OKI SEMICONDUCTOR

Dear customers,

About the change in the name such as "Oki Electric Industry Co. Ltd." and "OKI" in documents to OKI Semiconductor Co., Ltd.

The semiconductor business of Oki Electric Industry Co., Ltd. was succeeded to OKI Semiconductor Co., Ltd. on October 1, 2008. Therefore, please accept that although the terms and marks of "Oki Electric Industry Co., Ltd.", “Oki Electric”, and "OKI" remain in the documents, they all have been changed to "OKI Semiconductor Co., Ltd.". It is a change of the company name, the company trademark, and the logo, etc. , and NOT a content change in documents.

October 1, 2008

OKI Semiconductor Co., Ltd.

OKI SEMICONDUCTOR CO., LTD.

550-1 Higashiasakawa-cho, Hachioji-shi, Tokyo 193-8550, Japan

http://www.okisemi.com/en/

OKI®

Oki, Network Solutions for a Global Society

FEUL674000-03

ML674000
User's Manual

32-bit General Purpose ARM-based Microcontroller

Issue Date: February 4, 2003

Preface

This Development Specification contains hardware and software specifications of Oki Electric's ML674000 32-bit microcontroller. The manuals shown below are also available, and should be consulted a necessary.

ARM Architecture Reference Manual

- Description of ARM instruction set architecture

ARM7TDMI Data Sheet

- Description of ARM7TDMI instruction set
- Description of ARM7TDMI operation

ARM Software Development Toolkit (Ver. 2.50) User's Guide

- Guidance on software development and debug environment using ARM development environment SDT (Software Development Toolkit)

ARM Software Development Toolkit (Ver. 2.50) Reference Guide

- Reference for software development and debug environment using ARM development environment SDT (Software Development Toolkit)

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

Notation

This manual uses the following notational conventions.

Type	Notation	Meaning
■ Numerals	0x*nn*	Hexadecimal number
■ Units	word, WORD	1 word = 32 bits
	byte, BYTE	1 byte = 8 bits
	M (mega-)	10^6
	K (Kilo-)	$2^{10} = 1024$
	k (kilo-)	$10^3 = 1000$
	m (milli-)	10^{-3}
	μ (micro)	10^{-6}
	n (nano-)	10^{-9}
	s	second(s)
■ Terms	“H” level	The VIH or VOH voltage level stipulated in the Electrical Characteristics as the voltage high signal level
	“L” level	The VIL or VOL voltage level stipulated in the Electrical Characteristics as the voltage low signal level

Table of Contents

Chapter 1 Introduction

Chapter 2 CPU

Chapter 3 Address Mapping

Chapter 9 Built-In Memory

Chapter 10 External Memory Controller

Chapter 11 Direct Memory Access Controller (DMAC)

Chapter 12 GPIO

Chapter 13 Watchdog Timer (WDT)

Chapter 14 Timers

Chapter 15 PWM Generator

Chapter 16 SIO

Chapter 17 UART with FIFO(16byte)

Chapter 18 Analog-to-Digital Converter

Chapter 19 Electrical Characteristics

Appendixes

Revision History

Chapter 1

Introduction

Chapter 1 Introduction

1.1 Features

This high-performance CMOS 32-bit general-purpose microcontroller combines the 32-bit ARM7TDMI™ core, a RISC CPU developed by Advanced RISC Machines Limited (ARM), with a DMA controller, serial ports, PWM generator, analog-to-digital converter, 16-bit timers, and other peripheral functions on a single LSI.
In addition to 32-bit data processing, this LSI includes internal RAM and onboard peripherals that make it ideal for such embedded control applications as PC peripherals and communication terminals.
Finally, there is a built-in external memory controller for directly connecting ROM, SRAM, SDRAM, other memory types, and peripheral devices.
The following is a list of features.

- CPU
  - — 32-bit RISC CPU (ARM7TDMI)
  - — Little-endian byte order
  - — Operating frequency: 1 MHz to 33 MHz
  - — Instruction set: Free switching between a highly dense 32-bit instruction set and a 16-bit subset offering higher object code efficiency
  - — General-purpose registers: 32-bit × 31
  - — Barrel shifter: Simultaneous ALU and barrel shift operations in the same instruction
  - — Multiplier (32-bit × 8-bit)
  - — JTAG interface for debugging
- Internal Memory
  - — 8 kilobytes of SRAM (2 K × 32-bit)
  - — Connected to processor bus for single-cycle access
- Interrupt Controller
  - — One fast interrupt (FIQ) source (external)
  - — 23 interrupt (IRQ) sources (19 internal and 4 external)
  - — Independent masking for each FIQ and IRQ source
  - — Independent interrupt priority level settings for each IRQ source
  - — Priority control blocking IRQ requests with priority levels at or below those for interrupt requests currently being processed
  - — Choice of level or edge sensing for external IRQ sources EXINT0 to EXINT3 (nIR28 to nIR31).
  - — Falling edge sensing for external FIQ sources.
- Timers
  - — One 16-bit system timer
  - — Six 16-bit auto reload timers
    Independent clock settings for each timer
    Independent choice of one shot or interval timer operation for each timer
  - — Maximum period: 30 ms or more
- Watchdog Timer
  - — One 16-bit timer
  - — Choice of interval or watchdog timer operation
  - — Choice of interrupt or reset upon overflow
  - — Maximum period: 200 ms or more
- GPIO
  - — Two 16-bit ports
  - — Individual settings for pin I/O direction
  - — Individual settings for pin interrupt requests
- PWM
  - — Two outputs with 16-bit resolution
  - — Maximum period; 30 ms or more

- Analog-to-Digital Converter
  - Eight channels of 10-bit resolution, each using consecutive comparison
  - Sample and hold function
  - Choice of scan or select operation
  - Conversion time: 5 µs to 25 µs
- DMA Controller
  - Two channels
  - Choice of fixed or round robin mode for channel priority order
  - Choice of cycle-steal or burst mode for requesting bus access
  - Choice of software or external DMA transfer requests
  - Maximum transfer count: 65,536
  - Data transfer sizes: 8-, 16-, and 32-bit
- External Memory Controller
  - ROM (FLASH) access
    - Supports 16-bit devices
    - Supports Flash memory
  - SRAM access
    - Supports 16-bit devices
    - Supports Asynchronous SRAM
  - DRAM access
    - Supports EDO DRAM and SDRAM
    - Supports 16-bit and 8-bit (only possible for EDO DRAM) devices
    - Supports distributed CAS before RAS (CBR) refresh
  - External I/O banks access
    - Two banks of external I/O space
    - Supports 8- and 16-bit devices
    - Supports external WAIT input signal (XWAIT) for I/O Bank 0
- SIO
  - Full duplex asynchronous operation
  - Built-in baud rate generator
- UART
  - 16550A-compatible asynchronous communications
  - 16-byte FIFO each for transmit and receive operations
  - Full duplex collision operation
  - Built-in baud rate generator
- Clock Signal
  - Connects to crystal (16 MHz to 33 MHz)
  - Also supports direct external clock input
- Power Management
  - STANDBY mode: Stop clock in software
  - HALT mode: Stop clock signals to CPU and other key components in software
  - Clock gear: Dynamically change the HCLK or CCLK frequency to 1/1, 1/2, 1/4, 1/8, or 1/16 times the base frequency in software
  - Functional blocks: Stop clock signals to individual function blocks

1.2 Functional Blocks

1.2.1 Block Diagram

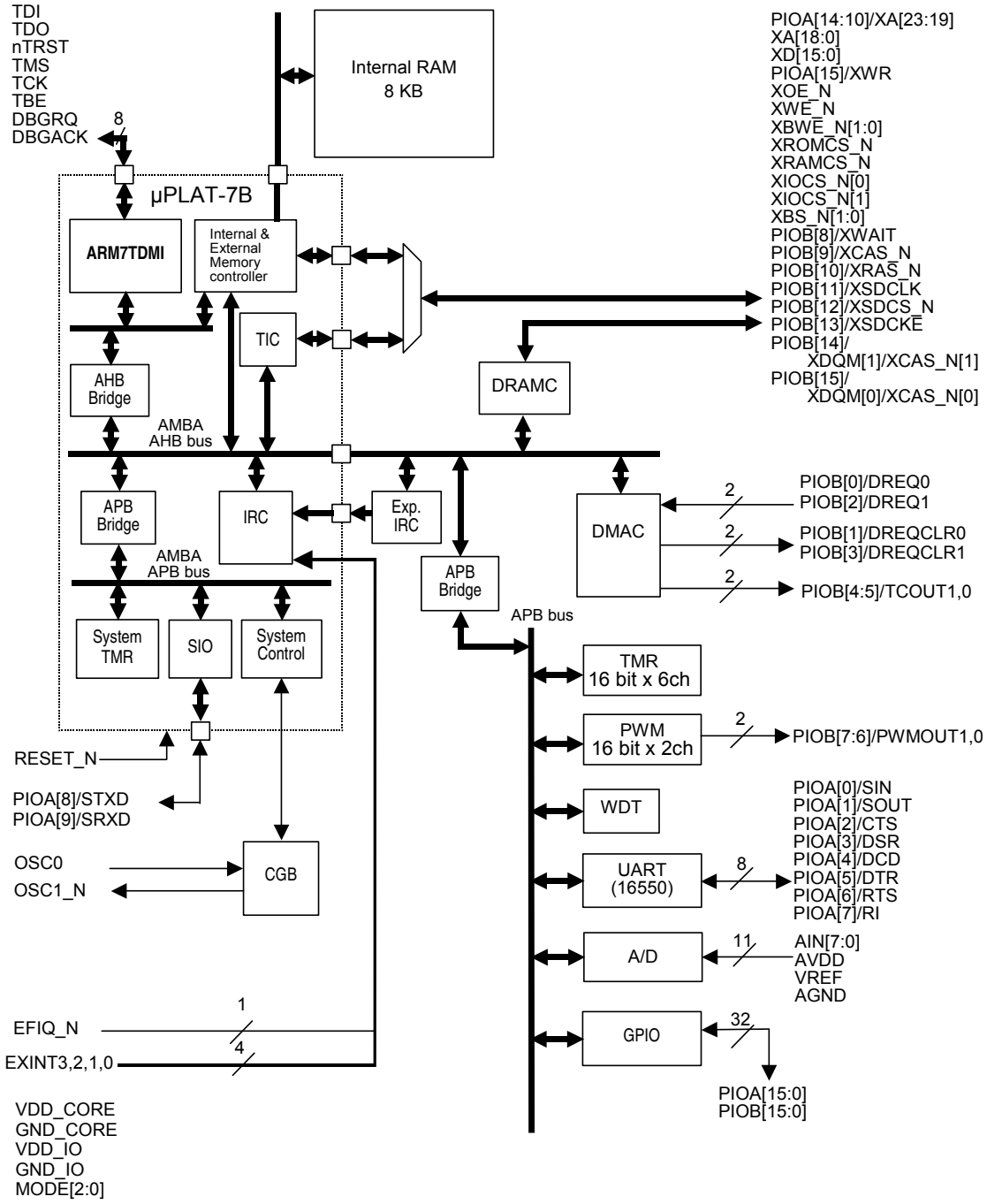


Figure 1.1 Block Diagram

1.3 Pins

1.3.1 Pin Layout

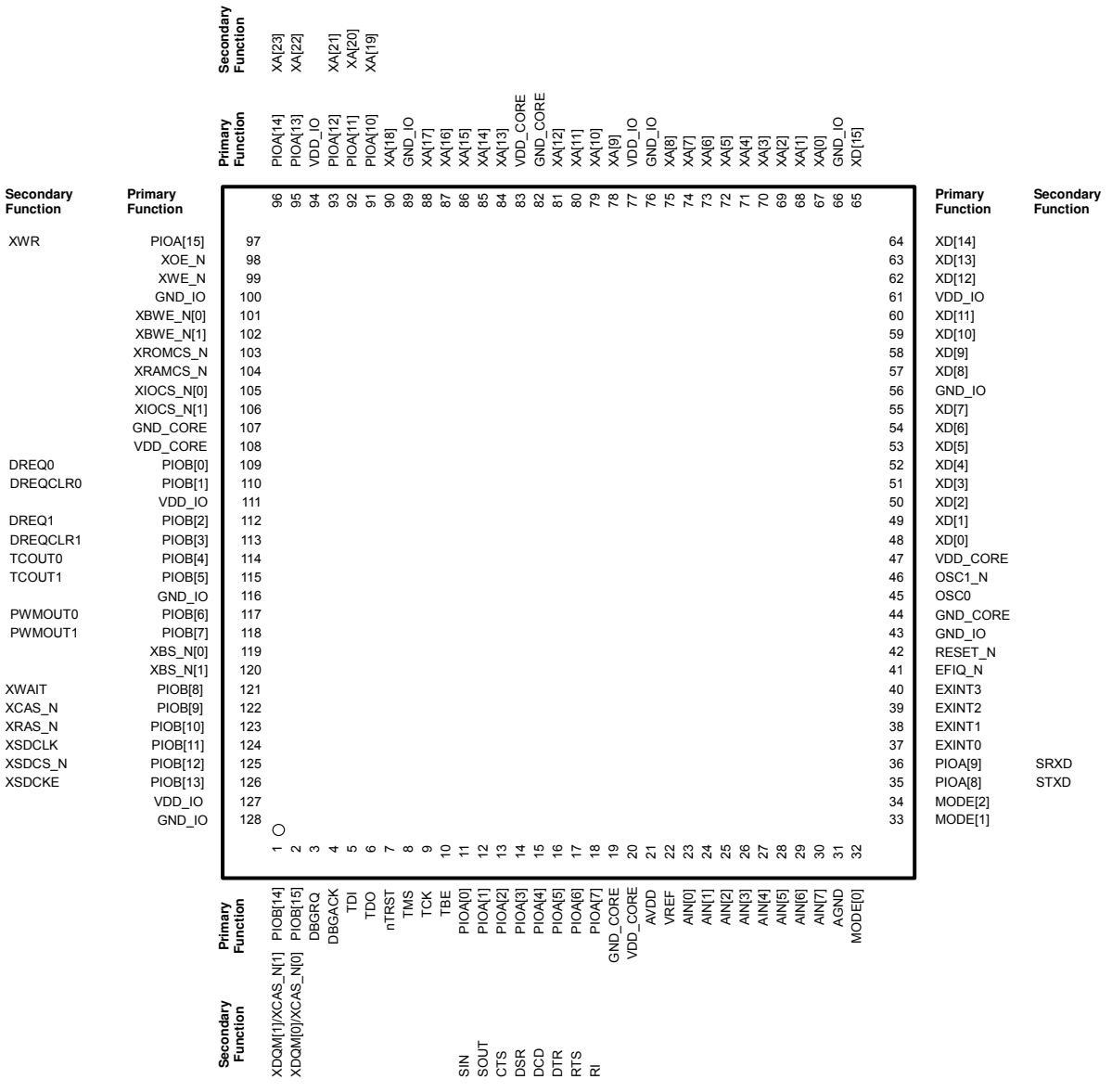


Figure 1.2 Pin Layout(TQFP)

