value determined according to the value of the reset signal of the IDE device.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PINCTL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCTL	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	RSTMODE	RESET
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X

Address: 0x84
Access: R/W
Access size: 8/16/32 bits

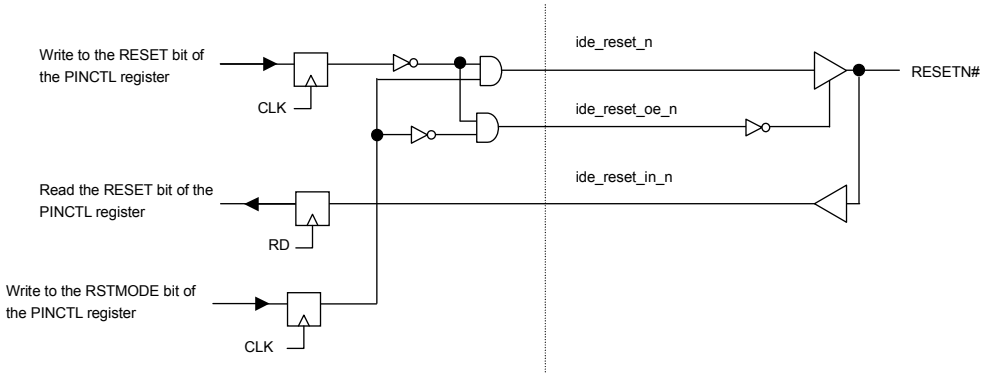
[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **RESET** (Reset pin signal) (Bit 0)
  When "1" is written to this bit, the IDE reset signal (RESETN#) is asserted (Low). When "0" is written to this bit, the IDE reset signal is deasserted (High). The actual pin value can be read by reading this bit. Therefore, when this bit is read for confirmation after "1" is written to this bit to assert (Low) the IDE reset signal, value "0" is read. Likewise, when this bit is read after "0" is written to this bit to deassert the IDE reset signal, value "1" is read.

RESET	Description
0	Write: The RESETN# signal is negated. (Set to High) Read: The RESETN# signal is Low.
1	Write: The RESETN# signal is asserted. (Set to Low) Read: The RESETN# signal is High.

(Note) The "IDE reset signal" refers to the reset signal (RESETN#) to the device. In this LSI, the IDE reset signal is configured as shown below, using the signals (ide_reset_in_n, ide_reset_n, ide_reset_oe_n) of the I/O buffer.



- **RSTMODE** (Reset pin drive mode) (Bit 1)
  This bit sets a drive method of the RESETN# signal.

RSTMODE	Description
0	Open collector
1	push-pull

25.2.2.17 IORDY Timer (IORDYTMR)

The IORDYTMR register is used for setting the IORDY timer.
Read/write operation is enabled for the IORDYTMR register by program control.
At reset, the register is set to 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IORDYTMR	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	IORDYTMR[17]	IORDYTMR[16]
At reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IORDYTMR[15]	IORDYTMR[14]	IORDYTMR[13]	IORDYTMR[12]	IORDYTMR[11]	IORDYTMR[10]	IORDYTMR[9]	IORDYTMR[8]	IORDYTMR[7]	IORDYTMR[6]	IORDYTMR[5]	IORDYTMR[4]	IORDYTMR[3]	IORDYTMR[2]	IORDYTMR[1]	IORDYTMR[0]
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x88
Access: R/W
Access size: 8/16/32 bits

[Note]
*: Reserved for future extension. In this LSI, "0" is read at read operation and the bits are ignored at write operation.

[Bit Description]

- **IORDYTMR[17:0]**(IORDY Timer configuration) (Bits 17-0)
  These bits set a timer value of the IORDY timer in a clock cycle. When the device waits for a signal by setting the IORDY signal to Low, the value of the IORDY timer counter is decremented.
  When the IORDY timer is set to "0" before the device sets the IORDY signal to High, an interrupt signal is asserted, the IORDYINT bit of the BMISP/BMISS register is set to "1", and the DIOW#/DIOR# signal is negated.
  When the value of this field is set to "0", the IORDY timer is disabled. The DIOW#/DIOR# signal is maintained in the Assert status as long as the IORDY signal is Low and no interrupt occurs.

25.3 Operation

25.3.1 PIO Transfer

25.3.1.1 Timing Mode

In PIO transfer, access is enabled in each mode (mode 0-4) by register setting.

Refer to "25.3.4 Timing Setting" for the method of setting each timing mode.

25.3.1.2 IORDY Signal

Sampling of the IORDY signal can be ignored by register setting.
Flow control by the IORDY signal is optional in PIO modes 0-2 and mandatory in modes 3-4. Set the control according to the mode used.

<IORDY timeout function>
A timeout value can be set using the IORDYTMR register. When IORDY timeout occurs, an interrupt signal (intr_n) to CPU is asserted. Refer to the specification of the IORDYTMR register and "25.3.3 Interrupt" for the IORDY timeout interrupt.

<Sampling point/recovery time of the IORDY signal>
When sampling of the IORDY signal is enabled (can be set by the IDETIMP/IDETIMS register), the sampling point and the recovery time can be set using the IDETIMP/IDETIMS register.

However, when "1" is set in the TIMORIDE bit of the TIMORIDE register, the setting of the REGSTB and REGRCVR registers influence the sampling point and the recovery time of the IORDY signal. The device checks the status of the IORDY signal after the clock value is set in the REGSTB register following assertion of the DIOW#/DIOR# signal. If the IORDY signal has not been asserted, the device waits for a signal (the DIOW#/DIOR# signal is kept asserted). That is, the strobe width of the DIOW#/DIOR# signal is set in the REGSTB register and the value also indicates the sample point of the IORDY signal. When the IORDY timer is set, the DIOW#/DIOR# signal is negated at the occurrence of timeout.
The REGRCVR register sets the recovery time for the DIOW#/DIOR# signal (negate of the DIOW#/DIOR# signal to assert of the next cycle), not the recovery time for the IORDY signal (from assert of the IORDY signal to assert of DIOW#/DIOR# of the next cycle).

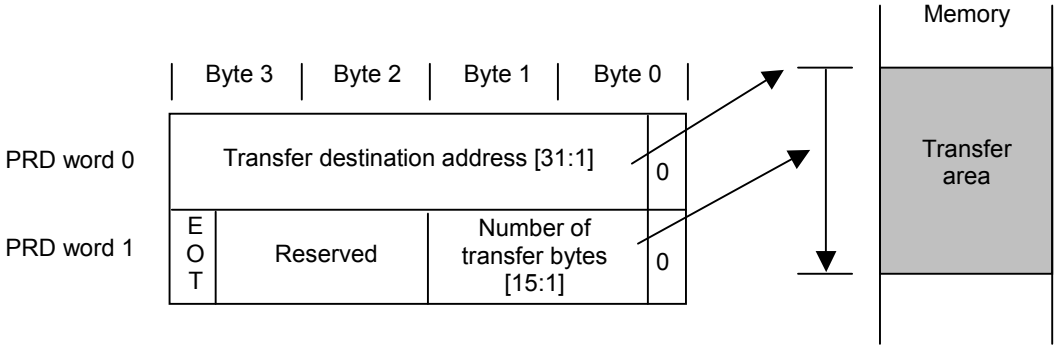
The setting of the PIO register (8-bit task file register) has been described above. For the PIO Data register, the setting of the DATSTB and DATRCVR registers influence the sampling point/recovery time of the IORDY signal.

25.3.2 Bus Master Function

At Multiword DMA transfer and UltraDMA transfer, this controller is used as an AHB bus master and performs DMA transfer between the AHB bus and the IDE bus. One bus master is available for each IDE channel (primary/secondary).
For performing DMA transfer, be sure to read the contents described in Section 25.3.4.

25.3.2.1 PRD (Physical Region Descriptor)

Control software must create a PRD (Physical Region Descriptor) table before starting DMA transfer. The PRD table comprises multiple PRD entries and is allocated in the main memory by the control software. PRD entries are descriptors that indicate areas in the main memory for which data transfer is to be executed. The format of PRD entries is shown below.



PRD word	Bit	Description
0	31-0	Transfer destination address (bit 0 is fixed to "0")
1	31	End of Descriptor Table
1	30-16	Reserved ("0")
1	15-0	Number of transfer bytes(Bit 0 is fixed to "0")

One PRD entry is expressed in 8 bytes and the bytes are segmented to the following three definitions.

(1) Transfer destination address (PRD word 0: Bits 31-0)
Indicates the first address of the transfer destination.

(2) Number of transfer bytes (PRD word 1: Bits 15-0)
Indicates the number of transfer bytes. The maximum transfer size is 64K bytes and value "0" indicates 64K bytes.

(3) End Of Descriptor Table (PRD word 1: Bit 31)
Indicates that this is the last entry of the PRD table. When this bit is "1", the entry is the last entry of the PRD table. When this bit is "0", there is a next PRD entry.

There are the following restrictions on location of PRD entries and the contents that are set in PRD.

(1) A PRD entry must be assigned on a 4-byte boundary in the main memory.
Assign each entry correctly on a 4-byte boundary by the control software. The first PRD address is set to the DMA descriptor pointer register (BMIDTPP/BMIDTPS), however, bits 1-0, which are in Read Only mode, are fixed to "00b".

(2) A PRD entry must not cross a 64K-byte boundary.
Assign each entry correctly by the control software.

(3) The transfer destination address must be an even number address.
Set an even number address correctly by the control software. Bit 0 of the transfer destination address must be "0".

(4) The number of transfer bytes must be an even number.
Set an even number of transfer bytes correctly by the control software. Bit 0 of the transfer size must be "0".

When DMA transfer is started, the controller reads the contents of PRD from the PRD address that is indicated by the DMA descriptor pointer register (BMIDTPP/BMIDTPS). Multiple PRD entries can be assigned. When the EOT bit of the PRD entry is "0", at completion of data transfer of the PRD entry, the controller automatically reads the PRD entry and continues data transfer. When the EOT bit in the PRD entry is "1", the controller terminates DMA operation at completion of data transfer for the PRD entry.

(Note) At termination of transfer of the entire data for the command, the device asserts an interrupt signal. After the last word data is written to the memory (or written out to the device), the controller sets the interrupt bit (INTRSTAT bit of BMISP/BMISS of the DMA status register) and clears the IDE active bit (IDEACT bit of BMISP/BMISS of the DMA status register).

(Note) Before the start of DMA transfer, assign PRD correctly to the address that is indicated by the DMA descriptor pointer register (BMIDTPP/BMIDTPS). Even though the CPU cache function has started, operation is not guaranteed unless PRD has been assigned in the memory.

(Note) Change of the contents of PRD corresponding to the DMA during DMA transfer is prohibited.

25.3.2.2 DMA Operation

This section describes operation of DMA transfer using PRD. A "single PRD mode" and a "multi PRD mode" are available for DMA transfer performed using PRD. This section describes the operation in "single PRD mode" using normal memory such as DRAM as the data transfer destination. The "single PRD mode" is used as the default mode of the controller. Refer to "25.3.2.3 Single PRD Mode and Multi PRD Mode" for description of a "Multi PRD mode" and differences between modes.

1) The control software allocates a PRD table on the memory.
   (Each PRD entry must be assigned observing the restrictions provided in 25.3.2.1 PRD (Physical Region Descriptor).)
2) The control software sets the address of the first PRD entry in the DMA descriptor pointer register (BMIDTPP/BMIDTPS).
   The DMA transfer direction is set in the DMADIR bit of the DMA control register (BMICP/BMICS).
   When the interrupt bit (INTRSTAT) and the error bit (DMAERROR) of the DMA control register (BMICP/BMICS) are set ON, the contents are cleared.
3) The control software issues a DMA transfer command to the device based on the ATA Standard.
4) When the control software sets the DMASTART bit of the DMA control register (BMICP/BMICS) to "1", the DMA function of the controller starts operation.
   The controller acquires the address of the first PRD entry and the number of transfer bytes from the DMA descriptor pointer register (BMIDTPP/BMIDTPS).
5) The controller executes DMA transfer for the DMA request from the device.
6) At termination of DMA transfer for one PRD entry, the controller reads the next PRD entry and executes DMA transfer in the same way. When the EOT bit of the PRD entry is set to "1" at completion of DMA transfer of a PRD entry, transfer of the entire data is completed.
7) At completion of transfer of the entire data for the DMA transfer command, an interrupt signal is asserted from the device.
8) The control software clears the Start/Stop bit (DMASTART) of the DMA control register (BMICP/BMICS). Then, the software checks for the occurrence of an error by reading the DMA status register (BMISP/BMISS).

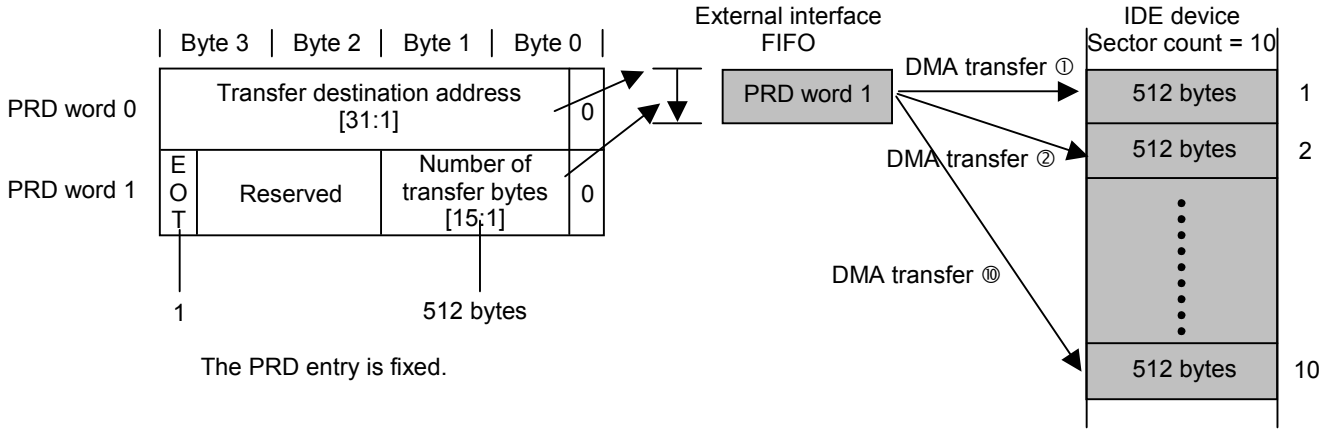
25.3.2.3 Single PRD Mode and Multi PRD Mode

In normal DMA transfer, the total transfer size that is indicated by a set of PRD entry groups and the data transfer size of the IDE device (indicated by the sector counter) are equal. That is, the entire data transfer size of the IDE device is transferred by one DMA transfer activation.
For instance, assume that the sector counter indicates 10 as the data transfer size of the IDE device. In this case, since the data transfer size is 5120 bytes, the data transfer size that is indicated by PRD is set to 5120 bytes. Even if multiple PRD entries are used, the value is set so that the transfer size becomes 5120 bytes. At completion of data transfer of 5120 bytes after the start of DMA transfer, the IDE device generates an interrupt and on receiving the interrupt signal, the controller notifies a data transfer completion interrupt to the CPU.
Such transfer mode is called a "single PRD mode" in the sense that a set of PRD entry groups (EOT=1 in only the last PRD) is used by one DMA transfer. When using normal RAM as the DMA transfer destination, use a single PRD mode.

For alternate DMA transfer, assume FIFO that is incorporated in the external interface circuit as the DMA transfer destination. In this case, access is not always enabled to FIFO, which is the DMA transfer destination. This controller does not support flow control by a wait request from a DMA transfer destination. Therefore, if DMA transfer is executed with a size greater than the FIFO of the DMA transfer destination, overflow/underflow occurs. The size of each DMA transfer must not exceed the FIFO size of the transfer destination.
In this case, DMA transfer may be executed multiple times for one ATA command. For instance, when the sector count of the ATA command is 10 and the FIFO size of the DMA transfer destination is 512 bytes, DMA transfer is executed ten times. Processing that executes DMA transfer multiple times for one ATA command is called a "multi PRD mode" in the sense that multiple PRD entry groups are used. When using FIFO mainly as a DMA transfer destination, use a multi PRD mode.

<Setting a multi PRD mode>
To set this controller to a multi PRD mode, set the MULTPRDT bit of the DMAFEAT register to "1". When "1" is set in the EOTINTEN bit of the DMAFEAT register, an interrupt signal is asserted in the CPU at termination of DMA transfer even if an interrupt signal from the IDE signal is not asserted. ("Termination of DMA transfer" refers to termination of DMA transfer for the PRD whose EOT bit is "1".) When the EOTINTEN bit of the DMAFEAT register is "0", only termination of DMA transfer does not cause assertion of an interrupt signal if an interrupt signal from the IDE device is not asserted. The EOTINT bit of the BMISP/BMISS register will not be set to "1" either. In this case, termination of DMA transfer can be checked by polling the IDEACT bit of the DMA status register (BMISP/BMISS).

<Example of processing in multi PRD mode>
This example introduces a method of processing where FIFO of 512 bytes is used for the external interface that is prepared separately as the DMA transfer destination and sector count = 10 is applied as the data transfer size of the IDE device. Assume that the MULTPRDT bit and the EOTINTEN bit of the DMAFEAT register have been set to "1" at initialization of the controller.



(1) The control software allocates a PRD table in the memory.
(Each PRD entry must be assigned complying with restrictions indicated in 25.3.2.1, "PRD (Physical Region Descriptor.")
In this example, the address of FIFO of the external address is set as the "transfer destination address", a size of 512 bytes is set as the "number of transfer bytes", and "1" is set as the "EOT" bit.

(2) The control software sets the address of the first PRD entry in the DMA descriptor pointer register (BMIDTPP/BMIDTPS).
The DMA transfer direction is set in the DMADIR bit of the DMA control register (BMICP/BMICS).
When the interrupt bit (INTRSTAT) and the error bit (DMAERROR) of the DMA control register (BMICP/BMICS) are set ON, the contents are cleared.

(3) The control software issues a DMA transfer command to the device based on the ATA Standard.
In this example, DMA transfer command is issued based on the sector count = 10.

(4) When the control software sets the DMASTART bit of the DMA control register (BMICP/BMICS) to "1", the DMA function of the controller starts operation.
The controller acquires the address of the first PRD entry and the number of transfer bytes from the DMA descriptor pointer register (BMIDTPP/BMIDTPS).

(5) The controller executes DMA transfer for the DMA request from the device.

(6) At termination of DMA transfer for one PRD entry, the controller reads the next PRD entry and executes DMA transfer in the same way. When the EOT bit of the PRD entry is "1" at completion of a PRD entry, this means that transfer of the entire data is completed.
In this example, since the EOT bit of PRD is "1", DMA transfer terminates with one PRD. At completion of DMA transfer, this controller generates an interrupt for the CPU. After checking that no error occurred by reading the DMA status register (BMISP/BMISS), the control software clears the interrupt by writing "1" in the EOTINT bit.
(When the EOTINTEN bit of the DMAFEAT register is "0", no interrupt occurs at termination of DMA transfer. In this case, check termination of DMA transfer by polling the DMA status register (BMISP/BMISS).)

DMA transfer ① in the diagram above is completed. Since the sector count is 10 in this example, DMA transfer is executed for 10 sectors by repeating the steps 1) and 2) and 4) to 6).

(7) At completion of transfer of the entire data for the DMA transfer command, an interrupt signal is asserted from the device.
This example shows that assertion of the interrupt from the device is indicated by setting the INTRSTAT bit of the DMA status register (BMISP/BMISS) to "1". That is, DMA transfer has been executed 10 times and the entire data transfer for the ATA command has been completed.
(Note) Since a multi PRD mode is applied, an interrupt is generated from the DMA controller at termination of 1 time of DMA transfer. Therefore, the interrupt signal (INTRQ) of the device caused by the termination of the command may not have been asserted at assertion of an interrupt signal. In this case, check assertion of the interrupt signal (INTRQ) of the device by polling the INTRSTAT bit of the status register (BMISP/BMISS).

(8) The control software clears the start/stop bit (DMASTART) of the DMA control register (BMICP/BMICS). Then, the software checks for the occurrence of an error by reading the DMA status register (BMISP/BMISS).

(Note) It is prohibited to clear the start/stop bit (DMASTART) of the DMA controller register at the start of the next DMA in Step 6). This bit can be cleared only when the entire data transfer for the ATA command is completed (Step 8).

(Note) Step 6 indicates "repetition of steps 1) and 2) and 4) to 6) for 10 sectors" to execute DMA transfer. However, the PRD table and the address setting can be omitted if they have not been changed from the previous DMA transfer.

25.3.2.4 Cautions Regarding DMA Transfer

1. Location of the PRD table

When locating the PRD table in the internal SRAM (SHBRAM), do not cross the 1KB boundaries (boundaries to divide the SRAM in units of 1KB starting from the first address).

2. DMA transfer in single PRD mode / single table

PRD transfer should be in units of 512 bytes. When designating the AHBRAM as the transfer destination, do not allocate the transfer domain in excess of the 1KB boundary.

3. DMA transfer in single PRD mode / multi table and also in multi PRD mode

3-A. When the transfer destination is an external memory

Transfer should be in units of 512 bytes, and for each table, allocate the transfer domain having the transfer size plus 3 words or more.

3-B. When the transfer destination is the USB-FIFO

The beginning of the address space designated as an end point must be specified as a DMA transfer destination address (address to be specified to the PRD), and the transfer size should be 512 bytes. For example, specify the address 0x7BB0_1000 for EPa.

3-C. When the transfer destination is the internal SRAM (SHBRAM)

The internal SRAM (AHBRAM) cannot be specified as a transfer designation. When transferring data to the internal SRAM (SHBRAM), use single PRD mode / single table.

25.3.3 Interrupt

This controller generates an interrupt to CPU in the following four cases. A CPU interrupt signal (intr_n) is asserted/negated synchronized with the AHB clock (hclk signal).

1) Termination of DMA transfer

<Single PRD mode>

In single PRD mode, a CPU interrupt signal (intr_n) is asserted when the following conditions are satisfied after execution of DMA transfer.

- The device asserted an IDE interface interrupt signal (INTRQ) and
- The last data in the DMA transfer that was activated is written to the memory or device (varies according to the transfer direction).

That is, at assertion of the controller interrupt signal, completion of the transfer of the entire data that is stored in the controller buffer is guaranteed.
In addition to assertion of an interrupt signal, the INTRSTAT bit of the BMISP/BMISS register is set to "1".
The interrupt can be cleared by writing "1" to the INTRSTAT bit. When "1" is written to the INTRSTAT bit, a CPU interrupt signal (intr_n) is negated. A CPU interrupt signal (intr_n) can be negated also by writing "1" to the INTRSTAT bit while the device interrupt signal (INTRQ) is in Assert status, disabling any further interrupt. (A device interrupt signal (INTRQ) can be negated by reading the status register of the device. A CPU interrupt signal (intr_n) can also be negated by negating a device interrupt signal (INTRQ) by reading the status while a CPU interrupt signal (intr_n) is asserted. In this case, the INTRSTAT bit is not cleared and remains "1".)

<Multi PRD mode>

In multi PRD mode, a CPU interrupt signal (intr_n) is asserted when the following condition is satisfied after execution of DMA transfer.

- The last data in the DMA transfer that was activated is written to the memory or the device (varies according to the transfer direction).

By setting the EOTINTEN bit and the MULTIPRDT bit of the DMAFEAT register to "1", a multi PRD mode can be set, enabling assertion of a CPU interrupt signal (intr_n) at termination of DMA transfer even if an interrupt signal (INTRQ) from the device is not asserted. In addition to assertion of an interrupt signal, the EOTINT bit of the BMISP/BMISS register is set to "1".
An interrupt can be cleared by writing "1" to the EOTINT bit. When "1" is written to the EOTINT bit, a CPU interrupt signal (intr_n) is negated.

Refer to "25.3.2 Bus Master Function" for description of a single PRD mode and multi PRD mode.

2) Termination of PIO transfer

- The device asserted an IDE interface interrupt signal (INTRQ).

When receiving an interrupt signal (INTRQ) from the device, this controller asserts an interrupt signal (intr_n) to the CPU. In PIO transfer, the INTRSTAT bit of the BMISP/BMISS register remains "0", without being set to "1".
The interrupt can be cleared by reading the status register of the device. When the status register is read, the device negates the interrupt signal (INTRQ). Consequently, this controller negates the CPU interrupt signal (intr_n).

3) IORDY timeout
   - IORDY timeout occurred.

   When IORDY timeout occurs while the IORDY timer is set by the IORDYTMR register, a CPU interrupt signal (intr_n) is asserted. (The DIOW#/DIOR# signal is negated.)
   In addition to assertion of the interrupt signal, the IORDYINT bit of the BMISP/BMISS register is set to "1".
   The interrupt can be cleared by writing "1" to the IORDYINT bit. When "1" is written to the IORDYINT bit, a CPU interrupt signal (intr_n) is negated.

4) Upon reception of an error response
   - When an AHB ERROR response is received from the AHB slave during DMA transfer

   When an AHB ERROR response is received from the AHB slave that is the transfer mate during DMA transfer, this controller stops DMA transfer and asserts the CPU interrupt signal (intr_n). If the interrupt condition is satisfied, an interrupt is generated regardless of being in single PRD mode or being in Multi-PRD mode.
   In addition to assertion of the interrupt signal, the HERRINT bit of the BMISP/BMISS register is set to "1".
   The interrupt can be cleared by writing "1" to the HERRINT bit. When "1" is written to the HERRINT bit, a CPU interrupt signal (intr_n) is negated.

   If this interrupt is generated, reset this controller by hardware. See also Section 25.3.5.3, "Notes on Use" for the reception of the ERROR response.

The following table shows interpretation of the DMA status register (BMISP/BMISS) at assertion of an interrupt.

Mode	HERRINT	EOTINT	IORDYINT	INTRSTAT	IDEACT	Description
DMA transfer (Singe PRD mode)	0	0	0	1	0	The IDE device asserted an interrupt signal (INTRQ) and DMA transfer is completed. Termination status where the size of the physical memory area indicated by PRD and the transfer size of the IDE device are equal.
DMA transfer (Single PRD mode)	0	0	0	1	1	Although the IDE device asserted an interrupt signal (INTRQ), DMA transfer is not completed. Termination status when the size of the physical memory area indicated by PRD is greater than the transfer size of the IDE device
DMA transfer (Multi PRD mode)	0	1	0	1	0	The IDE device asserted an interrupt signal (INTRQ) and DMA transfer is completed. Normal termination status where the size of the physical memory area indicated by PRD and the remaining transfer size of the IDE device are equal when the EOTINTEN bit of the DMAFEAT register is "1". (Status at termination of the last DMA for the ATA command)
DMA transfer (Multi PRD mode)	0	1	0	0	0	Although the IDE device has not asserted an interrupt signal (INTRQ), DMA transfer is completed. Termination status where the size of the physical memory area indicated by PRD is smaller than the remaining transfer size of the IDE device when the EOTINTEN bit of the DMAFEAT register is "1". (Status where there is a remaining transfer size of the IDA device at DMA transfer for the ATA command).
DMA transfer (Single PRD mode/Multi PRD mode)	1	0	0	0	0	An AHB ERROR response has been received from the AHB slave that is the transfer mate during DMA transfer, and DMA transfer is stopped.
PIO transfer	0	0	0	0	0	An interrupt occurred due to assertion of an interrupt signal (INTRQ) of the device. The interrupt status bit is not set.
PIO transfer (IORDYTMR is enabled)	0	0	1	0	0	IORDY timeout occurred in PIO transfer.

25.3.4 Setting Timing

The timing register of an IDE interface must be set appropriately according to the transfer mode (PIO mode 0-4, Multiword DMA mode 0-2, UltraDMA mode 0-4).
Two methods are available for setting timing according to the setting of the TIMORIDE bit of the TIMORIDE register. Select a suitable method according to the use environment.

1) TIMORIDE bit is 0 (Disabled): Refer to 25.3.4.1 Setting Basic Timing (TIMORIDE register is disabled).
2) TIMORIDE bit is 1 (Enabled): Refer to 25.3.4.2 Setting Detail Timing (TIMORIDE register is enabled).

When using the setting method described in "1) Setting Basic Timing", use the value set in the controller for each timing parameter. This value is determined based on the IDE clock (ide_clk signal) frequency of 66 MHz. Since the frequency of the IDE clock is 60 MHz in this LSI, the timing such as the strobe width changes according to the frequency proportion (the strobe width increases).

When using the setting method described in "2) Setting Detail Timing", each timing parameter can be set in the register. In this case, the parameters that are set in the controller are not used. Therefore, when, like this LSI, the frequency of the IDE clock is 60 MHz, set each timing register to set an optimum strobe width and so on.

25.3.4.1 Setting Basic Timing (TIMORIDE register is disabled)

Use the following registers to set the timing of the IDE interface (the TIMORIDE bit of the TIMORIDE register is "0").

- IDETIMP/IDETIMS (primary/secondary IDE timing control register)
  - DMAFTIM0/DMAFTIM1 and PIOFTIM0/PIOFTIM1 bits
  - RDYSMPL and RDYRCVRY fields
  - SLBTIMEN bit
- SIDETIM (slave IDE timing enable register)
- UDMATIM (UltraDMA timing control register)

Select a timing setting method of each transfer mode using the DMAFTIM0/DMAFTIM1 and PIOFTIM0/PIOFTIM1 bits of the IDETIMP/IDETIMS register.
The table below shows the differences of timing setting methods according to the combination of the DMAFTIMx bit and the PIOFTIMx bit. When setting the timing according to each transfer mode in PIO transfer and Multiword DMA transfer, set DMAFTIMx = 0 and PIOFTIMx = 1. Set the actual timing in the RDYSMPL and RDYRCVRY fields.

IDETIMP/IDETIMS		PIO	PIO	Multiword DMA	UltraDMA
DMAFTIM	PIOFTIM	Other than the Data register	Data register		
0	0	The settings of RDYSMPL and RDYRCVRY are valid.	Fixed to mode 0	Fixed to mode 0	The setting of UDMATIM register is valid.
0	1	The settings of RDYSMPL and RDYRCVRY are valid.	The settings of RDYSMPL and RDYRCVRY are valid.	The settings of RDYSMPL and RDYRCVRY are valid.	The setting of UDMATIM register is valid.
1	0	The settings of RDYSMPL and RDYRCVRY are valid.	Fixed to mode 0	Fixed to mode 0	The setting of UDMATIM register is valid.
1	1	The settings of RDYSMPL and RDYRCVRY are valid.	Fixed to mode 0	The settings of RDYSMPL and RDYRCVRY are valid.	The setting of UDMATIM register is valid.

The following table shows general settings in the RDYSMPL and RDYRCVRY fields in each transfer mode.

	PIO		DMA		UltraDMA
Register	IDETIMx		IDETIMx		UDMATIM
Field	RDYSMPL	RDYRCVRY	RDYSMPL	RDYRCVRY	TCYCxx
Mode 0	0	0	0	0	0
Mode 1	1	0	1	1	1
Mode 2	2	1	2	2	2
Mode 3	3	2	—	—	3
Mode 4	3	3	—	—	4

<Switching setting in each transfer mode>
As shown above, under the setting of DMAFTIMx = 0 and PIOFTIMx = 1, the timing in each transfer mode (PIO, PIO data, and multiword DMA) can be set in the RDYSMPL and RDYRCVRY fields.
However, since only one set of the RDYSMPL and RDYRCVRY fields is available, the fields must be set again when the timing in each transfer mode is changed. For instance, assume execution of the READ DMA command using mode 0 for accessing the PIO task file register (register other than the Data register) and mode 2 for multiword DMA transfer. In this case, the timing must be set each time using the following procedure.

(1) Set the RDYSMPL and RDYRCVRY fields for accessing the PIO task file register according to mode 0.
↓
(2) Access each task file register (write the READ DMA command to the command register after writing each parameter of the command).
↓
(3) Set the RDYSMPL and RDYRCVRY fields for multiword DMA transfer according to mode 2.
↓
(4) Start DMA.

<Setting the timing of device 1 (slave)>
For this LSI, device 1 (slave) cannot be used. So the following descriptions are not needed.
When the SLVTIMEN bit of the IDETIMP/IDETIMS register is set to "0", the settings of the RDYSMPL and RDYRCVRY fields of the IDETIMP/IDETIMS register become valid for both device 0 (master) and device 1 (slave).
When the SLVTIMEN bit is set to "1", the settings of the RDYSMPL and RDYRCVRY fields become valid for device 0 (master) only. Set the timing for device 1 (slave 1) in the SIDETIM register in the same way.

The differences according to the value set in the SLVTIMEN bit are provided below.

- The SLVTIMEN bit is "0"
  Setting timing of device 0 (master): The setting of the RDYSMPL and RDYRCVRY fields of the IDETIMP/IDETIMS registers are valid.
  Setting timing of device 1 (slave): The setting of the RDYSMPL and RDYRCVRY fields of the IDETIMP/IDETIMS registers are valid (same as device 0)

- SLVTIMEN bit is "1"
  Setting timing of device 0 (master): The setting of the RDYSMPL and RDYRCVRY fields of the IDETIMP/IDETIMS registers are valid
  Setting timing of device 1 (slave): The setting of the RDYSMPL and RDYRCVRY fields of the SIDETIM register are valid (set each in the primary and secondary).

This controller stores access to the Device/Head register of the previous ATA register as the condition for using the timing setting of device 1 (slave). Use the timing setting of device 1 (slave) for the access after "1" is written to the DEV bit of the Device/Head register. Refer to "25.3.4.3 Information on Timing Setting of Device 1 (Slave)" for details.

25.3.4.2 Setting Detail Timing (TIMORIDE register is enabled)

This section describes the method of setting the timing when "1" is set in the TIMORIDE bit of the TIMORIDE register. The following example shows the setting of timing when the frequency of the IDE clock (ide_clk signal) is 60 MHz.

<Enabling the TIMORIDE mode>
By setting 1 in the TIMORIDE bit of the TIMORIDE register, the TIMORIDE mode can be enabled. The DMAFTIM0/1 and PIOFTIM0/1 fields of the IDETIMP/IDETIMS register must also be set appropriately. The following table summarizes the settings required for enabling the TIMORIDE mode.

Register	TIMORIDE	IDETIMP/IDETIMS	
Field	TIMORIDE	DMAFTIM	PIOFTIM
Setting value	1	0	1

Setting for enabling the TIMORIDE mode

When each resgister is set as shown above, a TIMORIDE mode is set, enabling each timing setting register. Each timing can be set in the timing setting register in the clock count. The timing setting registers refer to the following:

REGSTB (8-bit register access strobe width)
REGRCVR (8-bit register access recovery time)
DATSTB (Data register access strobe width)
DATRCVR (Data register access recovery time)
DMASTB (DMA access strobe width)
DMARCVR (DMA access recovery time)
UDMASTB (UltraDMA access strobe width)
UDMATRP (UltraDMA ready-to-pause time)
UDMATENV (UltraDMA t_{ENV} timing parameter)

(1) PIO mode (other than Data register)
An Assert time of the DIOW#/DIOR# signal can be set in the REGSTB register.
A recovery time (from negate to assert) of the DIOW#/DIOR# signal can be set in the REGRCVR register.
Set the time appropriately according to the AC timing of the ATA Standard. The following table shows an example of timing setting.

	PIO mode (other than the Data register)		ATA Standard (min)		
Register	REGSTB	REGRCVR	Pulse width	Recovery time	Cycle time
Field	REGSTB0/1[6-0]	REGRCVR0/1[6-0]	t_2 (ns)	t_{2i} (ns)	t_0 (ns)
Mode 0	0x12 (19 clocks) ≅ 304 ns	0x12 (19 clocks) ≅ 304 ns	290	—	600
Mode 1	↑	0x05 (6 clocks) ≅ 96 ns	290	—	383
Mode 2	↑	0x01 (2 clocks) ≅ 32 ns	290	—	330
Mode 3	0x05 (6 clocks) ≅ 96 ns	0x06 (7 clocks) ≅ 112 ns	80	70	180
Mode 4	0x04 (5 clocks) ≅ 80 ns	0x02 (3 clocks) ≅ 48 ns	70	25	120

(Note) The values in the table are reference values provided as the guideline of the specification.
(Note) Field of the REGSTB register: Set the REGSTB0 and REGSTB1 fields for each of the master and slave devices.
(Note) Field of the REGRCVR register: Set the REGRCVR0 and REGRCVR1 fields for each of the master and slave devices.

(Note) The time of the setting value is calculated based on 1 clock = 16 ns for simplification. The value that is set will be (cycle time – 1). (Refer to the specification of each register for details.)
(Note) The recovery time of mode 0-2 is not specified. The setting value is calculated based on (cycle time – pulse width). When setting a different value for the pulse width, set the recovery time accordingly also.
(Note) Set the value for mode 3-4 so that it satisfies the Standard of cycle time. Therefore, if the pulse width and recovery time are set to the values very close to the standard values, the standard of the cycle time cannot be satisfied.

(2) PIO mode (Data register)
An assert time of the DIOW#/DIOR# signal can be set in the DATSTB register.
A recovery time (from negate to assert) of the DIOW#/DIOR# signal can be set in the DATRCVR register.
Set the time appropriately according to the AC timing of the ATA Standard. The following table shows an example of timing setting.

	PIO mode (Data register)		ATA Standard (min)		
Register	DATSTB	DATRCVR	Pulse width	Recovery time	Cycle time
Field	DATSTB0/1 [6-0]	DATRCVR0/1 [6-0]	t_2(ns)	t_{2i}(ns)	t_0(ns)
Mode 0	0x0A(11 clocks) ≅ 176 ns	0x1A(27 clocks) ≅ 432 ns	165	—	600
Mode 1	0x07(8 clocks) ≅ 128 ns	0x10(17 clocks) ≅ 272 ns	125	—	383
Mode 2	0x06(7 clocks) ≅ 112 ns	0x08(9 clocks) ≅ 144 ns	100	—	240
Mode 3	0x05 (6 clocks) ≅ 96 ns	0x06(7 clocks) ≅ 112 ns	80	70	180
Mode 4	0x04(5 clocks) ≅ 80 ns	0x02(3 clocks) ≅ 48 ns	70	25	120

(Note) The values in the table are reference values provided as the guideline of the specification.
(Note) Field of the DATSTB register: Set the DATSTB0 and DATSTB1 fields for each of the master and slave devices.
(Note) Field of the DATRCVR register: Set the DATRCVR0 and DATRCVR1 fields for each of the master and slave devices.
(Note) The time of the setting value is calculated based on 1 clock = 16 ns for simplification. The value that is set will be (cycle time – 1). (Refer to the specification of each register for details.)
(Note) The recovery time of mode 0-2 is not specified. The setting value is calculated based on (cycle time – pulse width). When setting a different value for the pulse width, set the recovery time accordingly also.
(Note) Set the value for mode 3-4 so that it satisifies the Standard of cycle time. Therefore, if the pulse width and recovery time are set to the values very close to the standard values, the standard of the cycle time cannot be satisfied.

(3) Multiword DMA mode

An assert time of the DIOW#/DIOR# signal can be set in the DMASTB register.
A recovery time (from negate to assert) of the DIOW#/DIOR# signal can be set in the DMARCVR register.
Set the time appropriately according to the AC timing of the ATA Standard. The following table shows an example of timing setting.

	Multiword DMA mode		ATA Standard (min)			
Register	DMASTB	DMARCVR	Pulse width	DIOR# Negate pulse width	DIOW# Negate Hold time	Cycle time
Field	DMASTB0/1 [6-0]	DMARCVR0/1 [6-0]	t_2(ns)	t_{Kr}(ns)	t_{Kw}(ns)	t_0(ns)
Mode 0	0x0D (14 clocks) ≅ 224 ns	0x10 (17 clocks) ≅ 272 ns	215	50	215	480
Mode 1	0x05 (6 clocks) ≅ 96 ns	0x04 (5 clocks) ≅ 80 ns	80	50	50	150
Mode 2	0x04 (5 clocks) ≅ 80 ns	0x02 (3 clocks) ≅ 48 ns	70	25	25	120

(Note) The values in the table are reference values provided as the guideline of the specification.
(Note) Field of the DMASTB register: Set the DMASTB0 and DMASTB1 fields for each of the master and slave devices.
(Note) Field of the DMARCVR register: Set the DMARCVR0 and DMARCVR1 fields for each of the master and slave devices.
(Note) The time of the setting value is calculated based on 1 clock = 16 ns for simplification. The value that is set will be (cycle time – 1). (Refer to the specification of each register for details.)
(Note) Set the value so that it satisifies the Standard of cycle time. Therefore, if the DIOR# negate pulse width and the DIOW# Negate Hold time are set to the values very close to the standard values, the standard of the cycle time cannot be satisfied.

(4) UltraDMA mode

A strobe width at UltraDMA Data Out transfer (Write) can be set in the UDMASTB register. The DIOR# signal is used for the strobe.
A ready-to-pause time (time from negate of the HDMARDY# signal to assert of the STOP signal) at UltraDMA Data In transfer (Read) can be set in the UDMATRP register.
The t_{ENV} parameter of the ATA Standard (from assert of the DMACK# signal to negate of the STOP signal or assert of the HDMARDY# signal) can be set in the UDMATENV register.
Set the time appropriately according to the AC timing of the ATA Standard. The following table shows an example of timing setting.

<UDMASTB register>

	UltraDMA mode	ATA Standard (min)
Register	UDMASTB	Cycle time
Field	UDMASTB0/1 [3-0]	t_{RP}(ns)
Mode 0	0x07(8 clocks) ≅ 128 ns	112
Mode 1	0x04(5 clocks) ≅ 80 ns	73
Mode 2	0x03(4 clocks) ≅ 64 ns	54
Mode 3	0x02(3 clocks) ≅ 48 ns	39
Mode 4	0x01(2 clocks) ≅ 32 ns	25

(Note) The values in the table are reference values provided as the guideline of the specification.
(Note) Field of the UDMASTB register: Set the UDMASTB0 and UDMASTB1 fields for each of the master and slave devices.
(Note) The time of the setting value is calculated based on 1 clock = 16 ns for simplification. The value that is set will be (cycle time – 1). (Refer to the specification of each register for details.)

<UDMATRP Register>

	UltraDMA mode	ATA Standard (min)
Register	UDMATRP	Ready-to-pause
Field	UDMATRP0/1 [4-0]	t_{RP}(ns)
Mode 0	0x0A(11 clocks) ≅ 176 ns	160
Mode 1	0x07(8 clocks) ≅ 128 ns	125
Mode 2	0x06(7 clocks) ≅ 112 ns	100
Mode 3	↑	100
Mode 4	↑	100

(Note) The values in the table are reference values provided as the guideline of the specification.
(Note) Field of the UDMATRP register: Set the UDMATRP0 and UDMATRP1 fields for each of the master and slave devices.
(Note) The time of the setting value is calculated based on 1 clock = 16 ns for simplification. The value that is set will be (cycle time – 1). (Refer to the specification of each register for details.)

<UDMATENV register>

	UltraDMA mode	ATA Standard (min-max)
Register	UDMATENV	Envelope time
Field	UDMATENV0/1 [3-0]	t_{ENV}(ns)
Mode 0	0x02(3 clocks) ≅ 48 ns	20-70
Mode 1	↑	20-70
Mode 2	↑	20-70
Mode 3	0x01(2 clocks) ≅ 32 ns	20-55
Mode 4	↑	20-55

(Note) The values in the table are reference values provided as the guideline of the specification.
(Note) Field of the UDMATENV register: Set the UDMATENV0 and UDMATENV1 fields for each of the master and slave devices.
(Note) The time of the setting value is calculated based on 1 clock = 16 ns for simplification. The value that is set will be (cycle time – 1). (Refer to the specification of each register for details.)

25.3.4.3 Information on Setting Timing of Device 1 (Slave)

For this LSI, device 1 (slave) cannot be used. So the following descriptions are not needed.
As described in 25.3.4.1, "Setting Basic Timing (TIMORIDE register is disabled)" and 25.3.4.2, "Setting Detail Timing (TIMORIDE register is enabled)," this controller controls the timing of the IDE interface for device 1 (slave) separately from device 0 (master). The controller monitors write operation to the Device/Head register of the ATA register to determine whether device 0 (master) timing or device 1 (slave) timing is used for accessing the IDE device. That is, the controller stores the value written to the DEV bit of the Device/Head register in the internal section and determines the timing for accessing the IDE device according to the value.