	A	B	C	D	E	F	G	H	J	K	L	M	N	
13	NC	NC	NC	PIOA[12]/ XA[21]	XA[18]	XA[16]	GND_ CORE	XA[8]	XA[5]	XA[2]	GND_IO	XD[15]	NC	13
12	PIOA[15]/ XWR	PIOA[14]/ XA[23]	VDD_IO	GND_IO	XA[15]	XA[14]	XA[10]	GND_IO	XA[7]	XA[4]	XA[1]	NC	XD[14]	12
11	XOE_N	GND_IO	NC	PIOA[11]/ XA[20]	PIOA[10]/ XA[19]	VDD_ CORE	XA[12]	XA[9]	XA[3]	XA[0]	NC	VDD_IO	XD[13]	11
10	XBWE_ N[0]	XROM CS_N	XWE_N	PIOA[13]/ XA[22]	XA[17]	XA[13]	XA[11]	VDD_IO	XA[6]	XD[12]	XD[10]	GND_IO	XD[11]	10
9	XRAM CS_N	XIOCS_ N[1]	XBWE_ N[1]	XIOCS_ N[0]						XD[7]	XD[9]	XD[5]	XD[8]	9
8	GND_ CORE	VDD_ CORE	PIOB[1]/ DREQCLR 0	PIOB[0]/ DREQ0						XD[3]	XD[2]	XD[4]	XD[6]	8
7	PIOB[4]/ TCOUT0	VDD_IO	PIOB[3]/ DREQCLR 1	PIOB[2]/ DREQ1						XD[0]	XD[1]	NC	NC	7
6	XBS_N[0]	PIOB[6]/ PWMOUT0	PIOB[5]/ TCOUT1	GND_IO						NC	VDD_ CORE	OSC1_N	OSC0	6
5	PIOB[9]/ XCAS_N	PIOB[7]/ PWMOUT 1	PIOB[10]/ XRAS_N	XBS_N[1]						GND_IO	EXINT3	GND_ CORE	RESET_N	5
4	PIOB[12]/ XSDCS_N	PIOB[8]/ XWAIT	PIOB[11]/ XSDCLK	VDD_IO	TCK	PIOA[2]/ CTS	PIOA[5]/ DTR	VDD_ CORE	AIN[0]	AIN[7]	EXINT0	EFIQ_N	EXINT2	4
3	NC	PIOB[13]/ XSDCKE	NC	DBGRQ	TDO	PIOA[3]/ DSR	PIOA[6]/ RTS	GND_ CORE	AIN[3]	AIN[4]	PIOA[8]/ STXD	EXINT1	PIOA[9]/ SRXD	3
2	NC	GND_IO	DBGACK	nTRST	TBE	PIOA[1]/ SOUT	PIOA[4]/ DCD	NC	AVDD	AIN[1]	AIN[6]	NC	MODE[2]	2
1	NC	PIOB[14]/ XDQM[1]/ XCAS_N[1]	PIOB[15]/ XDQM[0]/ XCAS_N[0]	TDI	TMS	PIOA[0]/ SIN	PIOA[7]/ RI	VREF	AIN[2]	AIN[5]	AGND	MODE[0]	MODE[1]	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	

Notes: Don't connect NC pins with others.

Figure 1.3 Pin Layout(LFBGA)

1.3.2 Pin List

Pin Number		Primary Function			Secondary Function		
TQFP	LFBGA	Pin Name	I/O	Function	Pin Name	I/O	Function
1	B1	PIOB[14]	I/O	General-purpose port (with interrupt function)	XDQM[1]/ XCAS_N[1]	O	I/O mask/CAS (MSB)
2	C1	PIOB[15]	I/O	General-purpose port (with interrupt function)	XDQM[0]/ XCAS_N[0]	O	I/O mask/CAS (LSB)
3	D3	DBGRQ	I	Debugging input signal	—	—	
4	C2	DBGACK	O	Debugging output signal	—	—	
5	D1	TDI	I	JTAG data input	—	—	
6	E3	TDO	O	JTAG data output	—	—	
7	D2	nTRST	I	JTAG reset	—	—	
8	E1	TMS	I	JTAG mode select	—	—	
9	E4	TCK	I	JTAG clock	—	—	
10	E2	TBE	I	Test input signal	—	—	
11	F1	PIOA[0]	I/O	General-purpose port (with interrupt function)	SIN	I	UART Serial Data In
12	F2	PIOA[1]	I/O	General-purpose port (with interrupt function)	SOUT	O	UART Serial Data Out
13	F4	PIOA[2]	I/O	General-purpose port (with interrupt function)	CTS	I	UART Clear To Send
14	F3	PIOA[3]	I/O	General-purpose port (with interrupt function)	DSR	I	UART Set Ready
15	G2	PIOA[4]	I/O	General-purpose port (with interrupt function)	DCD	I	UART Carrier Detect
16	G4	PIOA[5]	I/O	General-purpose port (with interrupt function)	DTR	O	UART Data Terminal Ready
17	G3	PIOA[6]	I/O	General-purpose port (with interrupt function)	RTS	O	UART Request To Send
18	G1	PIOA[7]	I/O	General-purpose port (with interrupt function)	RI	I	UART Ring Indicator
19	H3	GND_CORE	GND	Core ground	—	—	
20	H4	VDD_CORE	VDD	Core power supply	—	—	
21	J2	AVDD	VDD	Analog-to-digital converter power supply	—	—	
22	H1	VREF	I	Analog-to-digital converter reference voltage	—	—	
23	J4	AIN[0]	I	Analog-to-digital converter analog input	—	—	
24	K2	AIN[1]	I	Analog-to-digital converter analog input	—	—	
25	J1	AIN[2]	I	Analog-to-digital converter analog input	—	—	
26	J3	AIN[3]	I	Analog-to-digital converter analog input	—	—	
27	K3	AIN[4]	I	Analog-to-digital converter analog input	—	—	
28	K1	AIN[5]	I	Analog-to-digital converter analog input	—	—	
29	L2	AIN[6]	I	Analog-to-digital converter analog input	—	—	
30	K4	AIN[7]	I	Analog-to-digital converter analog input	—	—	
31	L1	AGND	GND	GND for A/D converter	—	—	
32	M1	MODE[0]	I	Mode setting	—	—	
33	N1	MODE[1]	I	Mode setting	—	—	
34	N2	MODE[2]	I	Mode setting	—	—	
35	L3	PIOA[8]	I/O	General-purpose port (with interrupt function)	STXD	O	SIO transmit data output
36	N3	PIOA[9]	I/O	General-purpose port (with interrupt function)	SRXD	I	SIO receive data input
37	L4	EXINT0	I	Interrupt input	—	—	
38	M3	EXINT1	I	Interrupt input	—	—	
39	N4	EXINT2	I	Interrupt input	—	—	
40	L5	EXINT3	I	Interrupt input	—	—	
41	M4	EFIQ_N	I	FIQ input	—	—	

Pin Number		Primary Function			Secondary Function		
TQFP	LFBGA	Pin Name	I/O	Function	Pin Name	I/O	Function
42	N5	RESET_N	I	Reset input	—	—	
43	K5	GND_IO	GND	I/O ground	—	—	
44	M5	GND_CORE	GND	Core ground	—	—	
45	N6	OSC0	I	Oscillator input	—	—	
46	M6	OSC1_N	O	Oscillator output	—	—	
47	L6	VDD_CORE	VDD	Core power supply	—	—	
48	K7	XD[0]	I/O	External data bus	—	—	
49	L7	XD[1]	I/O	External data bus	—	—	
50	L8	XD[2]	I/O	External data bus	—	—	
51	K8	XD[3]	I/O	External data bus	—	—	
52	M8	XD[4]	I/O	External data bus	—	—	
53	M9	XD[5]	I/O	External data bus	—	—	
54	N8	XD[6]	I/O	External data bus	—	—	
55	K9	XD[7]	I/O	External data bus	—	—	
56	M10	GND_IO	GND	I/O ground	—	—	
57	N9	XD[8]	I/O	External data bus	—	—	
58	L9	XD[9]	I/O	External data bus	—	—	
59	L10	XD[10]	I/O	External data bus	—	—	
60	N10	XD[11]	I/O	External data bus	—	—	
61	M11	VDD_IO	VDD	I/O power supply	—	—	
62	K10	XD[12]	I/O	External data bus	—	—	
63	N11	XD[13]	I/O	External data bus	—	—	
64	N12	XD[14]	I/O	External data bus	—	—	
65	M13	XD[15]	I/O	External data bus	—	—	
66	L13	GND_IO	GND	I/O ground	—	—	
67	K11	XA[0]	O	External address output	—	—	
68	L12	XA[1]	O	External address output	—	—	
69	K13	XA[2]	O	External address output	—	—	
70	J11	XA[3]	O	External address output	—	—	
71	K12	XA[4]	O	External address output	—	—	
72	J13	XA[5]	O	External address output	—	—	
73	J10	XA[6]	O	External address output	—	—	
74	J12	XA[7]	O	External address output	—	—	
75	H13	XA[8]	O	External address output	—	—	
76	H12	GND_IO	GND	I/O ground	—	—	
77	H10	VDD_IO	VDD	I/O power supply	—	—	
78	H11	XA[9]	O	External address output	—	—	
79	G12	XA[10]	O	External address output	—	—	
80	G10	XA[11]	O	External address output	—	—	
81	G11	XA[12]	O	External address output	—	—	
82	G13	GND_CORE	GND	Core ground	—	—	
83	F11	VDD_CORE	VDD	Core power supply	—	—	
84	F10	XA[13]	O	External address output	—	—	
85	F12	XA[14]	O	External address output	—	—	

Pin Number		Primary Function			Secondary Function		
TQFP	LFBGA	Pin Name	I/O	Function	Pin Name	I/O	Function
86	E12	XA[15]	O	External address output	—	—	
87	F13	XA[16]	O	External address output	—	—	
88	E10	XA[17]	O	External address output	—	—	
89	D12	GND_IO	GND	I/O ground	—	—	
90	E13	XA[18]	O	External address output	—	—	
91	E11	PIOA[10]	I/O	General-purpose port (with interrupt function)	XA[19]	O	External address output
92	D11	PIOA[11]	I/O	General-purpose port (with interrupt function)	XA[20]	O	External address output
93	D13	PIOA[12]	I/O	General-purpose port (with interrupt function)	XA[21]	O	External address output
94	C12	VDD_IO	VDD	I/O power supply	—	—	
95	D10	PIOA[13]	I/O	General-purpose port (with interrupt function)	XA[22]	O	External address output
96	B12	PIOA[14]	I/O	General-purpose port (with interrupt function)	XA[23]	O	External address output
97	A12	PIOA[15]	I/O	General-purpose port (with interrupt function)	XWR	O	External bus data transfer direction
98	A11	XOE_N	O	Output enable (except SDRAM)	—	—	
99	C10	XWE_N	O	Write enable	—	—	
100	B11	GND_IO	GND	I/O ground	—	—	
101	A10	XBWE_N[0]	O	Write enable (LSB)			
102	C9	XBWE_N[1]	O	Write enable (MSB)	—	—	
103	B10	XROMCS_N	O	External ROM chip select	—	—	
104	A9	XRAMCS_N	O	External RAM chip select	—	—	
105	D9	XIOCS_N[0]	O	I/O bank 0 chip select	—	—	
106	B9	XIOCS_N[1]	O	I/O bank 1 chip select	—	—	
107	A8	GND_CORE	GND	Core ground	—	—	
108	B8	VDD_CORE	VDD	Core power supply	—	—	
109	D8	PIOB[0]	I/O	General-purpose port (with interrupt function)	DREQ0	I	DMA request signal (Ch 0)
110	C8	PIOB[1]	I/O	General-purpose port (with interrupt function)	DREQCLR0	O	DREQ clear signal (Ch 0)
111	B7	VDD_IO	VDD	I/O power supply	—	—	
112	D7	PIOB[2]	I/O	General-purpose port (with interrupt function)	DREQ1	I	DMA request signal (Ch 1)
113	C7	PIOB[3]	I/O	General-purpose port (with interrupt function)	DREQCLR1	O	DREQ clear signal (Ch 1)
114	A7	PIOB[4]	I/O	General-purpose port (with interrupt function)	TCOUT0	O	DMAC Terminal Count (CH 0)
115	C6	PIOB[5]	I/O	General-purpose port (with interrupt function)	TCOUT1	O	DMAC Terminal Count (CH 1)
116	D6	GND_IO	GND	I/O ground	—	—	
117	B6	PIOB[6]	I/O	General-purpose port (with interrupt function)	PWMOUT0	O	PWM output (Ch 0)
118	B5	PIOB[7]	I/O	General-purpose port (with interrupt function)	PWMOUT1	O	PWM output (Ch 1)
119	A6	XBS_N[0]	O	External bus byte select (LSB)	—	—	
120	D5	XBS_N[1]	O	External bus byte select (MSB)	—	—	
121	B4	PIOB[8]	I/O	General-purpose port (with interrupt function)	XWAIT	I	WAIT input for IO bank 0
122	A5	PIOB[9]	I/O	General-purpose port (with interrupt function)	XCAS_N	O	Column address strobe (SDRAM)
123	C5	PIOB[10]	I/O	General-purpose port (with interrupt function)	XRAS_N	O	Row address strobe (SDRAM/EDO)
124	C4	PIOB[11]	I/O	General-purpose port (with interrupt function)	XSDCLK	O	SDRAM clock
125	A4	PIOB[12]	I/O	General-purpose port (with interrupt function)	XSDCS_N	O	SDRAM chip select
126	B3	PIOB[13]	I/O	General-purpose port (with interrupt function)	XSDCKE	O	Clock enable (SDRAM)
127	D4	VDD_IO	VDD	I/O power supply	—	—	
128	B2	GND_IO	GND	I/O ground	—	—	

Notes: A1,C3,H2,M2,K6,M7,N7,M12,N13,L11,C13,B13,A13,C11,A3,A2 pins of LFBGA packaged version are NC pins. These pins must be left unconnected.

1.3.3 Pin Descriptions

Pin Name	I/O	Description	Primary/ Secondary	Logic
System				
RESET_N	I	Reset input	—	Negative
OSC0	I	Crystal connection or external clock input. Connect a crystal (16 MHz to 33 MHz), if used, to OSC0 and OSC1_N.	—	
OSC1_N	O	Crystal connection. Leave this pin unconnected if using external clock input.	—	
TBE	I	Test pin. Drive at High level.	—	Negative
Debugging support.				
DBGRQ	I	Debugging pin. Normally connect to ground.	—	Positive
DBGACK	O	Debugging pin. Normally leave open.	—	Positive
TCK	I	Debugging pin. Normally connect to ground.	—	—
TMS	I	Debugging pin. Normally drive at High level.	—	Positive
nTRST	I	Debugging pin. Normally connect to ground.	—	Negative
TDI	I	Debugging pin. Normally drive at High level.	—	Positive
TDO	O	Debugging pin. Normally leave open.	—	Positive
General-purpose I/O ports				
PIOA[15:0]	I/O	General-purpose port. Not available for use as port pins when secondary functions are in use.	Primary	Positive
PIOB[15:0]	I/O	General-purpose port. Not available for use as port pins when secondary functions are in use. Note that enabling DRAM controller with MODE[2:0] inputs permanently configures PIOB[15:9] for their secondary functions, making them unavailable for use as port pins.	Primary	Positive

Pin Name	I/O	Description	Primary/ Secondary	Logic
External Bus				
XA[23:19]	O	Address bus to external RAM, external ROM, external I/O banks, and external DRAM. After a reset, these pins are configured for their primary function (PIOA[14:10]).	Secondary	Positive
XA[18:0]	O	Address bus to external RAM, external ROM, external I/O banks, and external DRAM	—	Positive
XD[15:0]	I/O	Data bus to external RAM, external ROM, external I/O banks, and external DRAM	—	Positive
External bus control signals				
XROMCS_N	O	ROM bank chip select	—	Negative
XRAMCS_N	O	SRAM bank chip select	—	Negative
XIOCS_N[0]	O	I/O bank 0 chip select	—	Negative
XIOCS_N[1]	O	I/O bank 1 chip select	—	Negative
XOE_N	O	Output enable/read enable	—	Negative
XWE_N	O	Write enable	—	Negative
XBS_N[1:0]	O	Byte select: XBS_N[1] for MSB; XBS_N[0] for LSB	—	Negative
XBWE_N[0]	O	LSB write enable	—	Negative
XBWE_N[1]	O	MSB write enable	—	Negative
XWR	O	Data transfer direction for external bus, used when connecting to Motorola I/O devices. This represents the secondary function of pin PIOA[15], produced by setting bit 7 in the port control (GPCTL) register to "1."	Secondary	—
XWAIT	I	External I/O bank 0 WAIT signal. This input permits access to devices slower than register settings.	Secondary	Positive
External bus control signals (DRAM)				
XRAS_N	O	Row address strobe. Used for both EDO DRAM and SDRAM.	Secondary	Negative
XCAS_N	O	Column address strobe signal (SDRAM)	Secondary	Negative
XSDCLK	O	SDRAM clock (same frequency as internal system clock)	Secondary	—
XSDCKE	O	Clock enable (SDRAM)	Secondary	—
XSDCS_N	O	Chip select (SDRAM)	Secondary	Negative
XDQM[1]/ XCAS_N[1]	O	Connected to SDRAM: DQM (MSB) Connected to EDO DRAM: column address strobe signal (MSB)	Secondary	Positive/ Negative
XDQM[0]/ XCAS_N[0]	O	Connected to SDRAM: DQM (LSB) Connected to EDO DRAM: column address strobe signal (LSB)	Secondary	Positive/ Negative

Pin Name	I/O	Description	Primary/ Secondary	Logic
DMA control signals				
DREQ0	I	Ch 0 DMA request signal, used when DMA controller configured for DREQ type	Secondary	Positive
DREQCLR0	O	Ch 0 DREQ signal clear request. The DMA device responds to this output by negating DREQ.	Secondary	Positive
TCOUT0	O	Indicates to Ch 0 DMA device that last transfer has started	Secondary	Positive
DREQ1	I	Ch 1 DMA request signal, used when DMA controller configured for DREQ type	Secondary	Positive
DREQCLR1	O	Ch 1 DREQ signal clear request. The DMA device responds to this output by negating DREQ.	Secondary	Positive
TCOUT1	O	Indicates to Ch 1 DMA device that last transfer has started	Secondary	Positive
SIO				
STXD	O	SIO transmit signal	Secondary	Positive
SRXD	I	SIO receive signal	Secondary	Positive
UART				
SIN	I	Serial data input	Secondary	Positive
SOUT	O	Serial data output	Secondary	Positive
CTS	I	Clear To Send. Indicates that modem or data set is ready to transfer data. Bit 4 in modem status register reflects this input.	Secondary	Negative
DSR	I	Data Set Ready. Indicates that modem or data set is ready to establish a communications link with UART. Bit 5 in modem status register reflects this input.	Secondary	Negative
DCD	I	Data Carrier Detect. Indicates that modem or data set has detected data carrier signal. Bit 7 in modem status register reflects this input.	Secondary	Negative
DTR	O	Data Terminal Ready. Indicates that UART is ready to establish a communications link with modem or data set. Bit 0 in modem control register controls this output.	Secondary	Negative
RTS	O	Request To Send. Indicates that UART is ready to transfer data to modem or data set. Bit 1 in modem control register controls this output.	Secondary	Negative
RI	I	Ring Indicator. Indicates that modem or data set has received telephone ring indicator. Bit 6 in modem status register reflects this input.	Secondary	Negative

Pin Name	I/O	Description	Primary/ Secondary	Logic
PWM signals				
PWMOUT0	O	Ch 0 PWM output	Secondary	Positive
PWMOUT1	O	Ch 1 PWM output	Secondary	Positive
Analog-to-digital converter				
AIN[0]	I	Ch 0 analog input	—	
AIN[1]	I	Ch 1 analog input	—	
AIN[2]	I	Ch 2 analog input	—	
AIN[3]	I	Ch 3 analog input	—	
AIN[4]	I	Ch 4 analog input	—	
AIN[5]	I	Ch 5 analog input	—	
AIN[6]	I	Ch 6 analog input	—	
AIN[7]	I	Ch 7 analog input	—	
VREF	I	Analog-to-digital converter convert reference voltage	—	
AVDD		Analog-to-digital converter power supply	—	
AGND		Analog-to-digital converter ground	—	
Interrupt signals				
EXINT3, 2,1,0	I	External interrupt input signals	—	Positive/ Negative
EFIQ_N	I	External fast interrupt input signal. Interrupt controller connects this to CPU FIQ input.	—	Negative
MODE				
MODE[2:0]	I	Operating mode control signals	—	
Power supplies				
VDD_CORE		Core power supply	—	
VDD_IO		I/O power supply	—	
GND_CORE		Core ground	—	
GND_IO		I/O ground	—	

1.3.4 Pin States

Table 1.1 summarizes the output states for output and I/O pins during a reset; Table 1.2, for STANDBY mode.

Table 1.1 Output States During Reset

Pin Name	Reset with DRAM controller disabled	Reset with DRAM controller enabled
PIOA[15:0]	High impedance	High impedance
PIOB[8:0]	High impedance	High impedance
PIOB[15](XDQM[0]/XCAS_N[0])	High impedance	High level
PIOB[14](XDQM[1]/XCAS_N[1])	High impedance	High level
PIOB[13](XSDCKE)	High impedance	High level
PIOB[12](XSDCS_N)	High impedance	High level
PIOB[11](XSDCLK)	High impedance	Low level
PIOB[10](XRAS_N)	High impedance	High level
PIOB[9](XCAS_N)	High impedance	High level
XD[15:0]	High impedance	High impedance
XA[18:0]	Low level	Low level
XOE_N	High level	High level
XWE_N	High level	High level
XBWE_N[0]	High level	High level
XBWE_N[1]	High level	High level
XROMCS_N	High level	High level
XRAMCS_N	High level	High level
XIOCS_N[0]	High level	High level
XIOCS_N[1]	High level	High level
XBS_N[1:0]	High level	High level
DBGACK	Low level	Low level
TDO	High impedance	High impedance

MODE0 pin input during a reset switches the DRAM controller on and off.

- DRAM controller disabled: MODE[2] = Low, MODE[0] = High
- DRAM controller enabled: MODE[2] = Low, MODE[0] = Low

TDO pin is always high-impedance outside JTAG mode.

Table 1.2 Output States in STANDBY Mode

Pin Name	STANDBY Mode
PIOA[15:0] (primary function)	No change
PIOA[9:0] (secondary function)	No change
PIOA[14:10] (secondary function: XA[23:19])	Low level
PIOA[15] (secondary function: XWR)	High level
PIOB[15:0] (primary function)	No change
PIOB[15] (secondary function: XDQM[0]/XCAS_N[0])	High level
PIOB[14] (secondary function: XDQM[1]/XCAS_N[1])	High level
PIOB[13] (secondary function: XSDCKE)	High level
PIOB[12] (secondary function: XSDCS_N)	High level
PIOB[11] (secondary function: XSDCLK)	Low level
PIOB[10] (secondary function: XRAS_N)	High level
PIOB[9] (secondary function: XCAS_N)	High level
XD[15:0]	High impedance
XA[18:0]	Low level
XOE_N	High level
XWE_N	High level
XBWE_N[0]	High level
XBWE_N[1]	High level
XROMCS_N	High level
XRAMCS_N	High level
XIOCS_N[0]	High level
XIOCS_N[1]	High level
XBS_N[1:0]	High level
DBGACK	Low level
TDO	High impedance

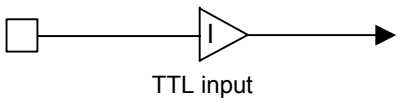
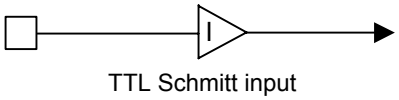
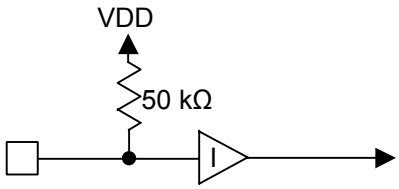
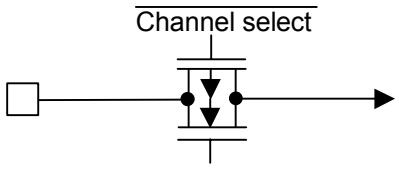
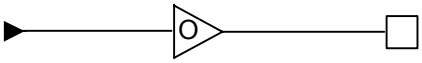
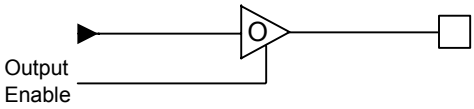
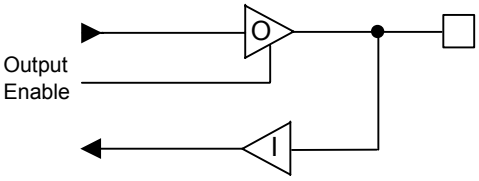
TDO pin is always high-impedance outside JTAG mode.

1.3.5 Pin Structure and Treatment

Pin Structure

Table 1.3 summarizes the structures.

Table 1.3 Pin Structures

Input pins	DBGRQ, TDI, nTRST, TMS, TCK, TBE	 TTL input
	EFIQ_N, MODE[2:0], EXINT3,2,1,0	 TTL Schmitt input
	RESET_N	 TTL Schmitt input with 50 kΩ pull-up resistance
	AIN[7:0]	 Analog-to-digital converter input
Output pins	XA[18:0], XWE_N, XOE_N, XBWE_N[1:0], XROMCS_N, XRAMCS_N, XIOCS_N[1:0], XBS_N[1:0], DBGACK	
	TDO	
I/O pins	PIOA[15:0], PIOB[15:0], XD[15:0]	 TTL Schmitt I/O

Treatment of Unused Pins

Table 1.4 specifies the treatment of unused pins.

Table 1.4 Treatment of Unused Pins

Pin Name	Treatment
DBGRQ	Low level
DBGACK	Open
TDI, TMS,TBE	High level
nTRST, TCK	Low level
TDO	Open
PIOA[15:0], PIOB[15:0]	Configured for input: High or Low level Configured for output: open
AVDD, VREF	VDD_IO
AIN7 to AIN0	VREF or AGND
AGND	GND_IO
EXINT3,2,1,0	High level (Note)
EFIQ_N	High level
XA[18:0]	Open
XWE_N, XBWE_N[1:0], XRAMCS_N, XIOCS_N[1:0], XBS_N[1:0]	Open
MODE[0]	DRAM function enabled: Low level DRAM function disabled: High level
MODE[1]	Low level
MODE[2]	Low level

(Note) The EXINT3,2,1,0 pins default to Low level interrupts, so drive them at High level immediately after a reset to prevent spurious interrupt requests. If the user application system subsequently switches to High level interrupts, drive them at Low level.