The timing of device 0 (master) is used for accessing the IDE device after "0" is written to the DEV bit of the Device/Head register. The timing set in device 0 (master) is also used for accessing the Device/Head register for writing "0" in the DEV bit.
The timing set for device 1 (slave) is used for accessing the IDE device after "1" is written to the DEV bit of the Device/Head register. The timing set for device 1 (slave) is also used for accessing the Device/Head register for writing "1" to the DEV bit.
For read operation of the Device/Head register, the value of the DEV bit that was written previously determines whether the timing of device 0 (master) or device 1 (slave) is used.

25.3.5 AHB Interface

The AHB master function and the AHB slave function are available as the CPU interfaces. External connection of the AHB arbiter and the decoder is assumed.

25.3.5.1 AHB Master Interface

The functions of the AHB master interface are described below.

1) Supports a burst mode.
   Single transfer
   Undefined length increment burst
   4-beat increment burst
   8-beat increment burst
2) The entire burst transfer that crosses a 1K byte boundary is divided into two undefined length increment burst segments.
3) 16/32-bit transfer (HSIZE = 001 or 010)
4) No BUSY cycle (HTRANS = 01 does not occur)
5) HPROT is not supported.
6) Supports OKAY/SPLIT/RETRY response
7) HLOCK is asserted at fixed-length burst.

25.3.5.2 AHB Slave Interface

The functions of the AHB slave interface are described below.

1) The burst mode is not supported (HBURST = 000).
2) 8/16/32-bit transfer (HSIZE = 000, 001, or 010)
3) The SEQ cycle is handled in the same way as the NONSEQ cycle.
4) HPROT is not supported.
5) Supports OKAY response only (the response is always HRESP = 00)

25.3.5.3 Notes on Use

(1) About error response reception during operation as the AHB master

During DMA operation, this controller acts as the AHB master. When an error response is received as the AHB bus response from the AHB slave that is the transfer mate while this controller is operating as the AHB master, this controller asserts the CPU interrupt signal (intr_n). And at the same time, the HERRINT bit of the BMISP/BMISS register is set to "1", indicating that an error response has been received.

In general, if a situation occurs that an error response is received, the address setting of the DMA transfer destination is considered erroneous. One of such cases is that such an area that generates an error response as a default slave has been specified because of a software bug. Set the correct DMA transfer destination so as not to specify an error response generating area.

If an error response is received (if an error response interrupt occurs), this controller stops DMA transfer. Reset this controller by hardware if an error response is received. In addition, since this also means a stoppage of data transfer, perform recovery processing such as resetting the device as well as the controller.

(2) About software manipulation during DMA operation

An access to the ATA register during DMA operation is not supported. When accessing the ATA register, be sure to do it after completion of DMA transfer.

Also, do not change any value of the registers of this controller during DMA operation. Operation cannot be guaranteed if it is changed during DMA operation. It is possible, however, to read any register of this controller during DMA operation.

(3) About clock control during DMA operation

During DMA operation, never stop the HCLOCK (hclk signal) of AHB or the IDE core clock (ide_clk signal) or change their frequency. Operation cannot be guaranteed if the clocks are stopped or their frequency is changed during DMA operation.

(4) About AMBA power management mode during DMA operation

When this controller is connected to a system bus such as μPLATD, never change its power management mode during DMA operation. Changing power management mode means a transition to the WFI, HALT, or STOP mode. Operation cannot be guaranteed if the power management mode is changed during DMA operation.

25.3.6 Notes on Connecting the ATA Interface

The ATA interface of the ML696201/69Q6203 may cause an unexpected increase in supply current caused by driving or leaving the ATA interface bus floating.
The data bus of the ATA interface comprises a bi-directional port and when the bus is not accessed by either this controller or the hard disk, the bus goes into the floating state. When the voltage level of the floating pin reaches a value close to the intermediate level of the power supply voltage, the power supply current increases due to the current penetrating the input circuit.

To prevent such penetration current, it is recommended to connect pull-down resistors to the data bus of the ATA interface. Connect pull-down resistors of 1MΩ to the IDED[15:8] and IDED[6:0] pins. A pull-down resistor of 10MΩ is connected to the IDED[7] pin according to the ATA Standard.

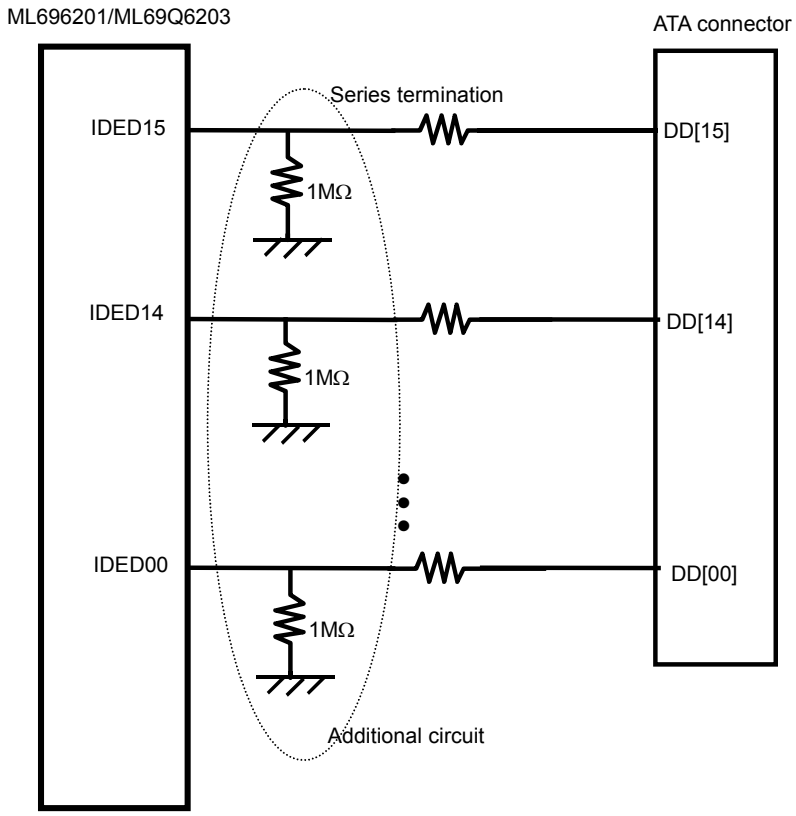


Figure 25-2 Example of Pull-Down Resistor Connections (Circuit for Guideline)

Chapter 26

HS USB

Chapter 26 HS USB

26.1 Overview

This LSI has a built-in high-speed USB module to provide an interface with Universal Serial Bus (USB) 2.0.

The USB module consists of a high-speed USB link module and a high-speed USB PHY module (physical layer) and supports USB2.0, the high-speed mode (480 Mbps), and USB1.1, the full-speed mode (12 Mbps).
It has seven embedded endpoints, and can support control transfer, interrupt transfer, and bulk transfer. The automatic data transfer control allows high speed transfer. For the automatic data transfer control, refer to Chapter 11.

The module is comprised of the following five blocks:
1) AMBA AHB interface block that can be connected to the CPU
2) Control register block
3) 4K-byte FIFO memory for packet data transfer
4) Serial Interface Engine Block
5) High-speed/full-speed USB-PHY transceiver (physical layer)

FEATURES:

- Can achieve a link function and a transceiver function (physical layer) supporting the USB 2.0 (and USB 1.1 supported) standard.

- Can be connected with a high-speed (480 Mbps)/full-speed (12 Mbps) USB transceiver.
  Note 1: Note that the LSI supply voltage should be limited in the USB connection. For details, refer to Chapter 30, Electrical Characteristics.
  Note 2: The Isochronous transfer cannot be used.

- Supports the master storage class.

- Supports six programmable endpoints (endpoint 0 is exclusively used for control transfer).

- The processing of standard device requests (other than GET_DESCRIPTOR, SET_DESCRIPTOR, and SYNC_FRAME) has been automated by hardware.

- Supports 4K bytes FIFO memory allowing multi-configuration.

- Supports 8-/16-/32-bit access for each endpoint (EP) FIFO.

- Comprised of the AMBA AHB interface block, the control register block, the 4K bytes FIFO memory for packet data transfer, the high-speed/full-speed USB transceiver interface block, and the serial interface engine (SIE) block.

- Can support high-speed data transfer using a double buffer.

- Supports UTMI 16-bit unidirectional mode.

- Provided with pullup resistors for full speed and termination resistors for high speed

26.1.1 Block Diagram

Figure 26-1 shows an overall block diagram of the USB and Figure 26-2 shows a block diagram of the transceiver section (physical layer) of the USB.

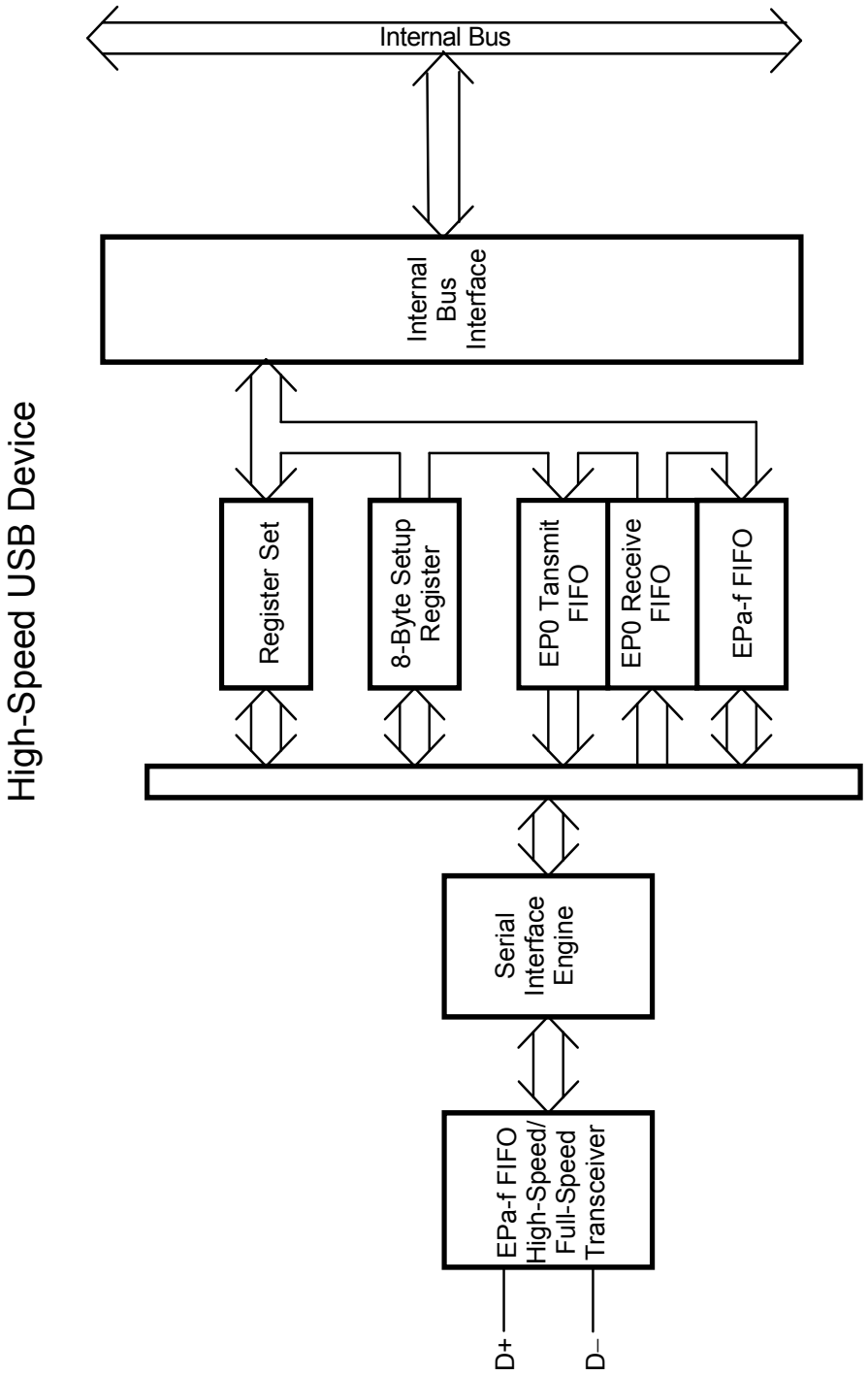


Figure 26-1 Block Diagram of the USB

26.1.2 Pin Descriptions

The following table shows the description of signals of this module.

LSI Pins *1	USB Module Pins	I/O	Description	Polarity	Initial State
USB_DP	dp	I/O	D+ USB up-stream pin		
USB_DM	dm	I/O	D– USB up-stream pin		
USB_REXT	rexti	Analog	Reference resistor connection pin. Connect a 12.4 kΩ±1% resistor between this pin and the USB GND.		
AVDDTX	avddtx	Power	Analog VDD pin for the transmitter.		
AGNDTX	avsstx	Power	Analog VSS pin for the transmitter.		
AVDDRX	avddrx	Power	Analog VDD pin for the receiver.		
AGNDRX	avssrx	Power	Analog VSS pin for the receiver.		
AVDDC	avddc	Power	Analog VDD pin for the bias circuit/PLL.		
AGNDC	avssc	Power	Analog VSS pin for the bias circuit/PLL.		
USB_VOREF	vorefi	Analog	Analog test pin. Leave this pin open.		
USB_ATEST1	atest1	Analog	Analog test pin. Leave this pin open.		
USB_ATEST0	atest0	Analog	Analog test pin. Leave this pin open.		
—	HCLK	I	AHB clock input signal.	—	—
—	HADDR[13:0]	I	Address input signal.	—	—
—	HWDATA[31:0]	I	Write data input signal.	—	—
—	HRDATA[31:0]	O	Read data input signal.	—	"L"
—	HREADYout	O	Ready output signal.	Positive	"L"
—	HREADYin	I	Ready input signal.	Positive	—
—	HRESP[1:0]	O	Response output signal for transfer.	Positive	"L"
—	HRESET	I	Reset input signal. Any write operation to the registers of this module is disabled for a period of 5ms after hardware reset is executed by this signal. Therefore, when writing to a register of this module, do so after a lapse of 5ms from the time of execution of hardware reset by this signal.	Negative	—
—	USBINT	O	Interrupt request output signal.	Negative	"H"
—	USBDREQ	O	Automatic data transfer request output signal.	Positive	"L"
—	USBDRQCLR	I	USB automatic data transfer request clear input signal.	Positive	—
—	USBTC	I	Final automatic data transfer notify signal.	Positive	—
—	PHYOpMode[1:0]	O	Operation mode designation output signal for USB PH. [1] [0] Description 0 0 Normal Operation 0 1 Non-Driving 1 0 Disable bit stuffing and NRZI encoding 1 1 Reserved	Positive	0 1
—	PHYPulldown	O	Output signal to enable the built-in pull-down resistor (both on the D+ and D– sides) for the USB PHY module.	Positive	"L"

*1: USB Module pins for LSI Pin columns indicated with "—" are connected to LSI internal control pins.

26.1.3 Registers

This section describes the register mapping of this block.

- If write access is made to the reserved addresses, operation cannot be guaranteed. Therefore, implement control so as not to make write access to the reserved addresses.
- Operation is not affected even if write access is made to an address not listed in the table below.
- If read access is made to the reserved addresses, data to be read is undefined.
- If read access is made to an address not listed in the table below, data to be read is "0".
- Operation cannot be guaranteed if "1" is written into a reserved field in any of the registers listed in the table below. To write into other fields, be sure to write "0" into this reserved field. If read access is made, data to be read is "0".

However, HREADY is responded to AHB in any of the above cases, and HRESP [1:0] = 00 (OKAY) is output at that time.

Address	Name	Abbreviation	R/W	Initial value	Remark
0x7BB0_0000	Revision register	Rev	R	0xFE01	
0x7BB0_0004	System control register	SysCtl	R/W	0x0080	
0x7BB0_0008	Automatic data transfer configuration register	DMACfg	R/W	0x0000	
0x7BB0_000A	Automatic data transfer control register	DMACtl	R/W	0x0000	
0x7BB0_000C	Interrupt status register	IntStt	R/W	0x0000	
0x7BB0_0010	Interrupt enable register	IntEnb	R/W	0x0100	
0x7BB0_0020	Setup data 1 register	SetUpData1	R	0x0000	
0x7BB0_0022	Setup data 2 register	SetUpData2	R	0x0000	
0x7BB0_0024	Setup data 3 register	SetUpData3	R	0x0000	
0x7BB0_0026	Setup data 4 register	SetUpData4	R	0x0000	
0x7BB0_0028	Frame number register	FrameNum	R	0x0000	
0x7BB0_002C	Standard device request information register	SrdDevReqInfo	R	0x8000	
0x7BB0_0030	EP0 configuration 1 register	EP0Cfg1	R/W	0x0000	
0x7BB0_0034	EP0 control register	EP0Ctl	R/W	0x0001	
0x7BB0_0036	EP0 status register	EP0Stt	R/W	0x0000	
0x7BB0_0038	EP0 receive byte count register	EP0RxCnt	R	0x0000	
0x7BB0_003C	EP0 transmit byte count register	EP0TxCnt	R/W	0x0000	
0x7BB0_0040	EPa configuration 1 register	EPaCfg1	R/W	0x0000_0000	
	EPa configuration 2 register	EPaCfg2			
0x7BB0_0044	EPa FIFO assignment register	EPaFIFOAsin	R/W	0x0000	
0x7BB0_0048	EPa control register	EPaCtl	R/W	0x0000	
0x7BB0_004C	EPa status register	EPaStt	R/W	0x0000	
0x7BB0_0050	EPa receive byte count 1 register	EPaRxCnt1	R	0x0000	
0x7BB0_0052	EPa receive byte count 2 register	EPaRxCnt2	R	0x0000	
0x7BB0_0054	EPa transmit byte count 1 register	EPaTxCnt1	R/W	0x0000	
0x7BB0_0056	EPa transmit byte count 2 register	EPaTxCnt2	R/W	0x0000	
0x7BB0_0060	EPb configuration 1 register	EPbCfg1	R/W	0x0000_0000	
	EPb configuration 2 register	EPbCfg2			
0x7BB0_0064	Epb FIFO assignment register	EPbFIFOAsin	R/W	0x0000	
0x7BB0_0068	EPb control register	EPbCtl	R/W	0x0000	
0x7BB0_006C	EPb status register	EPbStt	R/W	0x0000	
0x7BB0_0070	EPb receive byte count 1 register	EPbRxCnt1	R	0x0000	
0x7BB0_0072	EPb receive byte count 2 register	EPbRxCnt2	R	0x0000	
0x7BB0_0074	EPb transmit byte count 1 register	EPbTxCnt1	R/W	0x0000	
0x7BB0_0076	EPb transmit byte count 2 register	EPbTxCnt2	R/W	0x0000	

Address	Name	Abbreviation	R/W	Initial value	Remark
0x7BB0_0080	EPc configuration 1 register	EPcCfg1	R/W	0x0000_0000	
	EPc configuration 2 register	EPcCfg2			
0x7BB0_0084	EPc FIFO assignment register	EPcFIFOAsin	R/W	0x0000	
0x7BB0_0088	EPc control register	EPcCtl	R/W	0x0000	
0x7BB0_008C	EPc status register	EPcStt	R/W	0x0000	
0x7BB0_0090	EPc receive byte count 1 register	EPcRxCnt1	R	0x0000	
0x7BB0_0094	EPc transmit byte count 1 register	EPcTxCnt1	R/W	0x0000	
0x7BB0_00A0	EPd configuration 1 register	EPdCfg1	R/W	0x0000_0000	
	EPd configuration 2 register	EPdCfg2			
0x7BB0_00A4	EPd FIFO assignment register	EPdFIFOAsin	R/W	0x0000	
0x7BB0_00A8	EPd control register	EPdCtl	R/W	0x0000	
0x7BB0_00AC	EPd status register	EPdStt	R/W	0x0000	
0x7BB0_00B0	EPd receive byte count 1 register	EPdRxCnt1	R	0x0000	
0x7BB0_00B4	EPd transmit byte count 1 register	EPdTxCnt1	R/W	0x0000	
0x7BB0_00C0	EPe configuration 1 register	EPeCfg1	R/W	0x0000_0000	
	EPe configuration 2 register	EPeCfg2			
0x7BB0_00C4	EPe FIFO assignment register	EPeFIFOAsin	R/W	0x0000	
0x7BB0_00C8	EPe control register	EPeCtl	R/W	0x0000	
0x7BB0_00CC	EPe status register	EPeStt	R/W	0x0000	
0x7BB0_00D0	EPe receive byte count register 1	EPeRxCnt1	R	0x0000	
0x7BB0_00D4	EPe transmit byte count register 1	EPeTxCnt1	R/W	0x0000	
0x7BB0_00E0	EPf configuration 1 register	EPfCfg1	R/W	0x0000_0000	
	EPf configuration 2 register	EPfCfg2			
0x7BB0_00E4	EPf FIFO assignment register	EPfFIFOAsin	R/W	0x0000	
0x7BB0_00E8	EPf control register	EPfCtl	R/W	0x0000	
0x7BB0_00EC	EPf status register	EPfStt	R/W	0x0000	
0x7BB0_00F0	EPf receive byte count 1 register	EPfRxCnt1	R	0x0000	
0x7BB0_00F4	EPf transmit byte count 1 register	EPfTxCnt1	R/W	0x0000	
0x7BB0_0100	EP0 transmit FIFO register	EP0TxFIFO	W	Unde-fined	
0x7BB0_0104	EP0 receive FIFO register	EP0RxFIFO	R	Unde-fined	
0x7BB0_1000 to 0x7BB0_13FF	EPa FIFO register	EPaFIFO	R/W	Unde-fined	Lower 10 bits of an address are ignored.
0x7BB0_1400 to 0x7BB0_17FF	EPb FIFO register	EPbFIFO	R/W	Unde-fined	Lower 10 bits of an address are ignored.
0x7BB0_1800 to 0x7BB0_1BFF	EPc FIFO register	EPcFIFO	R/W	Unde-fined	Lower 10 bits of an address are ignored.
0x7BB0_1C00 to 0x7BB0_1FFF	EPd FIFO register	EPdFIFO	R/W	Unde-fined	Lower 10 bits of an address are ignored.
0x7BB0_2000 to 0x7BB0_23FF	EPe FIFO register	EPeFIFO	R/W	Unde-fined	Lower 10 bits of an address are ignored.
0x7BB0_2400 to 0x7BB0_27FF	EPf FIFO register	EPfFIFO	R/W	Unde-fined	Lower 10 bits of an address are ignored.

26.2 Detailed Description of Registers

26.2.1 Revision Register (Rev)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rev	RevInv								Revision							
At hardware reset	Inversed value of revision No.								Revision No.							
At USB reset	Inversed value of revision No.								Revision No.							

Address: 0x7BB00000
Access: R
Access size: 16 bits

[Description of Bits]

- RevInv (Revision Invert): Bits 15 to 8
  These bits indicate the inversed value (FEh) of the revision number (bits 7 to 0) of this block.
- Revision: Bits 7 to 0
  These bits indicate the revision number (01h) of this block.

26.2.2 System Control Register (SysCtl)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SysCtl	—*	—*	—*	—*	—*	—*	—*	—*	Self Pwr	PU Ctl	PD Ctl	RWA	—*	—*	PD	SR
At hardware reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	(Note1)	(Note1)	(Note1)	(Note1)	0	0	(Note1)	0

Address: 0x7BB00004
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]

- SelfPwr (Self Power): Bit 7
  This bit sets that the LSI is bus-powered or self-powered as the USB device system. The setting value of this bit is transmitted to the USB host as the SelfPowered bit of the information read by the standard device request command (Get Status) from the USB host.
  · 0: Indicates that the LSI operates as a bus-powered device.
  · 1: Indicates that the LSI operates as a self-powered device.

- PUCtl (Pullup Control): Bit 6
  Pullup setting to D+ is executed after this bit is set to "1" when the PDCtl bit is "0".
  Be sure to set the PDCtl bit to "0" before setting this bit to "1", since setting this bit to "1" with the PDCtl bit being set to "1" is prohibited.
  · 0: PHYOpMode [1:0] signal becomes "0 1" forcibly.
  · 1: PHYOpMode [1:0] signal becomes "Normal Operation".

- PDCtl (Pulldown Control): Bit 5
  The static power supply current can be saved by setting this bit to "1" when the USB cable is not connected.
  Pulldown set to D+ and D– is performed after this bit is set to "1" with the PUCtl bit being set to "1".
  Be sure to set the PUCtl bit to "0" before setting this bit to "1", since setting this bit to "1" with the PUCtl bit being set to "1" is prohibited.
  · 0: The PHYPulldown signal becomes "0".
  · 1: The PHYOpMode [1:0] signal becomes "Normal Operation" forcibly and the PHYPulldown signal becomes "1".
- RWA (Remote Wakeup): Bit 4
  This bit is used to specify the execution of remote wakeup by writing "1" to this bit while PHYCLK is stopped (after a lapse of at least 3ms from the time of occurrence of a suspend interrupt following the setting of PD=1). The operation is not guaranteed when "1" is written to this bit while PHYCLK is operating.
  To execute remote wakeup, remote wakeup must be enabled using SET_FEATURE of the standard request in advance from the USB host.
  This bit is a write-only bit, and is always set to "0" at read.
- PD (Power Down): Bit 1
  This bit sets the power down mode.
  · 0: Does not implement power saving (stopping PHYCLK) when suspended.
  · 1: Implements power saving (stopping PHYCLK) when suspended.
- SR (Soft Reset): Bit 0
  This bit is used to specify the execution of a reset by writing "1" into this bit. It is functionally the same as a hardware reset.
  This bit is a write-only bit, and is always set to "0" at read.
  Any write operation to this module is disabled for a period of 5 ms immediately after "1" has been written to this bit. Thus, this module should be accessed after a lapse of 5 ms.

26.2.3 Automatic Data Transfer Configuration Register(DMACfg)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMACfg	—*	—*	—*	—*	DREQItvl				DMA Enb	—*	DMA_EP			—*	Rsv	—*
At hardware reset	0	0	0	0	0	0	0	0	0	0	000			0	0	0
At USB reset	0	0	0	0	0	0	0	0	(Note 1)	0	(Note 1)			0	0	0

Address: 0x7BB00008
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]

- DREQItvl (DREQ Interval): Bits 11 to 8
  These bits are used to specify the automatic data transfer request output interval (the interval between the USB DREQ signal negate to next the USBDREQ signal assert) using the following equation:
  Interval period = T x n (T = HCLK cycle x 2, n: value of this register)
  If n = 0, high-speed operation takes effect.
  Since this register is initialized to 0000b at USB reset, this register must be set again to resume automatic data transfer transfer under the same condition after USB reset.
- DMAEnb (DMA Enable): Bit 7
  This bit is used to specify automatic data transfer enable.
  · 0: Automatic data transfer disable
  · 1: Automatic data transfer enable
- DMA_EP (DMA EP): Bits 5 to 3
  These bits are used to specify the target EP for automatic data transfer.
  Bits 5 to 3 = 111: Specification prohibited
  Bits 5 to 3 = 110: Specifies EPf as the target EP for automatic data transfer.
  Bits 5 to 3 = 101: Specifies EPe as the target EP for automatic data transfer.
  Bits 5 to 3 = 100: Specifies EPd as the target EP for automatic data transfer.
  Bits 5 to 3 = 011: Specifies EPc as the target EP for automatic data transfer.
  Bits 5 to 3 = 010: Specifies EPb as the target EP for automatic data transfer.
  Bits 5 to 3 = 001: Specifies EPa as the target EP for automatic data transfer.
  Bits 5 to 3 = 000: Specifies that there is no EP for automatic data transfer.
- Rsv (Reserve): Bit 1
  This bit is a reserved bit.
  Do not write "1" into this bit. If "1" is written into this bit, operation cannot be guaranteed.
  To write into other bits of this register, be sure to write "0" into this bit.

26.2.4 Automatic Data Transfer Control Register (DMACtl)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMACtl	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	DMA Strt	—*	—*	Rsv	
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00	
At USB reset	0	0	0	0	0	0	0	0	0	0	0	(Note 1)	0	0	00	

Address: 0x7BB0000A
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]

- DMAStrt (DMA Start): Bit 4
  This bit is used to specify the start of automatic data transfer.
  · 0: Stop automatic data transfer
  · 1: Start automatic data transfer
  If an automatic data transfer request is generated while this bit is set to "1", a USBDREQ signal is made active.
  However, if an automatic data transfer request is generated while this bit is set to "0", a USBDREQ signal is not made active.
  This bit is automatically cleared to "0" when any of the following conditions occurs.
  A USBTC signal is input
  A short packet receive interrupt cause is generated
- Rsv (Reserve): Bits 1 and 0
  These bits are reserved bits.
  Do not write "1" into this bit. If "1" is written into this bit, operation cannot be guaranteed.
  To write into the DMAStrt bit of this register, be sure to write "0" into this bit.

26.2.5 Interrupt Status Register (IntStt)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntStt	—*	EPf Evt Stt	EPe Evt Stt	EPd Evt Stt	EPc Evt Stt	EPb Evt Stt	EPa Evt Stt	EP0 Evt Stt	—*	—*	—*	Awk Stt	Spend Stt	USB Rst Dast Stt	USB Rst Ast Stt	SOF Stt
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	0	0	0	0	0	0	(Note1)	(Note1)	0

Address: 0x7BB0000C
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.
(Note 2): Determined by the interrupt enable bit and the EP Dir value of the EP configuration. This bit is initialized to "1" only when interrupt is enabled and when EPDir = 1 is set.
(Note): Bits 8 to 14 are read only, so they are disabled when written.

[Description of Bits]

- EPxEvtStt (EPx Event Status): Bits 14 to 8
  If any one bit of the EPx status register (EPxStt) is set to "1" and the EPx event interrupt enable bit of the corresponding interrupt enable register (IntEnb) is valid, this bit is set to "1" and an interrupt is generated.
- Bits other than EPxEvt: Bits 4 to 0
  See Section 4.3, "Description of Interrupts".

Note that when the corresponding bit of the interrupt enable register (IntEnb) is set to "0", each bit retains "0" even if a cause to set each bit to "1" is generated.

26.2.6 Interrupt Enable Register (IntEnb)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntEnb	—*	EPf Evt Enb	EPe Evt Enb	EPd Evt Enb	EPc Evt Enb	EPb Evt Enb	EPa Evt Enb	EP0 Evt Enb	—*	—*	—*	Awk Enb	Spend Enb	USB Rst Dast Enb	USB Rst Ast Enb	SOF Enb
At hardware reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
At USB reset	0	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	0	0	0	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)

Address: 0x7BB00010
Access: R/W
Access size: 16 bits

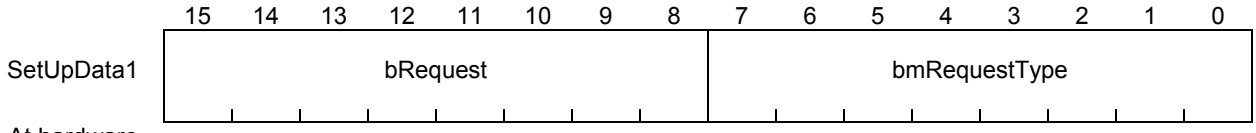
[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]

These bits are used to specify to enable/disable the interrupt that corresponds to each bit of the interrupt status register (IntStt).

- 1: Interrupt enable
- 0: Interrupt disable

26.2.7 Setup Data 1 Register (SetUpData1)



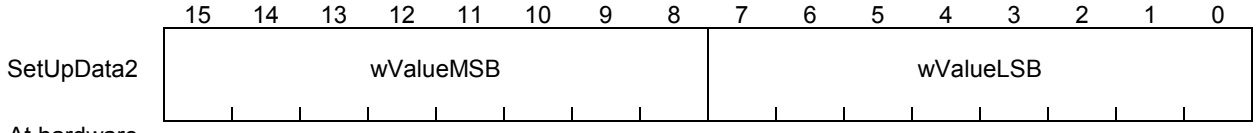
At hardware reset	00h	00h
At USB reset	00h	00h

Address: 0x7BB00020
Access: R
Access size: 16 bits

[Description of Bits]

- bRequest (Byte Request): Bits 15 to 8
- bmRequestType (Bit Map Request Type): Bits 7 to 0

26.2.8 Setup Data 2 Register (SetUpData2)



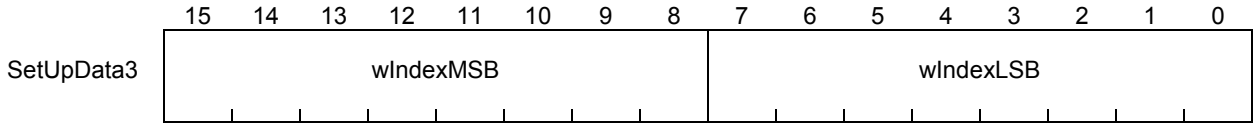
At hardware reset	00h	00h
At USB reset	00h	00h

Address: 0x7BB00022
Access: R
Access size: 16 bits

[Description of Bits]

- wValueMSB (Word Value MSB): Bits 15 to 8
- wValueLSB (Word Value LSB): Bits 7 to 0

26.2.9 Setup Data 3 Register (SetUpData3)



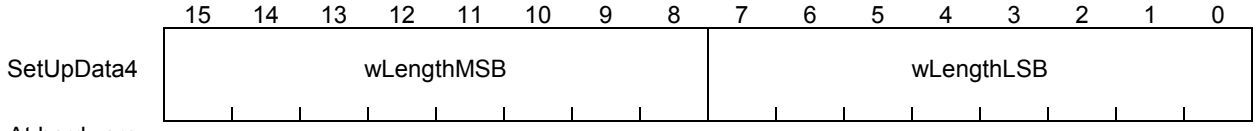
At hardware reset	00h	00h
At USB reset	00h	00h

Address: 0x7BB00024
Access: R
Access size: 16 bits

[Description of Bits]

- wIndexMSB (Word Index MSB): Bits 15 to 8
- wIndexLSB (Word Index LSB): Bits 7 to 0

26.2.10 Setup Data 4 Register (SetUpData4)



At hardware reset	00h	00h
At USB reset	00h	00h

Address: 0x7BB00026
Access: R
Access size: 16 bits

[Description of Bits]

- wLengthMSB (Word Length MSB): Bits 15 to 8
- wLengthLSB (Word Length LSB): Bits 7 to 0

26.2.11 Frame Number Register (FrameNum)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FrameNum	—*	—*	—*	—*	—*	FrmNum										
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BB00028
Access: R
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).

[Description of Bits]
- FrmNum (Frame Number): Bits 10 to 0
  When a start of frame (SOF) packet is transmitted from the USB host, this block automatically writes the frame number into this register.

26.2.12 Standard Device Request Information Register (SrdDevReqInfo)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SrdDev ReqInfo	H_F Spd	—*	—*	—*	Alt				If				Cfg			
At hardware reset	1	0	0	0	0000				0000				0000			
At USB reset	1	0	0	0	(Note 1)				(Note 1)				(Note 1)			

Address: 0x7BB0002C
Access: R
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).

[Description of Bits]
- H_FSpd (High_Full Speed): Bit 15
  This bit indicates the status of PHY that can be used. This bit is a read-only bit.
  Read this bit after the occurrence of a USB reset deassert interrupt, and determine HS (high speed) or FS (full speed).
  · 0: High-speed transceiver can be used.
  · 1: Full-speed transceiver can be used.
- Alt (Alternate): Bits 11 to 8
  The alternate setting to be requested by SET_INTERFACE, a standard device request, is stored. This field is updated when SET_INTERFACE is received. These bits are read-only bits.
- If (Interface): Bits 7 to 4
  The interface value to be requested by SET_INTERFACE, a standard device request, is stored. This field is updated when SET_INTERFACE is received. These bits are read-only bits.
- Cfg (Configuration): Bits 3 to 0
  The configuration value to be requested by SET_CONFIGURATION, a standard device request, is stored. This field is updated when SET_CONFIGURATION is received. These bits are read-only bits.

26.2.13 EP0 Configuration 1 Register (EP0Cfg1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP0Cfg1	—*	—*	—*	—*	—*	—*	MaxPktSize							—*	—*	—*
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	(Note 1)							0	0	0

Address: 0x7BB00030
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]

- MaxPktSize (Max. Packet Size): Bits 9 to 3
  These bits are used to specify the maximum packet size of a packet to be transferred at EP0. EP0 is an EP exclusively used for control transfer.
  It is necessary to make the same settings as those of the descriptor to be sent to the host.
  It is fixed to 64 bytes when in high speed mode.

26.2.14 EP0 Control Register (EP0Ctl)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP0Ctl	—	—	—	—	Stl	Stt Stg Chg Enb Enb	Set Itf Enb	Set Cfg Enb	Rsv	—	—	Shrt PktRx Enb	—	TxPkt Rdy Enb	RxPkt Rdy Enb	StUp Rdy Enb
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
At USB reset	0	0	0	0	0	0	0	0	0	0	0	(Note 1)	0	0	0	1

Address: 0x7BB00034
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]

- Stl (Stall): Bit 11
  If a data transfer request to EP0 arrives from the USB host when this bit is set to "1", a stall packet is returned.
  When a setup packet is received, this bit is automatically set to "0".
- Bits other than Stl: Bits 10 to 0
  These bits are used to specify to enable/disable each bit of the EP0 status register (EP0Stt).

If any of these bits is set to "1" and the applicable cause is generated, the corresponding bit of the EP0 status register (EP0Stt) is set to "1". For example, if EP0 receives a short packet when the ShrtPktRxEnb bit is set to "1", the ShrtPktRx bit of the EP0 status register (EP0Stt) is set to "1". Note that if any of these bits is set to "0", the corresponding bit of the EP0 status register (EP0Stt) retains "0".
For more information, see Section 26.3.3, "Description of Interrupts".

26.2.15 EP0 Status Register (EP0Stt)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP0Stt	—*	—*	—*	—*	—*	Stt Stg Chg Stt	Set Itf Stt	Set Cfg Stt	Rsv	—*	—*	Shrt PktRx Stt	—*	TxPkt Rdy Stt	RxPkt Rdy Stt	StUp Rdy Stt
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0x7BB00036
Access: R/W

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).

[Description of Bits]

For the description of each bit, see Section 26.3.3, "Description of Interrupts".
Note that when the corresponding bit of the EP0 control register (EP0Ctl) is set to "0", each bit retains "0" even if a cause to set each bit to "1" is generated.

26.2.16 EP0 Receive Byte Count Register (EP0RxCnt)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP0RxCnt	RxByteCunt															
At hardware reset	0000h															
At USB reset	(Note 1)															

Address: 0x7BB00038
Access: R
Access size: 16 bits

[Note]
(Note 1): Retains the status before USB reset.

[Description of Bits]
- RxByteCunt (Rx Byte Count): Bits 15 to 0
  These bits indicate the number of bytes received.

26.2.17 EP0 Transmit Byte Count Register (EP0TxCnt)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP0TxCnt	—*	—*	—*	—*	—*	—*	—*	—*	—*	TxByteCunt						
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	0	(Note 1)						

Address: 0x7BB0003C
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]
- TxByteCunt (Tx Byte Count): Bits 6 to 0
  These bits are used to specify the number of bytes to be transmitted.
  The transmission of invalid data can be suppressed by setting the number of bytes that should be transmitted per packet ("01h" in the case of one byte) in this register. If packets having the same number of bytes are to be transmitted, it is not necessary to set this register each time (as the previously set value is retained in this register).
  After writing the number of transmit data bytes in this register, write data into the data packet transfer FIFO, and then set the transmit PKTRDY to "1".

26.2.18 EPa - EPf Configuration 1 Register (EPa - EPfCfg1)

26.2.19 EPa - EPf Configuration 2 Register (EPa - EPfCfg2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPa - EPfCfg2	—*	—*	MaxPktSize											AltStMSB		
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	(Note1)											(Note1)		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPa - EPfCfg1	AltSt LSB	ItfNum				CfgNum				EPType		EPDir	EP Num			
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	(Note1)	(Note1)				(Note1)				(Note1)		(Note1)	(Note1)			

Address: 0x7BB00040, 0x7BB00060, 0x7BB00080, 0x7BB000A0, 0x7BB000C0, 0x7BB000E0, 0x7BB00042, 0x7BB00062, 0x7BB00082, 0x7BB000A2, 0x7BB000C2, 0x7BB000E2
Access: R/W
Access size: 32 bits

[Notes]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care). Writing "1" is prohibited.
(Note 1): Retains the status before USB reset.
(Note 2): When writing to EPxCfg1 and EPxCfg2, update both registers by 32-bit accessing at the same time or update both registers continuously by 16-bit accessing.
If only either register has been updated by 16-bit accessing, the set values cannot be reflected on hardware.

[Description of Bits]
- MaxPktSize (Max. Packet Size): Bits 29 to 19
  These bits are used to specify the maximum packet size.
  If the maximum packet sizes are different in High-Speed and Full-Speed, set this register to appropriate values after a USB bus reset deassert interrupt has occurred (the High-Speed or Full-Speed connection is recognized).
  When the applicable EP is in high speed mode and also during bulk transfer, the maximum packet size is fixed to 512 bytes (however, the written value can be read at read.)
- AltSt (Alternate Setting): Bits 18 to 15
  These bits are used to specify the alternate setting.
  This field consists of 4 bits; however, the values that can be set are 0 to 3. Do not set any value other than 0 to 3.
- ItfNum (Interface Number): Bits 14 to 11
  These bits are used to specify the interface number.
  This field consists of 4 bits; however, the values that can be set are 0 to 3. Do not set any value other than 0 to 3.

- CfgNum (Configuration Number): Bits 10 to 7
  These bits are used to specify the configuration number.
  This field consists of 4 bits; however, the values that can be set are 0 to 2. Do not set any value other than 0 to 2.
- EPType (EP Type): Bits 6 and 5
  These bits are used to specify the EP type. Since EPa to EPf cannot be used for Control transfer, do not set these bits to 00b.
  Bits 6, 5 = 1, 1: Interrupt
  Bits 6, 5 = 1, 0: Bulk
  Bits 6, 5 = 0, 1: Setting prohibited
  Bits 6, 5 = 0, 0: Control
- EPDir (EP Direction): Bit 4
  This bit is used to specify the transfer direction.
  · 1: IN transfer direction (this block → host)
  · 0: OUT transfer direction (host → this block)
- EPNum (EP Number): Bits 3 to 0
  These bits are used to specify the EP number.

26.2.20 EPa - EPfFIFO Assignment Register (EPa - EPfFIFOAsin)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPa - EPfFIFOAsin	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	—*	Buf Mod
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(Note 1)

Address: 0x7BB00044, 0x7BB00064, 0x7BB00084, 0x7BB000A4, 0x7BB000C4, 0x7BB000E4
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]

- BufMod (Buffer Mode): Bit 0
  This bit is used to specify the EP buffer mode.
  · 1: Single FIFO (one-sided FIFO)
  · 0: Double FIFO (double-sided FIFO)

26.2.21 EPa - EPf Control Register (EPa - EPfCtl)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPa - EPfCtl	Busy	ISO Mod Sel	Pre Frm InTk	FIFO Clr	Stl	—*	—*	—*	Rsv	Ovr Enb	NAK Res Enb	Shrt PktRx Enb	—*	TxPkt Rdy Enb	RxPkt Rdy Enb	—*
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	0	(Note 1)	(Note 1)	(Note 1)	0	(Note 1)	(Note 1)	0

Address: 0x7BB00048, 0x7BB00068, 0x7BB00088, 0x7BB000A8, 0x7BB000C8, 0x7BB000E8
Access: R/W
Access size: 16 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.