Chapter 2

CPU

Chapter 2 CPU

2.1 Overview

This LSI features the ARM7TDMI core, a RISC CPU developed by Advanced RISC Machines Limited (ARM). The ARM7TDMI core allows user application programs to freely switch, as necessary, between the ARM state running a 32-bit instruction set and the THUMB state running an alternate 16-bit set.

2.2 CPU Operation States

The CPU operates in either the ARM or THUMB state. Switching states does not affect the CPU operating mode or register contents.

- ARM state
  The CPU executes ARM instructions aligned at 32-bit word boundaries.
- THUMB state
  The CPU executes THUMB instructions aligned at 16-bit halfword boundaries.

2.2.1 State Transitions

The CPU changes processor states in two ways.

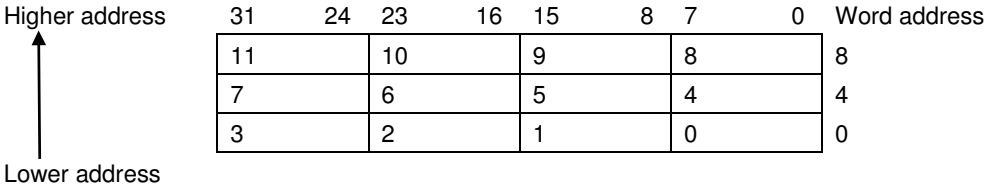
- BX instruction
  The program changes between ARM and THUMB states with a Branch and Exchange (BX) instruction.
- Exception processing
  The CPU always processes exceptions in the ARM state. If it was in the THUMB state when it accepted the exception, it automatically returns to THUMB state when it returns from the exception handler.

2.3 Address Space

This LSI uses 32-bit addresses to access a flat 4-gigabyte address space. This single address space contains both instructions and data.

2.4 Memory Format

Memory consists of bytes numbered sequentially from zero. Bytes 0 to 3 contain the first word of data; bytes 4 to 7, the second. The bytes making up a word are stored in little-endian format.



- The lowest byte is stored at the lowest address.
- The address of a word is the address of lowest byte.

Figure 2.1 Little-Endian Byte Addresses Within Words

Note

The CPU architecture supports both big- and little-endian byte orders, but this LSI uses only the little-endian order.

2.5 Instruction Length

The instruction length is either 32 bits (ARM state) or 16 (THUMB state).

2.6 Data Types

This LSI supports the data types byte (8 bits), halfword (16 bits), and word (32 bits).
Words must be aligned at 4-byte boundaries; halfwords, at 2-byte ones.

2.7 Processor Modes

This LSI supports seven processor modes.

User (usr):	Normal program execution state
FIQ (fiq):	High-speed interrupt handling
IRQ (irq):	General-purpose interrupt handling
Supervisor (svc):	Protected mode for the operating system
Abort (abt):	Prefetch abort (data or instruction) processing
System (sys):	Privileged user mode for the operating system
Undefined (und):	Undefined instruction exception processing

Mode changes are either under software control or in response to an interrupt or exception.
Most application programs run primarily in user mode, switching to non-user, privileged modes only in interrupt and exception handlers or to access protected resources.

2.8 Registers

The CPU has a total of 31 general-purpose 32-bit registers and six status registers. Not all registers are available simultaneously, however. The registers which a program can access depends on the CPU operation state and operating mode.

2.8.1 ARM State Registers

The ARM state provides access to 16 general-purpose registers and the current program status (CPSR) register at all times.
Non-user, privileged modes access their own private register banks and usually a private save program status register (SPSR).
Figure 2.2 indicates these private registers with shaded triangles.
The ARM state provides direct access to the sixteen registers R0 to R15. All but R15 are general-purpose registers.
There is also a seventeenth register, CPSR, which contains control and status information.

- Register 14
  This register functions as the subroutine link register because the branch with link (BL) instruction automatically copies the contents of R15 here. When not used for this purpose, this register is available for use as a general-purpose register.
  The corresponding bank registers R14_svc, R14_irq, R14_fiq, R14_abt, and R14_und hold the return address for an interrupt or exception handler or a BL instruction executed within such a handler.

- Register 15
  This register is reserved for use as the program counter (PC). In the ARM state, the program counter is in bits 31 to 2, and bits 0 and 1 are always zero. In the THUMB state, the program counter is in bits 31 to 1, and bit 0 is always zero.

- Register 16
  This is the current program status register (CPSR). It contains the four condition code flags and bits specifying the current operating mode.

The FIQ mode has seven bank registers (R8_fiq to R14_fiq) mapped to R8 to R14. Having these registers available means that many FIQ handlers do not have to save registers to the stack.
The User, IRQ, Supervisor, Abort, and Undefined modes have two bank registers mapped to R13 and R14. These modes use these registers as a private stack pointer and a link register.

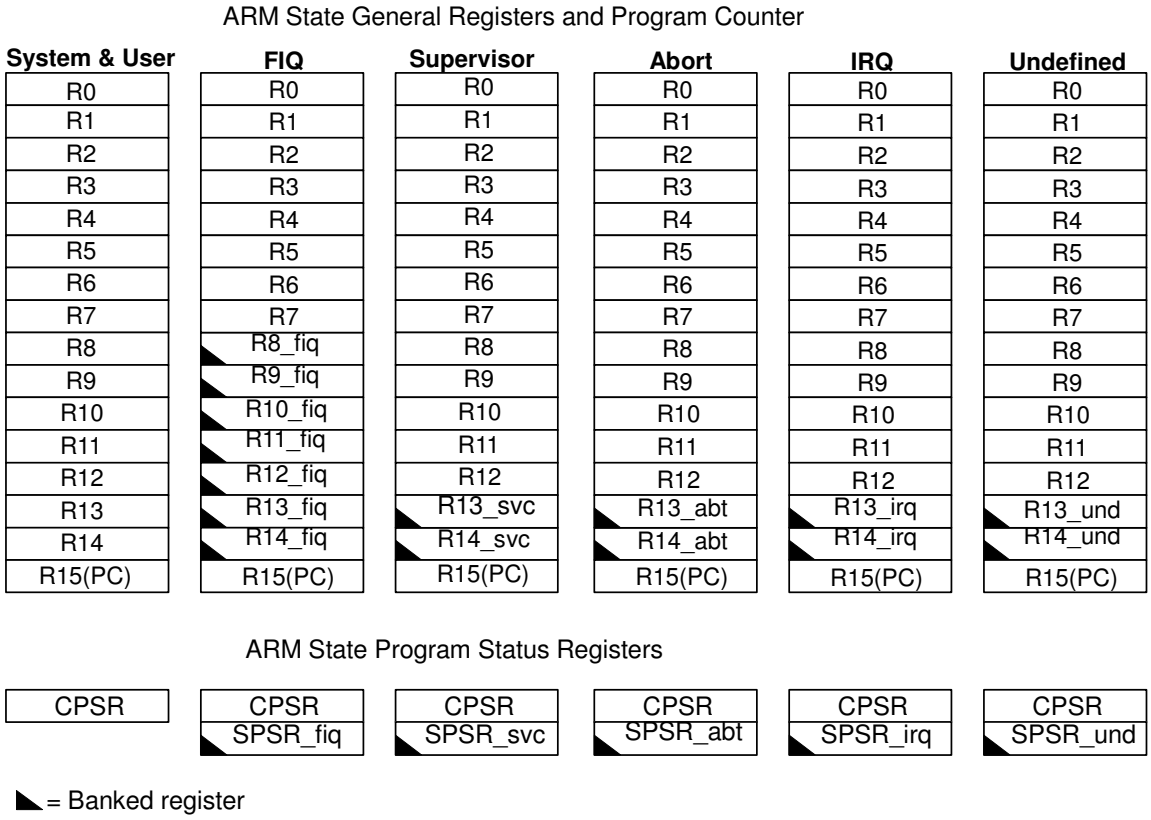


Figure 2.2 ARM State Registers

2.8.2 THUMB State Registers

The THUMB state uses a subset of the ARM state register set. The programmer has direct access to eight general-purpose registers (R0 to R7), the program counter (PC), stack pointer (SP), link register (LR), and CPSR. The privileged user modes access three private registers: stack pointer, link register, and save program status register (SPSR).
Figure 2.3 summarizes these registers.

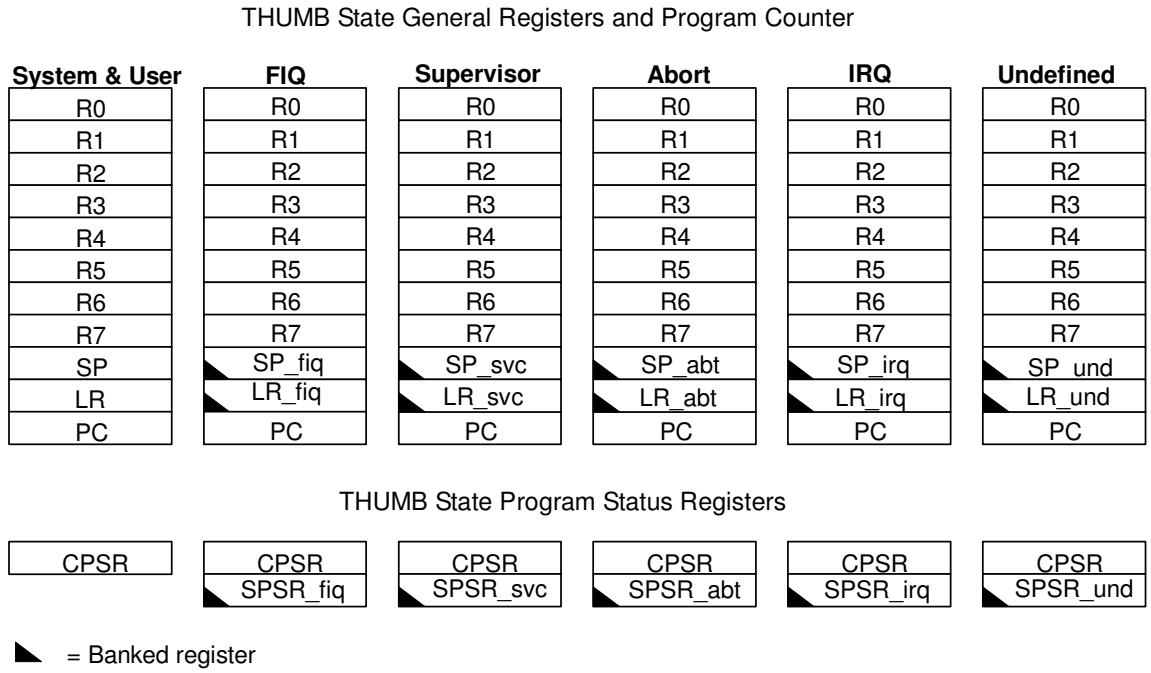


Figure 2.3 THUMB State Registers

2.8.3 Relationships Between ARM and THUMB State Registers

The THUMB state registers have the following relationships with the ARM state ones.

- The THUMB state general-purpose registers R0 to R7 are the same registers as their ARM state counterparts.
- The THUMB state status registers CPSR and SPSR are the same registers as their ARM state counterparts.
- The THUMB state stack pointer (SP) corresponds to the ARM state R13 register. The THUMB state link register (LR) corresponds to the ARM state R14 register.
- The THUMB state program counter (PC) corresponds to the ARM state R15 register.

Figure 2.4 summarizes these relationships.

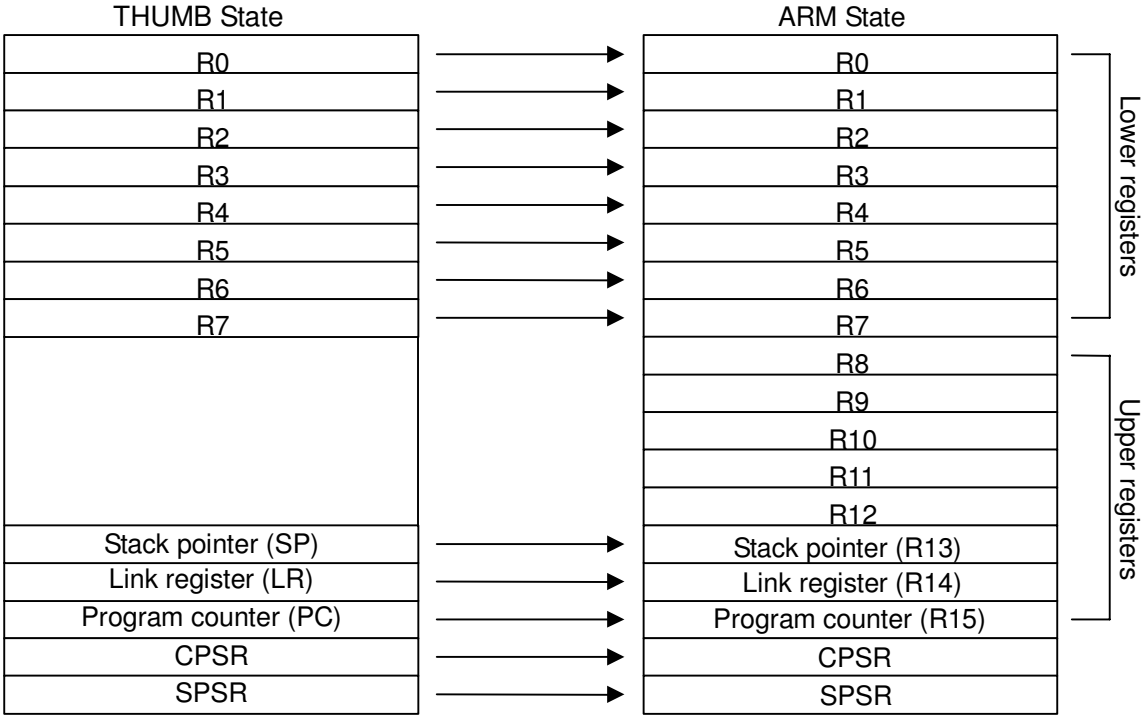


Figure 2.4 Mapping THUMB State Registers to ARM State Ones

2.8.4 Accessing Upper Registers from THUMB State

The THUMB state does not include the upper registers (R8 to R15) as part of the standard register set. Assembly language programs can, however, obtain limited access to them for use as high-speed temporary storage. There are special MOV instructions for transferring data from a lower register to an upper one and in the reverse direction. The CMP and ADD instructions also accept an upper register for comparison with or addition to a lower register.

2.9 Program Status Registers

The CPU has six status registers: the current program status register (CPSR) and five save program status registers (SPSRs).
These registers have the following functions.

- Status flags indicate the results of the last arithmetic instruction executed by the ALU.
- Control bits enable and disable interrupts.
- Mode bits specify the CPU operating mode.

Figure 2.5 gives the bit positions in a program status register.

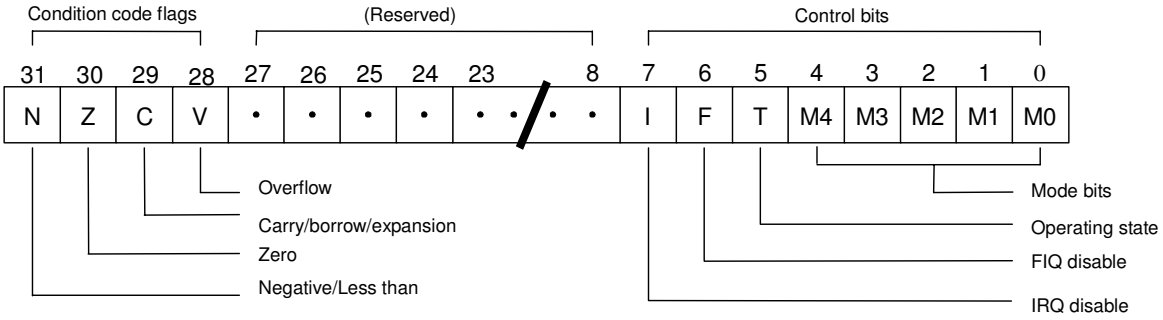


Figure 2.5 Program Status Register Format

2.9.1 Condition Code Flags

The N, Z, C, and V bits are condition code flags indicating the results of arithmetic and bitwise operations. The CPU tests these to determine whether to execute instructions. All ARM state instructions feature conditional execution. The only THUMB state instructions with it are conditional branches.

2.9.2 Control Bits

A program status register's lowest eight bits (I, F, T, and M[4:0]) are collectively called control bits. These change in response to exceptions. Software can manipulate them in privileged user modes.

- T bit
  This bit specifies the operation state: "1" for THUMB; "0" for ARM.
  Note that modifying this bit in software risks sending the CPU into an unpredictable state.

- Interrupt disable bits
  Setting the I or F bit to "1" disables IRQ or FIQ interrupts, respectively.

- Mode bits
  These five bits specify the CPU operating mode, as shown in Table 2.1.
  Note that not all combinations produce valid operating modes. Specifying one not listed risks sending the CPU into an unpredictable state.

Table 2.1 PSR Mode Bits

M[4:0]	Mode	THUMB State Registers	ARM State Registers
10000	User	R7..R0, LR, SP, PC, CPSR	R14..R0, PC, CPSR
10001	FIQ	R7..R0, LR_fiq, SP_fiq, PC, CPSR, SPSR_fiq	R7..R0, R14_fiq..R8_fiq, PC, CPSR, SPSR_fiq
10010	IRQ	R7..R0, LR_irq, SP_irq, PC, CPSR, SPSR_irq	R12..R0, R14_irq..R13_irq, PC, CPSR, SPSR_irq
10011	Supervisor	R7..R0, LR_svc,SP_svc, PC, CPSR, SPSR_svc	R12..R0, R14_svc..R13_svc, PC, CPSR, SPSR_svc
10011	Abort	R7..R0, LR_abt,SP_abt, PC, CPSR, SPSR_abt	R12..R0, R14_abt..R13_abt, PC, CPSR, SPSR_abt
11011	Undefined	R7..R0, LR_und, SP_und, PC, CPSR, SPSR_und	R12..R0, R14_und..R13_und, PC,CPSR
11111	System	R7..R0, LR, SP, PC, CPSR	R14..R0, PC,CPSR

2.9.3 Reserved Bits

Remaining program status register bits are reserved. Be careful not to modify their contents when changing the condition code flags or control bits.

2.10 Instruction Set Features

There are two instruction sets: 32-bit ARM instructions and 16-bit THUMB ones.

2.10.1 ARM Instruction Set

The ARM instruction set contains the following six main instruction types.

- Branch instructions
- Data processing instructions
- Status register transfer instructions
- Load/store instructions
- Coprocessor instructions
- Exception generation instructions

Most data processing instructions and one class of coprocessor instructions sometimes update the four condition code flags (sign, zero, carry, and overflow) in the current program status (CPSR).
Almost all ARM instructions include a 4-bit condition field. One such code ("Always") specifies unconditional execution. The others specify execution only if the corresponding condition is valid at the start of instruction execution. There is no execution if the condition is not met.

There are 14 such codes for testing the following.

- Equality and inequality
- Inequality (<, <=, >, or >=) after signed or unsigned arithmetic
- Individual condition code flags

2.10.2 THUMB Instruction Set

Each 16-bit THUMB instruction has a corresponding 32-bit ARM instruction with the same effect on the processor model. The twin design goals here are to boost performance in user applications using a 16 bits wide (or smaller) memory data bus and to increase code density.

THUMB instructions retain the same 32-bit architecture as ARM instructions--in particular, 32-bit arithmetic and 32-bit addresses for data access instructions and instruction fetch. They are, however, subject to a few access limitations. THUMB instructions share the first eight general-purpose registers, R0 to R7, with ARM instructions. The upper eight, R8 to R15, are generally off limits, but certain THUMB instructions have access to the program counter (ARM register 15), link register (ARM register 14), and stack pointer (ARM register 13).

ARM register 15 holds the program counter in bits 31 to 1. Bit 0 returns "0" for reads. Writes to bit 0 are ignored.

There are no THUMB analogs for the ARM instructions (MSR and MRS) for direct transfers to and from current program status register (CPSR) and saved program status registers (SPSR).

2.11 Addressing Modes

This CPU provides a rich selection of addressing modes for accessing memory.

2.11.1 Load/Store Instructions

Load/store instructions offer three basic addressing modes, each specifying a base register and an offset.

- Offset addressing mode
  The memory address consists of the offset added to or subtracted from the base register contents.
- Preindex addressing mode
  The memory address is the same as for the offset addressing mode. This address then overwrites the base register contents.
- Postindex addressing mode
  The memory address is the base register contents. The hardware then updates the base register by adding or subtracting the offset.

The offset is either an immediate value or the contents of an index register. There are options for shifting the contents of an index register before the addition or subtraction.

2.11.2 Multiple Load/Store Instructions

Multiple load instructions load two or more general-purpose registers from memory. Multiple store instructions store two or more general-purpose registers in memory.
These instructions offer four addressing modes.

- Increment after (IA)
  The base register contents are incremented after the transfer.
- Increment before (IB)
  The base register contents are incremented before the transfer.
- Decrement after (DA)
  The base register contents are decremented after the transfer.
- Decrement before (DB)
  The base register contents are decremented before the transfer.

2.12 Exceptions

An exception indicates the need to temporarily suspend normal program flow--to process an interrupt request from a peripheral. The CPU must therefore save the current CPU state before switching to the exception handler and subsequently restore it when the handler returns.
If there are simultaneous exceptions, a fixed priority system determines the order in which they are accepted. For further details, see Section 2.12.10 "Exception Priority Order."

2.12.1 Switching to Exception Handler

Switching to an exception handler involves the following operations.

1. The CPU saves the return address in the corresponding link register (R14_xxx).
   This return address is the current program counter (PC) contents plus an offset (2, 4, or 8) depending on the exception type and the CPU state at the time of the exception. For further details, see Table 2.2 "Returning from Exception Handlers."
   In the ARM state, the return address is simply the address of the next instruction. In the THUMB state, however, the offset sometimes changes to allow the exception handler to return with the same instruction--MOVS PC, R14_svc for a software interrupt, for example--regardless of the original state.
2. The CPU copies the current program status register (CPSR) contents to the corresponding save program status register (SPSR_xxx).
3. The CPU sets the CPSR mode bits to the exception handler's operating mode.
   To prevent multiple exceptions from interfering with proper system control, the hardware also sets both interrupt disable bits to "1."
4. The CPU loads the exception handler's entry point from the corresponding exception vector into PC to transfer control.

If the CPU is in the THUMB state, this step simultaneously switches it to the ARM state.

2.12.2 Returning from Exception Handler

Returning from an exception handler involves the following operations.

1. The CPU subtracts the specified offset (0, 4, or 8) from the link register contents and loads the result into the program counter (PC). This offset depends on the exception type.
2. The CPU copies the save program status register (SPSR_x) contents back into the current program status register (CPSR).

Note

This copy restores the interrupt disable bits to the states that they had prior to the exception. It also restores the T bit, so it is not necessary for the software to switch back to the THUMB state.

2.12.3 Summary of Exception Switching

Table 2.2 summarizes the instructions for returning from exception handlers and the return addresses saved in the corresponding R14 register.

Table 2.2 Returning from Exception Handlers

	Return Instruction	Return Address		Notes
		ARM	THUMB	
		R14_x	R14_x	
BL	MOV PC , R14	PC+4	PC+2	1
SWI	MOVS PC , R14_svc	PC+4	PC+2	1
UDEF	MOVS PC , R14_und	PC+4	PC+2	1
FIQ	SUBS PC , R14_fiq , #4	PC+4	PC+4	2
IRQ	SUBS PC , R14_irq , #4	PC+4	PC+4	2
PABT	SUBS PC , R14_abt, #4	PC+4	PC+4	1
DABT	SUBS PC , R14_abt, #8	PC+8	PC+8	3
RESET	NA	-	-	4

Notes: 1. The program counter contains the address of the instruction triggering the branch, software interrupt, or undefined instruction exception.
2. The program counter contains the address of the instruction superseded by acceptance of the interrupt request.
3. The program counter contains the address of the load/store instruction triggering the data abort.
4. After a reset, R14_svc contains an indeterminate value.

2.12.4 FIQ

This exception, for supporting a data transfer or channel process, provides an abundance of private registers to reduce the need to save registers and thus the context switching overhead.
The FIQ exception handler must, regardless of the original state (ARM or THUMB), terminate by executing the following instruction.

```
SUBS  PC , R14_fiq , #4
```

Setting the CPSR F bit to "1" disables FIQs. Note, however, that this bit is not accessible in User mode.

2.12.5 IRQ

This exception is for normal interrupt requests. Such interrupt requests have a lower priority than FIQ, so are masked during the FIQ sequence.
Setting the CPSR I bit to "1" disables IRQs. Note, however, that this bit is only accessible in non-user, privileged modes.
The IRQ exception handler must, regardless of the original state (ARM or THUMB), terminate by executing the following instruction.

```
SUBS  PC , R14_irq , #4
```

2.12.6 Aborts

An abort indicates failure to complete the current memory access. The CPU detects abort exceptions during the memory access cycle.
There are two types of aborts.

- Prefetch abort: during instruction prefetch
- Data abort: during data access

A prefetch abort marks the instruction as invalid, but the exception is not processed until the instruction reaches the head of the pipeline. If an intervening branch or other cause prevents execution, there is no abort exception.

The handling of a data abort depends on the instruction type.

- A single data transfer instruction (LDR or STR) proceeds as far as the base register update, if specified. The abort handler must watch out for this.
- A swap instruction (SWP) aborts without doing anything at all.
- A block data transfer instruction (LDM or STM) runs to completion, updating the base register, if specified.

If the instruction overwrites the base register contents with data--that is, includes the base register in the transfer list--the overwrite is aborted. Detection of an abort blocks all such register overwrites. One important consequence is that an aborted LDM instruction always preserves the contents of R15, the last register to be transferred.

The exception handler, after removing the cause of the abort, must, regardless of the state (ARM or THUMB), execute the appropriate instructions to restore the program counter (PC) and current program status (CPSR) register and re-execute the aborted instruction.

- Prefetch abort: SUBS PC, R14_abt, #4
- Data abort: SUBS PC, R14_abt, #8

2.12.7 Software Interrupts

A software interrupt (SWI) instruction switches the CPU to Supervisor mode, normally to request a special, supervisory function. The SWI handler must, regardless of the original state (ARM or THUMB), terminate by executing the following instruction.

```
MOV  PC , R14_svc
```

2.12.8 Undefined Instructions

Attempting to execute an instruction not supported by the CPU triggers an undefined instruction trap. This mechanism allows the programmer to expand the THUMB or ARM instruction set using software emulation. The trap handler, after emulating the failed instruction, must, regardless of the original state (ARM or THUMB), terminate by executing the following instruction.

```
MOVS  PC , R14_und
```

This instruction restores the current program status register (CPSR) and returns to the instruction following the undefined one.

2.12.9 Exception Vectors

Table 2.3 lists exception vector addresses.

Table 2.3 Exception Vectors

Address	Exception	Starting Mode
0x00000000	Reset	Supervisor
0x00000004	Undefined instruction	Undefined
0x00000008	Software interrupt	Supervisor
0x0000000C	Prefetch abort	Abort
0x00000010	Data abort	Abort
0x00000014	Reserved	Reserved
0x00000018	IRQ	IRQ
0x0000001C	FIQ	FIQ

2.12.10 Exception Priority Order

A fixed priority system determines the order in which simultaneous exceptions are accepted.

Highest priority:
1. Reset
2. Data abort
3. FIQ
4. IRQ
5. Prefetch abort

Lowest priority:
6. Undefined instruction, software interrupt

Not all exceptions can occur at once. Undefined instruction and software interrupt share a level because they are mutually exclusive, featuring nonoverlapping opcodes.

If a data abort occurs at the same time as a FIQ, and FIQs are enabled--that is, the CPSR's F flag is "0"--the CPU proceeds to the data abort handler and then immediately to the FIQ vector. Returning normally from the FIQ handler thus causes the data abort handler to resume execution.