[Description of Bits]

- Busy: Bit 15
  When this bit is set to "1", a NAK is always returned to the USB host regardless of the value of the packet ready bit.
  This bit is valid only if the applicable EP is set for bulk transfer or interrupt transfer by the EP control register. This bit becomes invalid when the applicable EP is set to a transfer other than bulk transfer or interrupt transfer by the EP control register.
- ISOModSel (Isochronous Mode Select): Bit 14
  This bit cannot be used for this LSI since the bit is used for the Isochronous transfer.
- PreFrmInTkn (Previous Frame In Token): Bit 13
  This bit cannot be used for this LSI since the bit is used for the Isochronous transfer.
- FIFOClr (FIFO Clear): Bit 12
  This bit is valid only if the applicable EP is set to the IN transfer direction (EPDir = 1).
  The transmit FIFO of the applicable EP is cleared by writing "1" into this bit. (This bit itself retains "0".)
  If FIFOClr is executed during the execution of DMA transfer, operation cannot be guaranteed.
  If data of the next packet is written immediately after the execution of FIFO Clear, this packet may be omitted. To prevent this problem, write data of the next packet after the occurrence of a NAK response interrupt (NAK Res Stt) of the applicable EP when the EP is set to bulk IN or interrupt IN transfer.
  If FIFOClr is executed during the automatic data transfer, operation cannot be guaranteed.
- Stl (Stall): Bit 11
  If a data transfer request to the applicable EP arrives from the USB host when "1" is written to this bit, a stall packet is returned. When the GetStatus command for the applicable EP is received after the stall packet is returned, Endpoint_Halt is returned. When this bit is read, "0" is always read out.
- Rsv (Reserve): Bit 7
  These bits are reserved bits.
  Do not write "1" into this bit. If "1" is written into this bit, operation cannot be guaranteed.
  To write into other bits of this register, be sure to write "0" into this bit.
- Bits other than above: Bits 6, 5, 4, 2 and 1
  See Section 26.3.3, "Description of Interrupts".

26.2.22 EPa - EPf Status Register (EPa - EPfStt)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPa - EPfStt	—*	—*	—*	—*	—*	—*	—*	—*	Rsv	Ovr Stt	NAK Res Stt	Shrt PktRx Stt	—*	TxPkt Rdy Stt	RxPkt Rdy Stt	—*
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	0	(Note 1)	(Note 1)	(Note 1)	0	(Note 2)	(Note 2)	0

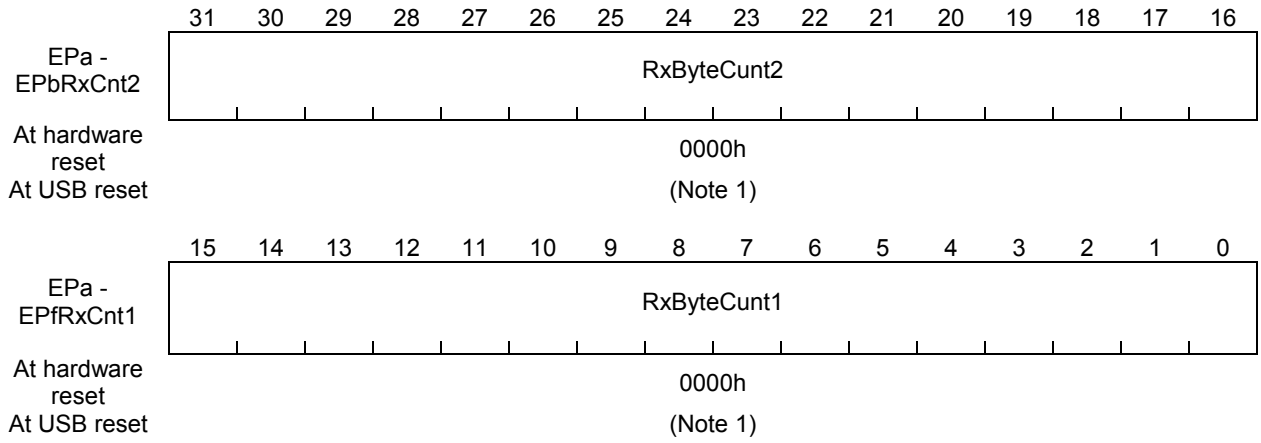
Address: 0x7BB0004C, 0x7BB0006C, 0x7BB0008C, 0x7BB000AC, 0x7BB000CC, 0x7BB000EC
Access: R/W
Access size: 32 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).
(Note 1): Retains the status before USB reset.
(Note 2) If EPs are specified for transmission, this bit is initialized to value "1". If EPs are specified for reception, this value is not affected by USB reset.

[Description of Bits]
For more information about each bit, see Section 26.3.3, "Description of Interrupts".
Note that when the corresponding bit of the EPa - EPf control register (EPa-EPfCtl) is set to "0", each bit retains "0" even if a cause that will cause to set each bit to "1" is generated.

26.2.23 EPa - EPf Receive Byte Count 1 Register (EPa - EPfRxCunt1)

26.2.24 EPa - EPb Receive Byte Count 2 Register (EPa - EPbRxCunt2)



Address: 0x7BB00050, 0x7BB00070, 0x7BB00090, 0x7BB000B0, 0x7BB000D0, 0x7BB000F0, 0x7BB00052, 0x7BB00072
Access: R
Access size: 32 bits

[Note]
(Note 1): Retains the status before USB reset.

[Description of Bits]

- RxByteCunt2 (Rx Byte Count2): Bits 31 to 16
  This bit cannot be used for this LSI since the bit is used for the Isochronous transfer.
- RxByteCunt1 (Rx Byte Count): Bits 15 to 0
  These bits indicate the number of bytes received.

26.2.25 EPa - EPf Transmit Byte Count 1 Register (EPa - EPfTxCnt1)

26.2.26 EPa - EPb Transmit Byte Count 2 Register (EPa - EPbTxCnt2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPa - EPbTxCnt2	—*	—*	—*	—*	—*	TxByteCunt2										
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPa - EPfTxCnt1	—*	—*	—*	—*	—*	TxByteCunt1										
At hardware reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
At USB reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

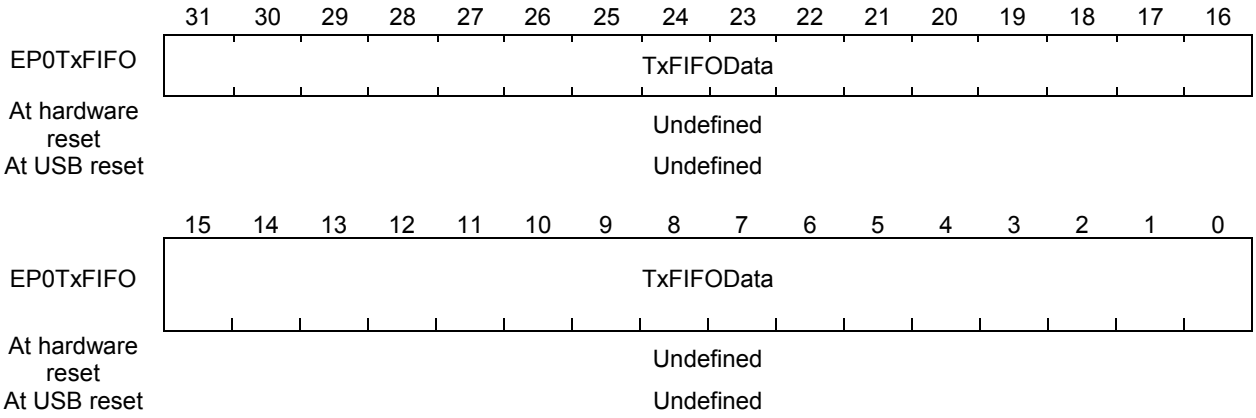
Address: 0x7BB00054, 0x7BB00074, 0x7BB00094, 0x7BB000B4, 0x7BB000D4, 0x7BB000F4, 0x7BB00056, 0x7BB00076
Access: R/W
Access size: 32 bits

[Note]
*: The bit data that is indicated by "—" returns "0" at read operation; however, it is recommended that the program do not assume "0" (don't care).

[Description of Bits]

- TxByteCunt2 (Tc Byte Count2): Bits 26 to 16
  This bit cannot be used for this LSI since the bit is used for the Isochronous transfer.
- TxByteCunt1 (Tx Byte Count): Bits 10 to 0
  These bits are used to specify the number of bytes to be transmitted.
  The transmission of invalid data can be suppressed by setting the number of bytes that should be transmitted per packet ("01h" in the case of one byte) in this register. If packets having the same number of bytes are to be transmitted, it is not necessary to set this register each time (as the previously set value is retained in this register).
  After writing the number of transmit data bytes in this register when in PIO mode, write the data into the data packet transfer FIFO, and then set the transmit PKTRDY to "1".
  When in automatic data transfer mode, write the number of transmit data bytes into this register, and then write data into the FIFO. However, it is prohibited to execute transmission of a NULL packet by the automatic data transfer by setting the value of this register to 0000h.

26.2.27 EP0 Transmit FIFO Register (EP0TxFIFO)



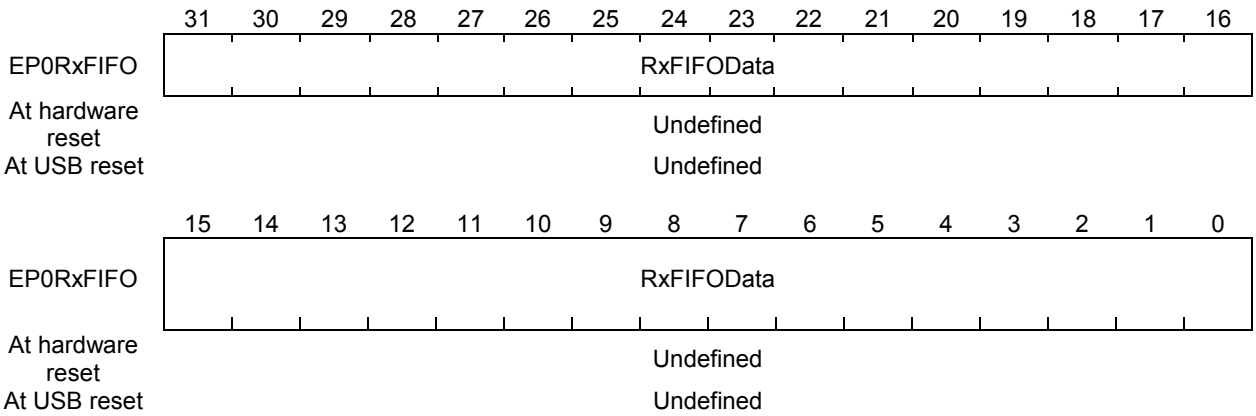
Address: 0x7BB00100
Access: W
Access size: 32 bits

Note: When data is written to EP0TxFIFO on an 8-bit or 16-bit basis, 4-byte data should be written continuously. That is, data should be written four times continuously, on the 8-bit basis and two times continuously on the 16-bit basis. If writing is not continued, transmitted data cannot be guaranteed.

[Description of Bits]

- TxFIFOData (Tx FIFO Data): Bits 31 to 0
  This is an FIFO register for transmit data.

26.2.28 EP0 Receive FIFO Register (EP0RxFIFO)



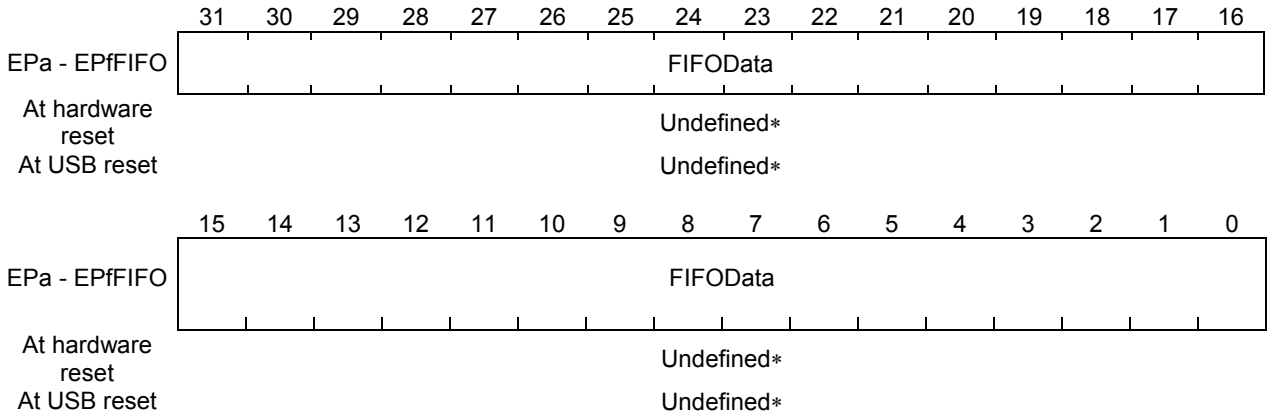
Address: 0x7BB00104
Access: R
Access size: 32 bits

Note: When data is read from EP0RxFIFO on an 8-bit or 16-bit basis, 4-byte data should be read continuously. That is, data should be read four times continuously on the 8-bit basis and two times continuously on the 16-bit basis. If reading is not continued, received data cannot be guaranteed.

[Description of Bits]

- RxFIFOData (Rx FIFO Data): Bits 31 to 0
  This is an FIFO register for receive data.

26.2.29 EPa - EPfFIFO Register (EPa - EPfFIFO)



Address: 0x7BB01000 - 0x7BB013FF, 0x7BB01400 - 0x7BB017FF, 0x7BB01800 - 0x7BB01BFF, 0x7BB01C00 - 0x7BB01FFF, 0x7BB02000 - 0x7BB023FF, 0x7BB02400 - 0x7BB027FF
Access: R/W
Access size: 32 bits

Note: When EPxFIFO is accessed on an 8-bit or 16-bit basis, it should be accessed for 4 bytes continuously. That is, EPxFIFO should be accessed four times continuously on the 8-bit basis and two times continuously on the 16-bit basis. If accessing is not continued, transmitted/received data cannot be guaranteed.

[Description of Bits]

- FIFOData (FIFO Data): Bits 31 to 0
  This is an FIFO register for transmit/receive data.
  This register is used when reading receive data (or writing transmit data).
  EPa and EPb can assign a FIFO of up to 4 packets for one EP, and EPc through EPf can assign an FIFO of up to 2 packets by the setting of the EPxFIFO assignment register (EPxFIFOAsin). (However, the total capacity should be 4352 bytes or less.)
  In the address when accessing this register, the lower 10 bits are ignored. Therefore, access can be made to this register by any of address increment, address wraparound and address fix methods.

26.3 Functional Descriptions

26.3.1 Description of EndPoints

This block can set up to seven endpoints (hereafter referred to as EP) using registers.
EP0 is exclusively used for control transfer in the transfer mode. However, EPa through Epf can be used for either of bulk transfer or isochronous transfer, as shown in the table below. Furthermore, either OUT or IN data transfer direction can be specified.

EP No.	FIFO size	Double FIFO	PIO mode	Automatic data transfer mode	Transfer type		
					Control	Interrupt	Bulk
0	64 bytes (fixed)	Not supported	Supported	Not supported	OUT	×	×
0	64 bytes (fixed)	Not supported	Supported	Not supported	IN	×	×
a	Programmable	Supported	Supported	Supported	X	OUT/IN	OUT/IN
b	Programmable	Supported	Supported	Supported	X	OUT/IN	OUT/IN
c	Programmable	Supported	Supported	Supported	X	OUT/IN	OUT/IN
d	Programmable	Supported	Supported	Supported	X	OUT/IN	OUT/IN
e	Programmable	Supported	Supported	Supported	X	OUT/IN	OUT/IN
f	Programmable	Supported	Supported	Supported	X	OUT/IN	OUT/IN

* "Two transactions" can be set as the number of transactions per μ frame. Only one transaction can be set for other EPs (EPc through EPf).

About FIFO Size:
FIFO size = (Maximum packet size) x (Number of sides) x (Number of transactions per μ frame)
The maximum packet size and the number of transactions per μ frame are set by the EPx configuration register.
If the maximum packet size is not a multiple of 4, round up to the nearest multiple of 4 and calculate the FIFO size.
(Example: If the maximum packet size is 65 bytes, use 68 bytes for calculation.)
The number of sides is set by the EPxFIFO assignment register.

The FIFO capacity for each transfer can be set using combinations listed in the table below. However, control transfer is fixed to 64 bytes for each IN and OUT of EP0. Furthermore, the total capacity of EPa through EPf cannot exceed 4352 bytes.
If the setting of the total capacity of EPa through EPf exceeds 4352 bytes, operation cannot be guaranteed.

Transfer type	Packet size	FIFO configuration
Control OUT	Up to 64 bytes	EP0 fixed. 64 bytes fixed.
Control IN	Up to 64 bytes	EP0 fixed. 64 bytes fixed.
Interrupt	Up to 1024 bytes	Can set any size to EPa through EPf. The maximum capacity per single-sided FIFO is 1024 bytes, and the maximum number of FIFO sides is two.
Bulk	Up to 512 bytes	Can set any size to EPa through EPf. The maximum capacity per single-sided FIFO is 512 bytes, and the maximum number of FIFO sides is two.

Note: The above settings are allowed regardless of high-speed or full-speed data transfer.
However, the size per single-sided FIFO of bulk transfer at high speed is fixed to 512 bytes.

The following table shows the calculation examples of the FIFO capacity.

EP No.	FIFO size	Number of FIFO sides specified	Transfer type	Number of transactions per μ frame	Calculation of FIFO capacity
0	64 bytes	Single side	Control OUT	1	64 bytes × single side × 1 transaction = 64 bytes
0	64 bytes	Single side	Control IN	1	64 bytes × single side × 1 transaction = 64 bytes
a	512 bytes	Double sides	Bulk OUT	2	512 bytes × double sides × 2 transactions = 2048 bytes
b	512 bytes	Double sides	Interrupt IN	2	512 bytes × double sides × 2 transactions = 2048 bytes
c	64 bytes	Single side	Interrupt OUT	1	64 bytes × single side × 1 transaction = 64 bytes
d	512 bytes	Double sides	Bulk OUT	1	512 bytes × double sides × 1 transaction = 1024 bytes
e	512 bytes	Double sides	Bulk IN	1	512 bytes × double sides × 1 transaction = 1024 bytes
f	Not used	—	—	—	0 bytes

The FIFO size of EP0 (fixed) and EPa to EPf (selectable) listed above totals 6336 bytes, which exceed 4352 bytes. Therefore, the register setting to use all of EP0 and EPa to EPf is impossible.
For example, if EPd and EPe are set as unused, the total capacity will be 4288 bytes.
Using the sample calculations listed above as a reference, set the total capacity of EPa through EPf to a value of 4352 bytes or less.

26.3.2 Description of Automatic Data Transfer

This section describes automatic data transfer. Note that the registers used for automatic data transfer are only the EPa through EPf FIFO registers (EPxFIFO). For the automatic data transfer, refer to Chapter 11.

26.3.2.1 Automatic Data Transfer Enable

Enabling/disabling automatic data transfer is specified by the automatic data transfer configuration register (DMACfg).

- If the DMAEnb bit is "0"
  Automatic data transfer is disabled and the USBDREQ sginal is not asserted.
- If the DMAEnb bit is "1"
  Automatic data transfer is enabled, and if the condition for generating the USB automatic data transfer request signal is satisfied, the USBDREQ signal is asserted.

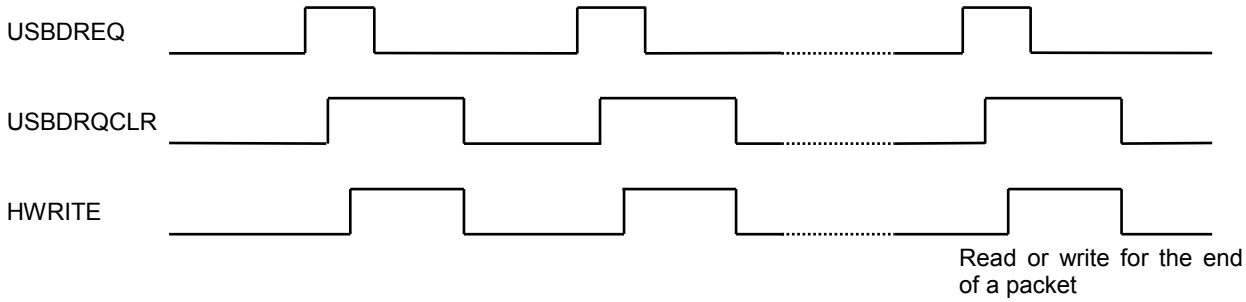
26.3.2.2 Selecting an EP That Uses Automatic Data Transfer

EPs that can use automatic data transfer are EPa through EPf. Which EP uses automatic data transfer can be specified by the DMA_EP bit of the automatic data transfer configuration register (DMACfg).

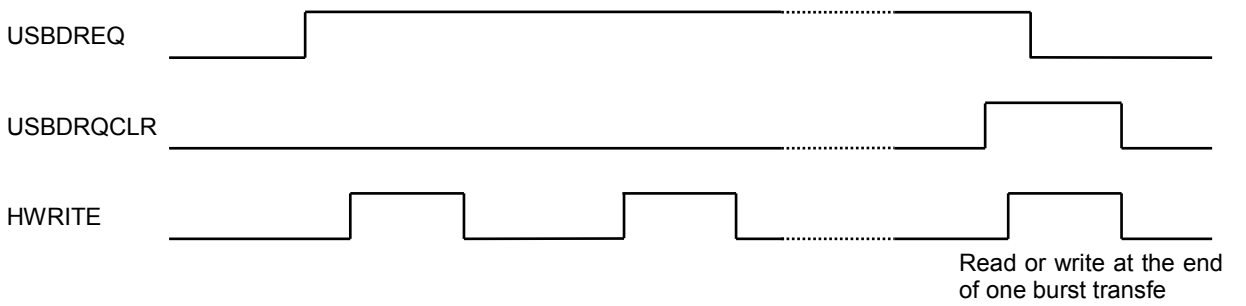
26.3.2.3 Transfer Mode of Automatic Data Transfer

This block supports two types of data transfer modes: cycle steal transfer mode and burst transfer mode.

- Cycle steal transfer mode
  When the DMAStrt bit of the automatic data transfer control register (DMACtl) is "1" and a receive (or transmit) data packet can be read (or written), the USBDREQ signal is asserted. When data of one byte data (one half-word/one word) is read (or written), the USBDREQ signal is deasserted.
  This operation is repeated until the read (or write) of one receive (or transmit) data packet is completed.



- Burst transfer mode
  When the DMAStrt bit of the automatic data transfer control register (DMACtl) is "1" and a receive (or transmit) data packet can be read (or written), the USBDREQ signal is asserted. The assert state is maintained during the period of one burst transfer (until USBDRQCLR is input). Once one burst transfer (USBDRQCLR is input) is completed, USBDREQ is deasserted. This operation is repeated until the read (or write) of one receive (or transmit) data packet is completed.



26.3.2.4 Starting/Stopping Automatic Data Transfer

This block starts automatic data transfer when the DMAStrt bit of the automatic data transfer control register (DMACtrl) is set to "1".

The DMAStrt bit is cleared to "0" if either of the following conditions occurs:

- A USBTC signal is input
- A packet stored in EPxFIFOEP (EPxFIFO) is a short packet
  Therefore, in the case of IN transfer, the microcontroller can set the transfer count exceeding the maximum packet size in automatic data transfer, and can recognize the end of automatic data transfer by the automatic data transfer end interrupt or a short packet receive interrupt from this block.
  In the case of OUT transfer, the microcontroller can set the required transfer count in the automatic data transfer control section, and can recognize the end of automatic data transfer by the automatic data transfer end interrupt.

26.3.2.5 Assert/Deassert Conditions of USBDREQ Signal

The following tables show the assert/deassert conditions of the USBDREQ signal. If the DMAEnb bit is "0 "(DMA disabled) or the DMAStrt bit is "0" (automatic data transfer stopped), the USBDREQ signal remains inactive even if these conditions occur. When forced termination of automatic data transfer is required during read or write operation of one packet of data, execute software reset (writing "1" to SR of SysCtrl) since no other method is available to deassert the USBDREQ signal.

- In the case of bulk transfer or interrupt transfer

EP direction	Assert/deassert condition of USBDREQ signal	
OUT (reception)	Assert condition	When this block detects the EOP of a data packet and valid data without error is stored in FIFO
	Deassert condition	When FIFO's receive data is read as many as the number of bytes indicated in the receive byte count register (RxCnt)
IN (transmission)	Assert condition	When it becomes possible to write transmit data into FIFO
	Deassert condition	When transmit data is written into FIFO as many as the number of bytes indicated in the transmit byte count register (TxCnt)

[Note] In the case of the cycle steal transfer mode, when data of one byte (one half-word/one word) is read (or written), USBDREQ is deasserted once because a USBDRQCLR signal is input. If data to be read (or written) still exists, the USBDREQ signal is asserted again.
In the case of the burst transfer mode, when data of one burst unit is read (or written), the USBDREQ signal is deasserted once because a USBDRQCLR signal is input. If the data to be read (or written) still exists, the USBDREQ signal is asserted again.

26.3.2.6 Setting the Packet Ready Bit at Automatic Data Transfer

When accessing the EPs (EPa through EPf) of this block via automatic data transfer, it is not necessary for the microcontroller to manipulate the packet ready bit of each EP.

- In the case of bulk-IN transfer or interrupt-IN transfer
  When data equal to the number of bytes set in the transmit byte count register (TxCnt) is written into FIFO by the automatic data transfer control section, this block automatically sets the packet ready bit.
  (Write the data into FIFO after writing the transmit data byte count into the transmit byte count register (TxCnt).)
- In the case of bulk-OUT transfer or interrupt-OUT transfer
  When data equal to the number of bytes set in the receive byte count register (RxCnt) is read from FIFO by the automatic data transfer control section, this block automatically resets the packet ready bit.

26.3.2.7 Automatic Data Transfer Width and Alignment

Whereas this block can specify 8-/16-/32-bit data bus width, only 32-bit access is allowed in this LSI. Do not specify 8-/16-bit data bus width.
When accessing, observe the following restrictions:

- If the data bus width is 32 bits
  Output (or input) valid data to all byte lanes (4 byte lanes) for a single access.
  In other words, set the number of transfer bytes to a byte count of an integer multiple of 4, and also set the transfer count to the "number of required transfer bytes/4".
  For example, the following transfer is prohibited in the case of 8-byte transfer.

Byte lane	Example 1			Example 2		
	1st transfer	2nd transfer	3rd transfer	1st transfer	2nd transfer	3rd transfer
HWDATA[31:24] HRDATA[31:24]	Valid data	Valid data	Invalid data	Valid data	Invalid data	Valid data
HWDATA[23:16] HRDATA[23:16]	Invalid data	Valid data	Valid data	Invalid data	Valid data	Valid data
HWDATA[15:8] HRDATA[15:8]	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
HWDATA[7:0] HRDATA[7:0]	Invalid data	Valid data	Valid data	Invalid data	Valid data	Valid data

26.3.3 Description of Interrupts

This block asserts a -USBINT signal, and notifies an interrupt to the microcontroller. The types of interrupt sources are as follows:

- Setup ready for 8-byte setup data
- EP0 receive/transmit packet ready
- EPa - EPf receive/transmit packet ready
- SOF
- USB bus reset assert
- USB bus reset deassert
- Device suspended state
- Device awake
- NAK response
- Short packet reception
- SET_CFG/SET_INTF response
- Status stage transition
- Overrun interrupt

The interrupt causes can be identified by the interrupt status register (IntStt)/EPx status register (EPxStt).
Furthermore, the interrupt causes can be masked individually by the interrupt enable register (IntEnb)/EPx control register (EPxCtl).
The following describes the interrupt causes, conditions and responses.

26.3.3.1 Setup Ready Interrupt

This interrupt occurs at control transfer when the USB host issues a device request by using a control pipe.
When 8-byte setup data is received in the setup stage of control transfer and this data is normally stored in the setup register group, this block sets the EP0 setup ready bit of the EP0 status register (EP0Stt) to "1".
At that time, if the setup ready interrupt enable bit of the EP0 control register (EP0Ctl) is set to "1" in advance, this block sets the setup ready interrupt status bit of the EP0 status register (EP0Stt) to "1" and makes a -USBINT signal active at the same time. After reading all setup data, the microcontroller can clear the setup ready interrupt status bit to "0" by writing "1" into the EP0 setup ready bit, and also can make a -USBINT sigal inactive.
The setup ready interrupt is enabled even after a system reset (hardware reset or software reset).
(Other interrupts are disabled after a system reset.)

Operation	Operation source	Description (condition, response, etc.)
Setup ready interrupt occurred	This block	When 8-byte setup control data is received from the host, and the received data is normally stored in the setup register group. ⇒ The microcontroller can then read the setup register group.
Setup ready interrupt ended	Microcontroller	When "1" is written into the EP0 setup ready bit of the EP0 status register (EP0Stt). If new 8-byte setup data is received before writing "1" into the EP0 setup ready bit, the interrupt is not deasserted. ⇒ The microcontroller discards the previously read setup data. Read new setup data.

This block automatically processes part of standard device requests. No setup ready interrupt occurs in the standard device requests to be processed automatically. Refer to the following table.

Request type	Request generating a setup ready interrupt
Standard device request	GET_DESCRIPTOR, SET_DESCRIPTOR, SYNCH_FRAME
Class request	All class requests
Vendor request	All vendor requests

26.3.3.2 EP0 Receive Packet Ready Interrupt

When one data packet is received from the USB host and the received data is normally stored into the receive FIFO of EP0, this block sets the EP0 receive packet ready bit of the EP0 status register (EP0Stt) to "1".
At that time, if the EP0 receive packet ready interrupt enable bit of the EP0 control register (EP0Ctl) is set to "1" in advance, this block sets the EP0 receive packet ready interrupt status bit of the interrupt status register (IntStt) to "1" and makes a -USBINT signal active at the same time. After reading all data from the receive FIFO, the microcontroller can clear the EP0 receive packet ready interrupt status bit to "0" by writing "1" into the EP0 receive packet ready bit, and also can make a -USBINT signal inactive.
This block recognizes the end of a packet when it receives an EOP (End of Packet) from either a full packet(*1) or short packet(*2).
(*1) Packet having data of the maximum packet size
(*2) Packet of the number of bytes that is less than the maximum packet size

Operation	Operation source	Description (condition, response, etc.)
EP0 receive packet ready interrupt occurred	This block	When this block detects the EOP of a receive data packet and stores valid data without error into the EP0 receive FIFO (EP0RxFIFO)
EP0 receive packet ready interrupt ended	Microcontroller	When "1" is written into the EP0 receive packet ready bit of the EP0 status register (EP0Stt) ⇒ This block can receive data at EP0.

26.3.3.3 EP0 Transmit Packet Ready Interrupt

When a write to the EP0 transmit FIFO (EP0TxFIFO) is ready, this block sets the EP0 transmit packet ready bit of the EP0 status register (EP0Stt) to "1".
At that time, if the EP0 transmit packet ready interrupt enable bit of the EP0 control register (EP0Ctl) is set to "1" in advance, this block sets the EP0 transmit packet ready interrupt status bit of the interrupt status register (IntStt) to "1" and makes a -USBINT signal active at the same time. The microcontroller sets the number of bytes it wants to send to the EP0 transmit byte count register (EP0TxCnt). After completing a write to the EP0 transmit FIFO (EP0TxFIFO), the microcontroller can clear the EP0 transmit packet ready interrupt status bit to "0" by writing "1" into the EP0 transmit packet ready bit, and can also make a -USBINT signal inactive.

Operation	Operation source	Description (condition, response, etc.)
EP0 transmit packet ready interrupt occurred	This block	When it becomes possible to write transmit data into the EP0 transmit FIFO (EP0TxFIFO). For the second and subsequent packets, an interrupt does not occur unless an ACK response is made from the host to the previously transmitted packet in addition to the above condition.
EP0 transmit packet ready interrupt ended	Microcontroller	When "1" is written into the EP0 transmit packet ready bit of the EP0 status register (EP0Stt) ⇒ This block can transmit data from EP0. (It can still transmit data even if the number of write data does not reach the maximum packet size.)

26.3.3.4 EPa - EPf Receive Packet Ready Interrupt

When one data packet is received from the USB host and the received data is normally stored into the receive FIFO of EPx, this block sets the EPx receive packet ready bit of the Epx status register (EPxStt) to "1".
At that time, if both the EPx event interrupt enable bit of the interrupt enable register (IntEnb) and the packet ready interrupt enable bit of the EPx control register (EPxCtl) are set to "1" in advance, this block sets the EPx event interrupt status bit of the interrupt status register (IntStt) and the packet ready interrupt status bit of the EPx status register (EPxStt) to "1" and makes a -USBINT signal active at the same time. After reading all data from the receive FIFO, the microcontroller can clear the packet ready interrupt status bit and the EPx event interrupt status bit to "0" by writing "1" into the EPx receive packet ready bit, and also can make a -USBINT signal inactive.(*1)
This block recognizes the end of a packet when it receives an EOP (End of Packet) from either a full packet (*2) or short packet (*3).

(∗1) If "1" has not been written into the applicable status bit while any of an overrun interrupt, NAK response interrupt and short packet reception interrupt has occurred, the EPx event interrupt status bit stays "1" and a -USBINT signal also stays active.
(∗2) Packet having data of the maximum packet size
(∗3) Packet of the number of bytes that is less than the maximum packet size

Operation	Operation source	Description (condition, response, etc.)
EPx receive packet ready interrupt occurred	This block	When receiving data, this block detects the EOP of a receive data packet and stores valid data without error into the EPx receive FIFO (EPxFIFO)
EPx receive packet ready interrupt ended	Microcontroller	When "1" is written into the EPx receive packet ready bit of the EPx status register (EPxStt) ⇒ This block can receive data at EPx.

26.3.3.5 EPa - EPf Transmit Packet Ready Interrupt

When a write to the EPx transmit FIFO (EPxFIFO) is ready, this block sets the EPx transmit packet ready bit of the EPx status register (EPxStt) to "1".
At that time, if both the EPx event interrupt enable bit of the interrupt enable register (IntEnb) and the packet ready interrupt enable bit of the EPx control register (EPxCtl) are set to "1" in advance, this block sets the EPx event interrupt status bit of the interrupt status register (IntStt) and the packet ready interrupt status bit of the EPx status register (EPxStt) to "1" and makes a -USBINT signal active at the same time. The microcontroller sets the number of bytes it wants to send to the EPx transmit byte count register (EPxTxCnt). After completing a write to the EPx transmit FIFO (EPxFIFO), the microcontroller can clear the packet ready interrupt status bit and the EPx event interrupt status bit to "0" by writing "1" into the EPx transmit packet ready bit, and can also make a -USBINT signal inactive.(*1)

(∗1) If "1" has not been written into the applicable status bit while any of an overrun interrupt, NAK response interrupt and short packet reception interrupt has occurred, the EPx event interrupt status bit stays "1" and a -USBINT signal also stays active.

Operation	Operation source	Description (condition, response, etc.)
EPx transmit packet ready interrupt occurred	This block	When it becomes possible to write transmit data into the EPx transmit FIFO (EPxFIFO). For the second and subsequent packets, an interrupt does not occur unless an ACK response is made from the host to the previously transmitted packet in addition to the above condition.
EPx transmit packet ready interrupt ended	Microcontroller	When "1" is written into the EPx transmit packet ready bit of the EPx status register (EPxStt) ⇒ This block can transmit data from EPx. (It can still transmit data even if the number of write data does not reach the maximum packet size.)

26.3.3.6 SOF Interrupt

This SOF interrupt is enabled if the SOF interrupt enable bit of the interrupt enable register (IntEnb) is set to "1". When this block detects an SOF (Start of Frame) on the USB bus, it sets the SOF interrupt status bit of the interrupt status register (IntStt) to "1" and makes a -USBINT signal active at the same time. The microcontroller can clear the SOF interrupt status bit to "0" by writing "1" into the SOF interrupt status bit, and can also make a -USBINT signal inactive.

Operation	Operation source	Description (condition, response, etc.)
SOF interrupt occurred	This block	When an SOF packet is detected on the USB bus
SOF interrupt ended	Microcontroller	When "1" is written into the SOF interrupt status bit of the interrupt status register (IntStt)

26.3.3.7 USB Bus Reset Assert Interrupt

This USB bus reset assert interrupt is enabled if the USB bus reset assert interrupt enable bit of the interrupt enable register (IntEnb) is set to "1". When this block detects a USB bus reset, it sets the USB bus reset assert interrupt status bit of the interrupt status register (IntStt) to "1" and makes a -USBINT signal active at the same time. The microcontroller can clear the USB bus reset assert interrupt status bit to "0" by writing "1" into the USB bus reset assert interrupt status bit, and can also make a -USBINT signal inactive.

Operation	Operation source	Description (condition, response, etc.)
USB bus reset assert interrupt occurred	This block	When a USB bus reset is detected
USB bus reset assert interrupt ended	Microcontroller	When "1" is written into the USB bus reset assert interrupt status bit of the interrupt status register (IntStt). An interrupt can be cleared even if the bus reset state continues.

26.3.3.8 USB Bus Reset Deassert Interrupt

This USB bus reset deassert interrupt is enabled if the USB bus reset deassert interrupt enable bit of the interrupt enable register (IntEnb) is set to "1". When this block detects the cancellation of a USB bus reset, it sets the USB bus reset deassert interrupt status bit of the interrupt status register (IntStt) to "1" and makes a -USBINT signal active at the same time.

The microcontroller can clear the USB bus reset deassert interrupt status bit to "0" by writing "1" into the USB bus reset deassert interrupt status bit, and can also make a -USBINT signal inactive.

Operation	Operation source	Description (condition, response, etc.)
USB bus reset deassert interrupt occurred	This block	When the cancellation of a USB bus reset is detected
USB bus reset deassert interrupt ended	Microcontroller	When "1" is written into the USB bus reset deassert interrupt status bit of the interrupt status register (IntStt)

26.3.3.9 Device Suspended State Interrupt

This device suspended state interrupt is enabled if the device suspended state interrupt enable bit of the interrupt enable register (IntEnb) is set to "1".
When this block detects a suspended state on the USB bus, it sets the device suspended state interrupt status bit of the interrupt status register (IntStt) to "1" and makes a -USBINT signal active at the same time. The microcontroller can clear the device suspended state interrupt status bit to "0" by writing "1" into the device suspended state interrupt status bit, and can also make a -USBINT signal inactive.

Operation	Operation source	Description (condition, response, etc.)
Suspended state interrupt occurred	This block	When a suspended state is detected on the USB bus. ⇒ When the PD bit of the system control register (SysCtl) is set to "1", this block automatically stops the internal clock after this interrupt occurs. Transition processing to the low supply current consumption mode can be initiated using this interrupt as a trigger.
Suspended state interrupt ended	Microcontroller	When "1" is written into the device suspended state interrupt status bit of the interrupt status register (IntStt)

26.3.3.10 Device Awake Interrupt

This device awake interrupt is enabled if the device awake state interrupt enable bit of the interrupt enable register (IntEnb) is set to "1". When this block detects a resume state (SE0 state immediately after K state) on the USB bus, it sets the device awake state interrupt status bit of the interrupt status register (IntStt) to "1" and makes a -USBINT signal active at the same time. The microcontroller can clear the device awake state interrupt status bit to "0" by writing "1" into the device awake state interrupt status bit, and can also make a -USBINT signal inactive.

Operation	Operation source	Description (condition, response, etc.)
Awake interrupt occurred	This block	When a resume state is detected on the USB bus
Awake interrupt ended	Microcontroller	When "1" is written into the device awake state interrupt status bit of the interrupt status register (IntStt)

This interrupt occurs when a resume state ends; however, oscillation in the power saving mode resumes when a resume state starts (when a K state is detected).

26.3.3.11 NAK Response Interrupt

This NAK response interrupt is valid in EPa through EPf that are set in the transmitting direction.
This interrupt is enabled if the EPx event interrupt enable bit of the interrupt enable register (IntEnb) is set to "1" and the NAK response interrupt enable bit of the EPx control register (EPxCtl) is set to "1". When this block makes a NAK response to an IN token from the USB host, it sets the EPx event interrupt status bit of the interrupt status register (IntStt) and the NAK response interrupt status bit of the EPx status register (EPxStt) to "1" and makes a -USBINT signal active at the same time. The microcontroller can clear the NAK response interrupt status bit and the EPx event interrupt status bit to "0" by writing "1" into the NAK response interrupt status bit, and can also make a -USBINT signal inactive.(*1)

(*1) If "1" has not been written into the applicable status bit while any of an overrun interrupt, short packet reception interrupt and packet ready interrupt has occurred, the EPx event interrupt status bit stays "1" and a -USBINT signal also stays active.

Operation	Operation source	Description (condition, response, etc.)
NAK response interrupt occurred	This block	When a NAK response is made to an IN token from the USB host
NAK response interrupt ended	Microcontroller	When "1" is written into the NAK response interrupt status bit of the EPx status register (EPxStt)

26.3.3.12 Short Packet Receive Interrupt

This short packet receive interrupt is valid in EPs that are set in the receiving direction.
This interrupt is enabled if the EPx event interrupt enable bit of the interrupt enable register (IntEnb) is set to "1" and the short packet receive interrupt enable bit of the EPx control register (EPxCtl) is set to "1". When this block receives a short packet, it sets the EPx event interrupt status bit of the interrupt status register (IntStt) and the short packet receive interrupt status bit of the EPx status register (EPxStt) to "1" and makes a -USBINT signal active at the same time. The microcontroller can clear the short packet receive interrupt status bit and the EPx event interrupt status bit to "0" by writing "1" into the short packet receive interrupt status bit, and can also make a -USBINT signal inactive.(*1)

(*1) If "1" has not been written into the applicable status bit while any of an overrun interrupt, NAK response interrupt and packet ready interrupt has occurred, the EPx event interrupt status bit stays "1" and a -USBINT signal also stays active.

Operation	Operation source	Description (condition, response, etc.)
Short packet receive interrupt occurred	This block	When this block receives a short packet
Short packet receive interrupt ended	Microcontroller	When "1" is written in the short packet receive interrupt status bit of the EPx status register (EPxStt)

For EPs that have a FIFO with double sides or more, this interrupt is generated when a short packet can be accessed from the microcontroller (i.e., the reading of the previous packet is completed, and the FIFO side that can be accessed from the microcontroller is switched to the FIFO side where a short packet is stored).
This interrupt is also generated for 0-byte short packets.

26.3.3.13 SET_CFG/SET_INTF Response Interrupt

This SET_CFG/SET_INTF response interrupt is enabled if the SET_CFG/SET_INTF response interrupt enable bit of the EP0 control register (EP0Ctl) is set to "1". If this block completes the transmission of an ACK packet(*1) in the setup stage of a SET_CONFIGURATION request (or SET_INTERFACE request), it sets the SET_CFG/SET_INTF response interrupt status bit of the EP0 status register (EP0Stt) to "1" and makes a -USBINT signal active at the same time.
The microcontroller can clear the SET_CFG/SET_INTF response interrupt status bit to "0" by writing "1" into the SET_CFG/SET_INTF response interrupt status bit, and can also make a -USBINT signal inactive.
(*1) The transmission of ACK packets is automatically performed by this block.

Operation	Operation source	Description (condition, response, etc.)
SET_CFG/SET_INTF response interrupt occurred	This block	When this block completes the transmission of an ACK packet in the setup stage of a SET_CONFIGURATION request (or SET_INTERFACE request)
SET_CFG/SET_INTF response interrupt ended	Microcontroller	When "1" is written into the SET_CFG/SET_INTF response interrupt status bit of the EP0 status register (EP0Stt)

The following items can be checked by reading the EP0 status register (EP0Stt) after this interrupt occurs:

- Cause of the occurrence of this interrupt (SET_CONFIGURATION request or SET_INTERFACE request)
- Configuration value of SET_CONFIGURATION request
- Interface value and alternate value of SET_INTERFACE request

26.3.3.14 Status Stage Transition Interrupt

This status stage interrupt is enabled if the status stage transition interrupt enable bit of the EP0 control register (EP0Ctl) is set to "1".

If a transition is made to the status stage during the execution of control transfer(*1) other than standard device requests to be processed automatically, this block sets the status stage transition interrupt status bit of the EP0 status register (EP0Stt) to "1" and makes a -USBINT signal active at the same time. The microcontroller can clear the status stage transition interrupt status bit to "0" by writing "1" into the status stage transition interrupt status bit, and can also make a -USBINT signal inactive.