Data abort has a higher priority than FIQ to ensure detection of transfer errors. The time for this exception entry should be added to worst-case FIQ latency calculations.

2.13 Resets

The following operations follow a system reset.

1. The CPU copies the current contents of the program counter (PC) and current program status register (CPSR) to R14_svc and SPSR_svc, overwriting the latter with indeterminate values.
2. The CPU loads CPSR with 10011 (Supervisor mode) in M[4:0], "1" in the I and F bits, and "0" in the T bit.
3. The CPU loads PC from address 0x0000 to fetch the next instruction.
4. The CPU resumes execution in the ARM state.

Chapter 3

Address Mapping

Chapter 3 Address Mapping

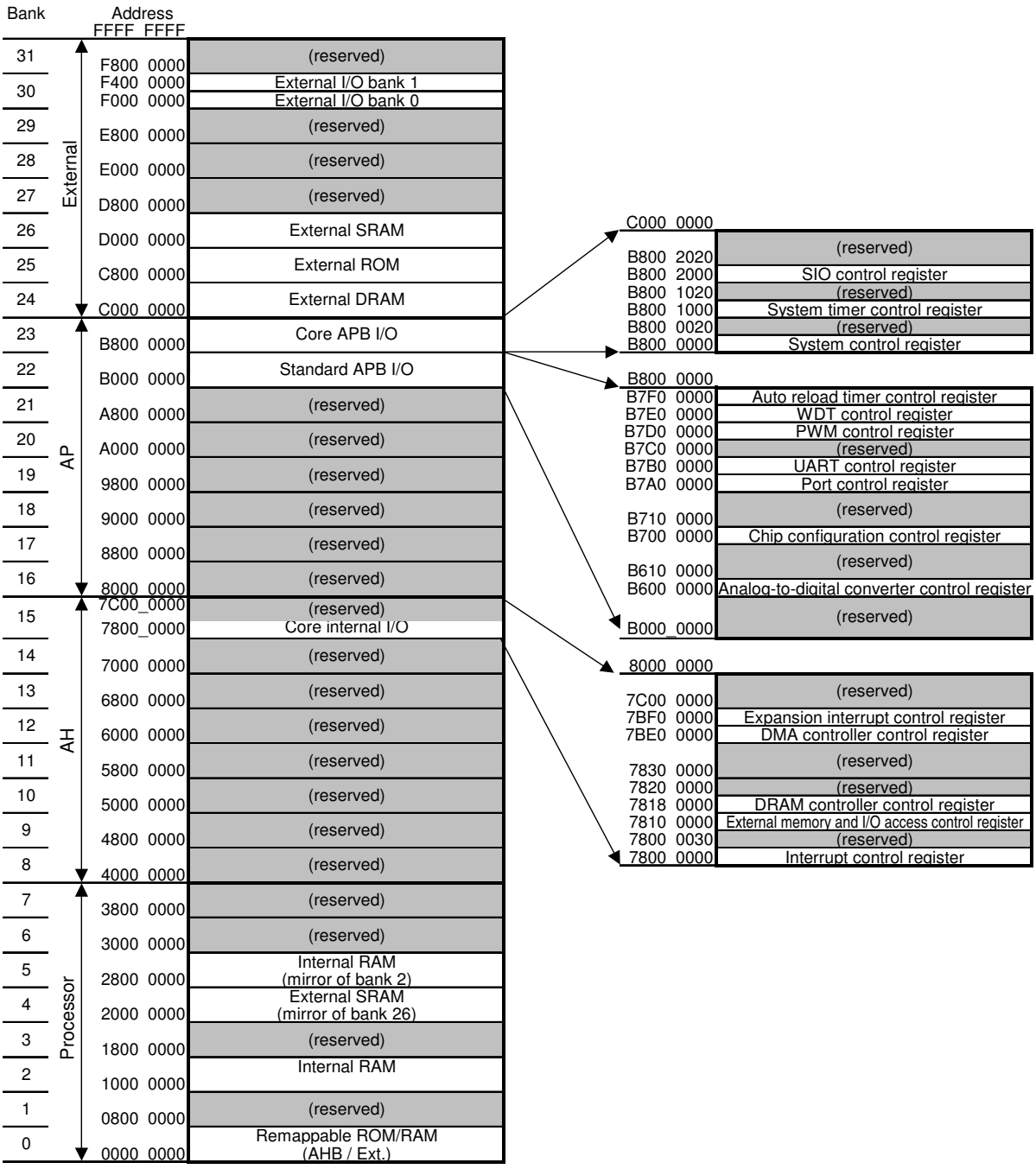
3.1 Overview

This LSI has a 4-gigabyte address space, divided into 32 equal banks.
Bank 0 (0x00000000 to 0x07FFFFFF) is remappable. A reset maps it to an external ROM for use as the boot device.

3.1.1 Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0xB8000010	Remap control register	RMPCON	R/W	32	0x00000000

3.2 Address Map



Note: Banks 4 and 5 respectively mirror banks 26 (external SRAM) and (internal RAM) so that software can treat the two as a single, contiguous memory region.

3.3 Register Descriptions

3.3.1 Remap Control Register (RMPCON)

This register controls mapping of bank 0 to built-in RAM and external memory (ROM/DRAM/SRAM) after booting is complete.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RMPCON	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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 Notes

Note

– *: These bits are for future expansion. They return "0" for reads. Writes to them are ignored.

This register features write protection to prevent errant software from accidentally overwriting register contents. The program must first write 0x3C to it immediately before updating its contents.

Bit Descriptions

- **RMPM[3:0]** (bits 0 to 3):
  These bits remap bank 0 to a different device after a boot.

RMPM[3:0]				Device
3	**2**	**1**	**0**	
0	x	x	x	External ROM
1	0	0	0	External SRAM
1	0	0	1	External DRAM
1	0	1	0	(reserved)
1	0	1	1	(reserved)
1	1	x	x	Built-in RAM

Operation is not guaranteed for a setting labeled "reserved."

Chapter 4

Chip Configuration

Chapter 4 Chip Configuration

4.1 Overview

This LSI provides mode selection (MODE[2:0]) pins controlling the test mode and the use of the following functional blocks.

- DRAM controller (DRAMC)

4.1.1 Pin List

Pin Name	I/O	Description
MODE[0]	I	DRAM controller mode setting
MODE[1]	I	Test mode setting
MODE[2]	I	Test mode setting

4.2 Mode Selection Pins (MODE[2:0])

High level MODE[0] input disables DRAM controller, reducing power consumption by cutting off their clock signals.

MODE[2]	MODE[1]	MODE[0]	DRAM controller block
L	L	L	Enable
L	L	H	Disable
H	X	X	(Reserved)
X	H	X	

Chapter 5

Clock Generator

Chapter 5 Clock Generator

5.1 Overview

This block supplies two clock signals: HCLK to the CPU and CPU interfaces and CCLK to the timer and other counters.
The clock gear provides software control of the frequency divisors for each signal. The choices are 1, 2, 4, 8, and 16.

Note:
For further details on the clock gear, see Chapter 7 "Power Management ".

5.1.1 Components

Figure 5.1 gives a block diagram for the clock generator.

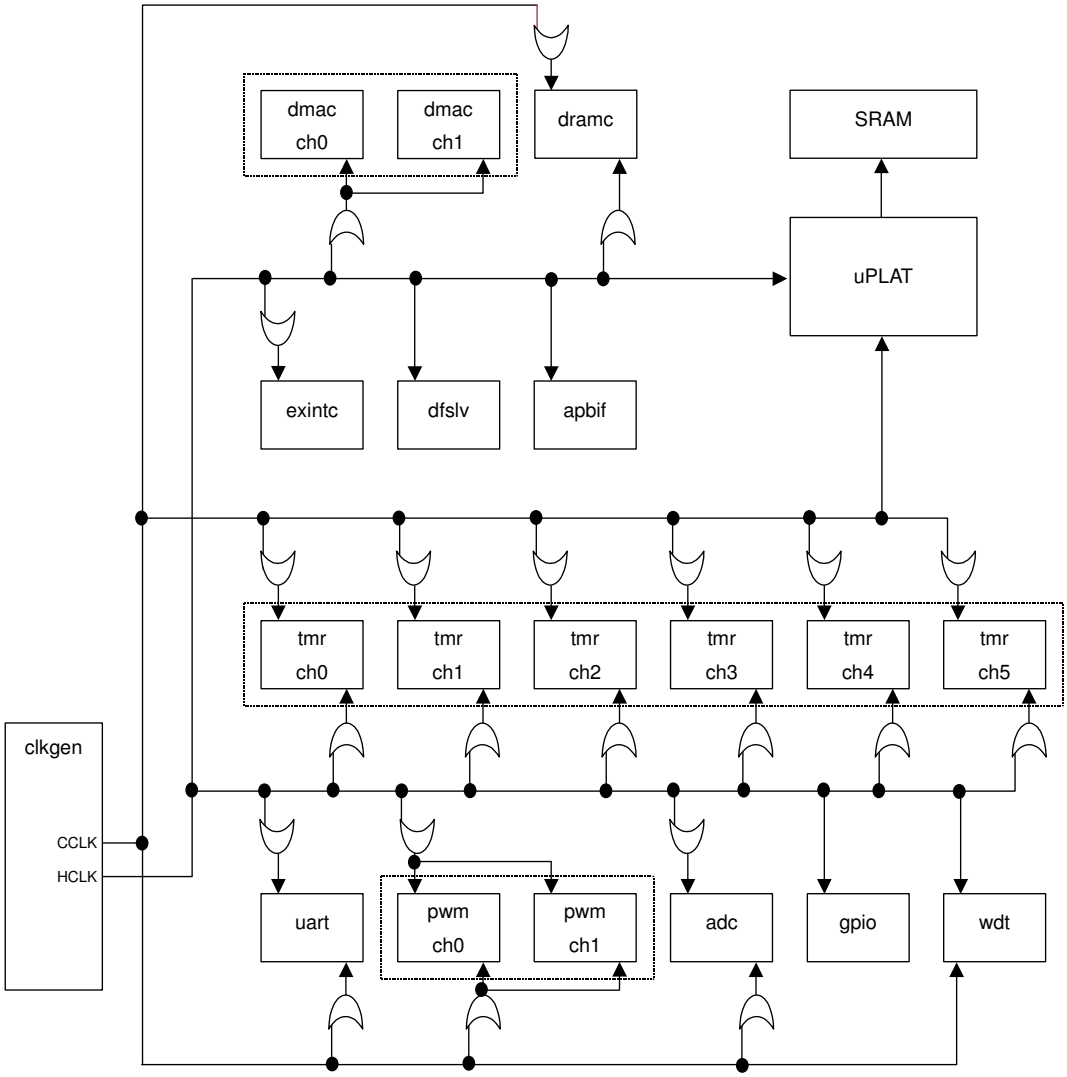


Figure 5.1 Clock Generator Block Diagram

The block clock control (BCKCTL) register controls the clock signals to the functional blocks using the OR gates shown in the clock lines.

5.1.2 Pin List

Pin Name	I/O	Description
OSC0	I	Crystal connection or external clock input
OSC1_N	O	Crystal connection. Leave this pin unconnected if using external clock input.

5.2 Sample Crystal Connections

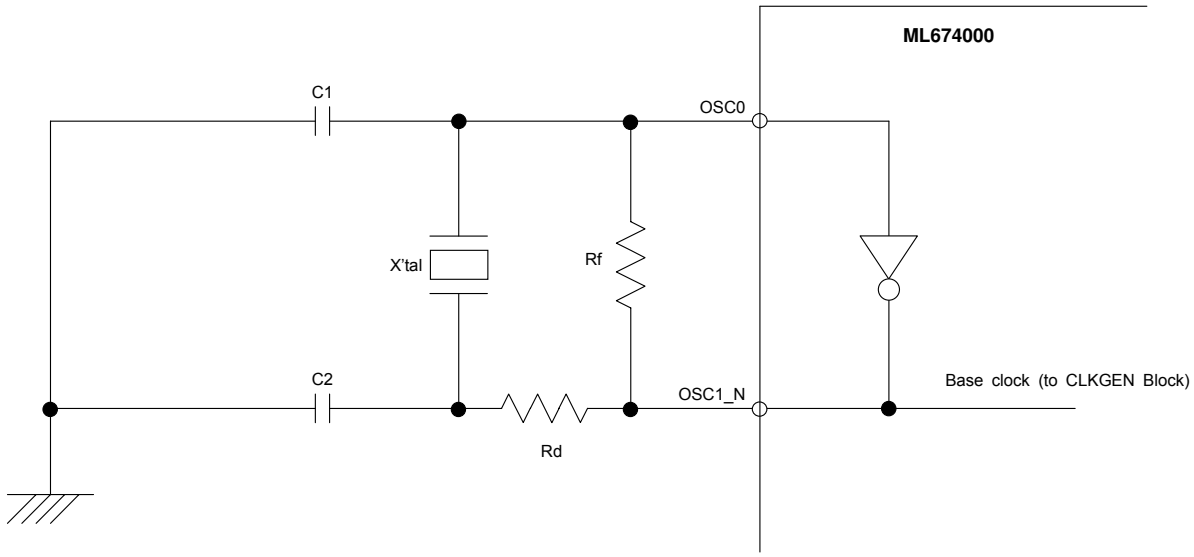


Figure 5.2 Sample Crystal Connections (1)

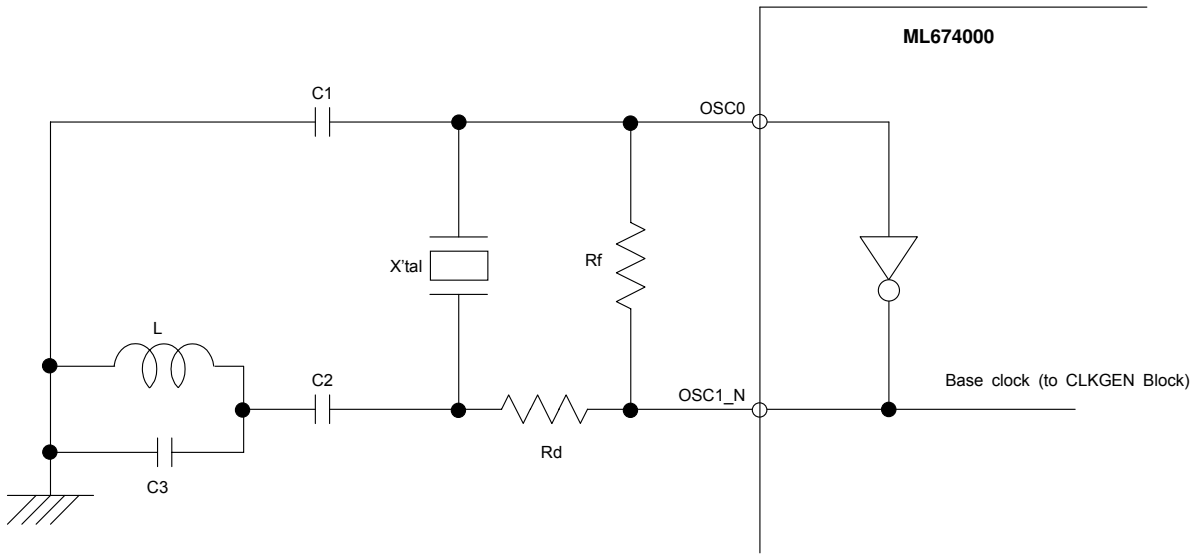


Figure 5.3 Sample Crystal Connections (2)

<Sample data of oscillation circuit>

The following table shows the condition for oscillation circuit, which OKI used for internal evaluation with EPSON's crystals at each frequency.