(*1) This interrupt occurs during any of control write transfer, control read transfer and no-data control transfer.

Operation	Operation source	Description (condition, response, etc.)
Status stage transition interrupt occurred	This block	When moving to the status stage while this block is executing control transfer. This block sends a NAK packet to a token to be transmitted from the host in the status stage while this interrupt is being generated.
Status stage transition interrupt ended	Microcontroller	When "1" is written into the status stage transition interrupt status bit of the EP0 status register (EP0Stt). (At that time, this interrupt is automatically disabled(*2).) This block sends an ACK packet to a token to be transmitted from the host in the status stage after this interrupt ends.

(*2) When the microcontroller writes "1" into the status stage transition interrupt status bit, the status stage transition interrupt enable bit is automatically cleared to "0" and this interrupt is disabled. To use this interrupt again, it is necessary to set the status stage transition interrupt enable bit to "1" again.

26.4 Timings

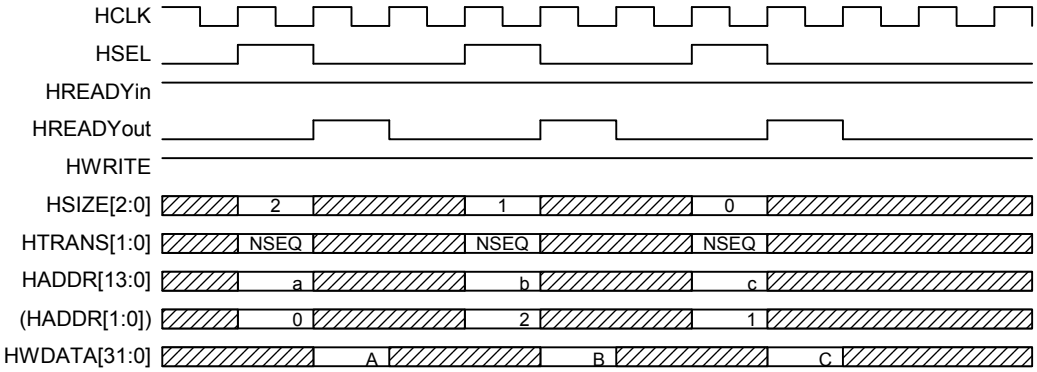
This chapter describes the operation timings of this block.

26.4.1 AHB Interface Timings

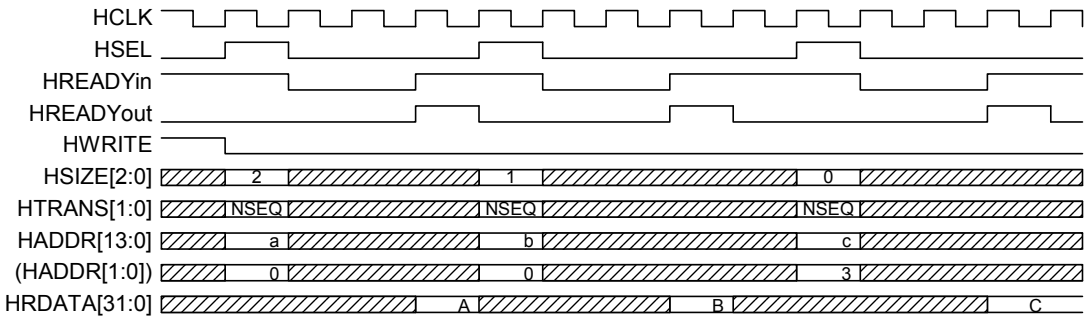
This section describes the operation timings of the AHB interface.

26.4.1.1 Register Access

(1) Write (AHB→Registers)

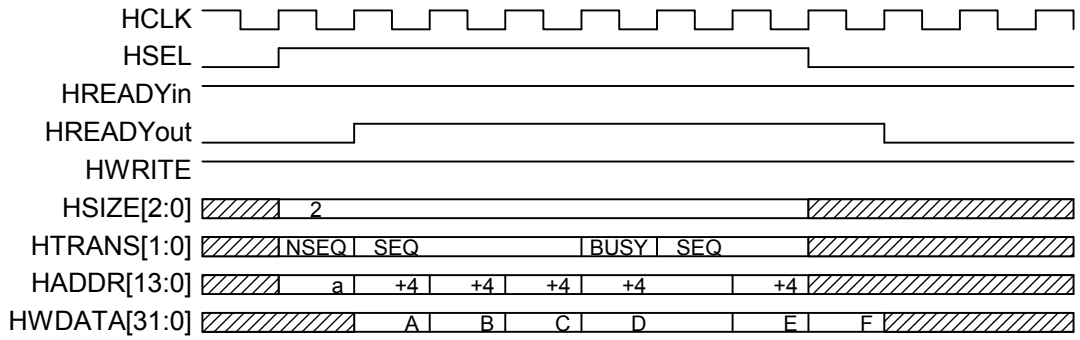


(2) Read (AHB←Registers)



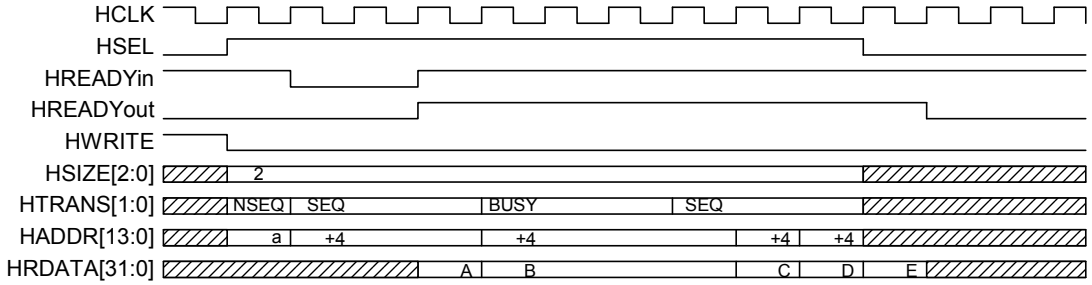
26.4.1.2 EPxFIFO Access

(1) Write (AHB→Registers)

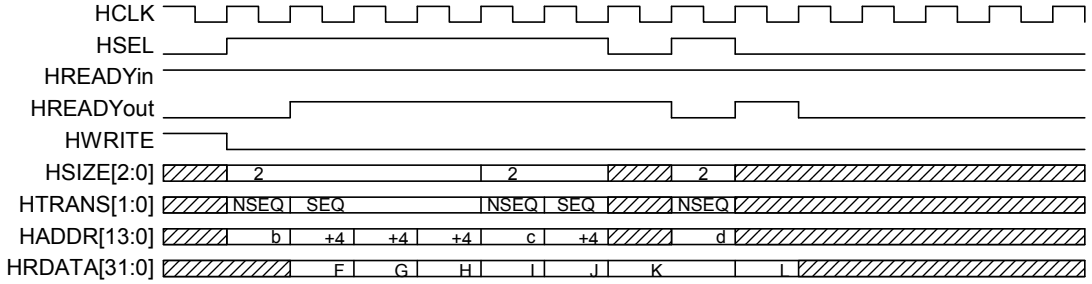


(2) Read (AHB←Registers)

When data at the beginning of a packet is read.



When data subsequent to the beginning of a packet is read.



A 2-clock wait is inserted when reading one double word at the beginning of a packet. Thereafter, no wait is inserted until the entire one packet is read.

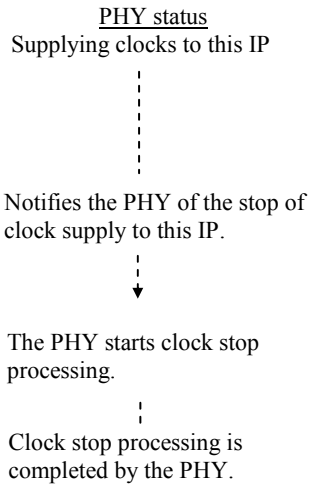
26.5 Notes on Use

26.5.1 How to Stop or Resume Clocks

This block operates with clocks from the USB PHY module (physical layer). Therefore, the clocks of this block can be stopped by stopping clocks supplied to the PHY, in addition to using the clock stop function (see Section 26.3.3.9 for more information) exclusively available to this block via suspend status detection on the USB bus. The following describes the procedure for stopping and resuming the clocks of this block by stopping/resuming clocks supplied to the PHY.

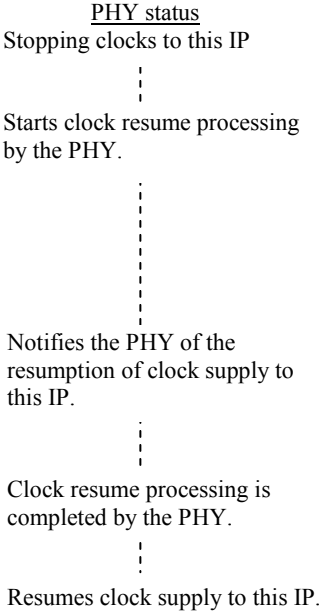
(1) Procedure for Stopping Clocks

- Clock stop setting
  Set the PUCtl bit to "0" and the PD bit to "1" of the system control register (SysCtl).
- Wait for the completion of clock stop processing executed internally in the PHY
  Wait at least 10 ms until the clock stop processing executed by the PHY is completed.
- Stop clock supply to the PHY
  Execute processing to stop the clock source that supplies clocks to the PHY.



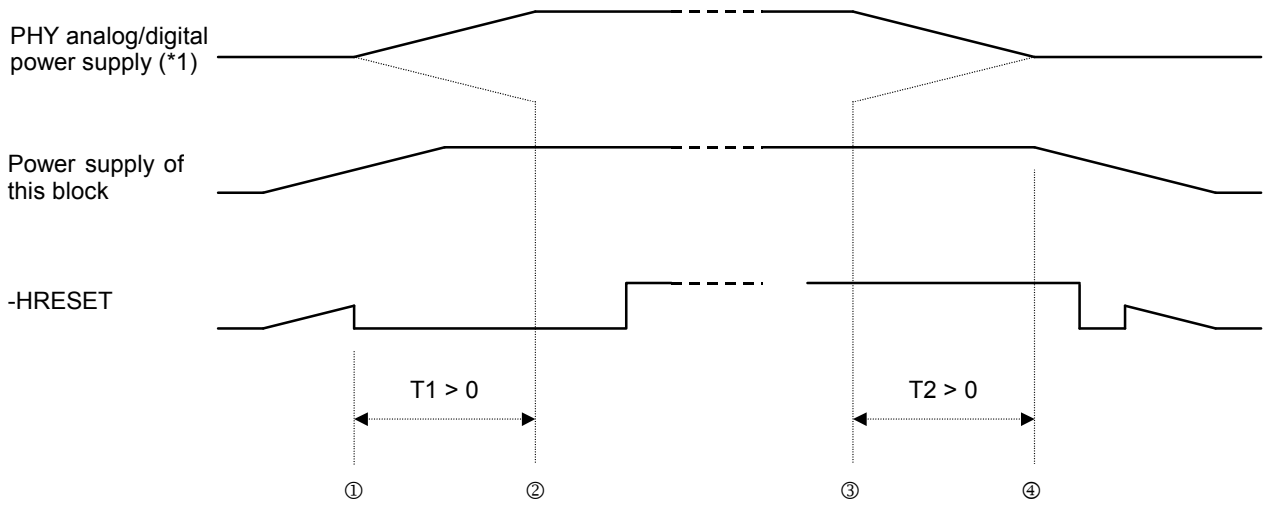
(2) Procedure for Resuming Clocks

- Resume clock supply to the PHY
  Execute processing to resume the clock source that supplies clocks to the PHY.
- Wait for the period of clock source oscillation stabilization time
  Wait until the oscillation of the clock source that supplies clocks to the PHY stabilizes(*1).
- Clock resume setting
  Set the PUCtl bit of the system control register (SysCtl) to "1"(*2).
- Wait for the completion of clock resume processing executed by the PHY
  Wait at least 5 ms until clock resume processing executed by the PHY is completed.



(*1): Because this wait time is dependent on the clock source in use, the user should determine the wait time.
(*2): The PD bit is included in the system control register (SysCtl). If the PUCtl bit is set to "1", overwrite the PD bit with "1" set when clock supply is stopped. If the PD bit is set to "0", overwrite the PD bit with "1" once, and then set it to "0" again.

26.5.2 Power-On Sequence

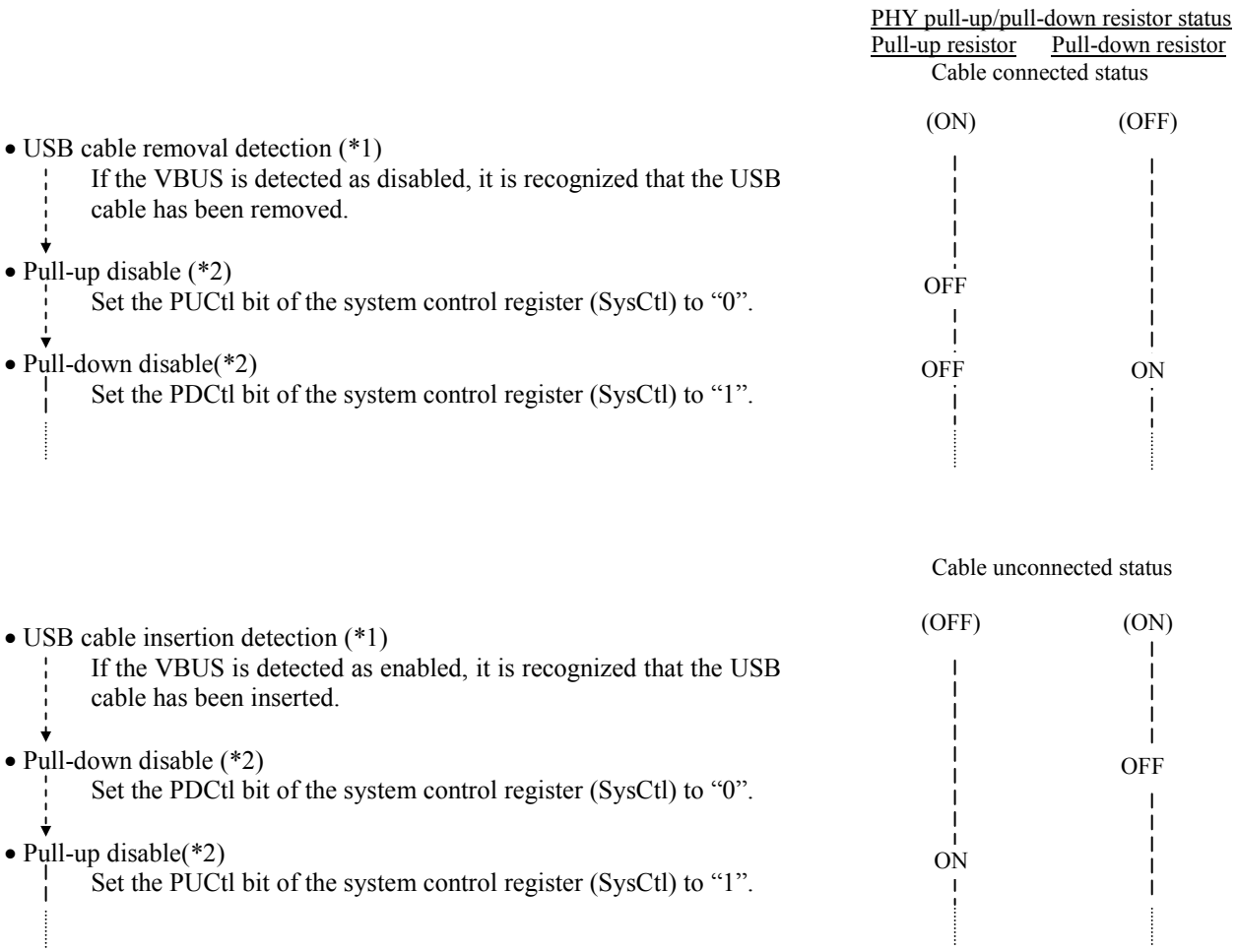


(∗1) Comply with the PHY Specification regarding the PHY digital/analog power-on and shutdown sequence.
- ① When HRESET is asserted, and when the analog power supply and digital power supply of the PHY are turned ON
- ② When the analog power supply and digital power supply of the PHY are turned ON
- ③ When the analog power supply and digital power supply of the PHY are shut down
- ④ When the power supply of this block is shut down

In the figure above, set T1 (from asserting HRESET to turning ON the analog power supply and digital power supply of the PHY) and T2 (from shutting down the analog power supply and digital power supply of the PHY to shutting down the power supply of this block) to "0" or higher.

26.5.3 Processing When a USB Cable Is not Connected

When a USB cable is not connected, the current when stationary can be reduced by setting the PDCtl bit to "1." The following shows the processing sequence in this case.



(*1): Whether the USB cable is removed or inserted is recognized by whether the VBUS is enabled or disabled. If the VBUS is enabled, it is recognized that the USB cable is inserted. If the VBUS is disabled, it is recognized that the USB cable is removed.

(*2): The PUCtl bit and PDCtl bit are assigned to the same system control register (SysCtl); however, be sure to set each of these bits individually.
For example, to execute "Pull-Up Disable to Pull-Down Enable" of the above sequence, it is necessary to set the applicable PUCtl bit to "0" and write it into the system control register (SysCtl) first, and then set the applicable PDCtl bit to "1" and write it into the system control register (SysCtl) next, rather than set the applicable PUCtl bit to "0" and the applicable PDCtl bit to "1", and then write them into the system control register (SysCtl) at once.

26.5.4 Processing When a USB Cable Is not Connected

USB 2.0 High-Speed is used to transmit signals using a 480 Mbps high-speed signal/400 mV small amplitude differential signal. Therefore, to design a board for a product embedded with this USB 2.0 PHY, observe the following points by conforming to the USB 2.0 Specification.
Be aware that the following items are merely notes on designing boards stated above, and are not intended to guarantee operations in any case. It is mandatory that the boards to be used by the user should be checked by the user in the actual operating environment as a final check.

26.5.4.1 USB Differential Signal (dp/dm)

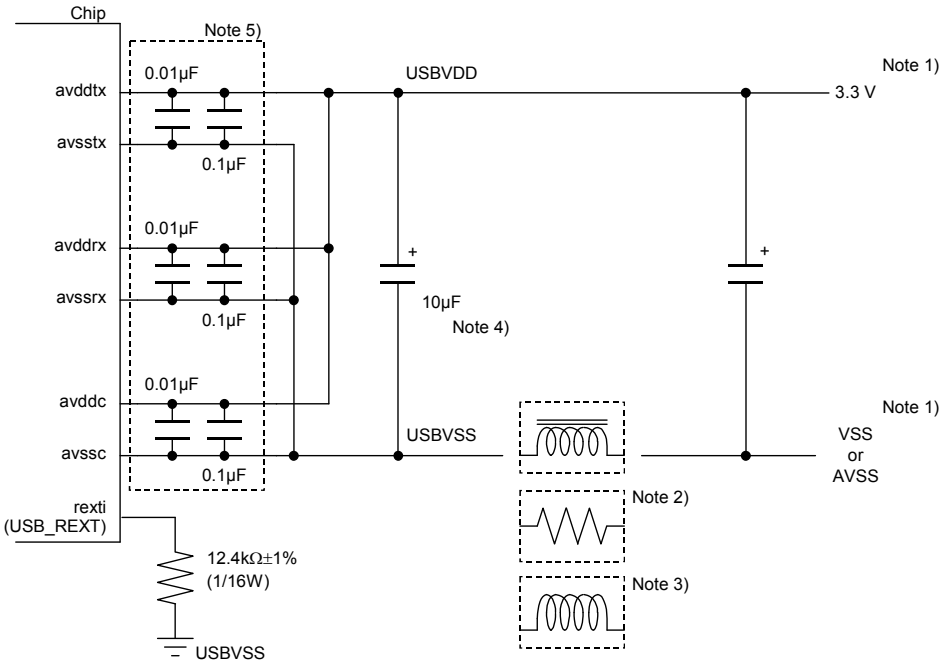
- The distance of a wire between the dp/dm pin and the USB receptacle should be less than 10 cm or less. (It is recommend that the wire length be less than 5 cm or less.)
- Route the wire of the dp/dm pin in parallel so that it has the equal length and width. Also, route in such a way that the difference in the wire lengths between the dp and dm pins is within 1 mm.
- Allocate as much distance as possible between the signal lines of the dp and dm pins and other signal lines (especially singal lines with significant changes such as clocks). Also, shielding both sides of the dp/dm pin with a ground will prevent them from being affected by other signals.
- Give consideration to the interlayer of the board and the wire width and thickness of the dp/dm pin so that the differential impedance and single end impedance are 90Ω and 45Ω, respectively.
- Construct a ground layer below the wiring layer of the dp/dm pin in order to adjust the wire impedance of the dp/dm pin.
- Wire the dp/dm pin on the same layer, and avoid constructing a through hole between the dp/dm pin and the USB receptacle.
- Avoid bending wires as much as possible. If wires should be bent, use a small bending angle as much as possible. For example, if a wire should be bent by 90 degrees, do not do so by bending it only once; instead, bend a wire by 30 degrees three times or by 45 degrees twice. Also, it is recommended that the edge area be rounded so that the wire width of the corner area is the same as the wire width of the linear area.
- Do not divide the power supply plane between the wiring of the dp/dm pin and the USB receptacle.
- Do not cross other signal lines between the wiring of the dp/dm pin and the USB receptacle.
- Do not branch the wiring of the dp/dm pin.

26.5.4.2 USB Power Supply

- Provide USBVDD/USBVSS dedicated to the analog power supply of the USB 2.0 PHY IP. Also, when separating on the same layer, provide a dedicated island for separating it.
- Set up USBVDD/USBVSS so that noise does not enter from other VDD/VSS via a filter. If there is another power supply for an analog circuit (AVDD/AVSS) on the chip as other VDD/VSS, connect USBVDD/USBVSS with AVDD/AVSS . In this case, repalce "VDD/VSS" with "AVDD/AVSS" in the description below.
- Connect USBVSS and VSS using a ferrite bead, resistor or inductor as shown in Figure 26-3. It is recommended that a ferrite bead be used. When using a resistor, set the resistance to 2Ω or larger to suppress the potential difference between USBVSS and VSS. When using an inductor, avoid resonance with a noise frequency.
- As shown in Figure 26-3, install a capacitor of 10μF or larger between USBVDD and USBVSS, and at least one capacitor each of 0.1μF and 0.01μF between avddtx and avsstx, between avddrx and avssrx, and between avddc and avssc near the chip. Also, connect the capacitors to USBVDD/USBVSS using through holes so that the distance between the capacitors and the pad of the chip becomes shorter.
- Do not use electrolytic capacitors, but use tantalum capacitors and ceramic capacitors that are superior in high frequency characteristic.
- If a regulator is shared by USBVDD/USBVSS and VDD/VSS, separate the power supply near the regulator output using a ferrite bead, resistor or inductor, as described previously. In this case, it is especially recommended that ferrite beads be used.

26.5.4.3 Reference Resistor (USB_REXT)

- Install a resistor of 12.4 kΩ ±1% (rated power: 1/16W) between USB_REXT and USBVSS near the chip.
- Because the USB 2.0 PHY IP operates using this resistor as the reference, install the USB 2.0 PHY IP near the chip, and prevent noise from entering. Also, under no circumstances, do not use resistors that exceed the resistance value tolerance.



Note 1) Connect the power supply to AVDD/AVSS if there is the analog circuit power supply (AVDD/AVSS) other than the digital circuit power supply. In such case, change "VDD/VSS" to "AVDD/AVSS" in the following description.

Note 2) When connecting USBVSS and VSS, keep the resistance to 2Ω or less to suppress the potential difference between USBVSS and VSS.

Note 3) When connecting an inductor across USBVSS and VSS, adjust the inductance to avoid resonance with a noise frequency according to the board capacitance.

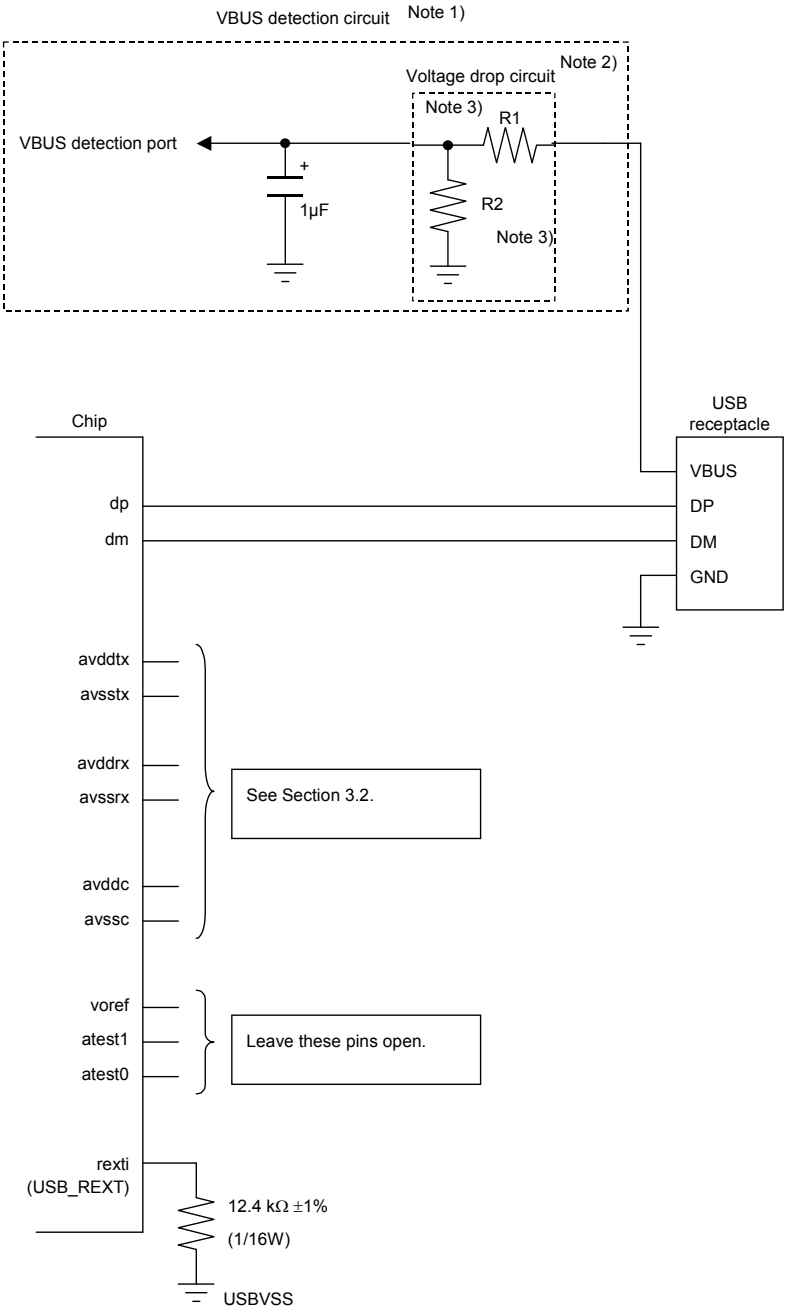
Note 4) Install a capacitor of 10µF or larger across USBVDD and USBVSS.

Note 5) Install at least one capacitor of 0.1µF and at least one capacitor of 0.01µF between avddtx and avsstx, between avddrx and avssrx, and between avddc and avssc in the vicinity of the chip. It is recommended that the capacitors be installed as many as possible.

The power supply separation method and number of capacitors to be installed vary depending on the environment used. Thus, adjust them according to the board used.

Figure 26-2 Example of Power Supply Circuit

26.5.5 Example of Application Circuit



Note 1) Install this circuit to detect a VBUS. Because there is no pin for detecting a VBUS in the USB 2.0 PHY IP, connect it to an interrupt port of the CPU.
Note 2) The VBUS voltage is 5V (typical). Install a voltage drop circuit according to the specification of the input port to be used for VBUS detection.
Note 3) Set the R1 and R2 values so that they are orders of 10kΩ.

Figure 26-3 Example of Application Circuit

26.5.6 Restriction on Using the USB HUB

Use only up to four USB hub levels when using the USB 2.0 module of this LSI.

26.5.7 Note on accessing EP FIFO register

The data from the EP FIFO, transferred by the CPU may be broken, when the CPU using the word access accesses the EP FIFO during the DMAC accesses another EP FIFO. To prevent this problem, the following two countermeasures are recommended.

1. Stop the DMAC accessing the EP FIFO, during the CPU access to the EP FIFO
Before the CPU reads the EP FIFO, save the content of the DMACfg register and the DMACtl register, then set to "0" into the DMAEnb bit of the DMACfg register and the DMAStrt bit of the DMACtl register.
After the CPU finish reading the EP FIFO, restore the content of the DMACfg register and the DMACtl register, then the DMAC restart the data transfer from the EP FIFO.

2. Not use the word access for transferring the data from the EP FIFO by the CPU
Use the halfword access or the byte access instead of using the word access for transferring the data from the EP FIFO by the CPU.

Chapter 27

Built-In Flash ROM

Chapter 27 Built-In Flash ROM

27.1 Overview

The ML69Q6203[*1] has built-in Flash ROM. The Flash ROM is electrically programmable, non-volatile memory; thus, this LSI provides simple and powerful, multiple programming methods. The features of the built-in Flash ROM are shown below.

∗1: The ML696201 has no built-in Flash ROM.

Features

- Size of built-in Flash ROM:
  ML69Q6203: 512 Kbytes (4 Mbits)
- Flash ROM can be read and programmed with a single power supply.
- Built-in Flash ROM programming methods:
  Programming using the JTAG (Joint Test Action Group) debug function
  Programming using the boot mode
- Programming unit: 2 bytes
- Erasing units:
  Sector erase: 2 Kbytes per sector
  Block erase: 64 Kbytes per block
  Chip erase: Entire area in a batch
- High-speed programming
  Programming time: 2 bytes/20 μs
- High-speed erasing
  Erasing time:
  - Sector erase/block erase: 25 ms
  - Chip erase: 100 ms
- Write protect function
  Block protection: 16 Kbytes on the top address side
  Chip protection: Entire built-in Flash ROM area
- Time required to turn on or cancel write protect:
  For turning on block protection/chip protection: 20 μs
  For canceling block protection/chip protection: 25 ms
- Read cycle time: 50 ns
- Control by JEDEC-conformed SDP (Software Data Protect) command sequence
- Highly reliable read/write
  Sector writing: 10,000 cycles
  Data holding period: 10 years

27.1.1 Block Diagram

Figure 27-1 shows the connection of the built-in Flash ROM control signals.

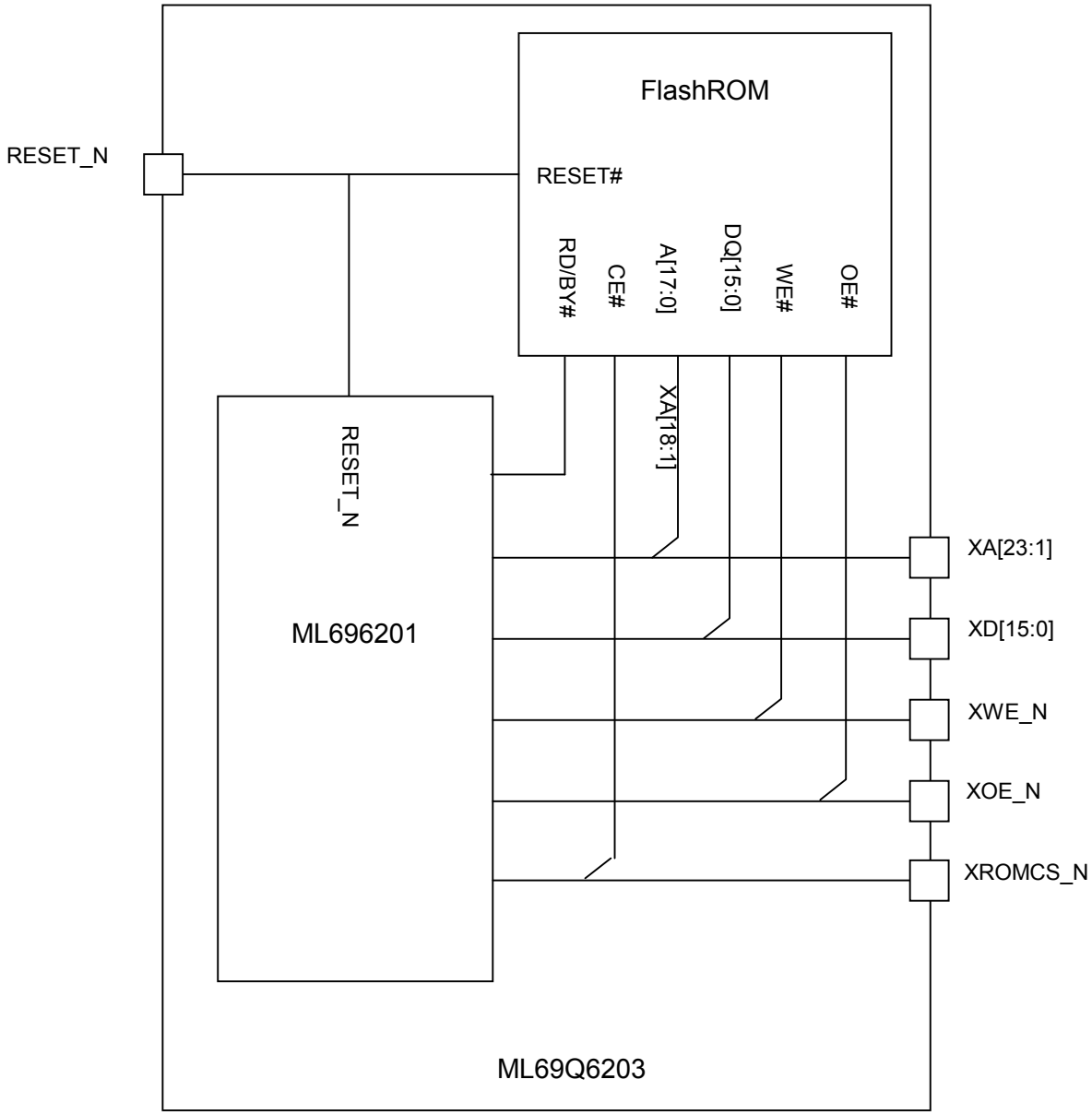


Figure 27-1 Connection of the Built-In Flash ROM Control Signals

27.2 SDP Command Sequence Control for Built-In Flash ROM

27.2.1 Outline of Operation

The built-in Flash ROM is equipped with a command register to facilitate interface and control. Reading, erasing, programming, and other functions required for built-in Flash ROM are executed via the command register.
Commands are written into the command register by the SDP (Software Data Protect) command sequence employing standard JEDEC commands. A command list is shown below, and Sections 27.2.2 to 27.2.11 gives the functional description.

Command Sequence List

Command	Req. cycle	1st cycle		2nd cycle		3rd cycle		4th cycle		5th cycle		6th cycle	
		Address Note 1	Data Note 2	Address Note 1	Data Note 2	Address Note 1	Data Note 2	Address Note 1	Data Note 2	Address Note 1	Data Note 2	Address Note 1	Data Note 2
Read/Reset	1	XXX	F0										
	3	555	AA	2AA	55	555	F0	RA Note 3	RD Note 3				
Software ID Entry/Verify Protect	3	555	AA	2AA	55	555	90	RA Note 4	RD Note 4				
Program	4	555	AA	2AA	55	555	A0	PA Note 3	PD Note 3				
Sector Erase	6	555	AA	2AA	55	555	80	555	AA	2AA	55	SA Note 3	30
Block Erase	6	555	AA	2AA	55	555	80	555	AA	2AA	55	BA Note 3	50
Chip Erase	6	555	AA	2AA	55	555	80	555	AA	2AA	55	555	10
Block Protect	4	555	AA	2AA	55	555	E0	XXX	00				
Chip Protect	4	555	AA	2AA	55	555	D0	XXX	00				
Cancel Protect	4	555	AA	2AA	55	555	E0	XXX	01				

Note 1: An address should be input by shifting it to the left by 1 bit.
- Address format: XA11 to XA1 (hexadecimal notation)
- XA17 to XA12 are don't cares.

Note 2:
- Data format: DQ7 to DQ0 (hexadecimal notation)
- DQ15 to DQ8 are don't cares.

Note 3:
- RA: Read address, PA: Program address, SA: Sector address (XA17 to XA11)
- RD: Read data, PD: Program data, SD: Sector data

Note 4:
- RA: Read address, RD: Read data. Refer to Sections 27.4.8 and 27.4.9.

[Note]
Be sure to use half-word (16-bit) access for a command sequence.

27.2.2 Command Entry

For command entry, use the JEDEC-conformed SDP command sequence. After the SDP command sequence is finished, the selected operation is initiated automatically. If an incorrect address or data is entered in the SDP command sequence, the SDP command sequence is suspended and the mode returns to read mode.

27.2.3 Read/Reset (Software Reset)

The Read/Reset command is used to end the Software ID Entry or Verify Protect operation, or stop an erase or programming operation.
A Read/Reset operation is carried out by entering an SDP command of either one cycle or three cycles to the command register. The mode then returns to the read mode.
The Flash ROM automatically enters the read mode when the power is turned on.

27.2.4 Erase

An erase operation is executed by entering an SDP command of six cycles—the Sector Erase command, Block Erase command, or Chip Erase command—to the command register, and is ended automatically by the control of the internal timer within the Flash ROM.
During an erase operation, DATA_N poling that performs detection of internal operations, the toggle bit, hardware reset, and software reset become valid.
The Sector Erase command places the selected 2-Kbyte area in the built-in Flash ROM in the "1" state.
The Block Erase command places the selected 64-Kbyte area in the built-in Flash ROM in the "1" state.
The Chip Erase command places the entire built-in Flash ROM area in the "1" state.

27.2.5 Programming

A programming operation is carried out by entering an SDP command of four cycles—the Program command—to the command register, and is ended automatically by the control of internal timing.
During a programming operation, DATA_N poling that performs the detection of internal operations, the toggle bit, hardware reset, and software reset become valid.
Note that the memory area at the address where programming is to be carried out must be placed in the erased state prior to programming.
Programming is performed in units of 16 bits (2 bytes).

Caution:
It is prohibited to re-program to the address of the area where programming has already been carried out without erasing the existing program first.

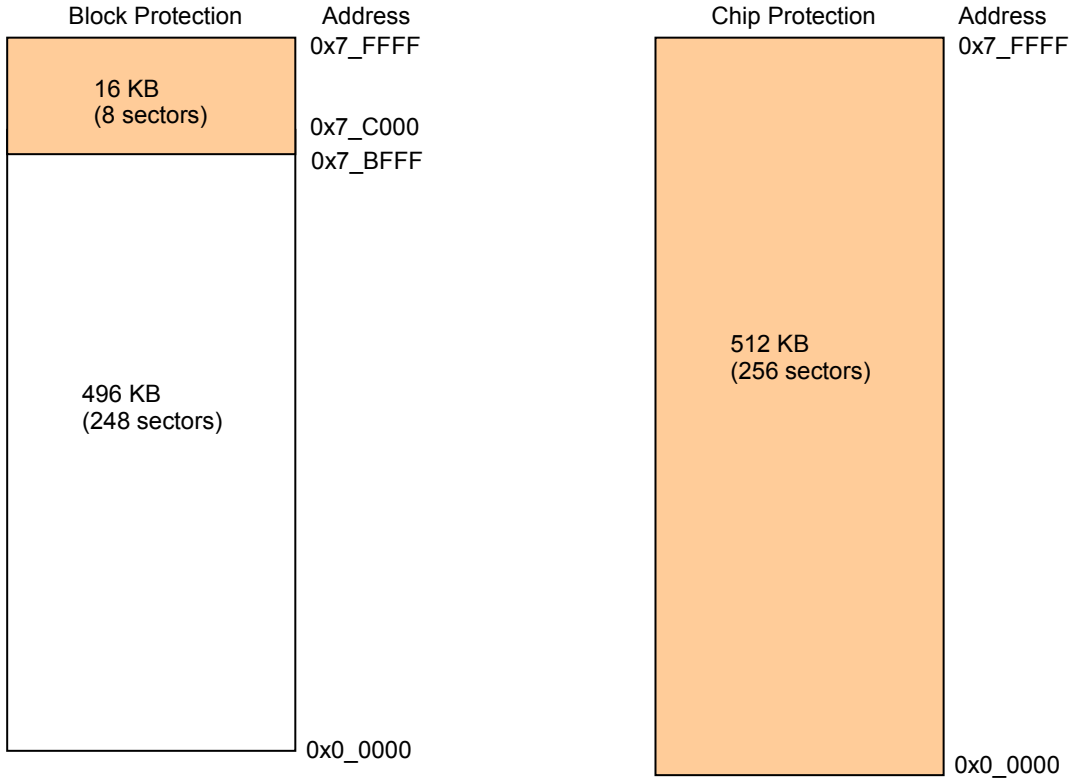
27.2.6 Write Protect

The write protect operation disables erase and programming operations in the specified areas.
There are two types of write protect: block protection and chip protection. Each type of write protect is activated by entering an SDP command of four cycles—the Block Protect or Chip Protect command—to the command register.
Block Protect protects an address space of 16 Kbytes (8 sectors) from the top address side, and Chip Protect protects the address space of the entire chip area.
Neither block protection nor chip protection will be canceled even if the power is turned off.

27.2.7 Cancellation of Write Protect

Write protect is canceled by entering an SDP command of four cycles—the Cancel Protect command—to the command register.
Executing the Cancel Protect command cancels both block protect and chip protect at the same time.

Areas to be write-protected in ML69Q6203



27.2.8 Product Identification/Software ID Entry

The product identification/software ID entry operation outputs the manufacturer code and device code, which are then read by the programming device, making it possible to select erase and programming algorithms suitable for the device.

The manufacturer code and device code are output by an SDP command of four cycles—the Software ID Entry command—to the command register. The following contents are output by the last read operation:

	Address	Output	Description
ML69Q6203	0x0000	0x0062	Manufacturer code
	0x0001	0x0002	Device code (ML69Q6203)

This operation ends when the Read/Reset command is entered.

27.2.9 Write-Protection Verification (Verify Protect Command)

Whether or not Flash ROM is write-protected is verified by entering an SDP command of four cycles—the Verify Protect command—to the command register. The following contents are output by the last read operation:

	Address	Output	Description
ML69Q6203	0x0002	0x0001	Protected.
		0x0000	Not protected.

This operation ends when the Read/Reset command is entered.

27.2.10 Hardware Reset

The hardware reset function stops an erase or programming operation, or cancels the mode entered by an SDP command.

27.2.11 Detecting the End of an Erase or Programming Operation

The end of an erase or programming cycle can be detected by DATA_N polling or the toggle bit.

(1) DATA_N Polling

DATA_N polling can be used to detect the end of an erase or programming cycle.
During an erase cycle, "0" is read into DQ7*, and upon completion of the erase cycle, "1" is read into DQ7.
During a programming cycle, an inverted value of the data loaded last is read into DQ7, and upon completion of the programming cycle, the value of data loaded last is read into DQ7.
DATA_N polling can be monitored any time during an erase or programming cycle. To avoid erroneous detections by DATA_N polling, verify the two consecutive detections of the end.
Note that, in order to make DATA-N polling function correctly, it is essential to erase the existing program first before programming.
Figure 27-2 shows the flowchart for detecting the end of an erase or programming cycle by using DATA_N polling:

*: DQ7 is bit 7 of a built-in Flash ROM data output. See Figure 27-1.