Frequency [Hz]	Crystal part number	Operating temperature range [°C]	Oscillation circuit	Circuit constants						Oscillation mode
				Rf [Ω]	Rd [Ω]	C1 [pF]	C2 [pF]	C3 [pF]	L [μH]	
16M to 30M	FA-365 FA-368 MA-406	-40 to +85	Figure 5.2	1M	100	40	40	—	—	Fundamental
30M to 33M	CA-301 MA-406 MA-505 MA-506	-40 to +85	Figure 5.3	1M	100	15	15	30	10	3x Over tone

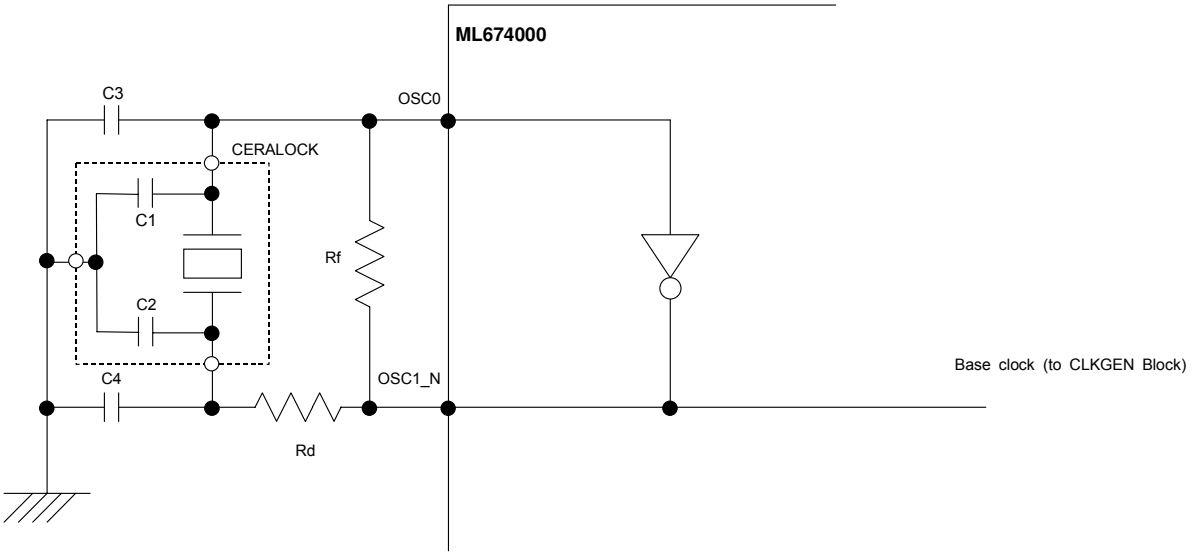


Figure 5.4 Sample Ceralock Connections

<Sample data of oscillation circuit>

The following table shows the condition for oscillation circuit, which OKI used for internal evaluation with Murata Manufacture's CERALOCKs at each frequency.

Frequency [Hz]	Type	Product	Reference value						Reference operation condition	
			C1*1 [pF]	C2*1 [pF]	C3 [pF]	C4 [pF]	Rf [Ω]	Rd [Ω]	Voltage range[V]	Temperature [°C]
12.000M	SMD	CSTCE12M0G55-R0	(33)	(33)	Open	Open	1M	680	VDD_CORE 2.25 to 2.75 VDD_IO 3.0 to 3.6	-40 to +85
16.000M	SMD	CSTCE16M0V53-R0	(15)	(15)	33	33	1M	150	VDD_CORE 2.25 to 2.75 VDD_IO 3.0 to 3.6	-40 to +85
20.000M	New SMD	CSTCG20M0V53-B0	(15)	(15)	33	33	1M	100	VDD_CORE 2.25 to 2.75 VDD_IO 3.0 to 3.6	-40 to +85
24.000M	New SMD	CSTCG24M0V53-R0	(15)	(15)	33	33	1M	100	VDD_CORE 2.30 to 2.75 VDD_IO 3.0 to 3.6	-40 to +85
30.000M	New SMD	CSTCG30M0V53-R0	(15)	(15)	33	33	1M	68	VDD_CORE 2.30 to 2.75 VDD_IO 3.0 to 3.6	-40 to +85
33.000M	New SMD	CSTCG33M0V53-R0	(15)	(15)	33	33	1M	68	VDD_CORE 2.30 to 2.75 VDD_IO 3.0 to 3.6	-40 to +85

(Note) *1: CST** Type build-in C1 and C2.

Chapter 6

Reset Control

Chapter 6 Reset Control

6.1 Overview

The RSTSTATUS bit in the watchdog status (WDSTAT) register indicates to the software the reason for the reset: external reset (RESET_N) pin input or watchdog timer overflow.
The minimum RESET_N pulse width for triggering a system reset/initialization is 20 clock cycles.

6.1.1 Pin List

Pin Name	I/O	Description
RESET_N	I	External reset input

6.2 Reset Types

6.2.1 External Reset Input

Driving the RESET_N pin at Low level for a pulse width at least 20 clock cycles triggers a reset. When first applying the power or waking the LSI from STANDBY mode, however, increase the pulse width to at least 10 ms to allow the crystal oscillator sufficient time to stabilize.
This type of reset leaves 0x0000 in the WDSTAT register.

6.2.2 Watchdog Timer Overflow

Setting the ITM and OFINTMODE bits in the watchdog time base counter control (WDTBCON) register to "0" and "1," respectively, causes watchdog timer overflow to trigger a system reset.
This type of reset leaves 0x0001 in the WDSTAT register.
This type also leaves the following LSI settings unchanged: I/O port direction (input/output), I/O port function (primary/secondary), I/O port output levels, and the oscillation stabilization interval specified in the clock wait (CKWT) register.

6.3 Operational Description

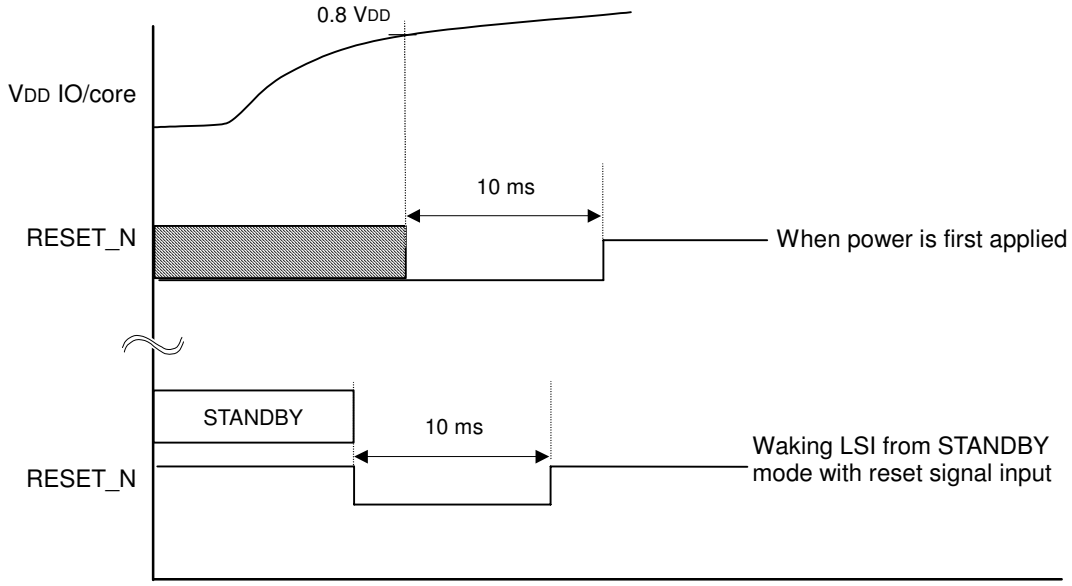


Figure 6.1 Reset Signal Timing

Chapter 7

Power Management

Chapter 7 Power Management

7.1 Overview

This LSI was designed with advanced power saving features to enable flexibility in optimizing power consumption. As such, a great level of configurability has been built into the power management block. This is achieved by varying the frequency of clock signal to different blocks or by stopping the clock signal entirely to certain designated blocks.

To save power, a user application system that does not need the DRAM controller, can save energy by disabling this unit through configuring the Mode Selection (MODE[2:0]) pins as explained in chapter 4 of this manual. This has the effect of disable the clock signal input to this block and thus shutting them down. Note, however, that this modification cannot be undone in software. Furthermore, manipulating the input signals in hardware, after the MCU has been powered up, not only does not work, but also risks unreliable operation.

Note:
For further details on the MODE signals, see Chapter 4 "Chip Configuration."

Two CPU modes allow software to selectively reduce power consumption: STANDBY stops the system clock oscillation entirely; HALT mode stops the clock signals to the following functional blocks: CPU, system bus, bus control circuitry, and memory controller interfaces to built-in RAM and external memory.
The software can save energy by stopping clock signals to individual function blocks.
The software can also save energy by dynamically changing the HCLK or CCLK frequency to 1/1, 1/2, 1/4, 1/8, or 1/16 times the base frequency.

Note:
For further details, see "Clock Gear" below.

7.2 Power Management Functions

The following Table summarizes the power management functions available to software; Figure 7.1 gives a state transition diagram.

Operating Mode		Stopping or Changing Clock	Restarting clock signals
RUN	All functional blocks operative		
RUN	Stop clock signals to individual functional blocks	Software control	Software
RUN	Clock gear	Software control	Software
HALT	Stop clock signals to CPU, system bus, etc.	Software control	Interrupt or reset
STANDBY	Stop clock oscillation	Software control	External interrupt, port input, or reset

Note:
HALT mode stops the clock signals to the following functional blocks: CPU, system bus, bus control circuitry, and memory controller interfaces to built-in RAM and external memory.

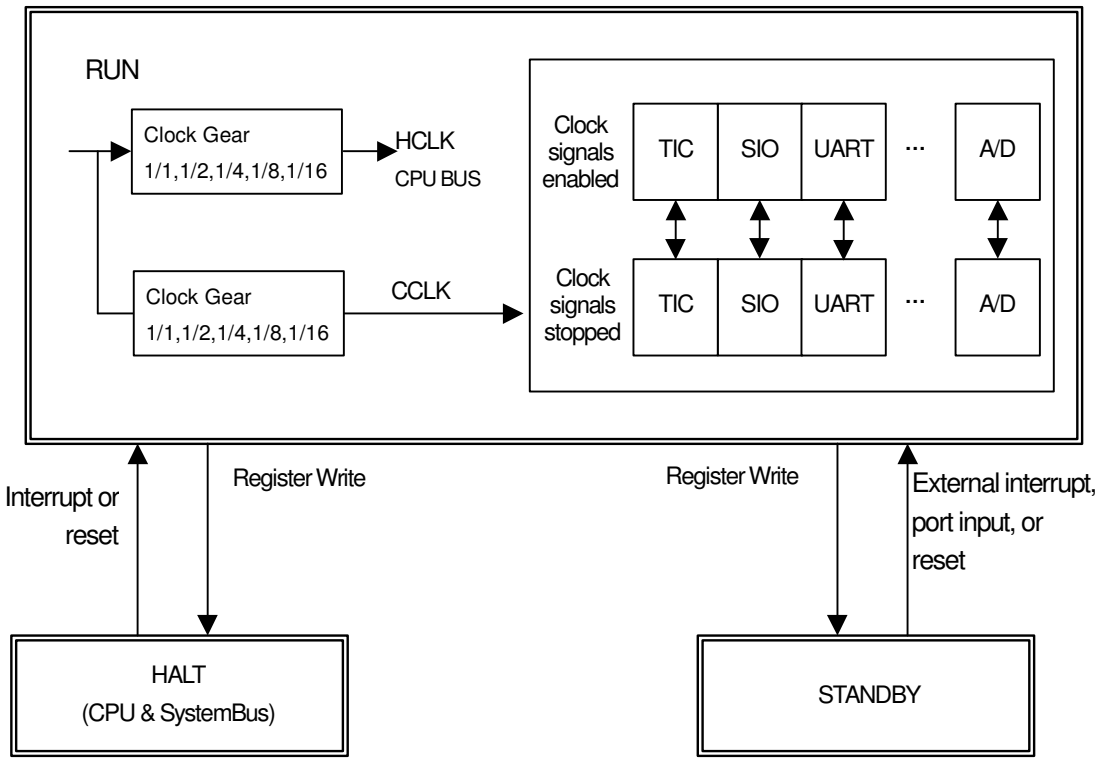


Figure 7.1 State Transition Diagram

Figure 7.2 illustrates the timing for shifting to and from the HALT and STANDBY modes.