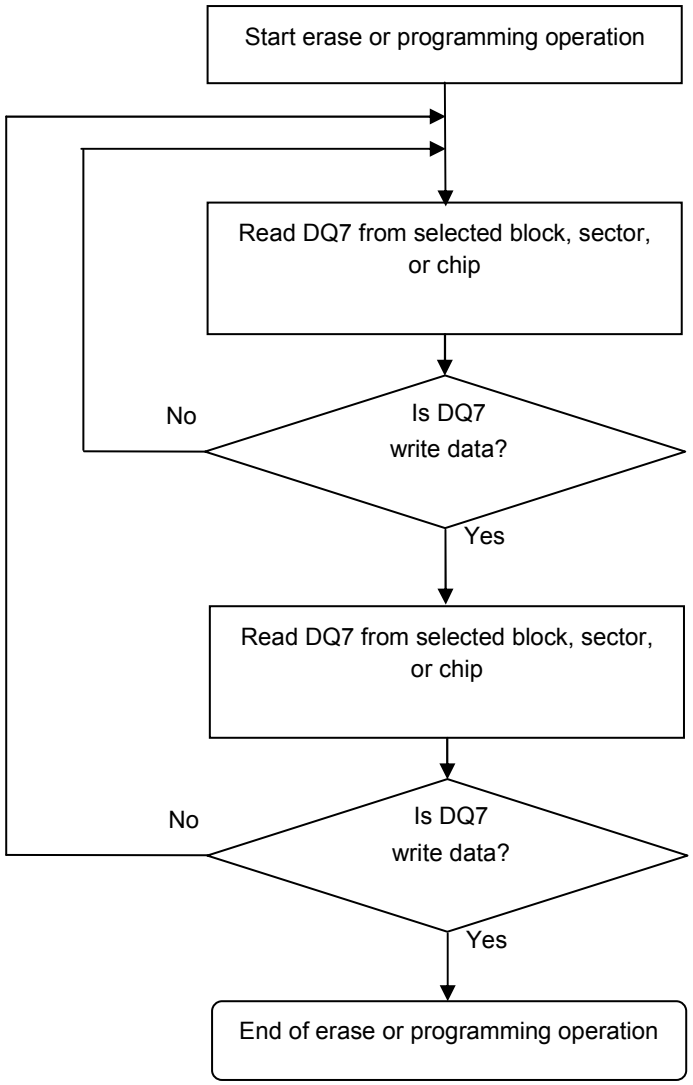


Figure 27-2 Flowchart of Detecting DATA_N Polling for End of Operation

(2) Toggle Bit

The toggle bit can be used to detect the end of an erase or programming cycle.
During an erase or programming cycle, "0" and "1" are continuously read out from DQ6* alternately. When an erase or programming cycle finishes, toggling stops and the cycle returns to the normal read cycle. The toggle bit can be monitored any time during an erase or programming cycle. To avoid erroneous detections by the toggle bit, verify the two consecutive detections of the end.
Figure 27-3 shows the flowchart for detecting the end of an erase or programming cycle by using the toggle bit.

*: DQ6 is bit 6 of a built-in Flash ROM data output. See Figure 27-1.

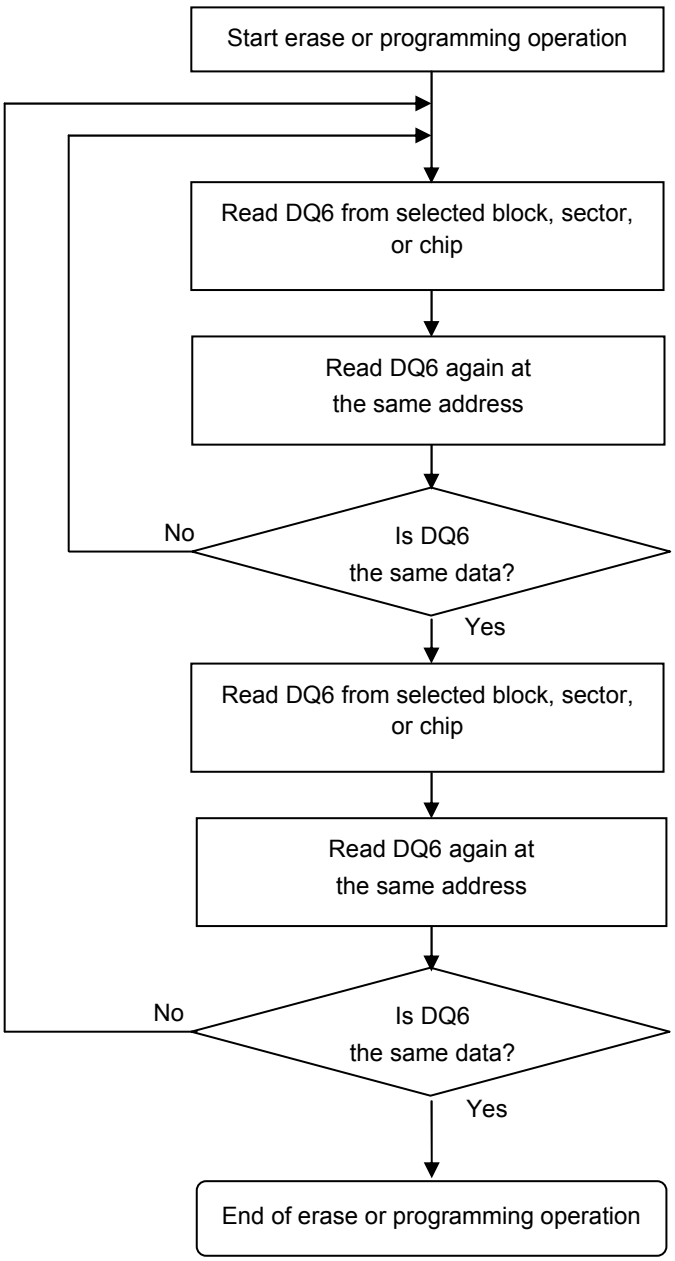


Figure 27-3 Flowchart of Detecting End of Operation by Toggle Bit

27.3 Built-In Flash ROM Programming

27.3.1 Outline of Built-In Flash ROM Programming

The following two methods are available to program the built-in Flash ROM of this LSI.

- Programming using the JTAG debug function
  This is a method of programming the built-in Flash ROM using the JTAG debug function. This method allows on-board programming.
  Using the built-in debug function, this method loads the built-in the Flash ROM programming routine into the built-in RAM or an external SRAM on the board via the JTAG interface, executes the routine, and programs the built-in Flash ROM.

- Programming using the boot mode
  This is a method of programming the built-in Flash ROM using USB from a PC. This method allows on-board programming.
  By using software on a PC, it is easy to write a program into the built-in Flash ROM via USB.

For the setting of the access timing to the built-in Flash ROM, see Chapter 10, External Memory Controller.

27.3.2 Method of Programming the Built-In Flash ROM Using the JTAG Debug Function

This method programs the built-in Flash ROM using the JTAG debug function. In this method, the built-in Flash ROM is programmed by loading the built-in Flash ROM programming routine into the built-in RAM or an external SRAM on the board via the JTAG interface and then executing the routine.
The built-in Flash ROM is programmed when the built-in Flash ROM programming routine issues a command conforming to the JEDEC standard SDP command sequence. The downloading and starting of the built-in Flash ROM programming routine and the transfer of application data to be written into the built-in Flash ROM are controlled by using the built-in debug function via the JTAG interface.

(1) Pin Settings

The following table lists the pin settings necessary to use the JTAG debug function (see Chapter 29).

Pin name	Setting
BOOT[1]	"L"
BOOT[0]	"L"

(2) Connection Method

Connect the JTAG pin with the host, such as a PC, or JTAG-compatible Flash Programmer via a JTAG debug interface tool.
For more information about the connections of the JTAG connectors on the user board, see Chapter 29.

Notes:
- The built-in Flash ROM programming routine should be provided by the customer himself/herself.
- By changing the built-in Flash ROM program routine, it is possible to program not only the built-in Flash ROM but an external flash memory mounted on the board.

27.3.3 Method of Programming the Built-In Flash ROM Using the Boot Mode

LAPIS Semiconductor provides software (FBP69Q6203) developed for writing programs into the built-in Flash ROM of the ML69Q6203. Using this software makes it easy to write a program to the ML69Q6203 connected to a PC via USB. For how to use the software, refer to the FBP69Q6203 User's Guide.

(1) Pin Settings

The table below lists the pin settings necessary to start in the boot mode at the time of reset.
As for the JTAG pin, set it to the setting when it is not used.

Pin name	Setting
BOOT[1]	"H"
BOOT[0]	"L"

27.4 Timing of Operations

The following diagrams show the access timing to the built-in Flash ROM.
The timing shows internal signal wave forms of the LSI as reference examples.

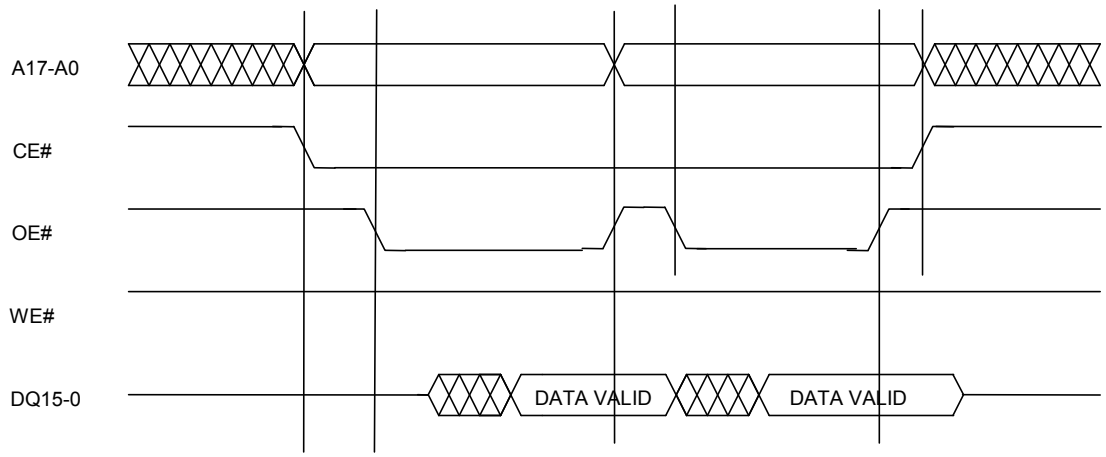


Figure 27-4 Read Cycle

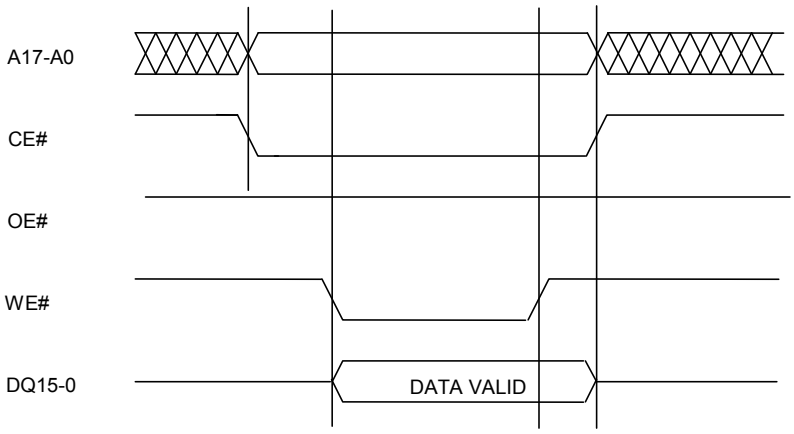


Figure 27-5 Command Entry

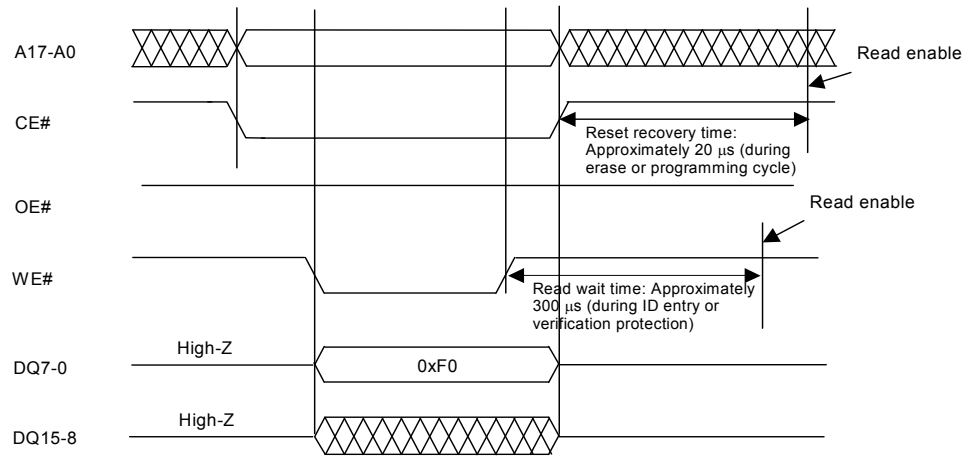


Figure 27-6 Read/Reset 1

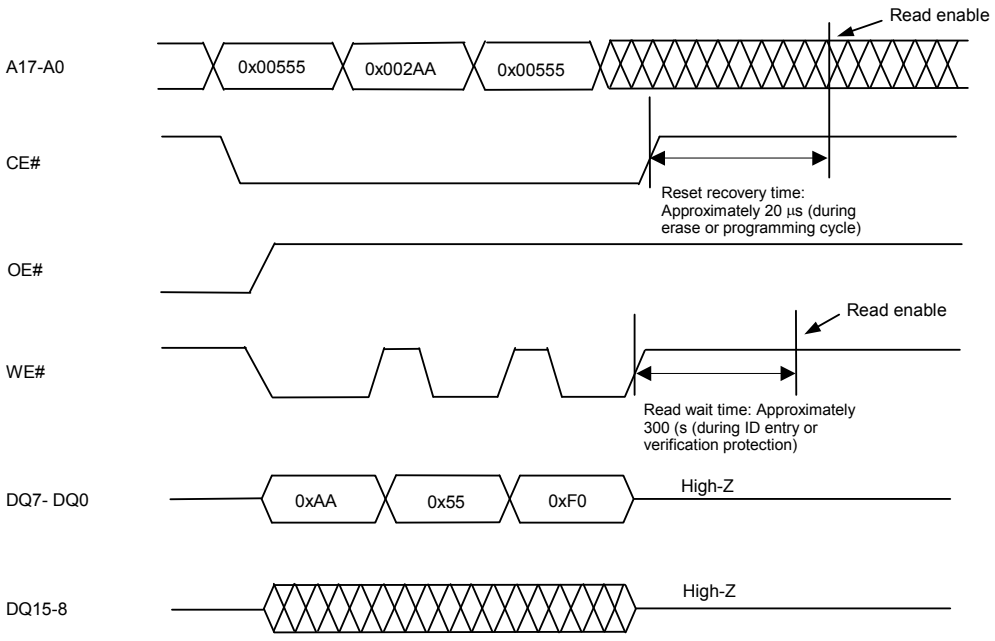


Figure 27-7 Read/Reset 2

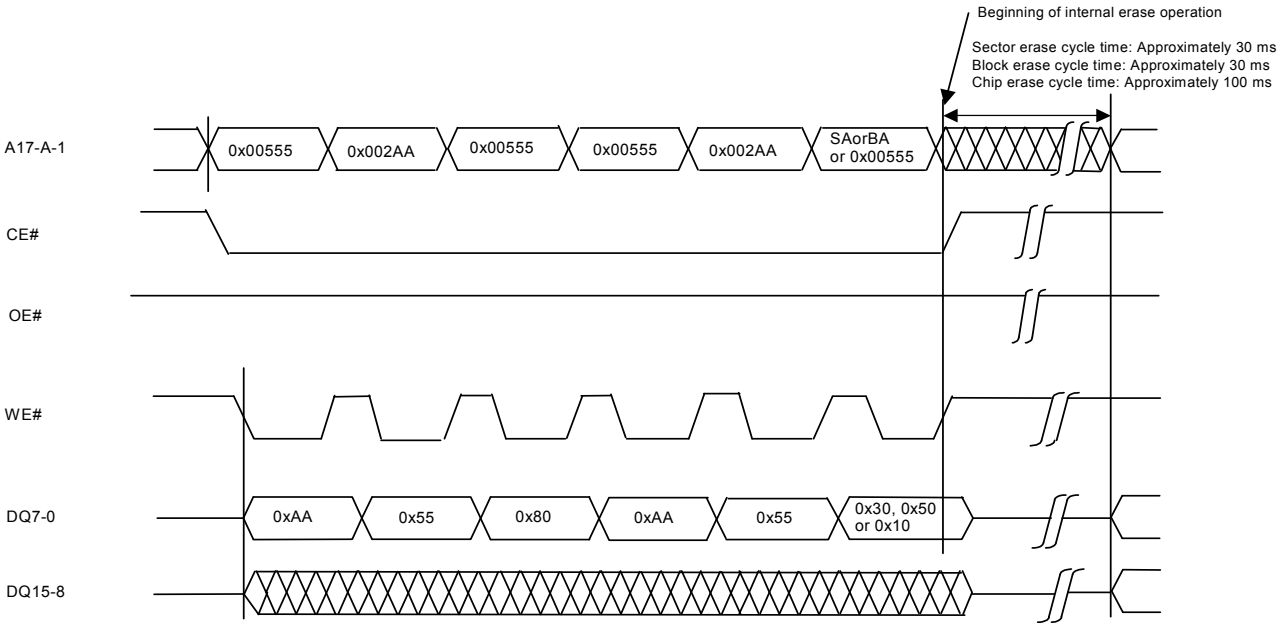


Figure 27-8 WE# Control, Selector Erase, Block Erase, and Chip Erase

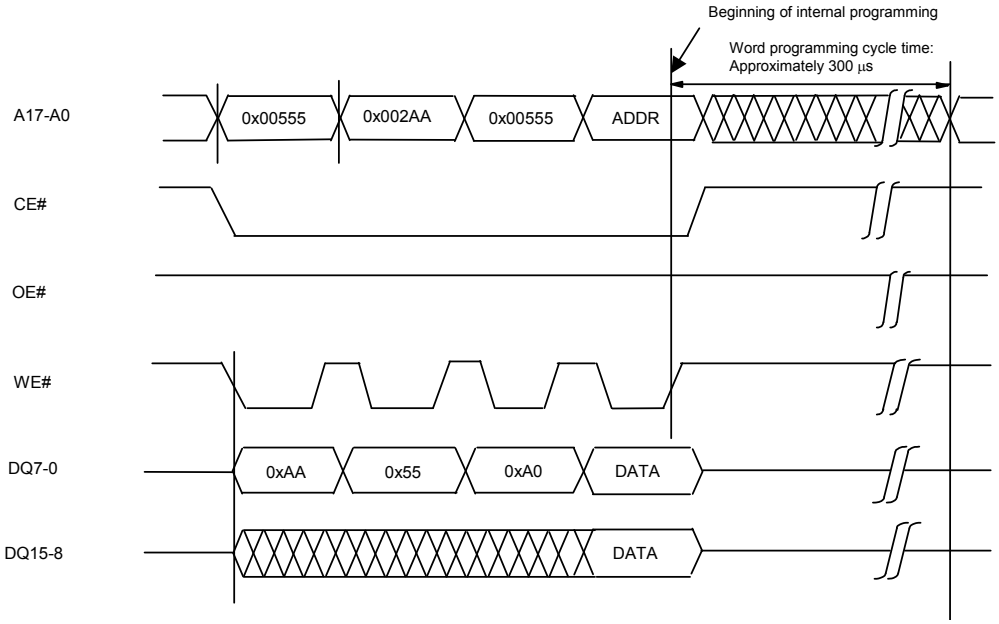


Figure 27-9 WE# Control Word Programming

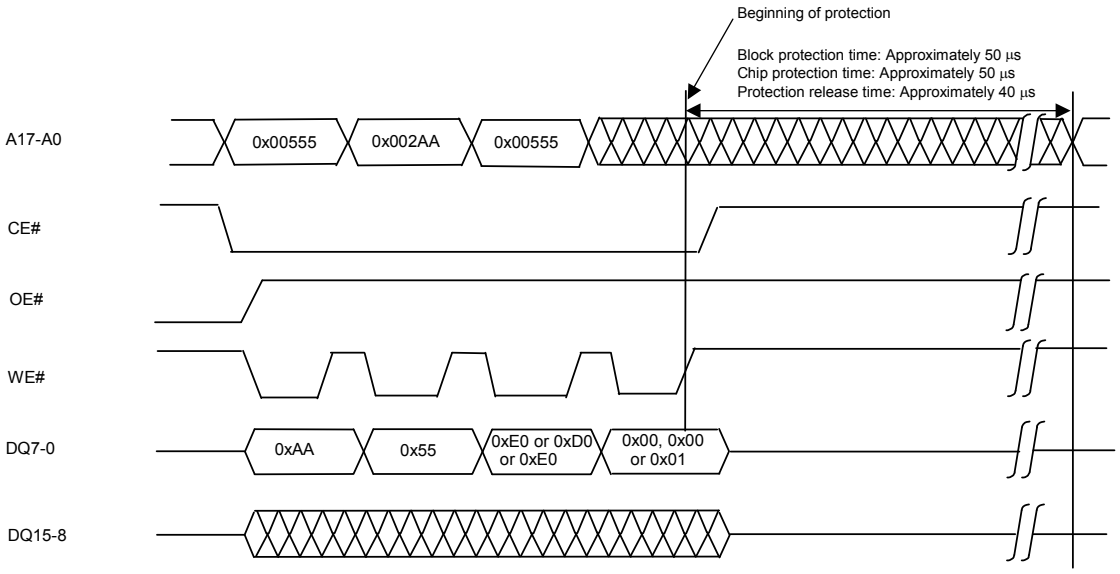


Figure 27-10 Block Protection, Chip Protection, and Protection Release

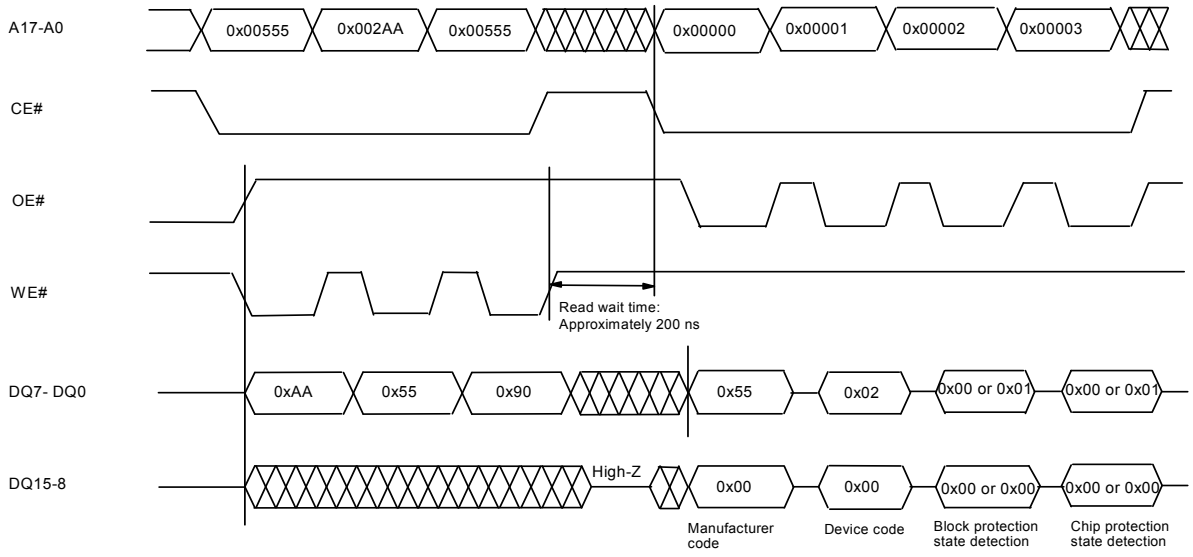
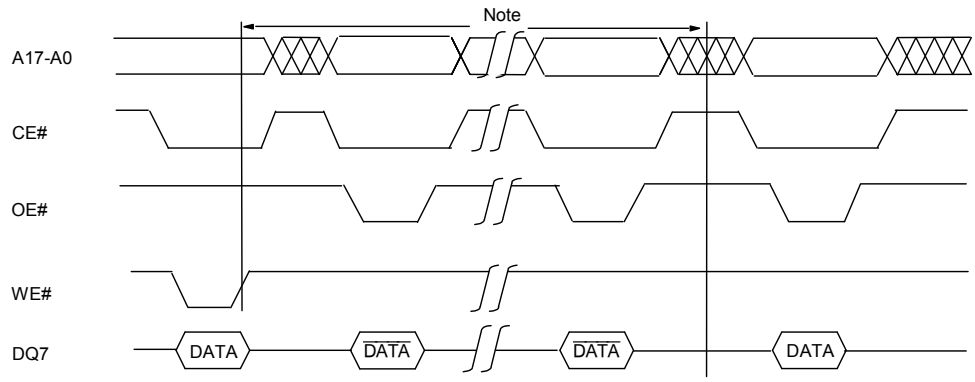
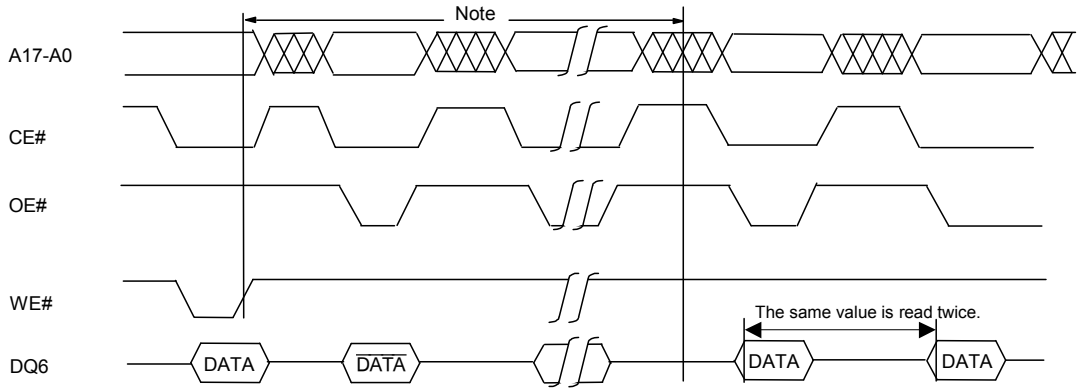


Figure 27-11 Software ID Entry/Verification Protection Read



Note: This timing depends on mode of operation used, and the word programming cycle, sector erase cycle, block erase cycle, or chip erase cycle time is applied.

Figure 27-12 DATA# Polling (DQ 7)



Note: This timing depends on mode of operation used, and the word programming cycle, sector erase cycle, block erase cycle, or chip erase cycle time is applied.

Figure 27-13 Toggle Bit (DQ6)

Chapter 28

JTAG

Chapter 28 JTAG

28.1 Overview

This LSI provides an on-board debugging function via the JTAG interface.

28.1.1 List of Pins

Pin name	I/O	Description
TDI	I	Test data input
TDO	O	Test data output
nTRST	I	TAP controller reset signal
TMS	I	Test mode select
TCK	I	Test clock input
RTCK	O	Test clock output

28.2 On-board Debugging Function

28.2.1 Requirements

This LSI has a built-in debugging function. The function can perform on-board debugging by installing a JTAG connector on the board. Prepare the following for on-board debugging.

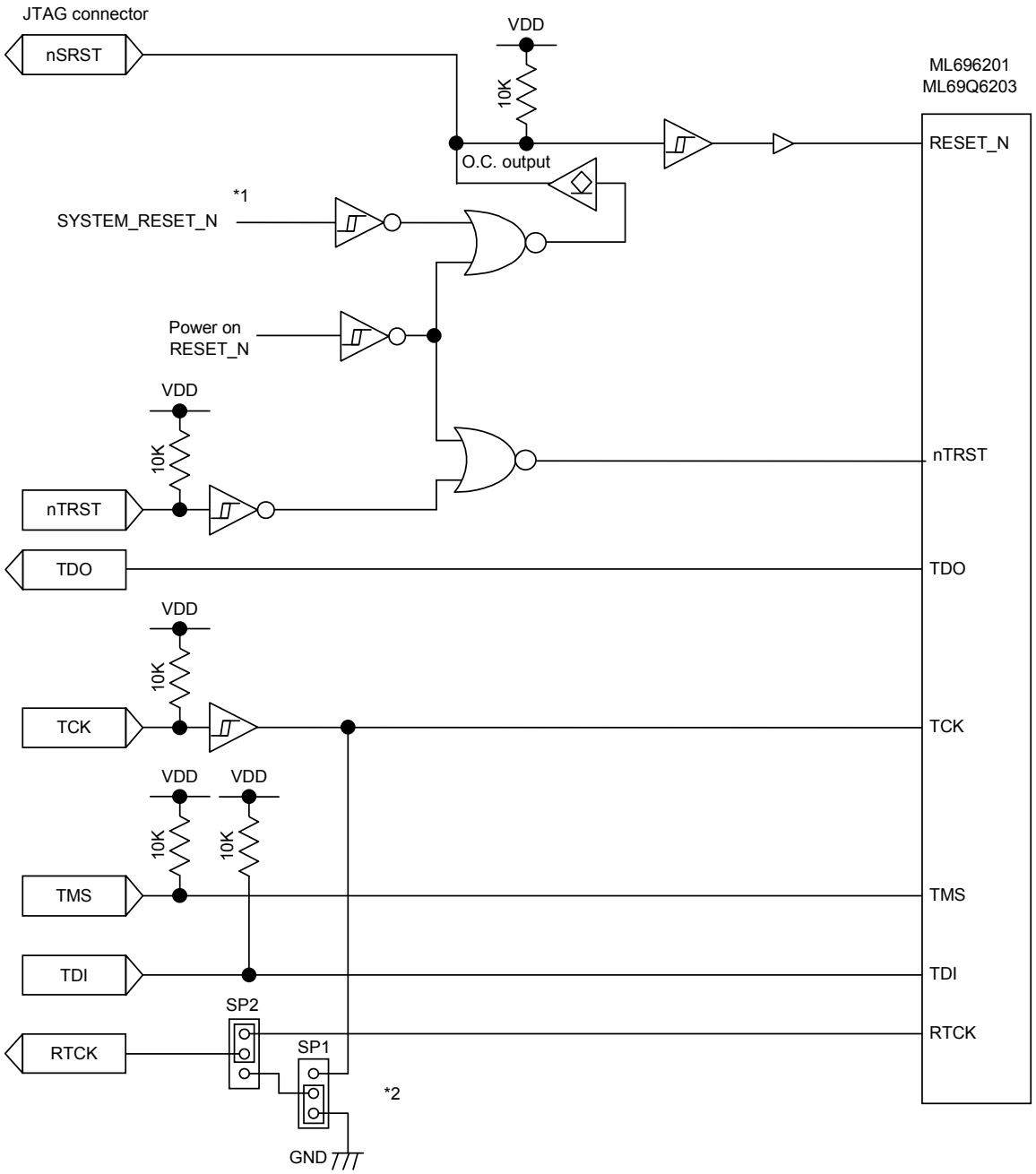
- JTAG interface tool (referred to as JTAG-ICE) such as Multi ICE or Real View ICE of ARM
- Debugger (such as Real View Debugger) that supports the above interface
- Host PC: PC with the above debugger installed
- Cables

28.2.2 Connection

Figure 28-1 shows an example of a connection circuit between this LSI and JTAG-ICE.
When connecting this LSI and a JTAG interface tool, set the JTAG interface tool as follows.

- Adaptive = on

For further details, refer to the JTAG-ICE Manual.



*1
Do not input SYSTEM_RESET to nTRST. When SYSTEM_RESET is input to nTRST while JTAG is used, the JTAG circuit is reset regardless of nTRST from ICE, causing malfunctioning of ICE.
*2
Normally, connect RTCK that is input from LSI. However, TCK needs to be looped back or fixed to GND depending on the ICE used.

Figure 28-1 Example of Connection Circuit with JTAG-ICE

Chapter 29

Electrical Characteristics

Chapter 29 Electrical Characteristics

29.1 Absolute Maximum Ratings

Parameter	Symbol	Condition	Rating	Unit
Digital power supply voltage (CORE)	V_{DDCORE}	All GNDs = 0 V Ta = 25°C	−0.3 to +2.0	V
Digital power supply voltage (I/O)	V_{DD_IO}		−0.3 to +4.6	
Input voltage	V_I		−0.3 to V_{DD_IO}+0.3	
Output voltage	V_O		−0.3 to V_{DD_IO}+0.3	
PLL section power supply voltage	V_{DDPLL}		−0.3 to +2.0	
RTC section power supply voltage	V_{DDRTC}		−0.3 to +2.0	
USB section power supply voltage 1	AV_{DDRX}		−0.3 to +4.6	
USB section power supply voltage 2	AV_{DDTX}		−0.3 to +4.6	
USB section power supply voltage 3	AV_{DDC}		−0.3 to +4.6	
Flash section power supply voltage	V_{DDFLA}		−0.3 to +4.6	
Analog power supply voltage	AV_{DD}		−0.3 to +4.6	
Analog input voltage	V_{AI}		−0.3 to AV_{DD}	
Power dissipation	P_D	Ta = 70°C per package	400	mW
Storage temperature	T_{STG}	—	−50+150	°C

29.2 Recommended Operating Conditions

(GND = 0 V)

Parameter	Symbol	Condition		Min.	Typ.	Max.	Unit
Digital power supply voltage (CORE)	V_{DD_CORE}	$V_{DDPL} = V_{DD\_CORE}$ $V_{DDRTC} = V_{DD\_CORE}$	USB not used	1.35	1.5	1.65	V
PLL section power supply voltage	V_{DDPLL}		USB FS mode used	1.425	1.5	1.575	
RTC section power supply voltage	V_{DDRTC}		USB HS mode used	1.45	1.5	1.55	
Digital power supply voltage (I/O)	V_{DD_IO}	$V_{DD\_IO} \geq V_{DD\_CORE}$		2.7	3.3	3.6	
Analog power supply voltage	AV_{DD}	$A_{VDD} = V_{DD\_IO}$					
USB section power supply voltage 1	AV_{DDRX}	$AV_{DDRX} = AV_{DDTX} = AV_{DDC}$		3.0	3.3	3.6	
USB section power supply voltage 2	AV_{DDTX}						
USB section power supply voltage 3	AV_{DDC}						
Flash section power supply voltage (*1)	V_{DDFLA}			2.7	3.3	3.6	
CPU Operating frequency	F_{OSC}			—	—	120	MHz
Operating frequency	f_{c48}			—	48.00	—	MHz
	f_{c11}			—	11.2896	—	MHz
	f_{c32}			—	32.768	—	kHz
Ambient temperature	Ta	USB not used		−30	25	+70	°C
		USB FS mode used		−20	25	+70	
		USB HS mode used		0	25	+50	

Notes:
*1: The Flash section power supply voltage is specified only for the ML69Q6203.

- Operation cannot be guaranteed if neither output pin nor any I/O pin configures as output short-circuit is occured.

- Power-On/Power-Off Sequence
  When turning the power on, either turn on all power supplies at the same time or turn on the core power supply first and then turn the other power supplies on.
  Turn off all power suppies at once or turn off all other power first then turn off the core power when the power is turned off
  It is possible, while holding only the RTC power supply turned on, to turn off the other power supplies. For details see Chapter 7, "Power Consumption Control and System Control."

29.3 DC Characteristics

(V_{DD_CORE} = 1. 35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
"H" input voltage 1 (*1)	V_{IH1}	—	2.2	—	V_{DD_IO}+0.3	V
"L" input voltage 1 (*2)	V_{IL1}	—	–0.3	—	+0.6	V
"H" input voltage 2 (*3) (5 V tolerant)	V_{IH2}	—	2.2	—	5.5	V
"H" input voltage 3 (*4)	V_{IH3}	—	0.8×V_{DD_RTC}	—	V_{DD_RTC} +0.3	V
"L" input voltage 3 (*4)	V_{IL3}	—	–0.3	—	0.2×V_{DD_RTC}	V
"H" input voltage 4 (*5)	V_{IH4}	—	0.8×V_{DD_IO}	—	V_{DD_IO} +0.3	V
"L" input voltage 4 (*5)	V_{IL4}	—	–0.3	—	0.2×V_{DD_IO}	V
"H" output voltage 1 (*6)	V_{OH1}	*7	2.2	—	—	V
"H" output voltage 2 (*8)	V_{OH2}	V_{DD_IO} = 3.0 to 3.6 V	2.2	—	—	V
"L" output voltage 1 (*6)	V_{OL1}	*7	—	—	0.4	V
"H" output voltage 3 (*9)	V_{OH3}	—	0.8×V_{DD_RTC}	—	—	V
"L" output voltage 3 (*9)	V_{OL3}	—	—	—	0.2×V_{DD_RTC}	V
"H" output voltage 4 (*10)	V_{OH4}	—	0.8×V_{DD_IO}	—	—	V
"L" output voltage 4 (*10)	V_{OL4}	—	—	—	0.2×V_{DD_IO}	V
Input leakage current 1 (*11)	I_{L1}	V_I = 0V/V_{DDIO}	–10	—	+10	μA
Input leakage current 2 (*11)	I_{L2}	V_I = 0V/5.5V	–10	—	+10	μA
Input leakage current 3 (*12)	I_{L3}	V_I = 0V	–200	–66	–10	μA
Input leakage current 4 (*12)	I_{L4}	V_I = V_{DDIO}	–10	—	+10	μA
Input leakage current 5 (*13	I_{L5}	V_I = 0V	–10	—	+10	μA
Input leakage current 6 (*13)	I_{L6}	V_I = V_{DDIO}	10	66	200	μA
Input leakage current 7 (*14)	I_{L7}	V_I = 0V/V_{DDIO}	–1	—	+1	μA
Input leakage current 8 (*15)	I_{L8}	V_I = 0V/V_{DDIO}	–0.15	—	+0.15	μA
CORE supply current	I_{DD_CORE}	*16	—	75	110	mA
IO supply current	I_{DD_IO}	*16	—	20	30	mA
RTC section supply current	I_{DDRTC}	*17	—	7	120	μA

Applicable pins:

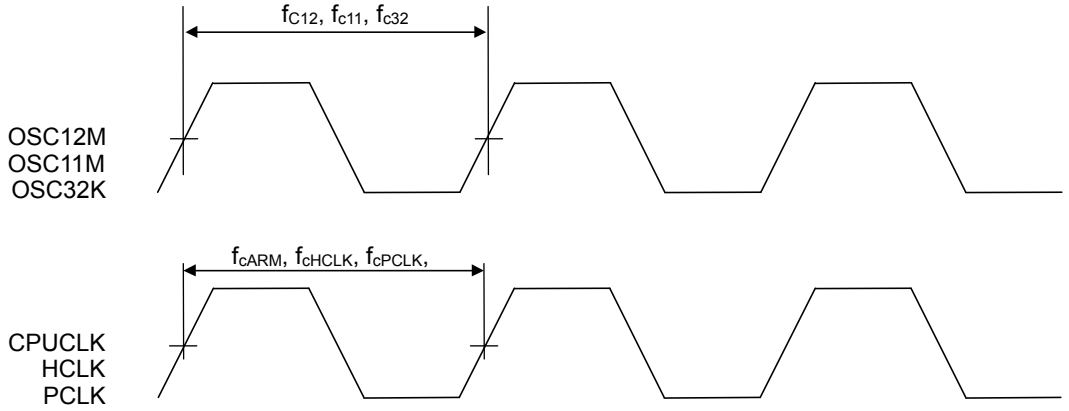
*1 Normal pins (including pulled-up/pulled-down pins)
*2 Normal pins (including pulled-up/pulled-down pins)
*3 Tolerant pins
*4 RTC_TESTMODE, 32K_TESTMODE, OSC32K0
*5 OSC48M0, OSC11M0
*6 Normal pins (including pull-up/down)
*7 For IOH/L, those specified as 2 mA in the Pin List indicate ±2 mA; those specified as 4 mA in the Pin List indicate ±4 mA; those specified as 6 mA in the Pin List indicate ±6 mA.
*8 IDE-related pins
*9 32 kHz oscillation pins
*10 48 MHz or 11 MHz oscillation pins
*11 Normal pins
*12 pull-up pins
*13 pull-down pins
*14 48 MHz or 11 MHz oscillation pins
*15 32 kHz oscillation pins
*16 When operating the peripheral other than USB(CPUCLK=120MHz,HCLK=60MHz,PCLK=30MHz).
*17 When the power supplies other than the RTC section power supply are turned off.

29.4 Interface Characteristics by Functions

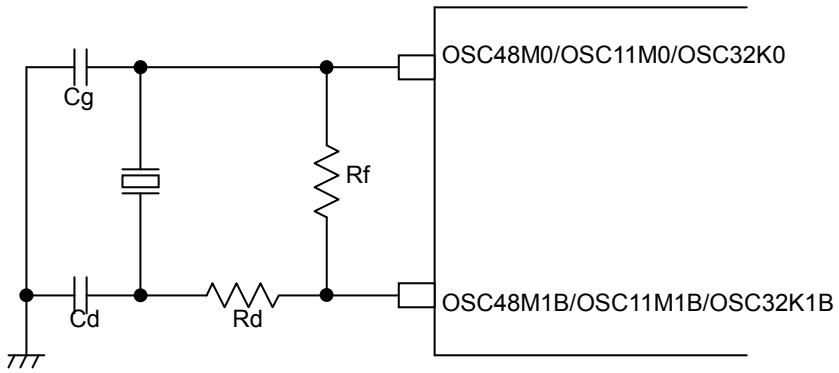
29.4.1 Clock Timing

(V_{DDCORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit	Remarks
ARM maximum operating frequency	f_{ARM}	—	—	—	120	MHz	
HCLK maximum operating frequency	f_{HCLK}	—	—	—	60	MHz	
PCLK maximum operating frequency	f_{PCLK}	—	—	—	30	MHz	
OSC48M pin oscillation frequency	f_{c48}	*1	—	48.00	—	MHz	
OSC48M pin duty	$Duty_{48}$	—	40	50	60	%	
OSC11M pin oscillation frequency	f_{c11}	*1	—	11.2896	—	MHz	
OSC11M pin duty	$duty_{11}$	—	40	50	60	%	
OSC32K pin oscillation frequency	f_{c32}	*1	—	32.768	—	kHz	
OSC32K pin duty	$duty_{32}$	—	40	50	60	%	



*1 Configuration of the evaluation circuit for the crystal oscillation section



- Circuit constants

	Rf	Rd	Cd/Cg	Crystal used
48 MHz	1 MΩ	100Ω	10pF	DSX840G manufactured by Daishinku Corp. (Master clock oscillation)
11 MHz	1 MΩ	300Ω	30pF	DSX840G manufactured by Daishinku Corp. (Master clock oscillation)
32 kHz	4 MΩ	470kΩ	20pF	DMX-26S manufactured by Daishinku Corp.

Since the optimum value of a circuit constant varies depending on the crystal mounted on the board and the value of the load capacitance, so use the above values for reference only.
The customer is encouraged to perform sufficient evaluation in order to determine circuit constants for actual products.

29.4.2 Reset Timing

(V_{DDCORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit	Remarks
RESET_N pulse width 1	t_{RSTW1}		10	—	—	μs	At power-on; Except when during the HALT/STANDBY mode
RESET_N pulse width 2	t_{RSTW2}	(*1)	Oscillation stabilization time	—	—	—	At power-on; After the HALT/STANDBY mode is released

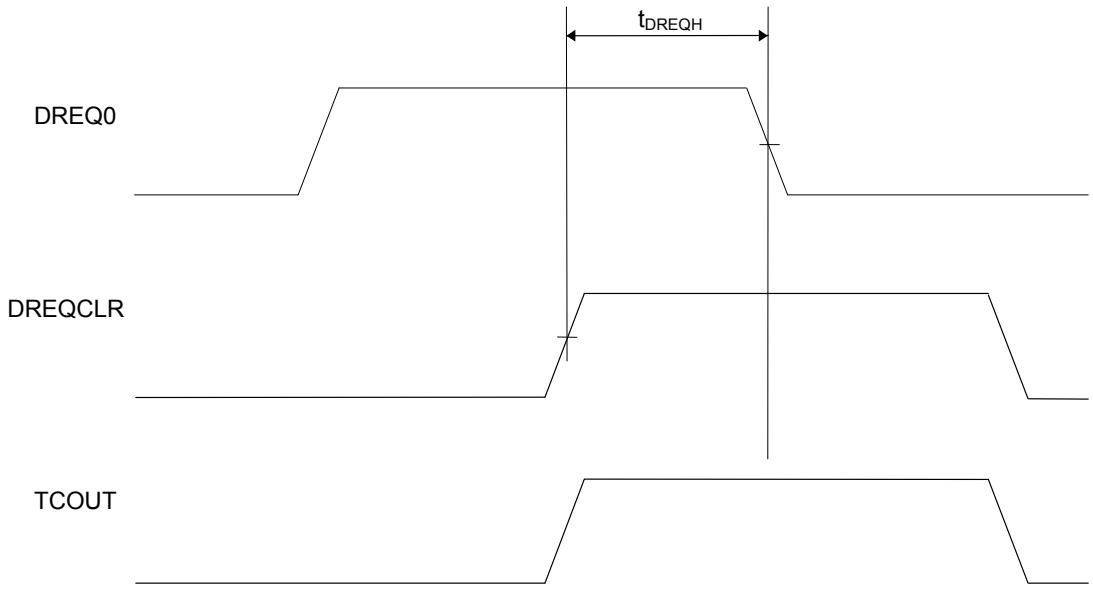
*1 When turning the power on, power needs to be applied with the system being in a reset state. If the system is reset after power-on, a malfunction may occur during the period in which a reset is not asserted.

29.4.3 DMA Request Timing

(V_{DDCORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit	Remarks
DREQ Hold Time	t_{DREQH}	—	Tc	—	—	ns	

* Tc = hclk cycle



29.4.4 SRAM/ROM/IO Access Timing

(V_{DDCORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

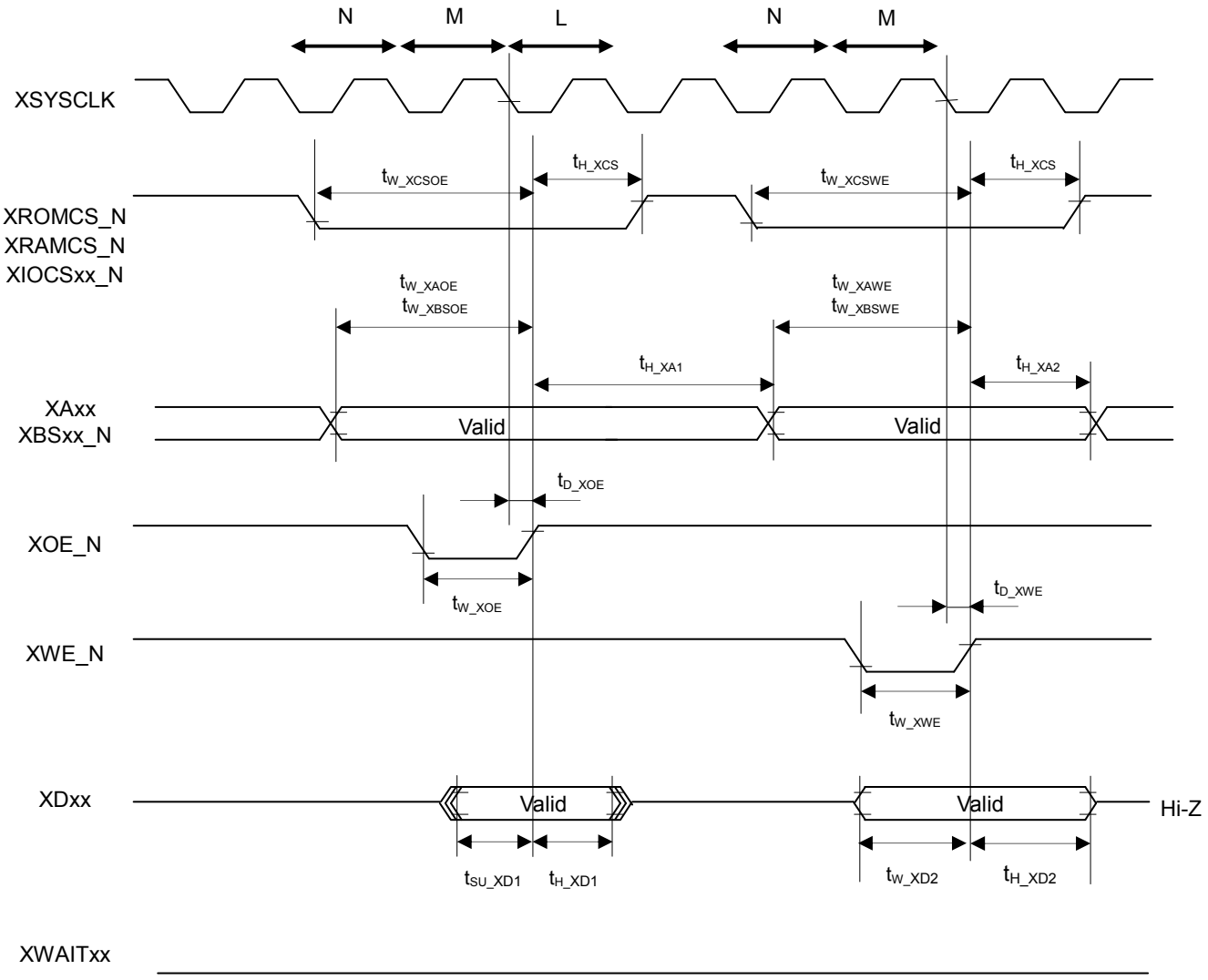
Parameter	Symbol	Min.	Max.	Unit	Remarks
XOE_N signal delay time	t_{D_XOE}	-7	+12	ns	
XOE_N signal pulse width	t_{W_XOE}	M*t_C -8	M*t_C +9		
X**CS_N - XOE_N delay time	t_{W_XCSOE}	(M+N)*t_C -7	(M+N)*t_C +9		
X**CS_N signal hold time	t_{H_XCS}	t_C -7	—		
XA - XOE_N delay time (normal access)	t_{W_XAOE}	(M+N)*t_C -7	(M+N)*t_C +9		
XA - XOE_N delay time (page mode)		K*t_C -7	K*t_C +9		
XA signal delay time	t_{D_XA}	-6	+12		
XA signal hold time at read	t_{H_XA1}	(L+1)*t_C -8	—		
XBSx_N - XOE_N delay time	t_{W_XBSOE}	(M+N)*t_C -7	(M+N)*t_C +9		
XD signal input setup time at read	t_{SU_XD1}	18	—		
XD signal input hold time at read	t_{H_XD1}	0	—		
XD signal input setup time at continuous read	t_{SU_XD3}	18	—		
XD signal input hold time at continous read	t_{H_XD3}	0	—		
XWAITx signal input setup time	t_{SU_XWAIT}	18	—		
XWAITx signal input hold time	t_{H_XWAIT}	0	—		
XWE_N signal delay time	t_{D_XWE}	-7	+12		
XWE_N signal pulse width	t_{W_XWE}	M*t_C -8	M*t_C +7		
X**CS_N - XWE_N delay time	t_{D_XCSWE}	(M+N)*t_C -7	(M+N)*t_C +8		
XA - XWE_N delay time	t_{D_XAWE}	(M+N)*t_C -6	(M+N)*t_C +11		
XA signal hold time at write	t_{H_XA2}	t_C -8	—		
XBSx_N - XWE_N delay time	t_{D_XBSWE}	(M+N)*t_C -7	(M+N)*t_C +9		
XD signal setup time at write	t_{SU_XD2}	M*t_C -8	M*t_C +8		
XD signal hold time at write	t_{H_XD2}	t_C -8	—		

Notes:
* Tc = hclk cycle
* N means address setup. N = 1 or 2 (varies depending on the RAMTYPE[2:0] setting)
* M means RE/WE pulse width. M = 1 or 2 or 3 or 5 or 8 or 10 or 16 (varies depending on the RAMTYPE[2:0] setting)
* L means data off timing. L = 1 or 2 or 3 or 4 or 5 or 7 (varies depending on the RAMTYPE[2:0] setting)
* K means burst timing. K = 1 or2 or 3 or 5 or 6 or 9 (varies depending on the RAMTYPE[2:0] setting)
* CL = 32pF (Reference information: The load dependency for output pin delay is 0.18 ns/pF [Typical] with the 4 mA buffer.)

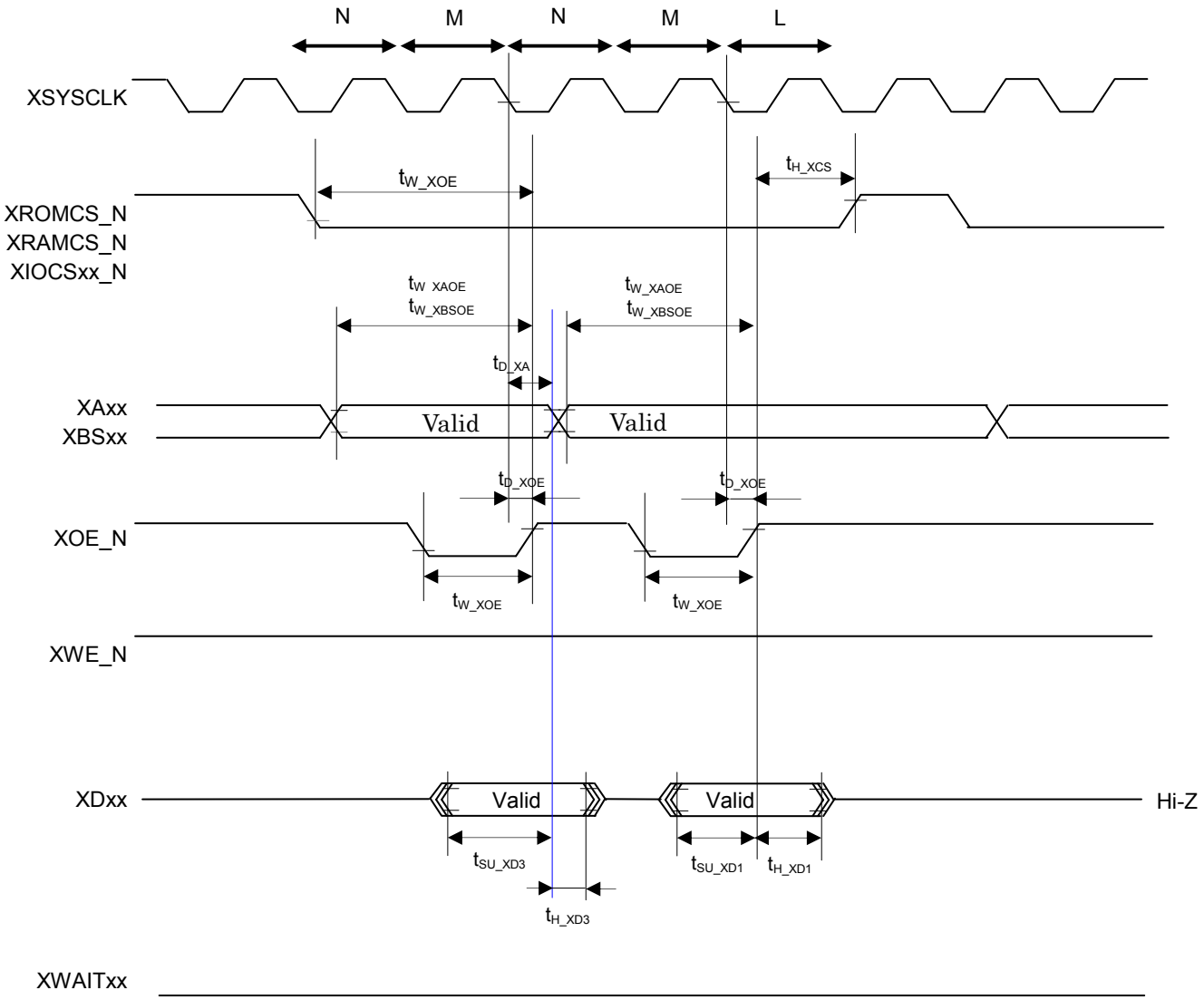
- ROMAC/RAMAC/IOAC registers and parameters for timings that can be set (OE/WE pulse width, read off time)

  Timings that can be set depend on the memories and devices connected. An optimum setting must be made in line with the bus speed.
  Refer to Chapter 10, External Memory Controller, for setting timing parameters.

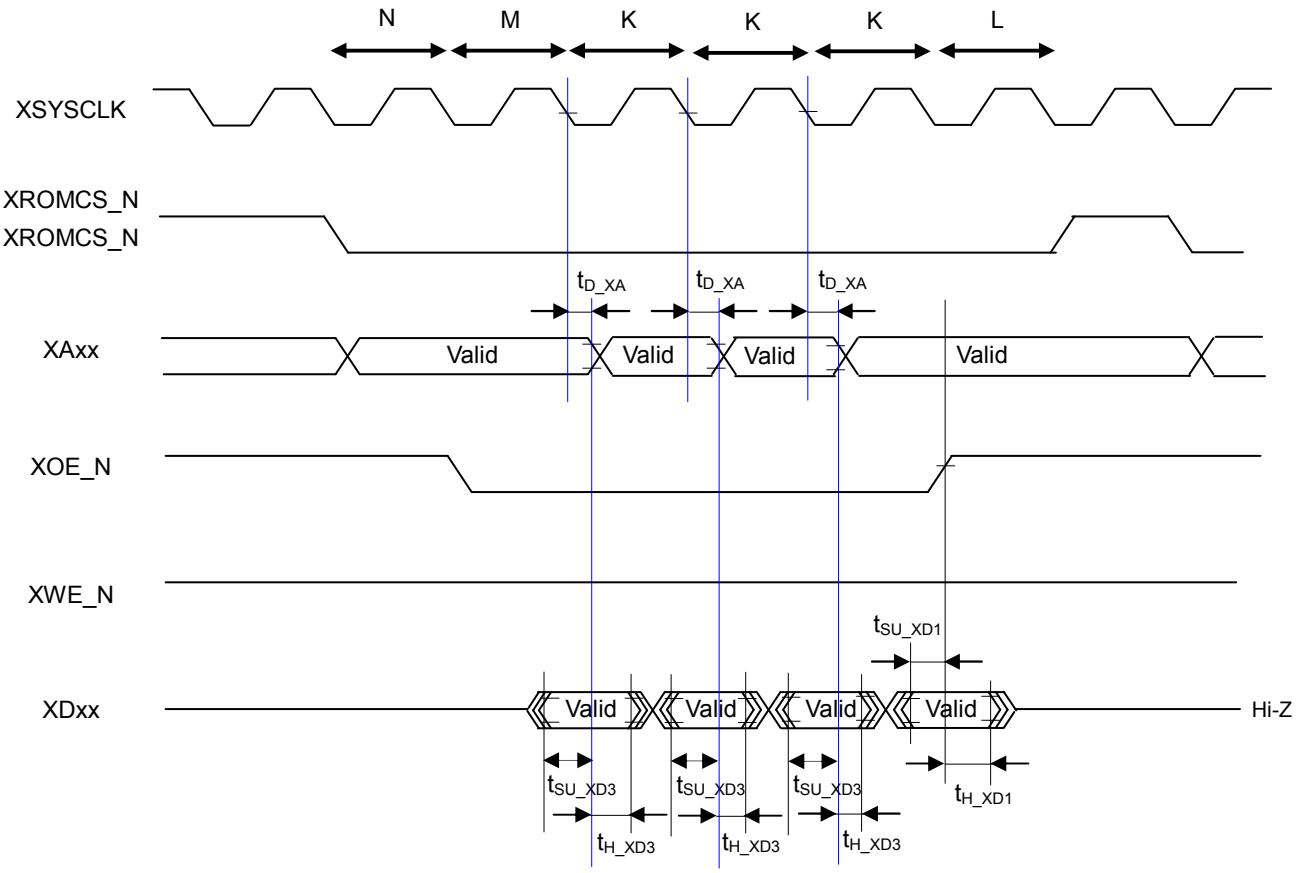
- ROM/RAM/IO single access



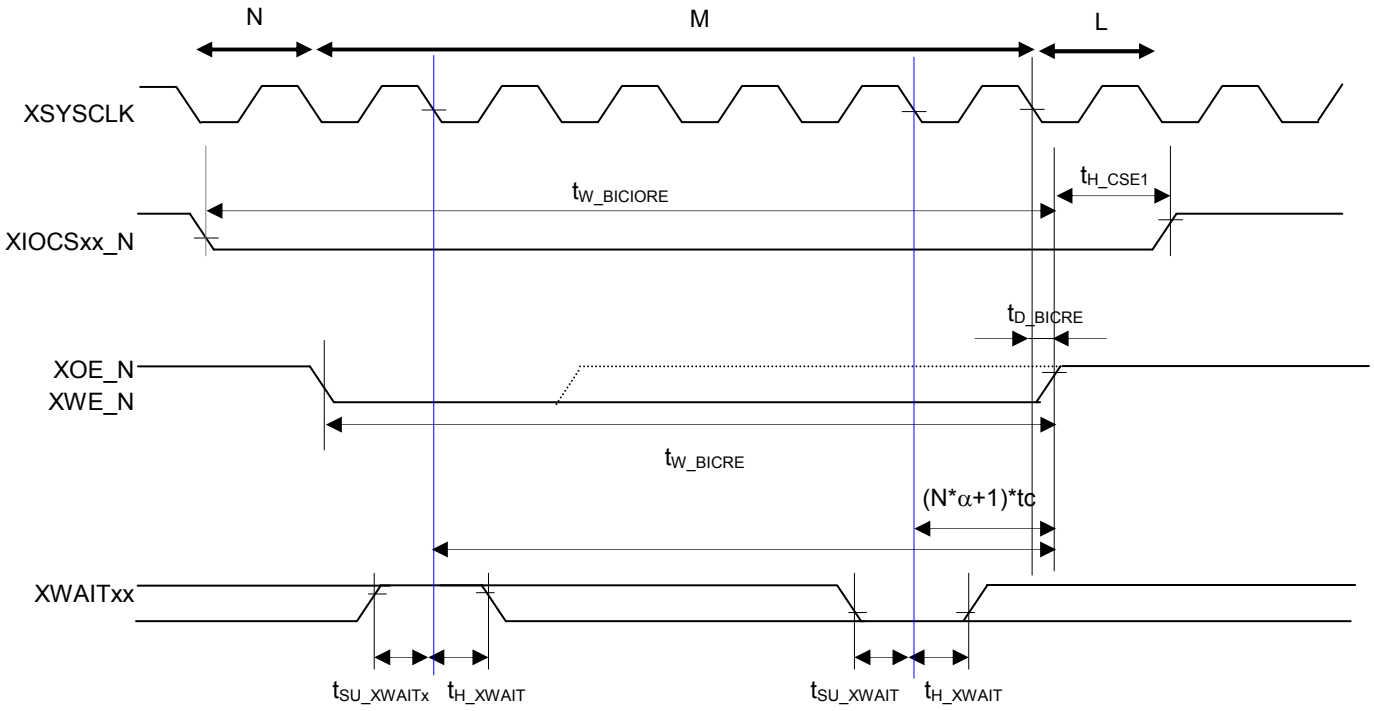
- ROM/RAM/IO continuous access



- ROM/RAM page read access



- I/O section WAIT access
  (With EXT_BIC_WAIT_N asserted, IOTYPE[2:0] = 3'b001, and EXI_BIC_REN_N pulse width = 4)



Notes:
- XWAITx is evaluated at the timing edge that is effect 1 clock before the XOE_N/XWE_N pulse is negated.
- α indicates the number of times that XWAITx was evaluated when at a High level (α = 0, 1, 2,…)

29.4.5 SDRAM access timing

(V_{DDCORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6V, Ta = –30 to +70°C)

Parameter	Symbol	Min.	Max.	Unit	Remarks
XSDCS_N signal delay time	t_{D_SDCS}	$0.5t_C$–5	$0.5t_C$+5	ns	
XSDCKE signal delay time	t_{D_SDCKE}	$0.5t_C$–5	$0.5t_C$+5		
XDQMx signal delay time	t_{D_SDDQM}	$0.5t_C$–5	$0.5t_C$+5		
XRAS_N signal delay time	t_{D_SDRAS}	$0.5t_C$–5	$0.5t_C$+5		
XCAS_N signal delay time	t_{D_SDCAS}	$0.5t_C$–5	$0.5t_C$+5		
RASCAS minimum delay time	t_{D_SDRC}	$n_{SD1}t_C$	—		
RAS signal active time	t_{A_SDRAS}	$n_{SD2}t_C$	—		
RAS signal precharge time	t_{P_SDRAS}	$n_{SD3}t_C$	—		
XWE_N signal delay time	t_{D_SDWE}	$0.5t_C$–5	$0.5t_C$+5		
XDxx input setup time	t_{SU_SDXDI}	6	—		
Xdxxi input hold time	t_{H_SDXDI}	0	—		
XAxx signal delay time	t_{D_SDXA}	$0.5t_C$–5	$0._{5tC}$+5		
Xdxx output delay time	t_{D_SDXDO}	$0.5t_C$–5	$0.5t_C$+5		
Xdxx output hold time	t_{H_SDXDO}	$0.5t_C$–5	—		
XDxx output enable time	t_{E_SDXDO}	$0.5t_C$–5	$0.5t_C$+5		
XDxx output disable time	t_{DE_SDXDO}	$0.5t_C$–5	—		

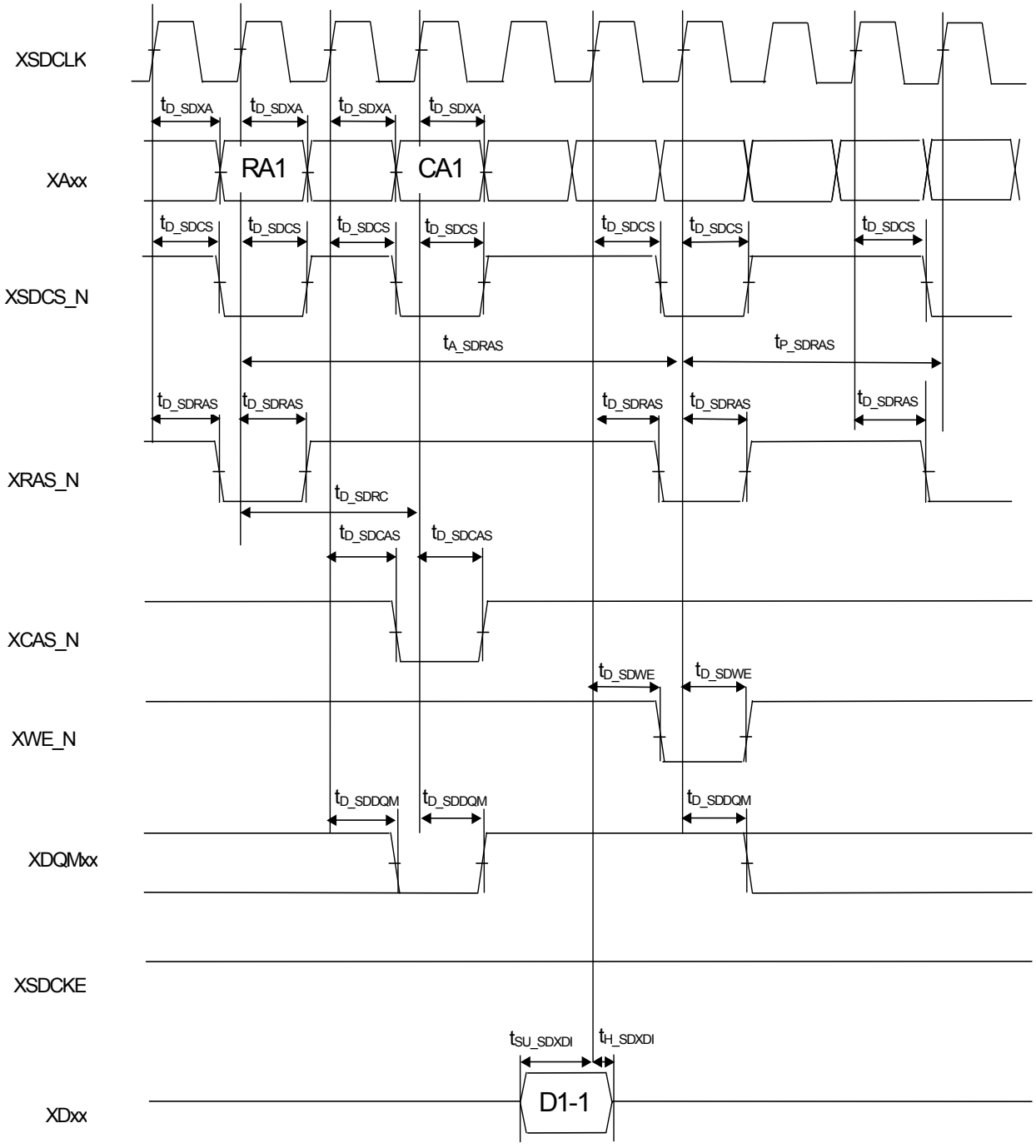
Notes:
- Tc = hclk cycle
- nSD1 = tRCD (parameter that can be set with the DRPC register)
- nSD2 = tRAS (parameter that can be set with the DRPC register)
- nSD3 = tRP (parameter that can be set with the DRPC register)
- CL = 18pF (Reference information: The load dependency for output pin delay is 0.18 ns/pF [Typical] with the 4 mA buffer.)

- Setting and parameters for the DRPC register (tRCS, tRAS, tRP, tDPL)

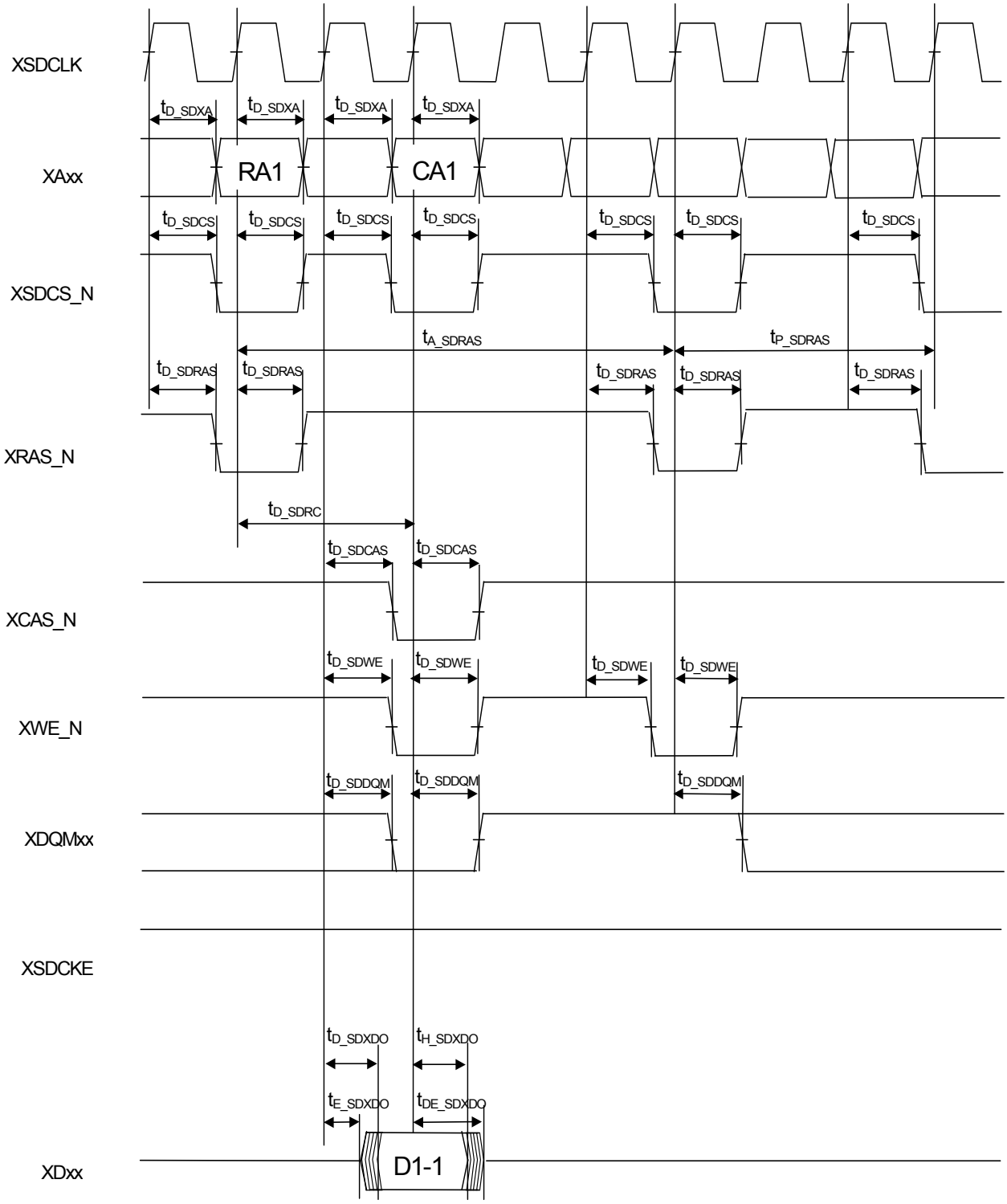
DRAMSPEC: Setting DRAM characteristics parameters

SDRAM		tRCD	tRAS	tRP	tDPL	
DRAMSPEC	0000	1	2	1	1	High-speed DRAM Low frequency
[3:0]	0001	1	3	1	1	↑ ↑
	0010	2	3	2	1	| |
	0011	2	4	2	1	| |
	0100	2	4	2	2	| |
	0101	2	5	2	1	| |
	0110	2	5	2	2	| |
	0111	2	5	3	1	| |
	1000	3	5	3	2	↓ ↓
	1001	3	6	3	2	Low-speed DRAM High frequency
	1010	<Reserved>				Operation is unpredictable if this setting is made
	:	:				:
	1111	<Reserved>				Operation is unpredictable if this setting is made

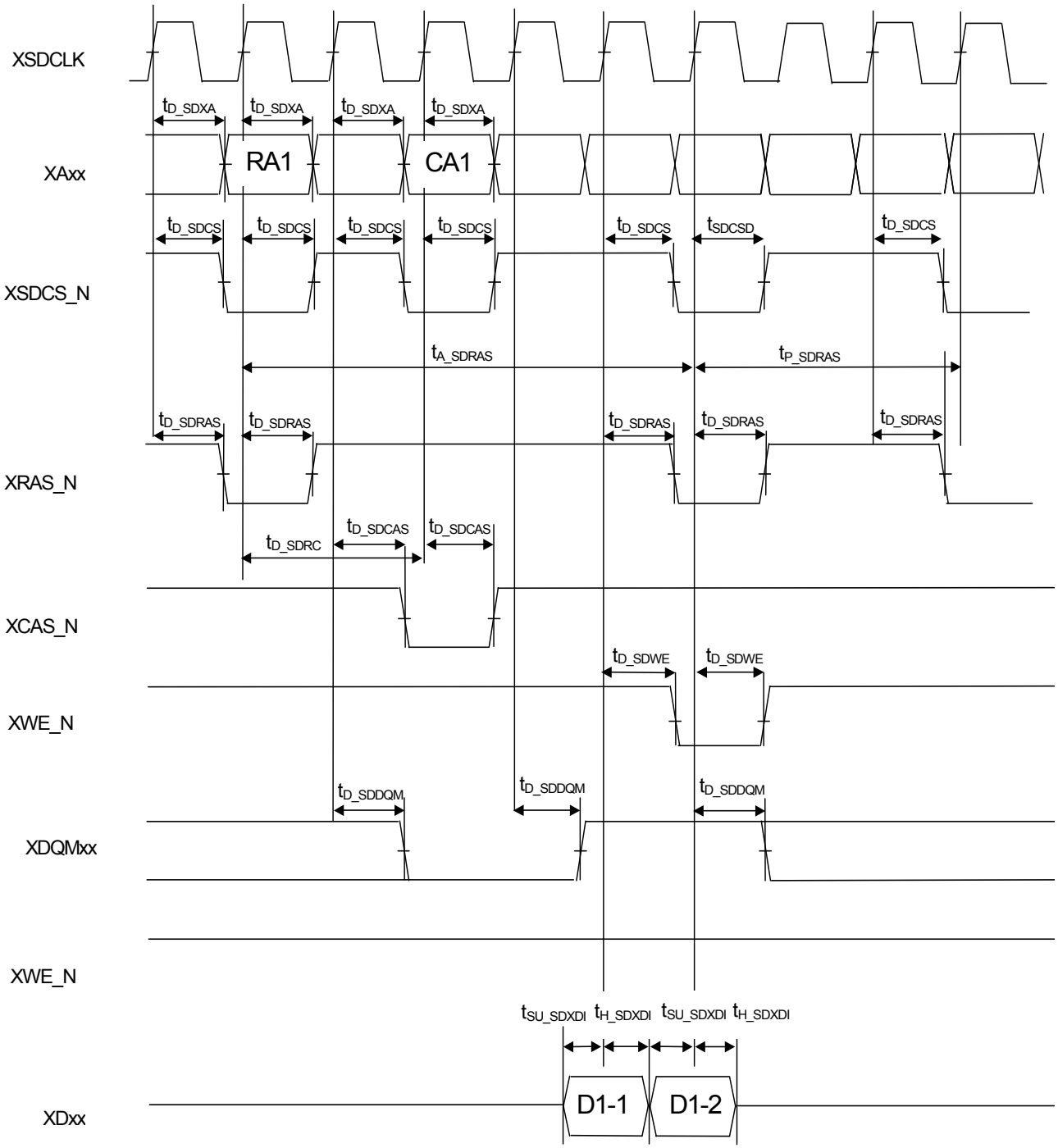
- External bus setting – SDRAM read cycle
  (SDRAM with 16-bit bus width; byte/half-word access)



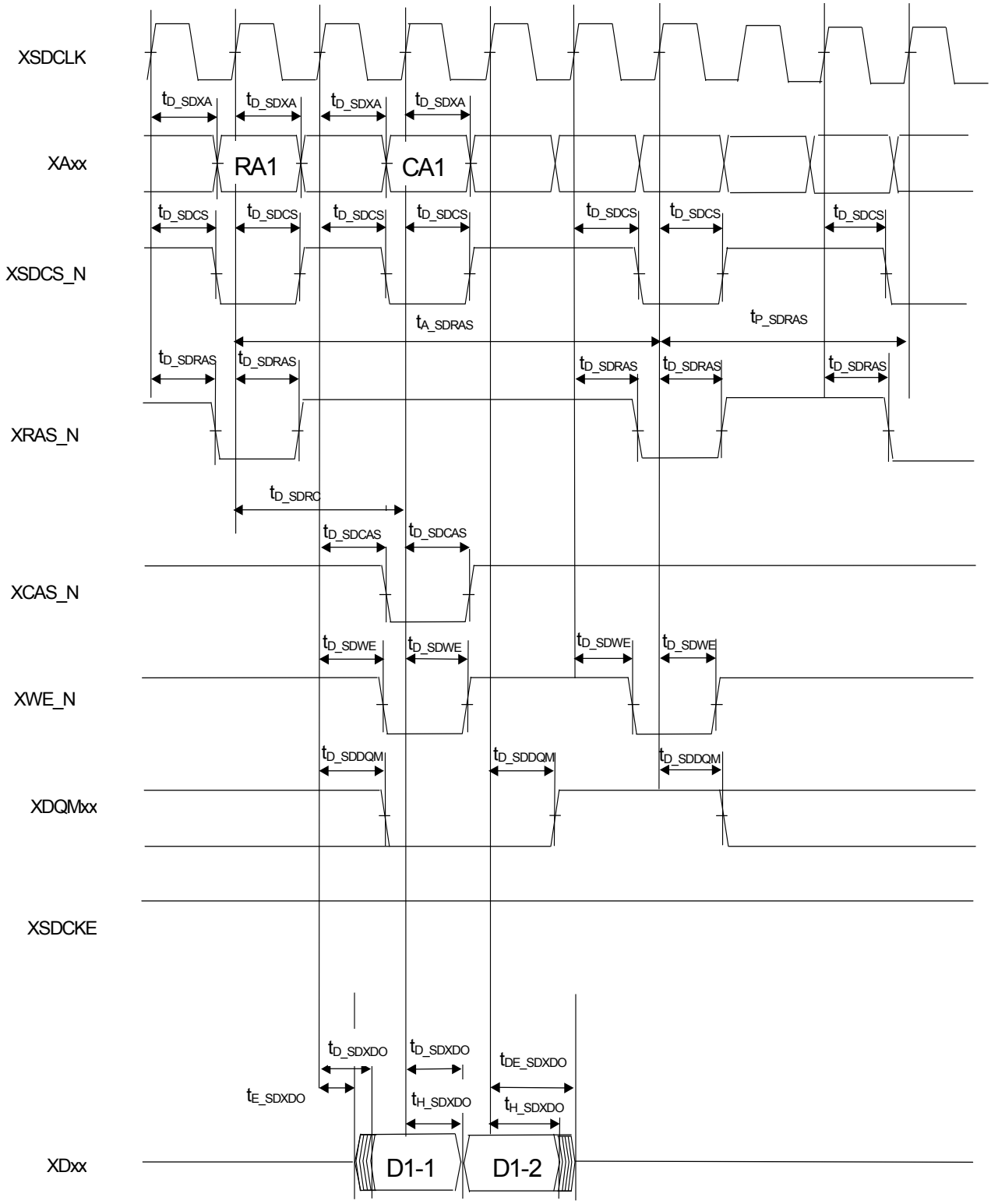
- External bus timing – SDRAM write cycle
  (SDRAM with 16-bit bus width; byte/half-word access)



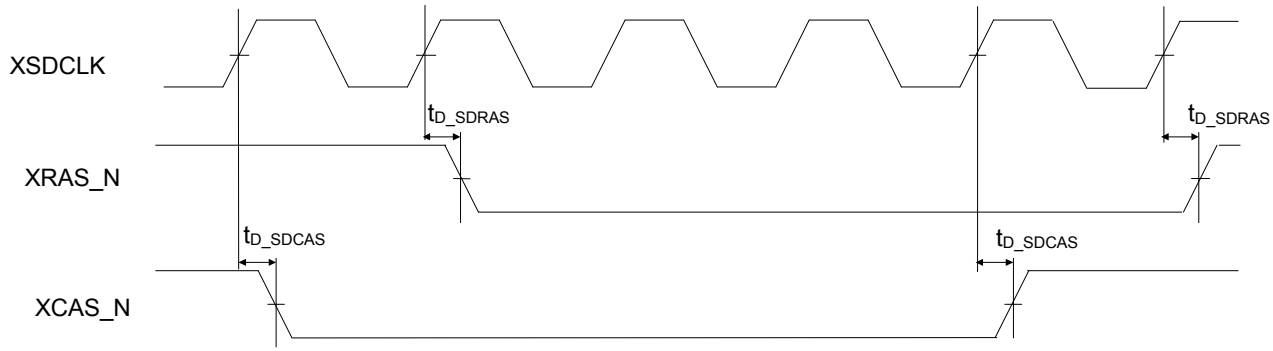
- External bus timing – SDRAM read cycle
  (SDRAM with 16-bit bus width; word access)



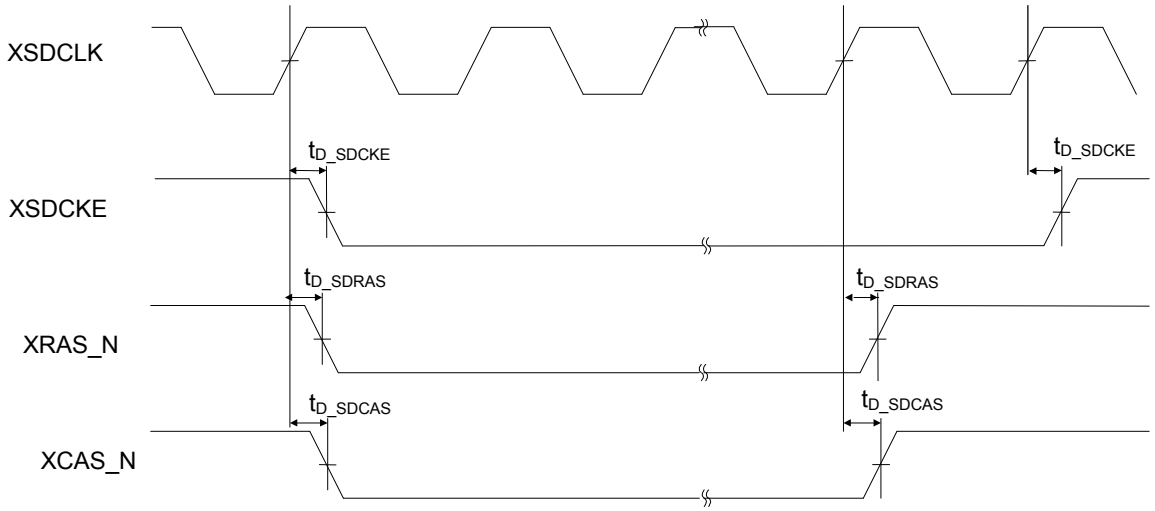
- External bus timing – SDRAM write cycle
  (SDRAM with 16-bit bus width; word access)



- External bus timing – CAS-before-RAS (CBR) refresh



- External bus timing – Self refresh



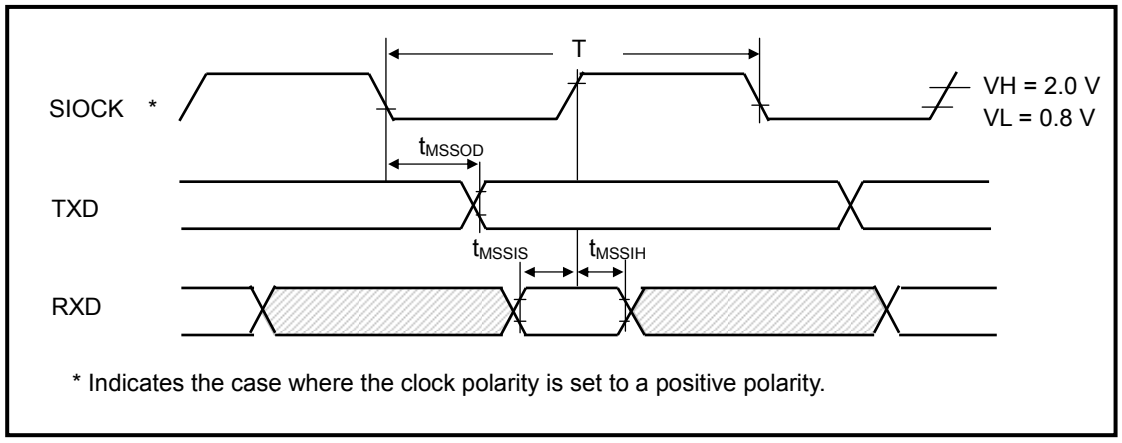
29.4.6 Synchronous Serial Interface Timing

Using the internal register, master mode or slave mode can be selected for this synchronous serial interface. A serial clock polarity can be selected as well.
When the clock polarity is set to a positive polarity, transmit data is shifted out at a falling edge of the clock and receive data is shifted in at a rising edge of the clock. When data transmission/reception is completed by 8 bits, the clock stops at a High level and the last output data is retained.
When the clock polarity is set to a negative polarity, data for transmission is output at a rising edge of the clock and data for reception is input at a falling edge of the clock. When data transmission/reception is completed by 8 bits, the clock stops at a Low level and the last output data is retained.
The waveforms in the figures below show cases where the clock polarity is set to a positive polarity.

- Master mode

(V_{DD_CORE} = 1.35 to 1.65V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit	Remarks
Serial clock cycle	T	C_L =50pF	66.67	—	ns	
Output data delay time	t_{MSSOD}		—	20		
Input data setup time	t_{MSSIS}		40	—		
Input data hold time	t_{MSSIH}		0	—		

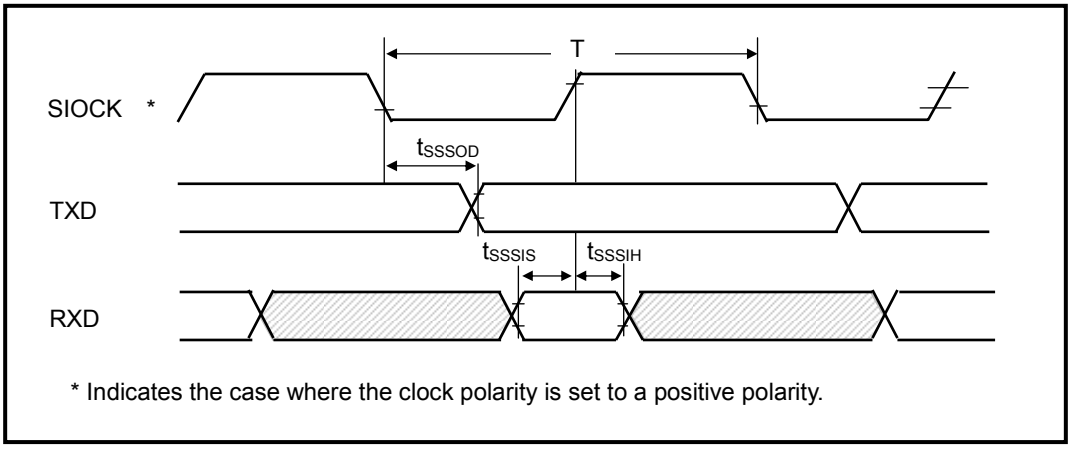


* Indicates the case where the clock polarity is set to a positive polarity.

- Slave mode

(V_{DD_CORE} = 1.35 to 1.65V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit	Remarks
Serial clock cycle	T	C_L =50pF	66.67	—	ns	
Output data delay time	t_{SSSOD}		—	40		
Input data setup time	t_{SSSIS}		20	—		
Input data hold time	t_{SSSIH}		20	—		



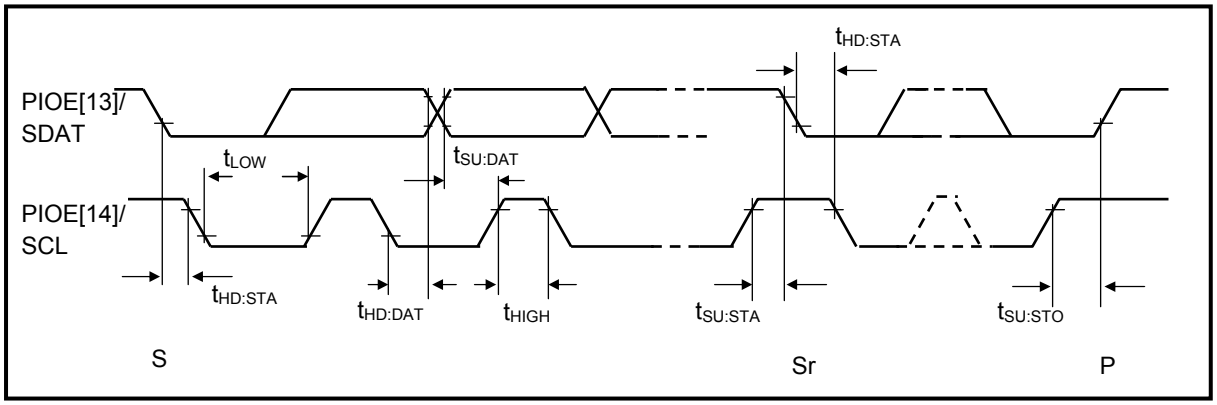
* Indicates the case where the clock polarity is set to a positive polarity.

29.4.7 I2C Bus Timing

(V_{DD_CORE} = 1.35 to 1.65V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Standard mode		Fast mode		Unit
		Min.	Max.	Min.	Max.	
SCL clock frequency	f_{SCL}	—	100	—	400	kHz
SCL clock "L" period	f_{LOW}	4.7	—	1.3	—	μs
SCL clock "H" period	f_{HIGH}	4.0	—	0.6	—	μs
Hold time (repetitive) "START" condition After this period, the first clock pulse is generated.	$t_{HD:STA}$	4.0	—	0.6	—	μs
Setup time for repetitive "START" condition	$t_{SU:STA}$	4.7	—	0.6	—	μs
Data hold time	$t_{HD:DAT}$	5.0	—	—	—	μs
Data setup time	$t_{SU:DAT}$	250	—	100 (*1)	—	μs
Setup time for "STOP" condition	$t_{SU:STO}$	4.0	—	0.6	—	μs

*1 Although a Fast-mode I2C bus device can be used in a standard I2C system, the required condition tSU:DAT ≥ 250 ns must be satisfied. Satisfying this condition implies that that device automatically does not extend a "L" period of the SCL signal.
If a device does not extend a "L" period of the SCL signal, next data must be output to the SDA line before trmax + tSU:DAT = 1000 + 250 = 1250ns elapses after the SCL line is released.



29.4.8 I2S Transceiver/Receiver Timing

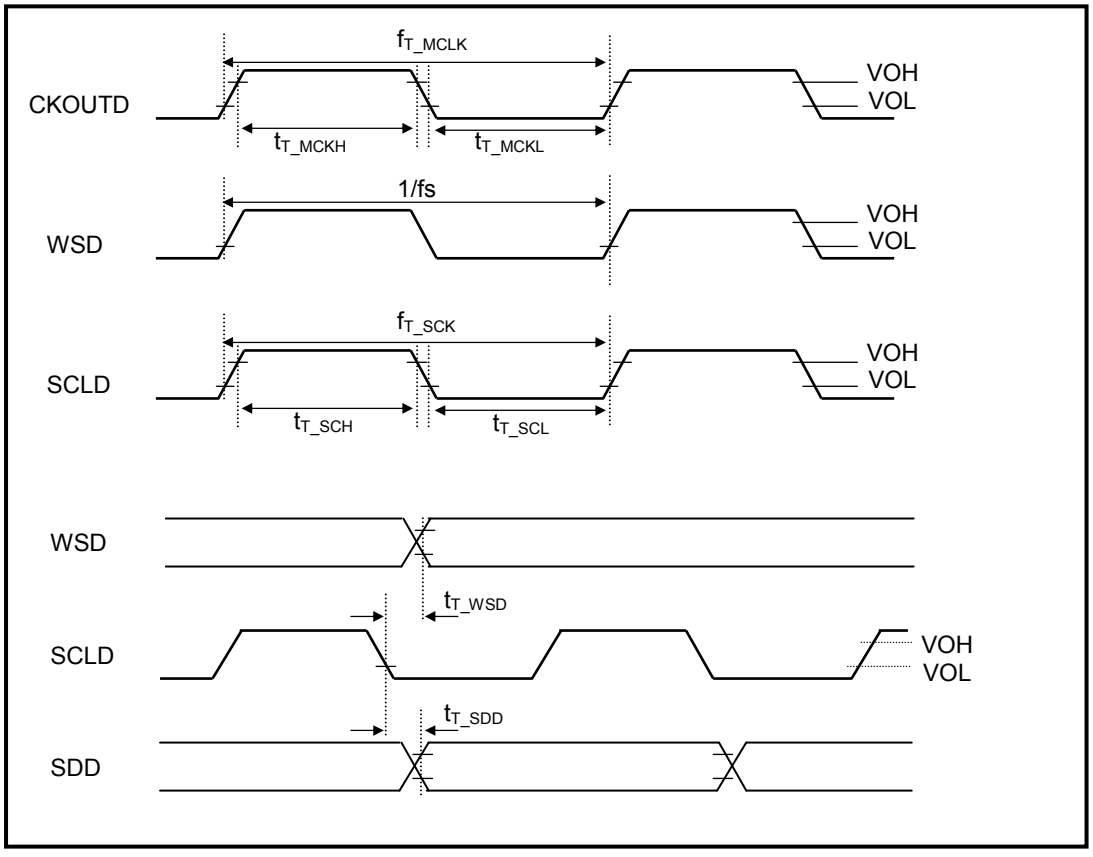
- Transceiver (master mode only)

(V_{DD_CORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	fs=32k	fs=44.1k	fs=48k	Unit	Remarks
Master clock frequency (256fs)	f_{T_MCLK}	—	8.2106	11.2896	12.31597	MHz	Dividing ratio: 1/1, 1/2, 1/4
Sampling frequency	fs		32	44.1	48	kHz	
Serial clock period	f_{T_SCK}		—	32fs	—	kHz	

(V_{DD_CORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	typ.	Max.	Unit	Remarks
WS output delay time	t_{T_WSD}	C_L = 50pF	–50	—	+50	ns	
Serial data (SDD) output delay time	t_{T_SDD}		–50	—	+50	ns	



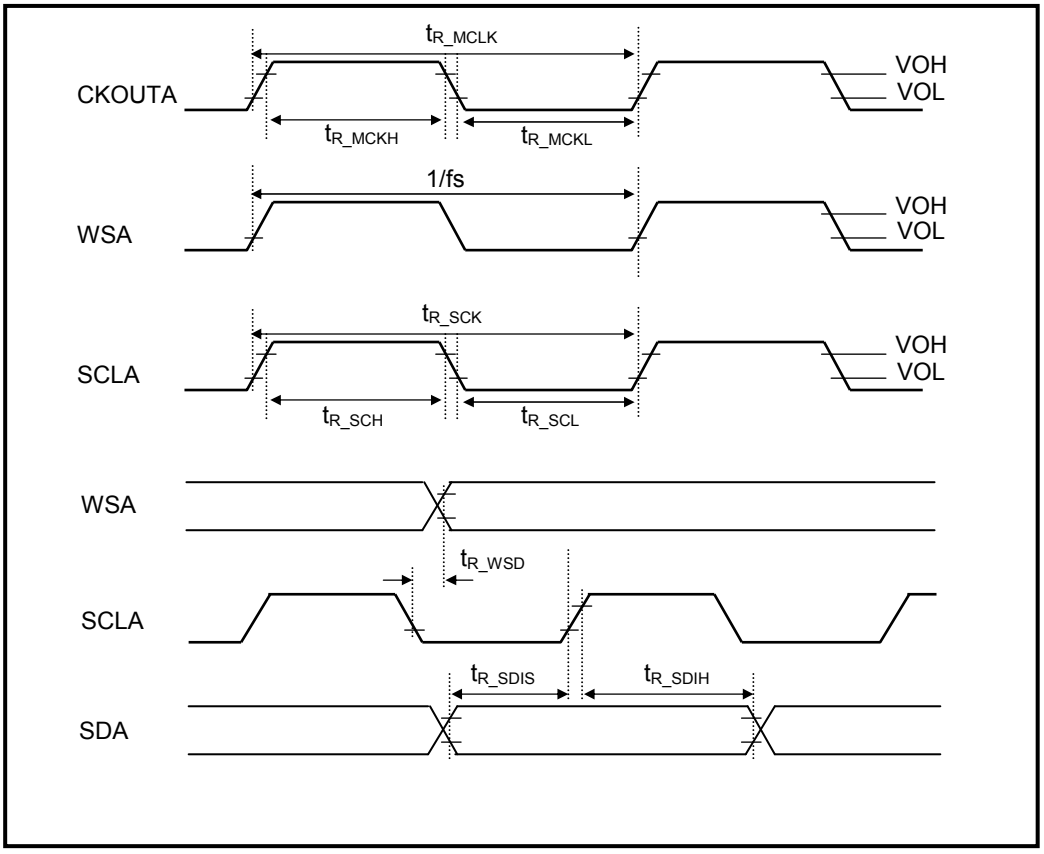
- Receiver (master mode only)

(V_{DD_CORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	fs = 32k	fs = 44.1k	fs = 48k	Unit	Remarks
Master clock frequency (256fs)	f_{R_MCLK}	—	8.2106	11.2896	12.31597	MHz	Dividing ratio: 1/1, 1/2, 1/4
Sampling frequency	fs		32	44.1	48	kHz	
Serial clock period	f_{R_SCK}		—	32fs	—	kHz	

(V_{DD_CORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.		Max.	Unit	Remarks
Serial clock period	f_{R_SCK}	CL = 50pF	—	32fs	—	kHz	
WS output delay time	t_{R_WSD}		–50	—	+50	ns	
Serial data (SDA) setup time	t_{R_SDIS}		50	—	—	ns	
Serial data (SDD) hold time	t_{R_SDIH}		0	—	—	ns	



29.4.9 NAND Flash Interface

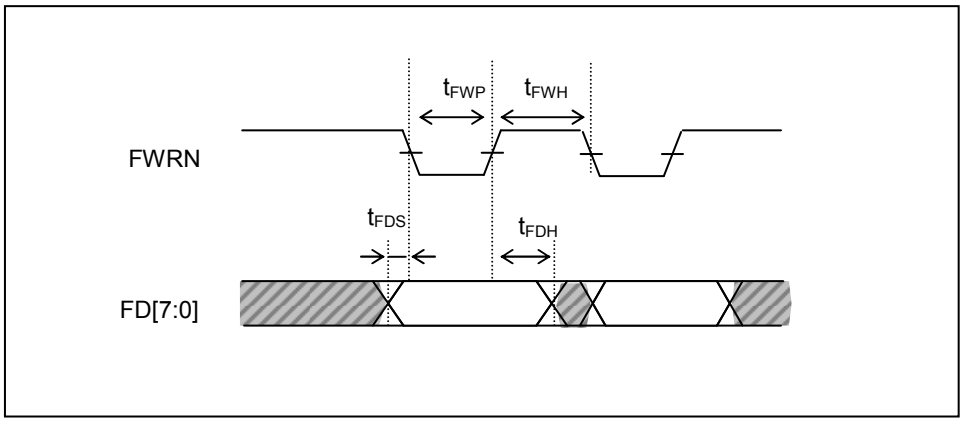
(V_{DD_CORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit	Remarks
FWRN pulse width	tFWP	CL = 50pF	(0.5*w1+1) Tc –10	—	ns	
FWRN "H" output hold time	tFWH		(0.5*w1+1) Tc –10	(0.5*w1+1) Tc +10		
Write data setup time	tFDS		0	—		
Write data hold time	tFDH		w3*Tc –10	—		
FRDN pulse width	tFRP		r1*Tc –10	—		
FRDN "H" output hold time	tFREH		r2*Tc –13	—		
Read data setup time	tFREA		21.5	—		
Read data output hold time	tFRHZ		0	—		

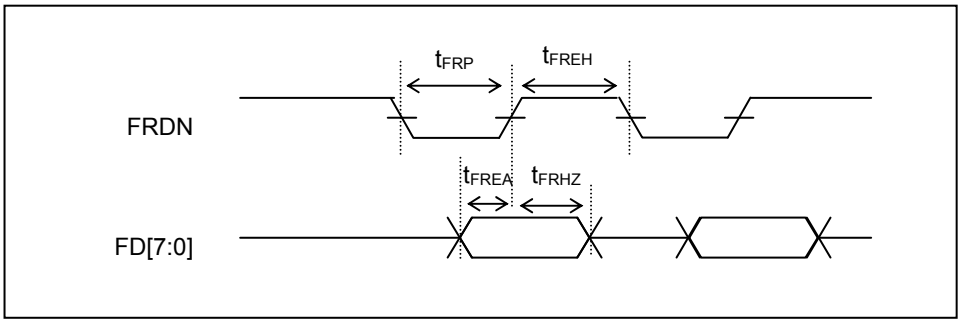
Tc = pclk cycle
Substitute the following values for w1, w2 and w3 (for writing by the sequencer) and for r1 and r2 (for reading by the sequencer):

w1 : w1 = 0 for no write-wait mode; w1 = 1 for 1-write-wait mode; w1 = 2 for 2-write-wait mode
w3 : w3 = 0.5 for no write-wait mode; w3 = 1 for 1-write-wait mode; w3 = 1 for 2-write-wait mode
r1 : r1 = 1.5 for no read-wait mode; r1 = 2 for 1-read-wait mode; r1 = 3 for 2-read-wait mode
r2 : r2 = 0.5 for no read-wait mode; r2 = 1 for 1-read-wait mode; r2 = 1 for 2-read-wait mode

- Write transfer
  Set the number of waits to 1 wait or 2 waits when PCLK = 30 MHz so that the timing for SmartMedia will be satisfied.



- Read transfer
  Set the number of waits to 1 wait or 2 waits when PCLK = 30 MHz so that the timing for SmartMedia will be satisfied.



29.4.10 USB Pin Characteristics

29.4.10.1 Recommended Operating Conditions

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Analog power supply voltage	AVDD	—	3.0	3.3	3.6	V
Digital power supply voltage	VDD	FS mode	1.425	1.50	1.575	V
		HS mode	1.45	1.50	1.55	
Operating temperature	Ta	FS mode	-20	—	+50	°C
		HS mode	0	—	+50	
Reference clock	F_{OSC}	—	48MHz ±50ppm or less			
External reference resistance	R_{exti}	—	12.4 kΩ ±1%			

29.4.10.2 DC Characteristics

Parameter	Condition		Symbol	Min.	Typ.	Max.	Unit
Supply current	HS mode	Recommended Operating Conditions	AVDD	—	65	75	mA
			VDD	—	25	28	mA
	FS mode		AVDD	—	22	26	mA
			VDD	—	12	15	mA
	Suspend mode (When refclk operating)		AVDD	—	200	1400	µA
			VDD	—	150	1400	µA
	Suspend mode (When refclk stopped)		AVDD	—	0.5	1000	µA
			VDD	—	20	1000	µA

29.4.11 A/D Converter Characteristics

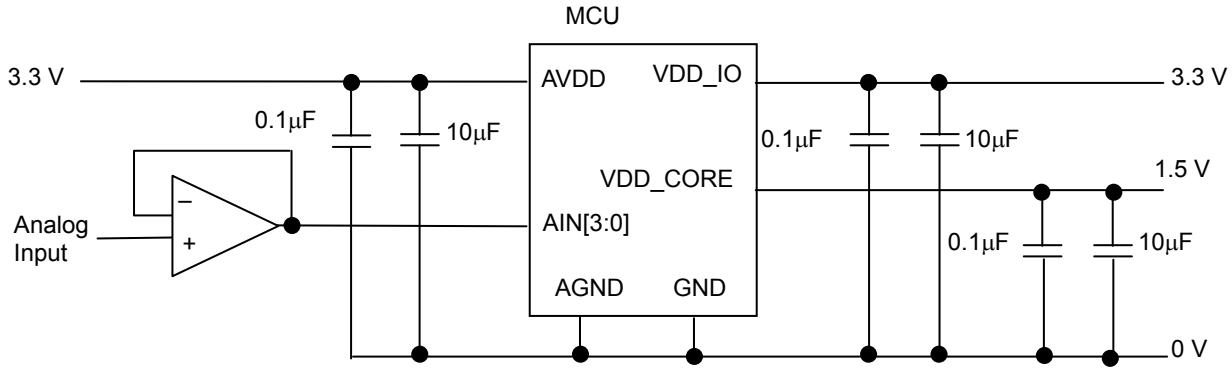
A/D Converter Characteristics

(V_{DD_CORE} = 1.35 to 1.65 V, V_{DD_IO} = 2.7 to 3.6 V, Ta = –30°C to +70°C)

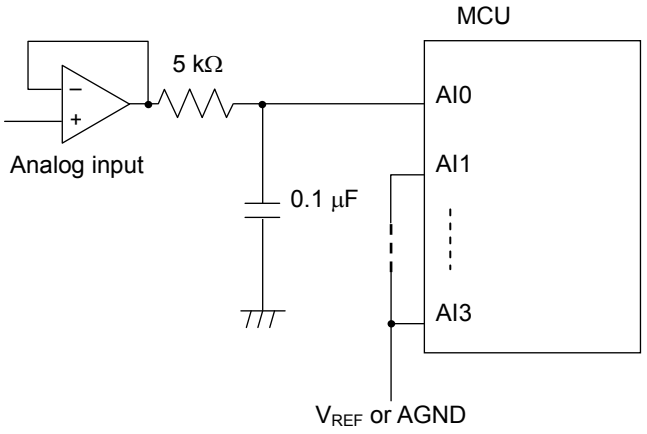
Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	n	—	—	—	10	bit
Linearity error	E_L	Analog input source impedance Ri≤1 kΩ	—	±1	±4	LSB
Differential linearity error	E_D		—	±1	±4	
Zero scale error	E_{ZS}		—	0	±4	
Full-scale error	E_{FS}		—	0	±4	
Conversion time 1	t_{CONV1}	Analog input source impedance Ri≤1 kΩ	2.5	—	31.25	μs
Throughput rate 1			32	—	400	kHz
Conversion time 2	t_{CONV2}	Analog input source impedance Ri≤5 kΩ	10	—	31.25	μs
Throughput rate 2			32	—	100	kHz
Conversion time 3	t_{CONV3}	Analog input source impedance Ri≤10 kΩ	25	—	31.25	μs
Throughput rate 3			32	—	40	kHz

Definition of Terminology

Resolution	Resolution is the value of minimum discernable analog input. With 10 bits, since 2^{10} = 1024, resolution of "(V_{REF} – AGND) ÷ 1024" is possible.
Linearity error	Linearity error is the difference between ideal conversion characteristics and actual conversion characteristics of a 10-bit A/D converter (not including quantization error). Ideal conversion characteristics can be obtained by dividing the voltage between V_{REF} and AGND into 1024 equal steps.
Differential linearity error	Differential linearity error indicates the smoothness of conversion characteristics. Ideally, the range of analog input voltage that corresponds to 1 converted bit of digital output is 1 LSB = (VREF – AGND) ÷ 1024. Differential error is the difference between this ideal bit size and bit size of an arbitrary point in the conversion range.
Zero scale error	Zero scale error is the difference between ideal conversion characteristics and actual conversion characteristics at the point where the digital output changes from 000H to 001H.
Full-scale error	Full-scale error is the difference between ideal conversion characteristics and actual conversion characteristics at the point where the digital output changes from 3FEH to 3FFH.



Measuring Circuit 1



Crosstalk is the difference between the A/D conversion results when the same analog input is applied to AI0 through AI3 and the A/D conversion results of the circuit on the left.

Measuring Circuit 2

Chapter 30

List of Registers

Chapter 30 List of Registers

30.1 List of Registers

Address	Register name	Symbol	R/W	Size	Initial value [H]	Reference page
0xB800_0000	ID Register	IDR	R	32	0x0000_0100	7-5
0xB800_0004	Clock Stop Register	CLKSTP	R/W	32	0x0000_0000	7-6
0xB800_0008	Clock (CGB) Control Register 0	CGBCNT0	R/W	32	0x0000_0000	7-8
0xB800_000C	Clock Supply Wait Time Control Register	CKWT	R/W	32	0x0000_00FF	7-10
0xB800_0010	Remap Control Register	RMPCON	R/W	32	0x0000_0000	7-12
0xB800_0014	<Reserved>	—	—	—	—	—
0xB800_0018	Clock (CGB) Control Register 1	CGBCNT1	—	32	0x0000_0000	7-14
0xB800_001C	Clock (CGB) Control Register 2	CGBCNT2	R/W	32	0x0000_0000	7-15
0x7800_0000	IRQ Register	IRQ	R	32	0x0000_0000	8-6
0x7800_0004	IRQ Software Register	IRQS	R/W	32	0x0000_0000	8-7
0x7800_0008	FIQ Register	FIQ	R	32	0x0000_0000	8-8
0x7800_000C	FIQRAW Register	FIQRAW	R	32	Depends on FIQ_N interrupt input	8-9
0x7800_0010	FIQ Enable Register	FIQEN	R/W	32	0x0000_0000	8-10
0x7800_0014	IRQ Number Register	IRN	R	32	0x0000_0000	8-11
0x7800_0018	Current IRQ Level Register	CIL	R/W	32	0x0000_0000	8-12
0x7800_001C	(Reserved)	—	—	—	—	—
0x7800_0020	IRQ Level Control Register 0	ILC0	R/W	32	0x0000_0000	8-13
0x7800_0024	IRQ Level Control Register 1	ILC1	R/W	32	0x0000_0000	8-15
0x7800_0028	Current IRQ Level Clear Register	CILCL	W	32	—	8-17
0x7800_002C	Current IRQ Level Encode Register	CILE	R	32	0x0000_0000	8-18
0x7BF0_0004	IRQ Clear Register	IRCL	W	32	—	8-19
0x7BF0_0010	IRQA Register	IRQA	R/W	32	0x0000_0000	8-20
0x7BF0_0014	IRQ Detect Mode Setting Register	IDM	R/W	32	0x0000_0000	8-22
0x7BF0_0018	IRQ Level Control Register	ILC	R/W	32	0x0000_0000	8-24
0x7810_0000	Bus Width Control Register	BWC	R/W	32	0x0000_0008	10-3
0x7810_0004	External ROM Access Control Register	ROMAC	R/W	32	0x0000_0007	10-5
0x7810_0008	External SRAM Access Control Register	RAMAC	R/W	32	0x0000_0007	10-7
0x7810_000C	External I/O Bank 0 Access Control Register	IO0AC	R/W	32	0x0000_0007	10-8
0x7810_0010	External I/O bank 1 Access Control Register	IO1AC	R/W	32	0x0000_0007	10-9
0x7818_0000	DRAM Bus Width Control Register	DBWC	R/W	32	0x0000_0000	10-10
0x7818_0004	DRAM Control Register	DRMC	R/W	32	0x0000_0000	10-11
0x7818_0008	DRAM Property Parameter Control Register	DRPC	R/W	32	0x0000_0000	10-13
0x7818_000C	SDRAM Mode Register	SDMD	R/W	32	0x0000_0001	10-14
0x7818_0010	DRAM Command Register	DCMD	R/W	32	0x0000_0000	10-15
0x7818_0014	DRAM Refresh Cycle Control Register 0	RFSH0	R/W	32	0x0000_0000	10-16
0x7818_001C	DRAM Refresh Cycle Control Register 1	RFSH1	R/W	32	0x0000_0000	10-17
0x7818_0018	DRAM Power Down Control Register	PDWC	W	32	0x0000_0003	10-18
0x7BE0_0000	DMA Mode Register	DMAMOD	R/W	32	0x0000_0000	11-5
0x7BE0_0004	DMA Status Register	DMASTA	R	32	0x0000_0000	11-6
0x7BE0_0008	DMA Termination Status Register	DMAINT	R	32	0x0000_0000	11-7
0x7BE0_0100	DMA Channel Mask Register 0	DMACMSK0	R/W	32	0x0000_0001	11-10
0x7BE0_0104	DMA Transfer Mode Register 0	DMACTMOD0	R/W	32	0x0000_0040	11-11
0x7BE0_0108	DMA Transfer Source Address Register 0	DMACSAD0	R/W	32	0x0000_0000	11-13

Address	Register name	Symbol	R/W	Size	Initial value [H]	Reference page
0x7BE0_010C	DMA Transfer Destination Address Register 0	DMACDAD0	R/W	32	0x0000_0000	11-14
0x7BE0_0110	DMA Transfer Count Register 0	DMACSIZ0	R/W	32	0x0000_0000	11-15
0x7BE0_0114	DMA Termination Status Clear Register 0	DMACCINT0	W	32	—	11-15
0x7BE0_0200	DMA Channel Mask Register 1	DMACMSK1	R/W	32	0x0000_0001	11-10
0x7BE0_0204	DMA Transfer Mode Register 1	DMACTMOD1	R/W	32	0x0000_0040	11-11
0x7BE0_0208	DMA Transfer Source Address Register 1	DMACSAD1	R/W	32	0x0000_0000	11-13
0x7BE0_020C	DMA Transfer Destination Address Register 1	DMACDAD1	R/W	32	0x0000_0000	11-14
0x7BE0_0210	DMA Transfer Count Register 1	DMACSIZ1	R/W	32	0x0000_0000	11-15
0x7BE0_0214	DMA Termination Status Clear Register 1	DMACCINT1	W	32	—	11-15
0x7BE0_0300	DMA Channel Mask Register 2	DMACMSK2	R/W	32	0x0000_0001	11-10
0x7BE0_0304	DMA Transfer Mode Register 2	DMACTMOD2	R/W	32	0x0000_0040	11-11
0x7BE0_0308	DMA Transfer Source Address Register 2	DMACSAD2	R/W	32	0x0000_0000	11-13
0x7BE0_030C	DMA Transfer Destination Address Register 2	DMACDAD2	R/W	32	0x0000_0000	11-14
0x7BE0_0310	DMA Transfer Count Register 2	DMACSIZ2	R/W	32	0x0000_0000	11-15
0x7BE0_0314	DMA Termination Status Clear Register 2	DMACCINT2	W	32	—	11-15
0x7BE0_0400	DMA Channel Mask Register 3	DMACMSK3	R/W	32	0x0000_0001	11-10
0x7BE0_0404	DMA Transfer Mode Register 3	DMACTMOD3	R/W	32	0x0000_0040	11-11
0x7BE0_0408	DMA Transfer Source Address Register 3	DMACSAD3	R/W	32	0x0000_0000	11-13
0x7BE0_040C	DMA Transfer Destination Address Register 3	DMACDAD3	R/W	32	0x0000_0000	11-14
0x7BE0_0410	DMA Transfer Count Register 3	DMACSIZ3	R/W	32	0x0000_0000	11-15
0x7BE0_0414	DMA Termination Status Clear Register 3	DMACCINT3	W	32	—	11-15
0xB7A0_0000	Port A Output Register	GPPOA	R/W	32	Undefined	12-7
0xB7A0_0004	Port A Input Register	GPPIA	R	32	Depends on the pin status	12-8
0xB7A0_0008	Port A Mode Register	GPPMA	R/W	32	0x0000_0000	12-9
• • • • • • • • • •	<Reserved>	—	—		—	—
0xB7A0_0020	Port B Output Register	GPPOB	R/W	32	Undefined	12-7
0xB7A0_0024	Port B Input Register	GPPIB	R	32	Depends on the pin status	12-8
0xB7A0_0028	Port B Mode Register	GPPMB	R/W		0x0000_0000	12-9
• • • • • • • • • •	<Reserved>	—	—		—	—
0xB7A0_0040	Port C Output Register	GPPOC	R/W	32	Undefined	12-7
0xB7A0_0044	Port C Input Register	GPPIC	R	32	Depends on the pin status	12-8
0xB7A0_0048	Port C Mode Register	GPPMC	R/W	32	0x0000_0000	12-9
• • • • • • • • • •	<Reserved>	—	—		—	—
0xB7A0_0060	Port D Output Register	GPPOD	R/W	32	Undefined	12-7
0xB7A0_0064	Port D Input Register	GPPID	R	32	Depends on the pin status	12-8
0xB7A0_0068	Port D Mode Register	GPPMD	R/W	32	0x0000_0000	12-9
• • • • • • • • • •	<Reserved>	—	—	—	—	—
0xB7A0_0080	Port E Output Register	GPPOE	R/W	32	Undefined	12-7
0xB7A0_0084	Port E Input Register	GPPIE	R	32	Depends on the pin status	12-8
0xB7A0_0088	Port E Mode Register	GPPME	R/W	32	0x0000_0000	12-9
0xB7A0_008C	Port E Interrupt Enable Register	GPIEE	R/W	32	0x0000_0000	12-10
0xB7A0_0090	Port E Interrupt Polarity Register	GPIPE	R/W	32	0x0000_0000	12-11
0xB7A0_0094	Port E Interrupt Status Register	GPISE	R/W	32	0x0000_0000	12-12
0xB7A0_0098	Port E Interrupt Mode Register	GPIME	R/W	32	0x0000_0000	12-13

Address	Register name	Symbol	R/W	Size	Initial value [H]	Reference page
• • • • • • • • • •	<Reserved>	—	—	—	—	—
0xB7A0_00A0	Port F Output Register	GPPOF	R/W	32	Undefined	12-7
0xB7A0_00A4	Port F Input Register	GPPIF	R	32	Depends on the pin status	12-8
0xB7A0_00A8	Port F Mode Register	GPPMF	R/W	32	0x0000_0000	12-9
0x8000_0008	PIO Pin Switching Register	PIOCTL	R/W	32	0x0000_0000	12-14
0xB7E0_0000	Watchdog Timer Control Register	WDTCON	W	32	—	13-2
0xB7E0_0004	Time Base Counter Control Register	WDTBCON	R/W	32	0x0000_0000	13-3
0xB7E0_0014	Status Register	WDSTAT	R/W	32	0x0000_0000	13-5
0xB800_1004	System Timer Enable Register	TMEN	R/W	32	0x0000_0000	14-4
0xB800_1008	System Timer Reload Register	TMRLR	R/W	32	0x0000_0000	14-5
0xB800_1010	System Timer Overflow Register	TMOVFR	R/W	32	0x0000_0000	14-6
0xB7F0_0000	Timer 0 Control Register	TIMECNTL0	R/W	32	0x0000_0000	14-7
0xB7F0_0004	Timer 0 Base Register	TIMEBASE0	R/W	32	0x0000_0000	14-9
0xB7F0_0008	Timer 0 Counter Register	TIMECNT0	R	32	0x0000_0000	14-10
0xB7F0_000C	Timer 0 Compare Register	TIMECMP0	R/W	32	0x0000_FFFF	14-11
0xB7F0_0010	Timer 0 Status Register	TIMESTAT0	R/W	32	0x0000_0000	14-12
0xB7F0_0020	Timer 1 Control Register	TIMECNTL1	R/W	32	0x0000_0000	14-7
0xB7F0_0024	Timer 1 Base Register	TIMEBASE1	R/W	32	0x0000_0000	14-9
0xB7F0_0028	Timer 1 Counter Register	TIMECNT1	R	32	0x0000_0000	14-10
0xB7F0_002C	Timer 1 Compare Register	TIMECMP1	R/W	32	0x0000_FFFF	14-11
0xB7F0_0030	Timer 1 Status Register	TIMESTAT1	R/W	32	0x0000_0000	14-12
0xB7F0_0040	Timer 2 Control Register	TIMECNTL2	R/W	32	0x0000_0000	14-7
0xB7F0_0044	Timer 2 Base Register	TIMEBASE2	R/W	32	0x0000_0000	14-9
0xB7F0_0048	Timer 2 Counter Register	TIMECNT2	R	32	0x0000_0000	14-10
0xB7F0_004C	Timer 2 Compare Register	TIMECMP2	R/W	32	0x0000_FFFF	14-11
0xB7F0_0050	Timer 2 Status Register	TIMESTAT2	R/W	32	0x0000_0000	14-12
0xB7D0_0000	PWM Register	PWR	R/W	32	0x0000_0000	15-3
0xB7D0_0004	PWM Cycle Register	PWCY	R/W	32	0x0000_0000	15-4
0xB7D0_0008	PWM Counter	PWC	R/W	32	0x0000_0000	15-5
0xB7D0_000C	PWM Control Register	PWCON	R/W	32	0x0000_0000	15-6
• • • • • • • • • •	<Reserved>	—	—	—	—	—
0xB7D0_003C	PWM Interrupt Status Register	PWINTSTS	R/W	32	0x0000_0000	15-8
0xB800_2000	Transmit/Receive Buffer Register	SIOBUF	R/W	32	0x0000_0000	16-3
0xB800_2004	SIO Status Register	SIOSTA	R/W	32	0x0000_0000	16-4
0xB800_2008	SIO Control Register	SIOCON	R/W	32	0x0000_0000	16-6
0xB800_200C	Baud Rate Control Register	SIOBCN	R/W	32	0x0000_0000	16-8
0xB800_2010	(Reserved)	—	—	—	—	—
0xB800_2014	Baud Rate Timer Register	SIOBT	R/W	32	0x0000_0000	16-9
0xB800_2018	SIO Test Control Register	SIOTCN	R/W	32	0x0000_0000	16-10
• • • • • • • • • •	<Reserved>	—	—	—	—	—
0xB600_0004	A/D Control 1 Register	ADCON1	R/W	32	0x0000_0000	17-3
0xB600_0008	A/D Control 2 Register	ADCON2	R/W	32	0x0000_0003	17-4
0xB600_000C	A/D Interrupt Control Register	ADINT	R/W	32	0x0000_0000	17-5
• • • • • • • • • •	<Reserved>	—	—	—	—	—
0xB600_0014	A/D Result 0 Register	ADR0	R/W	32	0x0000_0000	17-6
0xB600_0018	A/D Result 1 Register	ADR1	R/W	32	0x0000_0000	17-6
0xB600_001C	A/D Result 2 Register	ADR2	R/W	32	0x0000_0000	17-6
0xB600_0020	A/D Result 3 Register	ADR3	R/W	32	0x0000_0000	17-6
0xB7B1_0000	SSIO Transmit/Receive Buffer Register 0	SSIOBUF0	R/W	32	Undefined	18-4
0xB7B1_0004	SSIO Transmit/Receive Status Register 0	SSIOSTA0	R/W	32	0x0000_0000	18-5
0xB7B1_0008	SSIO Transmit/Receive Control Register 0	SSIOCON0	R/W	32	0x0000_0000	18-6
0xB7B1_000C	SSIO Interrupt Request Register 0	SSIOINTR0	R/W	32	0x0000_0002	18-8
0xB7B1_0010	SSIO Interrupt Enable Register 0	SSIOINTEN0	R/W	32	0x0000_0000	18-9
0xB7B1_0014	SSIO DMA Transfer Register 0	SSIODMAC0	R/W	32	0x0000_0000	18-10
0xB7B1_0018	SSIO Test Control Register 0	SSIOTSCON0	R/W	32	0x0000_0000	18-11
• • • • • • • • • •	<Reserved>	—	—	—	—	—

Address	Register name	Symbol	R/W	Size	Initial value [H]	Reference page
0xB7B1_0020	SSIO Transmit/Receive Buffer Register 1	SSIOBUF1	R/W	32	Undefined	18-4
0xB7B1_0024	SSIO Transmit/Receive Status Register 1	SSIOSTA1	R/W	32	0x0000_0000	18-5
0xB7B1_0028	SSIO Transmit/Receive Control Register 1	SSIOCON1	R/W	32	0x0000_0000	18-6
0xB7B1_002C	SSIO Interrupt Request Register 1	SSIOINTR1	R/W	32	0x0000_0002	18-8
0xB7B1_0030	SSIO Interrupt Enable Register 1	SSIOINTEN1	R/W	32	0x0000_0000	18-9
0xB7B1_0034	SSIO DMA Transfer Register 1	SSIODMAC1	R/W	32	0x0000_0000	18-10
0xB7B1_0038	SSIO Test Control Register 1	SSIOTSCON1	R/W	32	0x0000_0000	18-11
・・・・・・・・・・	<Reserved>	—	—	—	—	—
0xB7B0_0000	I2C Bus Control Register	I2CCON	R/W	32	0x0000_0000	19-3
0xB7B0_0004	I2C Bus Slave Address Mode Setting Register	I2CSAD	R/W	32	0x0000_0000	19-5
0xB7B0_0008	I2C Bus Transfer Speed Setting Register	I2CCLR	R/W	32	0x0000_0000	19-6
0xB7B0_000C	I2C Bus Status Register	I2CSR	R	32	0x0000_0000	19-7
0xB7B0_0010	I2C Bus Interrupt Request Register	I2CIR	R/W	32	0x0000_0000	19-8
0xB7B0_0014	I2C Bus Interrupt Mask Register	I2CIMR	R/W	32	0x0000_0001	19-9
0xB7B0_0018	I2C Bus Transmit/Receive Data Setting Register	I2CDR	R/W	32	0x0000_0000	19-10
0xB7C0_0000	RTC Register	RTCREG	R/W	32	Undefined	20-3
0xB7C0_0004	RTC Control Register	RTCCON	R/W	32	Undefined	20-4
0xB7C0_0008	RTC Compare Register	RTCCMP	R/W	32	Undefined	20-5
0xB7C0_000C	RTC Scratch Pad Register	RTCSCRP	R/W	32	Undefined	20-6
0xB7C0_0010	RTC Status Register	RTCST	R/W	32	Undefined	20-7
0x8000_0000	Power Down Mask Register	PDMASK	R/W	32	0x0000_0000	21-2
0x8000_0004	Module Clock Stop Register	MCKST	R/W	32	0x0000_0000	21-3
0x8000_0008	PIO Pin Switching Register	PIOCTL	R/W	32	0x0000_0000	21-5
0x8000_000C	I2S Control Register	I2SCNTL	R/W	32	0x0000_0000	21-8
0x8000_0010	Other Configuration Register	OCONFIG	R/W	32	0x0000_0000	21-11
0x8000_0014	Test Register	TST	R/W	32	0x0000_0000	21-12
0x8200_0000	I2S Transceiver FIFO Register	I2SFIFOO	R/W	32	Undefined	22-3
0x8200_0004	I2S Transceiver Control Register 0	I2SCONO0	R/W	32	0x0000_0000	22-4
0x8200_0008	I2S Transceiver Control Register 1	I2SCONO1	R/W	32	0x0000_0000	22-6
0x8200_000C	I2S Transceiver Almost Full Threshold Value Setting Register	I2SAFRO	R/W	32	0x0000_0900	22-8
0x8200_0010	I2S Transceiver Almost Empty Threshold Value Setting Register	I2SAERO	R/W	32	0x0000_0000	22-9
0x8200_0014	I2S Transceiver Interrupt Mask Register	I2SIMRO	R/W	32	0x0000_000F	22-10
0x8200_0018	I2S Transceiver Interrupt Status Register	I2SISTO	R/W	32	0x0000_0000	22-11
0x8200_001C	I2S Transceiver FIFO Write Address Register	I2SWADRO	R/W	32	0x0000_0000	22-12
0x8200_0020	I2S Transceiver FIFO Read Address Register	I2SRADRO	R/W	32	0x0000_0000	22-13
0x8200_0024	I2S Transceiver FIFO Occupied Data Size Register	I2SDNOO	R	32	0x0000_0000	22-14
0x8400_0000	I2S Receiver FIFO Register	I2SFIFOI	R	32	0x0000_0000	23-3
0x8400_0004	I2S Receiver Control Register 0	I2SCONI0	R/W	32	0x0000_0000	23-4
0x8400_0008	I2S Receiver Control Register 1	I2SCONI1	R/W	32	0x0000_0000	23-6
0x8400_000C	I2S Receiver Almost Full Threshold Value Setting Register	I2SAFRI	R/W	32	0x0000_0900	23-7
0x8400_0010	I2S Receiver Almost Empty Threshold Value Setting Register	I2SAERI	R/W	32	0x0000_0000	23-8
0x8400_0014	I2S Receiver Interrupt Mask Register	I2SIMRI	R/W	32	0x0000_000F	23-9
0x8400_0018	I2S Receiver Interrupt Status Register	I2SISTI	R/W	32	0x0000_0000	23-10

Address	Register name	Symbol	R/W	Size	Initial value [H]	Reference page
0x8400_001C	I2S Receiver FIFO Write Address Register	I2SWADRI	R/W	32	0x0000_0000	23-11
0x8400_0020	I2S Receiver FIFO Read Address Register	I2SRADRI	R/W	32	0x0000_0000	23-12
0x8400_0024	I2S Receiver FIFO Occupied Data Size Register	I2SDNOI	R	32	0x0000_0000	23-13
0x8600_0000	Media Bank Register	MBANK	R/W	32	0x0000_0000	24-5
0x8600_0004	Media Sequencer Control Register	MSCTRL	R/W	32	0x0000_0000	24-7
0x8600_0008	Media Sequencer Wait Register	MSWAIT	R/W	32	0x0000_0000	24-10
0x8600_000C	Media Sequencer Status Register	MSSTS	R	32	0x0000_001E	24-11
0x8600_0010	Media Sequencer Interrupt Enable Register	MINTENBL	R/W	32	0x0000_0000	24-13
0x8600_0014	Media Sequencer Error Status Register	MSERR	R	32	0x0000_0000	24-14
0x8600_0018	Media Command Register	MMCMD	W	32	—	24-15
0x8600_001C	Media Address Register	MMADR	W	32	—	24-16
0x8600_0020	Media Select Register	MMSEL	R/W	32	0x0000_0000	24-17
0x8600_0024	Media Data Read Control Register	MMRDCTL	R/W	32	0x0000_0000	24-18
0x8600_0028	Media Section Option Register	MOPTION	R/W	32	0x0000_0000	24-19
0x8600_002C	Media Section Read Data Storage Register	MMRDDATA	R	32	0x0000_0000	24-20
0x8600_0030	ECC Line Parity Register 1	ECCLP1	R	32	0xFFFF_FFFF	24-21
0x8600_0034	ECC Column Parity Register 1	ECCCP1	R	32	0x003F_003F	24-22
0x8600_0038	ECC Error Pointer Register 1	ECCERR1	R	32	0x0000_0000	24-23
• • • • • • • • • •	<Reserved>	—	—	—	—	—
0x8600_0040	Redundancy Part Reserve Data Register 1	HREV1	R/W	32	0x0000_0000	24-24
0x8600_0044	Redundancy Part Data/Block Status & Block Address Register 1	HSTAD1	R/W	32	0x0000_0000	24-25
0x8600_0048	Redundancy Part ECC2-High Register & Redundancy Part ECC2-Low/Block Address 2 Register	HECC2	R/W	32	0x0000_0000	24-27
0x8600_004C	Redundancy Part ECC1-High/Block Address 2 Register & Redundancy Part ECC1-Low Register	HECC1	R/W	32	0x0000_0000	24-29
0x8600_0050	ECC Line Parity Register 3	ECCLP3	R	32	0xFFFF_FFFF	24-21
0x8600_0054	ECC Column Parity Register 3	ECCCP3	R	32	0x003F_003F	24-22
0x8600_0058	ECC Error Pointer Register 3	ECCERR3	R	32	0x0000_0000	24-23
• • • • • • • • • •	<Reserved>	—	—	—	—	—
0x8600_0060	Redundancy Part Reserve Data Register 3	HREV3	R/W	32	0x0000_0000	24-24
0x8600_0064	Redundancy Part Data/Block Status & Block Address Register 3	HSTAD3	R/W	32	0x0000_0000	24-25
0x8600_0068	Redundancy Part ECC4-High Register & Redundancy Part ECC4-Low/Block Address 4 Register	HECC4	R/W	32	0x0000_0000	24-27
0x8600_006C	Redundancy Part ECC3-High/Block Address 4 Register & Redundancy Part ECC3-Low Register	HECC3	R/W	32	0x0000_0000	24-29
0x8600_0070	ECC Line Parity Register 5	ECCLP5	R	32	0xFFFF_FFFF	24-21
0x8600_0074	ECC Column Parity Register 5	ECCCP5	R	32	0x003F_003F	24-22
0x8600_0078	ECC Error Pointer Register 5	ECCERR5	R	32	0x0000_0000	24-23
• • • • • • • • • •	<Reserved>	—	—	—	—	—
0x8600_0080	Redundancy Part Reserve Data Register 5	HREV5	R/W	32	0x0000_0000	24-24
0x8600_0084	Redundancy Part Data/Block Status & Block Address Register 5	HSTAD5	R/W	32	0x0000_0000	24-25

Address	Register name	Symbol	R/W	Size	Initial value [H]	Reference page
0x8600_0088	Redundancy Part ECC6-High Register & Redundancy Part ECC6-Low/Block Address 6 Register	HECC6	R/W	32	0x0000_0000	24-27
0x8600_008C	Redundancy Part ECC5-High/Block Address 6 Register & Redundancy Part ECC5-Low Register	HECC5	R/W	32	0x0000_0000	24-29
0x8600_0090	ECC Line Parity Register 7	ECCLP7	R	32	0xFFFF_FFFF	24-21
0x8600_0094	ECC Column Parity Register 7	ECCCP7	R	32	0x003F_003F	24-22
0x8600_0098	ECC Error Pointer Register 7	ECCERR7	R	32	0x0000_0000	24-23
・・・・・・・・・・	<Reserved>	—	—	—	—	—
0x8600_00A0	Redundancy Part Reserve Data Register 7	HREV7	R/W	32	0x0000_0000	24-24
0x8600_00A4	Redundancy Part Data/Block Status & Block Address Register 7	HSTAD7	R/W	32	0x0000_0000	24-25
0x8600_00A8	Redundancy Part ECC8-High Register & Redundancy Part ECC8-Low/Block Address 8 Register	HECC8	R/W	32	0x0000_0000	24-27
0x8600_00AC	Redundancy Part ECC7-High/Block Address 8 Register & Redundancy Part ECC7-Low Register	HECC7	R/W	32	0x0000_0000	24-29
0x8600_01xx	Media Data Register	MMDATA	R/W	32	Undefined	24-31
0x8800_0000 to 0x8800_0FFC	NAND Flash Buffer Memory	—	R/W	8/16/32	Undefined	24-32
0x7B90_0000 to 0x7B90_0001(*1)	Primary IDE Channel DMA Control	BMICP	R/W	8/16(*3)	0x0000	25-8
0x7B90_0002 to 0x7B90_0003(*1)	Primary IDE Channel DMA Status	BMISP	R/W	8/16(*3)	0x0000	25-9
0x7B90_0004 to 0x7B90_0007(*1)	Primary IDE Channel DMA Descriptor Pointer	BMIDTPP	R/W	8/16/32	0x0000_0000	25-13
0x7B90_0008 to 0x7B90_0009(*2)	Secondary IDE Channel DMA Control	BMICS	R/W	8/16(*4)	0x0000	25-8
0x7B90_000A to 0x7B90_000B(*2)	Secondary IDE Channel DMA Status	BMISS	R/W	8/16(*4)	0x0000	25-9
0x7B90_000C to 0x7B90_000F(*2)	Secondary IDE Channel DMA Descriptor Pointer	BMIDTPS	R/W	8/16/32	0x0000_0000	25-13
0x7B90_0010 to 0x7B90_0017	Reserved register (*5)	—	—	—	—	—
0x7B90_0018 to 0x7B90_001B	DMA Controller Feature Configuration Register	DMAFEAT	R/W	8/16/32	0x0000_0000	25-14
0x7B90_001C to 0x7B90_001F	IDE Clock Control Register	CLKCTL	R/W	8/16/32	0x0000_0000	25-16
0x7B90_0020 to 0x7B90_0023	Reserved register (*5)	—	—	—	—	—
0x7B90_0040 to 0x7B90_0041	Primary Channel IDE Timing Control	IDETIMP	R/W	8/16(*6)	0x0000_0000	25-17
0x7B90_0042 to 0x7B90_0043(*8)	Secondary Channel IDE Timing Control	IDETIMS	R/W	8/16(*6)	0x0000_0000	25-17
0x7B90_0044	Slave IDE Timing Enable	SIDETIM	R/W	8	0x00	25-21
0x7B90_0047	IDE Cable ID Status	IDESTAT	R	8	0b0000_00xx(*9)	25-23
0x7B90_0048 to 0x7B90_0049	UltraDMA Control	UDMACTL	R/W	8/16(*7)	0x0000	25-24
0x7B90_004A to 0x7B90_004B	UltraDMA Timing Control	UDMATIM	R/W	8/16(*7)	0x0000	25-26
0x7B90_0050 to 0x7B90_0053	IDE Timing Override Control	TIMORIDE	R/W	32	0x0000_0000	25-28
0x7B90_0054 to 0x7B90_0057	8-Bit Register Access Strobe Width	REGSTB	R/W	32	0x0000_0000	25-29

Address	Register name	Symbol	R/W	Size	Initial value [H]	Reference page
0x7B90_0058 to 0x7B90_005B	8-Bit Register Access Recovery Time	REGRCVR	R/W	32	0x0000_0000	25-30
0x7B90_005C to 0x7B90_005F	Data Register Access Strobe Width	DATSTB	R/W	32	0x0000_0000	25-31
0x7B90_0060 to 0x7B90_0063	Data Register Access Recovery Time	DATRCVR	R/W	32	0x0000_0000	25-32
0x7B90_0064 to 0x7B90_0067	DMA Access Strove Width	DMASTB	R/W	32	0x0000_0000	25-33
0x7B90_0068 to 0x7B90_006B	DMA Access Recovery Time	DMARCVR	R/W	32	0x0000_0000	25-34
0x7B90_006C to 0x7B90_006F	UltraDMA Access Strobe Width	UDMASTB	R/W	32	0x0000_0000	25-35
0x7B90_0070 to 0x7B90_0073	UltraDMA ready-to-pause Time	UDMATRP	R/W	32	0x0000_0000	25-36
0x7B90_0074 to 0x7B90_0077	UltraDMA t_{ENV} Timing Parameter	UDMATENV	R/W	32	0x0000_0000	25-37
0x7B90_0078 to 0x7B90_0083	Reserved	—	—	—	—	—
0x7B90_0084 to 0x7B90_0087	Reset Pin Control	PINCTL	R/W	8/16/32	0x0000_000x (*10)	25-38
0x7B90_0088 to 0x7B90_008B	IORDY Timer	IORDYTMR	R/W	8/16/32	0x0000_0000	25-40
0x7B90_008C to 0x7B90_00FF	Reserved	—	—	—	—	—
0x7B90_01F0 to 0x7B90_01F7	Primary IDE command block	—	R/W		—	25-7
0x7B90_03F6	Primary IDE control block	—	R/W		—	25-7
0x7B90_0170 to 0x7B90_0177	Secondary IDE command block	—	R/W		—	25-7
0x7B90_0376	Secondary IDE control block	—	R/W		—	25-7
0x7BB0_0000	Revision Register	Rev	R	16	0xFE01	26-7
0x7BB0_0004	System Control Register	SysCtl	R/W	16	0x0000	26-7
0x7BB0_0008	DMA Configuration Register	DMACfg	R/W	16	0x0000	26-9
0x7BB0_000A	DMA Control Register	DMACtl	R/W	16	0x0000	26-10
0x7BB0_000C	Interrupt Status Register	IntStt	R/W	16	0x0000	26-11
0x7BB0_0010	Interrupt Enable Register	IntEnb	R/W	16	0x0100	26-12
0x7BB0_0020	Setup Data 1 Register	SetUpData1	R	16	0x0000	26-13
0x7BB0_0022	Setup Data 2 Register	SetUpData2	R	16	0x0000	26-13
0x7BB0_0024	Setup Data 3 Register	SetUpData3	R	16	0x0000	26-14
0x7BB0_0026	Setup Data 4 Register	SetUpData4	R	16	0x0000	26-14
0x7BB0_0028	Frame Number Register	FrameNum	R	16	0x0000	26-15
0x BB0_002C	Standard Device Request Information Register	StdDevReqInfo	R	16	0x8000	26-15
0x7BB0_0030	EP0 Configuration 1 Register	EP0Cfg1	R/W	16	0x0000	26-16
0x7BB0_0034	EP0 Control Register	EP0Ctl	R/W	16	0x0001	26-16
0x7BB0_0036	EP0 Status Register	EP0Stt	R/W	16	0x0000	26-18
0x7BB0_0038	EP0 Receive Byte Count Register	EP0RxCnt	R	16	0x0000	26-19
0x7BB0_003C	EP0 Transmit Byte Count Register	EP0TxCnt	R/W	16	0x0000	26-19
0x7BB0_0040	EPa Configuration 1 Register	EPaCfg1	R/W	16	0x0000	26-20
0x7BB0_0042	EPa Configuration 2 Register	EPaCfg2	R/W	16	0x0000	26-20
0x7BB0_0044	EPa FIFO Assignment Register	EPaFIFOAsin	R/W	16	0x0000	26-22
0x7BB0_0048	EPa Control Register	EPaCtl	R/W	16	0x0000	26-23
0x7BB0_004C	EPa Status Register	EPaStt	R/W	32	0x0000	26-27
0x7BB0_0050	EPa Receive Byte Count 1 Register	EPaRxCnt1	R	32	0x0000	26-28
0x7BB0_0052	EPa Receive Byte Count 2 Register	EPaRxCnt2	R	32	0x0000	26-28
0x7BB0_0054	EPa Transmit Byte Count 1 Register	EPaTxCnt1	R/W	32	0x0000	26-29
0x7BB0_0056	EPa Transmit Byte Count 2 Register	EPaTxCnt2	R/W	32	0x0000	26-29
0x7BB0_0060	EPb Configuration 1 Register	EPbCfg1	R/W	16	0x0000	26-20
0x7BB0_0062	EPb Configuration 2 Register	EPbCfg2	R/W	16	0x0000	26-20
0x7BB0_0064	EPb FIFO Assignment Register	EPbFIFOAsin	R/W	16	0x0000	26-22

Address	Register name	Symbol	R/W	Size	Initial value [H]	Reference page
0x7BB0_0068	EPb Control Register	EPbCtl	R/W	16	0x0000	26-23
0x7BB0_006C	EPb Status Register	EPbStt	R/W	32	0x0000	26-27
0x7BB0_0070	EPb Receive Byte Count 1 Register	EPbRxCnt1	R	32	0x0000	26-28
0x7BB0_0072	EPb Receive Byte Count 2 Register	EPbRxCnt2	R	32	0x0000	26-28
0x7BB0_0074	EPb Transmit Byte Count 1 Register	EPbTxCnt1	R/W	32	0x0000	26-29
0x7BB0_0076	EPb Transmit Byte Count 2 Register	EPbTxCnt2	R/W	32	0x0000	26-29
0x7BB0_0080	EPc Configuration 1 Register	EPcCfg1	R/W	16	0x0000	26-20
0x7BB0_0082	EPc Configuration 2 Register	EPcCfg2	R/W	16	0x0000	26-20
0x7BB0_0084	EPc FIFO Assignment Register	EPcFIFOAsin	R/W	16	0x0000	26-22
0x7BB0_0088	EPc Control Register	EPcCtl	R/W	16	0x0000	26-23
0x7BB0_008C	EPc Status Register	EPcStt	R/W	32	0x0000	26-27
0x7BB0_0090	EPc Receive Byte Count 1 Register	EPcRxCnt1	R	32	0x0000	26-28
0x7BB0_0094	EPc Transmit Byte Count 1 Register	EPcTxCnt1	R/W	32	0x0000	26-29
0x7BB0_00A0	EPd Configuration 1 Register	EPdCfg1	R/W	16	0x0000	26-20
0x7BB0_00A2	EPd Configuration 2 Register	EPdCfg2	R/W	16	0x0000	26-20
0x7BB0_00A4	EPd FIFO Assignment Register	EPdFIFOAsin	R/W	16	0x0000	26-22
0x7BB0_00A8	EPd Control Register	EPdCtl	R/W	16	0x0000	26-23
0x7BB0_00AC	EPd Status Register	EPdStt	R/W	32	0x0000	26-27
0x7BB0_00B0	EPd Receive Byte Count 1 Register	EPdRxCnt1	R	32	0x0000	26-28
0x7BB0_00B4	EPd Transmit Byte Count 1 Register	EPdTxCnt1	R/W	32	0x0000	26-29
0x7BB0_00C0	EPe Configuration 1 Register	EPeCfg1	R/W	16	0x0000	26-20
0x7BB0_00C2	EPe Configuration 1 Register	EPeCfg2	R/W	16	0x0000	26-20
0x7BB0_00C4	EPe FIFO Assignment Register	EPeFIFOAsin	R/W	16	0x0000	26-22
0x7BB0_00C8	EPe Control Register	EPeCtl	R/W	16	0x0000	26-23
0x7BB0_00CC	EPe Status Register	EPeStt	R/W	32	0x0000	26-27
0x7BB0_00D0	EPe Receive Byte Count 1 Register	EPeRxCnt1	R	32	0x0000	26-28
0x7BB0_00D4	EPe Transmit Byte Count 1 Register	EPeTxCnt1	R/W	32	0x0000	26-29
0x7BB0_00E0	EPf Configuration 1 Register	EPfCfg1	R/W	16	0x0000	26-20
0x7BB0_00E2	EPf Configuration 2 Register	EPfCfg2	R/W	16	0x0000	26-20
0x7BB0_00E4	EPf FIFO Assignment Register	EPfFIFOAsin	R/W	16	0x0000	26-22
0x7BB0_00E8	EPf Control Register	EPfCtl	R/W	16	0x0000	26-23
0x7BB0_00EC	EPf Status Register	EPfStt	R/W	32	0x0000	26-27
0x7BB0_00F0	EPf Receive Byte Count 1 Register	EPfRxCnt1	R	32	0x0000	26-28
0x7BB0_00F4	EPf Transmit Byte Count 1 Register	EPfTxCnt1	R/W	32	0x0000	26-29
0x7BB0_0100	EP0 Transmit FIFO Register	EP0TxFIFO	W	32	Undefined	26-30
0x7BB0_0104	EP0 Receive FIFO Register	EP0RxFIFO	R	32	Undefined	26-31
0x7BB0_1000 to 0x7BB0_13FF	EPa FIFO Register	EPaFIFO	R/W	32	Undefined	26-32
0x7BB0_1400 to 0x7BB0_17FF	EPb FIFO Register	EPbFIFO	R/W	32	Undefined	26-32
0x7BB0_1800 to 0x7BB0_1BFF	EPc FIFO Register	EPcFIFO	R/W	32	Undefined	26-32
0x7BB0_1C00 to 0x7BB0_1FFF	EPd FIFO Register	EPdFIFO	R/W	32	Undefined	26-32
0x7BB0_2000 to 0x7BB0_23FF	EPe FIFO Register	EPeFIFO	R/W	32	Undefined	26-32
0x7BB0_2400 to 0x7BB0_27FF	EPf FIFO Register	EPfFIFO	R/W	32	Undefined	26-32

(*1) Used Since the controller is set to the primary channel. In this LSI, the controller is fixed to the primary channel in the specification.
(*2) Used when the controller is set to the secondary channel. This register cannot be used in this LSI.
(*3) Access in 32-bit mode is possible by combining the BMICP register and BMISP register.
(*4) Access in 32-bit mode is possible by combining the BMICS register and the BMISS register. This register cannot be used in this LSI.
(*5) Read/Write access to reserved registers is prohibited. When the registers are accessed, the operation is not guaranteed.
(*6) Access in 32-bit mode is possible by combining the IDETIMP register and the IDETIMS register.
(*7) Access in 32-bit mode is possible by combining the UDMACTL register and the UDMATIM register.

(*8) Used when the controller is set to the secondary channel. This register cannot be used in this LSI.
(*9) The values of the npcblid and nscblid signals are reflected in the initial value of the IDESTAT register. Either of the two bits becomes valid according to whether the controller is set to the primary or secondary channel. Refer to the specification of the IDESTAT register for details.
(*10) The initial values of bits 31-1 of the PINCTL register are 0. The initial value of bit 0 is determined by the value of the reset signal of the IDE device.

[Note]
All registers should be accessed by the sizes specified in the table. Otherwise operation is not guaranteed.

Chapter 31

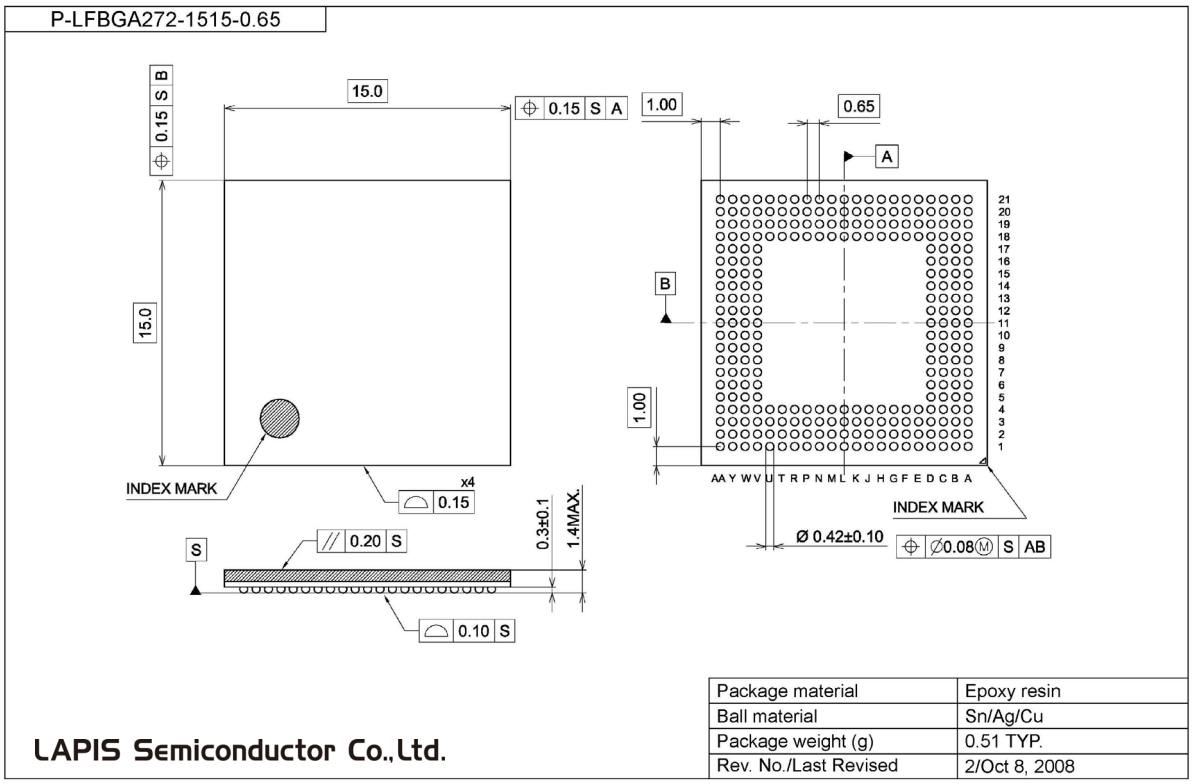
Package Dimensions

Chapter 31 Package Dimensions

31.1 Package Dimensions

272pin LFBGA(ML696201:P-LFBGA272-1515-0.65, ML69Q6203:P-LFBGA272-1515-0.65-1-MC)

(Unit: mm)



Package material	Epoxy resin
Ball material	Sn/Ag/Cu
Package weight (g)	0.51 TYP.
Rev. No./Last Revised	2/Oct 8, 2008

Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact ROHM's responsible sales person for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

Revision History

Revision History

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-01	Nov. 30, 2004	–	–	Final edition 1
FEUL696201-02	May 20, 2005	1-3	1-3	Changed 8 x 1 channel to 7 x 1 channel in the first sentence in the description of "GPIO".
		1-10 to 1-17	1-10 to 1-18	Added the "Output current" Column in Table 1-2.
		1-19	1-20	Added the note in the "Debug support" Section in the table in Section 1.3.3.
		1-22	1-23	Added the note in the descriptions of Symbols "VDDFLA" and "GNDFLA".
		6-2	6-2	Changed Figure 6-1, Reset Signal Timing.
		8-3	8-3	Added "_" to addresses in the "Address" Column and values in the "Initial value" Column in the table in Section 8.1.3.
		10-2	10-2	- Added "_" to the addresses in the "Address" Column and the values in the "Initial value" Column in the table in Section 10.1.2. - Changed the register name of Symbol "RAMAC" from "External RAM access control register" to "External SRAM access control register" in the table in Section 10.1.2.
		10-5	10-5	Added BRST to bit 4 of the ROMAC register.
		–	10-6	Added the description of the BRST bit.
		10-6	10-7	Added BRST to bit 4 in the RAMAC register.
		–	10-7	Added the description of the BRST bit.
		10-9	10-10	Changed the values of bits 0 and 1 at reset from 1 to 0 in the bit representation of the DBWC register.
		10-19	10-20	Modified contents and added the description of the XWAIT signal, in Section 10.3.4.
		–	10-29	Added Figure 10-9 and Figure 10-10.
		10-29	10-30	Changed Figure 10-9 and Figure 10-10 to Figure 10-11 and Figure 10-12, respectively.
		10-30	10-31	Changed Figure 10-11 and Figure 10-12 to Figure 10-13 and Figure 10-14, respectively.
		11-4	11-4	Added "_" to addresses in the "Address" Column and values in the "Initial value" Column in the table in Section 11.1.3.
		11-20	11-20	Changed the table in Section 11.3.6, Notes on Use

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-02	May 20, 2005	12-6	12-6	- Added "_" to the addresses in the "Address" Column in the table in Section 12.1.3. - Changed the symbols of "Port D mode register" from GPPM to GPPMD in the table. - Changed initial values of "Port A mode register", "Port B mode register", "Port C mode register", "Port D mode register", Port E mode register", "Port E interrupt enable register", "Port E interrupt polarity register", "Port E interrupt status register", "Port E interrupt mode register", "Port F mode register", and "PIO pin switching register" from 0x0000 to 0x0000_0000 in the table in Section 12.1.3.
		13-1	13-1	Added "_" to the addresses in the "Address" Column and changed the values in the "Initial value" Column, in the table in Section 13.1.2.
		14-3	14-3	Added "_" to addresses in the "Address" Columns and values in the "Initial value" Columns in the tables in Section 14.1.2.
		15-2	15-2	Added "_" to addresses in the "Address" Column and values in the "Initial value" Column and changed the register name of symbol "PWINTSTS", in the table in Section 15.1.3.
		16-2	16-2	Added "_" to addresses in the "Address" Column and changed the values in the "Initial value" Column, in the table in Section 16.1.3.
		17-2	17-2	Added "_" to addresses in the "Address" Column and changed the value in the "Size" Column and the values in the "Initial value" Column, in the table in Section 17.1.3.
		17-3, 4, 5, 6	17-3, 4, 5, 6	Changed the access size from 16 bits to 32 bits in the bit representations of the registers in Sections 17.2.1 to 17.2.4.
		18-3	18-3	Changed the symbols of "SSIO transmit/receive status register 0" and "SSIO transmit/receive status register 1" from SSIOST0 and SSIOST1 to SSIOSTA0 and SSIOSTA1 and added the "Size" Column, in the table in Section 18.1.3.
		18-4	18-4	Modified the content in Section 18.2.2.
		18-5	18-5	- Modified the description of the SSIO status registers in Section 18.2.3. - Corrected the heading of Section 18.2.3 from "SSIO Status Registers" to "SSIO Transmit/Receive Status Registers".
		18-6	18-6	Corrected the heading of Section 18.2.4 from "SSIO Control Registers" to "SSIO Transmit/Receive Control Registers".
		18-6, 8, 9, 10, 11	18-6, 8, 9, 10, 11	Changed 8-bit to 32-bit in the descriptions in Sections 18.2.4 to 18.2.8.
		18-16, 17	18-16, 17	Modified the contents in Sections 18.3.3, 18.3.3.1, and 18.3.3.2.
		19-2	19-2	Changed the register name from "I2C bus interrupt setting register" to "I2C bus interrupt mask register" in the table in Section 19.1.3.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-02	May 20, 2005	19-7	19-7	Changed the value at reset of bit 1 from 1 to 0 and "Access: R/W" to "Access: R" in the bit representation of the I2CSR register and modified the content of [Note], in Section 19.2.4.
		20-2	20-2	Changed the register name of Symbol "RTCREG" in the table in Section 20.1.3.
		21-1	21-1	Added the "Size" Column in the table in Section 21.1.3.
		21-5	21-5	Added "Access size: 32 bits" in the bit representation of the POICTL register in Section 21.2.3.
		22-14	22-14	Corrected I2SDNO to I2SDNOO in (3) in Condition 3 in [I2SDNOO Operation] in [Description of Bits] in Section 22.2.10.
		23-13	23-13	Modified contents of Condition 3 in [I2SDNOI Operation] in [Description of Bits] in Section 23.2.10
		24-4	24-4	Corrected the address and access size of "Media data register" in the table in Section 24.1.3.
		24-17	24-17	Corrected the register name in the bit representation from MMRDCTL to MMSEL in Section 24.2.9.
		24-21	24-21	Corrected values at reset from 0 to 1 in the bit representation of the ECCLPx register in Section 24.2.13.
		24-31	24-31	Changed the register name from MMSEL to MMDATA, the values at reset from 0 to x, and Address from "0x8600_01xx" to "0x8600_0100 to 0x8600_01FC", and modified the content of [Note], in the bit representation in Section 24.2.20.
		–	24-32	Added Section 24.2.21, NAND Flash Buffer Memory.
		–	–	Changed "device 0" to "device 0 (master)" and "device 1" to "device 1 (slave)" on most pages in Chapter 5.
		25-1	25-1	Modified the contents of the Feature Section.
		25-5	25-5	Added "_" to initial values and corrected names of registers in the "Name" Column, in the table in Section 25.1.3.1.
		25-6	25-6	Added "_" to initial values and changed the initial values of addresses "0x40-0x41" and "0x42-0x43" from 0x0000 to 0x0000_0000 in the table in Section 25.1.3.2.
		25-11	25-11	Added one sentence, "However, device 1 (slave) cannot be used for this LSI." in the description of the DMAEN1 bit.
		25-14	25-14	Corrected the heading of Section 25.2.1.4 from "DMA Controller Feature Configuration" to "DMA Controller Feature Configuration Register".
		25-16	25-16	Corrected the heading of Section 25.2.1.5 from "IDE Clock Control" to "IDE Clock Control Register".
		25-17	25-17	Corrected the heading of Section 25.2.2.1 from "Primary/Secondary IDE Timing Control" to "Primary/Secondary Channel IDE Timing Control".
		25-18 to 25-21	25-18 to 25-21	Added one sentence, "However, device 1 (slave) cannot be used for this LSI." in each of the descriptions of the PIOFTIM1, RDYSEN1, DMAFTIM1, SLVTMEN, RDYRCYP1[1:0], and RDYSMPP1[1:0] bits and Section 25.2.2.2.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-02	May 20, 2005	25-24	25-24	Added one sentence, "However, device 1 (slave) cannot be used for this LSI." in the description of the UDMAP1 bit.
		25-27	25-27	Added one sentence, "However, device 1 (slave) cannot be used for this LSI." in the description of the TCYCP1[2:0] bit.
		25-29 to 25-37	25-30 to 25-38	Added one sentence, "However, device 1 (slave) cannot be used for this LSI." in each of the descriptions of the REGSTB1[6:0], REGRCVR1[6:0], DATSBTB1[6:0], DATRCVR1[6:0], DMASTB1[6:0], DMACRCVR1[6:0], UDMASTB1[3:0], UDMATRP1[4:0], and UDMTENV1[3:0] bits.
		25-42	25-43	Added one sentence on the end of the description of Section 25.3.2.
		–	25-48	Added Section 25.3.2.4, "Cautions regarding DMA transfer".
		25-52	25-54	Added one sentence on the top of the <Setting the timing of device (slave)> Section.
		25-57	25-59	Added one sentence on the top of the Section 25.3.4.3.
		–	25-62	Added Section 26.3.6, Notes on Connecting the ATA Interface.
		26-1	26-1	Added one feature to the "FEATURES" Section in Section 26.1.
		26-4	26-4	- Added the description of USB Module pin "HRESET" in the table in Section 26.1.2. - Changed the resistor value in the description of LSI pin USB_REXT in the table in Section 26.1.
		26-5, 6	26-5, 6	- Added "_" to the address in the "Address" Column and corrected names of registers in the "Name" Column, in the table in Section 26.1.3. - Changed the initial value of address "0x7BB0_0004" from "0x0000" to "0x0080". - Eliminated the R/Ws and the initial values of addresses "0x7BB_00042", "0x7BB_00062", "0x7BB_00082", "0x7BB_000A2", "0X7BB_000C2", and "0x7BB_000E2", and the addresses themselves. - Changed the initial values of addresses "0x7BB_00040", "0x7BB_00060", "0x7BB_00080", "0x7BB_000A0", "0X7BB_000C0", and "0x7BB_000E0" from "0x0000" to "0x0000_0000".
		26-8	26-8	Changed "PDCtl : Bit 6" to "PDCtl : Bit 5" and modified the descriptions of the RWA, PD, and SR bits.
		26-9	26-9	Changed the description of the DREQItvl bits in [Description of Bits] in Section 26.2.3.
		26-11	26-11	Added (Note 2) in the bit representation of the IntStt register and the content of (Note), in Section 26.2.5.
		26-15, 16,19	26-15, 16,19	Modified the bit representations of the FrameNum register in Section 26.2.11, the EP0Cfg1 register in Section 26.2.13, and the EP0TxCnt register in Section 26.2.17.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-02	May 20, 2005	26-18	26-18	Eliminated bits 31 to 16 of the EP0Stt register in Section 26.2.15.
		26-20	26-20	- Modified the bit representations of the EPa-EPfCfg2 and EPa-EPfCfg1 registers in Sections 26.2.18 and 26.2.19. - Modified the description of the TPuF bits in [Description of Bits].
		26-21	26-21	Modified the description of the EPType bits in [Description of Bits].
		26-23, 26	26-23, 26	Modified the descriptions of the ISOodSel, PreFrmInTkn, and FIFOClr bits in Section 26.2.21.
		26-27	26-27	Modified (Note 2) in Section 26.2.22.
		26-29	26-29	- Modified the bit representations of the EPa-EPbTxCnt2 and EPa-EPfTxCnt1 registers in Sections 26.2.25 and 26.2.26. - Modified the description of the TxByteCunt1/2 bits in [Description of Bits].
		26-36	26-36	Modified the description in Section 26.3.2.5.
		26-51	26-51	Partially changed contents in Section 26.5.4.2, USB Power Supply.
		26-52	26-52	- Changed the resistor value from 12.4 to 12.8 in Section 26.5.4.3, Reference Resistor. - Changed the power supply circuit and notes in Figure 26-3.
		26-55	26-55	Corrected the word from "IP" to "module" in the sentence on the top line in No.4 in "Phenomenon" Column in the table.
		–	27-1 to 27-16	Added Chapter 27.
		–	28-1 to 28-3	Added Chapter 28.
		27-1 to 27-49	29-1 to 29-49	Changed Chapter 27 to Chapter 29 and accordingly changed the Section numbers.
		27-1	29-1	Changed the Symbol of Parameter "Digital power supply voltage" from V_{DDIO} to V_{DD_IO} in the table in Section 29.1.
		27-2	29-2	Changed the Symbol of Parameter "CPU operating frequency" from f_{OSC} to F_{OSC}.
		27-3	29-3	Changed the conditions of IL1, IL4, IL6, IL7, and IL8 in the table in Section 29.3.
		27-6	29-6	Changed "Ta = –40 to + 70°C" to "Ta = –30 to + 70°C" above the table in Section 29.4.2.
		27-45	29-45	- Eliminated the content of symbol R_{PU} in the table. - Changed the Typ. and Max. values for AVDD and VDD of Condition "HS mode" in the DC characteristics table in Section 29.4.11.2.
		–	30-1 to 30-8	Added Chapter 30.
		28-1	31-1	Changed Chapter 28 to Chapter 31 and accordingly changed the Section number.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-03	Mar. 6, 2006	1-2	1-2	Changed the contents of "High Speed USB" and "IDE Controller" in the "Features" Section in Section 1.1.
		1-8	1-8	Eliminated "/-" from the power supply pin symbols in the table.
		7-1	7-1	Partially added the description of Section 7.2.1.2.
		7-2	7-2	Modified the description of Section 7.2.2.1.
		7-2	7-2	Added Section 7.2.2.1.1 and Section 7.2.2.1.2.
		7-18	7-18	Modified the description of Section 7.5.5.
		10-19 and 10-20	10-19 and 10-20	Changed Sections 10.3.3 and 10.3.4 to Sections 10.3.4 and 10.3.3, respectively.
		–	10-24 to 10-32	Changed Section 10.3.7 from "Read OFF Time Control" to "Timing Settings" and added Section 10.3.8 as "Read OFF Time Control".
		–	10-32	Changed "Read Off time" to "Data Off time" in contents in Section 10.3.8.
		11-1	11-1	Changed the content of "DMA transfer request" in "Features" in Section 11.1.
		11-3	11-3	Modified the μPLAT946 block in Figure 11-2.
		11-16	11-16	Modified the content of Section 1 in Section 11.3.2, DMA activation Factors.
		12-2	12-2	Revised the clock signal name from "HCLK" to "PCLK" in Figure 12-1.
		12-3	12-3	Revised the clock signal name from "HCLK" to "PCLK" in Figure 12-2.
		12-7	12-7	Corrected the register name from GPPOF [15:8] to GPPOF [15:7] in the description of Section 12.2.1.
		12-9	12-9	- Corrected the register name from GPPMF [15:8] to GPPMF [15:7] in the description of Section 12.2.3. - Corrected the register name from GPPME[15:0] to GPPME[15:0] and added GPPMF[7:0] (bits 7 to 0), in [Bit Description].
		12-10	12-10	Corrected the register name from GPIED[15:12] to GPIEE[15:12] in Section 12.2.4.
		12-11	12-11	Corrected the register name from GPIPD[15:12] to GPIPE[15:12] in Section 12.2.5.
		12-13	12-13	Corrected the register name from GPIMD[15:12] to GPIME[15:12] in Section 12.2.7.
		12-17	12-17	- Partially modified the description of Section 12.3. - Corrected symbols PIOD15 and PIOD12 to PIOE15 and PIOE12, respectively, in the top line in Section 12.3.1. - Partially modified the description of Section 12.3.2, Primary/Secondary Function Control.
		18-1	18-1	Partially modified contents of "Features" in Section 18.1.
		18-3	18-3	Changed symbol "DMA" to "automatic data" in the table in Section 18.1.3.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-03	Mar. 6, 2006	18-7	18-7	Changed symbol "DMA" to "automatic data" in the description of SFTSLVx in Section 18.2.4.
		18-8	18-8	Changed symbol "DMA" to "automatic data" in the description of TRCENDx in Section 18.2.5.
		18-9	18-9	Changed symbol "DMA" to "automatic data" or "automatic data transfer" in Section 18.2.6.
		18-10	18-10	Changed symbol "DMA" to "automatic data" or "automatic data transfer control circuit" in Section 18.2.7.
		18-16	18-16	Modified contents of Sections 18.3.3 and 18.3.3.1.
		18-17	18-18	- Modified contents of Section 18.3.3.2. - Changed symbol DMA to "automatic data" in Section 18.3.3.2
		18-18	18-19	Changed symbol DMA to "automatic data" in Section 18.3.4.
		19-12	19-12	Modified the timing diagram in Section 19.3.6.
		20-8	20-8	- Changed Sections 20.3.1, 20.3.2, and 20.3.3 to Sections 20.3.2, 20.3.3, and 20.3.1, respectively. - Added Section 20.3.4, Note on Successive Accesses to Registers.
		24-1	24-1	Modified contents of "Features" in Section 24.1, Overview.
		24-8	24-8	Modified the description in the table of "HEAD[1:0]" bits in Section 24.2.2.
		24-24	24-24	Changed packet x to packet x +1 in "REVx+1D3[7:0]" and "REVx+1D4[7:0]" in [Bit Description] in Section 24.2.16.
		24-8	24-8	Partially eliminated the description of ECC (bit 6) in Section 24.2.2.
		24-9	24-9	Partially eliminated the description of PARITY (bit 7) in Section 24.2.2.
		25-1	25-1	Added one sentence to the description of Section 25.1, "Overview".
		25-16	25-16	Modified "IDE interface clock (ide_clk_in)" to "IDE interface clock (ide_clk signal)" in the description of the ICLKDIS bit in [Bit Description] in Section 25.2.15.
		25-49	25-49	Modified "AHB clock (clk signal)" to "AHB clock (hclk signal)" in the description of Section 25.3.3.
		25-52	25-52	Modified "IDE clock (clock signal "clk")" to "IDE clock (ide_clk signal)" in the description of Section 25.3.4.
		25-55	25-55	Modified "IDE clock (clock signal "clk")" to "IDE clock (ide_clk signal)" in the description of Section 25.3.4.2.
		25-61	25-61	Modified "HCLOCK (clk signal)" to "HCLK (hclk signal)" in the description of Section (3) in Section 25.3.5.3.
		26-1	26-1	Modified descriptions of Section 26.1, "Overview" and "Features".
		26-4	26-4	- Changed the resistor value from 12.8 to 12.4 in the description of LSI Pin "USB_REXT" in the table. - Changed symbols "DMA" and "DREQ" to "automatic data transfer" and "automatic data transfer request" respectively, in the table.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-03	Mar. 6, 2006	26-5	26-5	Changed symbol "DMA" to "Automatic data transfer" in the table.
		26-9	26-9	Changed symbols "DMA" and "DREQ" to "automatic data" and "automatic data transfer request" respectively, in Section 26.2.3.
		26-10	26-10	Changed symbols "DMA" and "DREQ" to "automatic data" and "automatic data transfer request" respectively, in Section 26.2.4.
		26-20	26-20	- Partially added the content in [Notes] in Section 26.2.19. - Eliminated the TPuF bit in [Description of Bits] in Section 26.2.19.
		26-21	26-21	Partially modified the description of the EPType bits.
		26-22	26-22	Partially eliminated the description of the BufMod bit in [Description of Bits] in Section 26.2.20.
		26-23 to 26-25	26-23	Changed the description of the ISOModeSel bit in [Description of Bits] in Section 26.2.21.
		26-26	26-23	- Changed the description of the PreFmInTkn bit in [Description of Bits] in Section 26.2.21. - Partially modified the description of the FIFOClr bit in [Description of Bits] in Section 26.2.21.
		26-28	26-25	Changed descriptions in [Description of Bits] in Section 26.2.24.
		26-29	26-26	Modified descriptions in [Description of Bits] in Section 26.2.26.
		26-33	26-30	Changed symbol "DMA" to "automatic data transfer" and eliminated Transfer type "Isochronous" in the two tables in Section 26.3.1.
		26-34	26-31	Changed Transfer types "Isochronous OUT" and "Isochronous IN" to "Bulk OUT" and "Interrupt IN" respectively, in the table.
		26-35	26-32	Changed symbol "DMA" and "DREQ" to "automatic data" and "automatic data transfer request" respectively, in Sections 26.3.2.1, 26.3.2.2, and 26.3.2.3.
		26-36	26-33	- Changed symbol "DMA" to "automatic data" in Section 26.3.2.4. - Changed descriptions of Section 26.3.2.5.
		26-37	26-34	Changed symbol "DMA" to "automatic data transfer" in Section 26.3.2.6.
		26-42	26-39	Eliminated the sentence related to isochronous transfer in Section 26.3.3.11.
		26-44	–	Eliminated Section 26.3.3.15.
		26-46	26-46	Changed "USB 2.0 PHY IP" to "USB 2.0 PHY" in the description of Section 26.5.4.
		26-51	26-48	Eliminated sentence "Connect - - - - Figure 26-3" in Section 26.5.4.2, USB Power Supply.
		26-52	26-49	- Modified the heading of Section 26.5.4.3 from "Reference Resistor (rexti) to "Reference Resistor (USB_REXT). - Changed figure 12.8 and word rexti to 12.4 and USB_REXT respectively, in the top sentence.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL696201-03	Mar. 6, 2006	26-54 and 26-55	26-50	Changed contents in Section 26.5.6 and 26.5.7.
		27-1	27-1	Changed "Sector writing: 1000 cycles" to "Sector writing: 10000 cycles" in "Highly reliable read/write" in "Features".
		29-2	29-2	Modified the table in Section 29.2 and added *1 in "Note" below the table.
		29-15	29-15	Changed "SDRAM read cycle" to "External bus timing - SDRAM read cycle" in Section 29.4.5.
		29-26 to 29-26	–	Eliminated Section 29.4.10.
		29-45	29-26	- Changed Section 29.4.11 to 29.4.10 - Changed Section 29.4.11.1 to 29.4.10.1 and added Parameter "External reference resistance" in the "Recommended Operating Conditions" table. - Changed Section 29.4.11.2 to 29.4.10.2. - Eliminated Section 29.4.11.3.
		29-46	29-47	Eliminated Sections 29.4.11.4 to 29.4.11.6.
		29-48	29-27	Changed Section 29.4.12 to 29.4.11.
		29-49	29-28	Changed VDD_CORE voltage from 1.8 V to 1.5 V in Measuring Circuit 1.
		30-6 and 30-7	30-6 and 30-7	Modified addresses in the "Address" Column in the table in Section 30.1.
FEUL696201-04	Jun. 30, 2006	29-2	29-2	The Min,Typ,and Max values of V_{DD_CORE}, V_{DDPL}, and V_{DDRTC} and Ta have been changed depending on USB operating modes in the table in Section 29.2.
		29-26	29-26	- The Min, Typ, and Max values of VDD and Ta have been changed by adding FS and HS modes in the "Conditions" Column in the table in Section 29.4.10.1. - The condions of VDD and Ta have been changed in the "DC Characteristics" table in Sections 29.4.10.20.
FEUL696201-05	Mar. 27, 2009	-	-	Changed OKI to OKI SEMICONDUCTOR.
		1-8 and 1-9	1-8 and 1-9	Changed the symbol of "R19" and "W1" from VSSFLA to GNDFLA in the table in Section 1.3.1.2.
		2-4	2-4	Modified the description of mirroring in Section 2.5.1.
		26-50	26-50	Eliminated Section 26.5.7, Note on compliance testing.
		26-50	26-50	Added Section 26.5.7, Note on accessing EP FIFO register.
		31-1	31-1	Changed the figure in Section 31.1, package height from "1.3MAX." to "1.4MAX.".