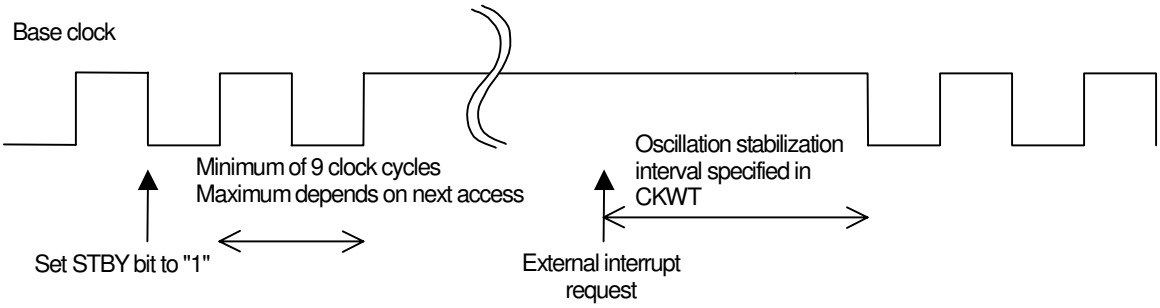


Figure 7.2 Timing Chart

Notes

1. The delay between setting the STBY bit in the CLKSTP register to "1" and stopping the clock depends on the contents of the ARM pipeline. Completing the next instruction's accesses takes a minimum of 9 clock cycles.
2. Waking from HALT mode takes only 3 to 6 clock cycles because there is no need to wait for the oscillation to stabilize before restoring clock signals to the functional blocks. The exact number depends on the wake-up signal.

7.2.1 Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0xB7000004	Block clock control register	BCKCTL	R/W	16	0x0000
0xB8000004	Clock stop register	CLKSTP	R/W	32	0x00000000
0xB8000008	Clock select register	CGBCNT0	R/W	32	0x00000000
0xB800000C	Clock wait register	CKWT	R/W	32	0x000000FF

7.3 Register Descriptions

7.3.1 Block Clock Control Register (BCKCTL)

The block clock control (BCKCTL) register controls the clock signals to the functional blocks.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCKCTL	– *	– *	– *	– *	– *	BCKCTL[10:0]										
After a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7000004
Access: R/W
Access size: 16 bits

Note
– *: These bits are for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **BCKCTL[0]** (bit 0):
  This bit controls the analog-to-digital converter clock signal.

BCKCTL[0]	Description
0	Supply
1	Stop

- **BCKCTL[1]** (bit 1):
  This bit controls the PWM block clock signal.

BCKCTL[1]	Description
0	Supply
1	Stop

- **BCKCTL[2]** (bit 2):
  This bit controls the timer 0 clock signal.

BCKCTL[2]	Description
0	Supply
1	Stop

- **BCKCTL[3]** (bit 3):
  This bit controls the timer 1 clock signal.

BCKCTL[3]	Description
0	Supply
1	Stop

- **BCKCTL[4]** (bit 4):
  This bit controls the timer 2 clock signal.

BCKCTL[4]	Description
0	Supply
1	Stop

- **BCKCTL[5]** (bit 5):
  This bit controls the timer 3 clock signal.

BCKCTL[5]	Description
0	Supply
1	Stop

- **BCKCTL[6]** (bit 6):
  This bit controls the timer 4 clock signal.

BCKCTL[6]	Description
0	Supply
1	Stop

- **BCKCTL[7]** (bit 7):
  This bit controls the timer 5 clock signal.

BCKCTL[7]	Description
0	Supply
1	Stop

- **BCKCTL[8]** (bit 8):
  This bit controls the DRAM controller clock signal.

BCKCTL[8]	Description
0	Supply
1	Stop

- **BCKCTL[9]** (bit 9):
  This bit controls the DMA controller clock signal.

BCKCTL[9]	Description
0	Supply
1	Stop

- **BCKCTL[10]** (bit 10):
  This bit controls the UART clock signal.

BCKCTL[10]	Description
0	Supply
1	Stop

7.3.2 Clock Stop Register (CLKSTP)

This register controls transitions to the HALT and STANDBY modes and the clock signals to two more functional blocks.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLKSTP	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *	– *	– *	– *	– *	– *	– *	– *	STBY	– *	– *	– *	– *	HALT	TIC	SIO
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

Notes

– *: These bits are for future expansion. They return "0" for reads. Writes to them are ignored.

This register features write protection to prevent errant software from accidentally overwriting register contents. The program must first write 0x0000003C to it immediately before updating its contents.

Bit Descriptions

- **SIO** (bit 0):
  This bit controls the SIO block clock signal.

SIO	Description
0	Supply
1	Stop

- **TIC** (bit 1):
  This bit controls the TIC block clock signal.

TIC	Description
0	Supply
1	Stop

- **HALT** (bit 2):
  Setting this bit to "1" shifts to HALT mode, stopping the clock signals to the following functional blocks: CPU, system bus, bus control circuitry, and memory controller interfaces to built-in RAM and external memory.

HALT	Description
0	Normal operation
1	Shift to HALT mode

- **STBY** (bit 7):
  Setting this bit to "1" shifts to STANDBY mode, stopping the system clock oscillation entirely.

STANDBY	Description
0	Normal operation
1	Shift to STANDBY mode

7.3.3 Clock Gear Control Register (CGBCNT0)

This register specifies the divisors for deriving the HCLK and CCLK clock signals from base clock.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CGBCNT0	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *	– *	– *	– *	– *	– *	– *	– *	– *	CCLKSEL			– *	HCLKSEL		
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

Notes

– *: These bits are for future expansion. They return "0" for reads. Writes to them are ignored.

This register features write protection to prevent errant software from accidentally overwriting register contents. The program must first write 0x0000003C to it immediately before updating its contents.

Bit Descriptions

- **HCLKSEL** (bits 0 to 2) and **CCLKSEL** (bits 4 to 6):
  These bit fields respectively specify the HCLK and CCLK frequency divisors.

HCLKSEL			Frequency divisor
2	1	0	
0	0	0	1/1
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	(reserved)
1	1	0	(reserved)
1	1	1	(reserved)

CCLKSEL			Frequency divisor
6	5	4	
0	0	0	1/1
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	(reserved)
1	1	0	(reserved)
1	1	1	(reserved)

7.3.4 Clock Wait Register (CKWT)

This register specifies the interval to wait, after waking the LSI from the STANDBY mode, for the oscillation to stabilize before restoring clock signals to the functional blocks.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKWT	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	CKWT			
	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

Notes

– *: These bits are for future expansion.

This register features write protection to prevent errant software from accidentally overwriting register contents. The program must first write 0x0000003C to it immediately before updating its contents.

Bit[31:8] return "0" for reads. Writes to them are ignored. Bit[7:4] return "1" for reads, and should be written "1" to them

Bit Descriptions

- **CKWT** (bits 0 to 3):
  These bits specify the amount of time to wait, after waking the LSI from the STANDBY mode, for the oscillation to stabilize before restoring clock signals to the functional blocks.
  The following Table gives rough values for 33 MHz operation. The 0000 setting is for External clock input with 1ckock wait of input clock to provide the chip internal clock. The setting 1111 is for using Crystal Oscillator with a wait between 10 ms and 24 ms regardless of the clock frequency.

CKWT				Wait Interval (33 MHz)
3	2	1	0	
0	0	0	0	30 ns(for External clock input to OSCI)
0	0	0	1	(reserved)
1	1	1	0	
1	1	1	1	10 ms to 24 ms(for Crystal Oscillator)

7.3.5 Stopping Clock Signals to Functional Blocks

This LSI provides software control over the clock signals to individual functional blocks.
The following Table lists the functional blocks with clock signal control.

Block	In software	With MODE pins	Notes
SIO	O		
TIC	O		
UART	O		
DMAC	O		Do not stop the clock signal while a DMA transfer is in progress.
DRAMC	O	O	Always switch DRAM to self refresh operation before stopping the clock signal in software.
TIMER	O (each timer individually)		Stopping the clock signal suspends the counter. If the timer is operational, restoring the clock signal causes counting to resume with that counter value.
PWM	O		Stopping the clock signal suspends the counter. If the PWM block is operational, restoring the clock signal causes counting to resume with that counter value.
A/D	O		Stop conversion before stopping the clock signal.

Do not access a functional block while its clock signal is stopped. Accessing the DMA controller or DRAM controller triggers an abort exception. Accessing other functional blocks risks unreliable operation.
If the clock signal to a functional block is stopped, shifting to HALT or STANDBY mode and back again restarts the clock signal only if the wake-up signal is a reset.

7.3.6 Clock Gear

Modifying the CCLK or HCLK divisor in the clock gear control (CGBCNT0) register dynamically changes the corresponding clock frequency to 1/1, 1/2, 1/4, 1/8, or 1/16 times the base frequency.
Changing the CCLK frequency with the clock gear affects the system timer, serial I/O (SIO) block, watchdog timer (WDT), timers, PWM block, UART, and DRAM refresh clock; changing the HCLK frequency does not.
If the clock gear is producing lower clock frequencies, shifting to HALT or STANDBY mode and back again restores the 1/1 settings only if the wake-up signal is a reset.
Always switch DRAM to self refresh operation before reducing the clock signal frequency below the minimum specified for reliable operation.

Notes for each function in using Clock Gear

SIO: When changing the CCLK, the communication through SIO should be stopped.

UART: When changing the CCLK, the communication through UART should be stopped.

WDT: The WDT interval will be changed when CCLK Clock Gear is changed.

DRAM: Refresh cycle setting should be set based on final CCLK frequency, before changing clock.
If refresh cycle is changed dynamically, must be set CCLK = 1/1 of clock gear temporally.

PWM: The PWM frequency is changed when the CCLK clock frequency is changed.

AD converter: The AD conversion should be stopped when changing CCLK clock frequency.

7.3.7 HALT Mode

HALT mode stops the clock signals to the following functional blocks: CPU, system bus, bus control circuitry, and memory controller interfaces to built-in RAM and external memory.
External interrupt requests wake the LSI from HALT mode and restart the clock signals. Recovery requires a maximum of 10 clock cycles.
Functional blocks whose clock signals do not stop remain operational, so most can also provide such interrupt requests. The DMA controller is an exception, however, because it cannot transfer data while the system bus is disabled. It is not possible, for example, to wait in HALT mode for a DMA transfer to end and wake the LSI with an interrupt request.

The following Table summarizes the events producing shifts to and from this mode.

Shift to HALT mode	Wake from HALT mode
Write "1" to HALT bit	Unmasked interrupt request • SIO • System Timer • UART • Timer • PWM • A/D • GPIO • External interrupt request Reset

7.3.8 STANDBY Mode

STANDBY mode stops the system clock oscillation entirely, stopping all internal clock signals and greatly reducing power consumption.

When the LSI wakes from STANDBY mode, it does not restart these internal clock signals until the oscillation stabilization interval specified in the clock wait (CKWT) register has elapsed. The maximum such interval is approximately 24 ms.

The following Table summarizes the events producing shifts to and from this mode.

Shift to STANDBY mode	Wake from STANDBY mode
Write "1" to STBY bit	Unmasked interrupt request • GPIO • External interrupt request (configured for level detection) Reset*

Notes

* In order to use the External Interrupt Request (EXINT) for exiting STANDBY mode, the EXINT should be configured for LEVEL Detection. If using GPIOs, the GPIO interrupt signal should be held active, until exiting STANDBY.

7.4 Using Power Management with DRAM

Writing "1" to BCKCTL[8] to stop the clock signal to the DRAM controller does not automatically activate DRAM self refresh operation. The software must explicitly request activation and deactivation by writing 110 and 111, respectively, to the DRCMD bits in the DRAM command (DCMD) register.
For SDRAM, self refresh operation stops XSDCLK output.
Accessing the DRAM address space during self refresh operation triggers an abort exception.

7.4.1 Activating Self Refresh Operation

The following is the procedure for activating self refresh operation before stopping the clock signal to the DRAM controller or switching to STANDBY mode.

1. Write 110 in the DCMD register DRCMD bits to activate self refresh operation.
   The DRAM controller then suspends distributed CAS before RAS (CBR) refresh operation and stops the SDRAM clock signal (XSDCLK) until the next deactivate request.

2. Stop the clock signal to the DRAM controller in software or shift to STANDBY mode.

7.4.2 Deactivating Self Refresh Operation

When the software or return from STANDBY mode restarts the clock signal to the DRAM controller, write 111 in the DCMD register DRCMD bits to deactivate self refresh operation.
The DRAM controller then restarts both the SDRAM clock signal XSDCLK and distributed CAS before RAS (CBR) refresh operation.

Chapter 8

Interrupt Controller

Chapter 8 Interrupt Controller

8.1 Overview

This LSI has an 8-level priority, individually maskable, highly configurable interrupt controller. The interrupt controller features are designed to provide flexibility to software programmer for designing an efficient interrupt handling routine.

The interrupt controller is connected to the nFIQ (fast interrupt request) and nIRQ (interrupt request) inputs of the ARM7TDMI processor. The processor nFIQ is only asserted in response to an external FIQ request through the pin EFIQ_N. The processor nIRQ is asserted in response to interrupt requests from internal peripherals or external pins EXINT0-EXINT3.

In total, the interrupt controller supports 23 interrupt (IRQ) sources and 1 fast interrupt (FIQ) source. Nineteen of the interrupt sources are coming from internal peripherals such as DMA, UART, SIO, etc. The other four interrupt sources are from external inputs; EXINT0, EXINT1, EXINT2, and EXINT3. In addition it supports one external fast interrupt (FIQ) source through the pin EFIQ_N.

The 8-level priority control feature allows the customer to define the interrupt priority level for the different interrupt sources.

External interrupt sources can be configured to be edge triggered or level triggered. Also, for edge triggered devices, the customer can configure the interrupt controller to trigger either on the negative or positive edge of the input.

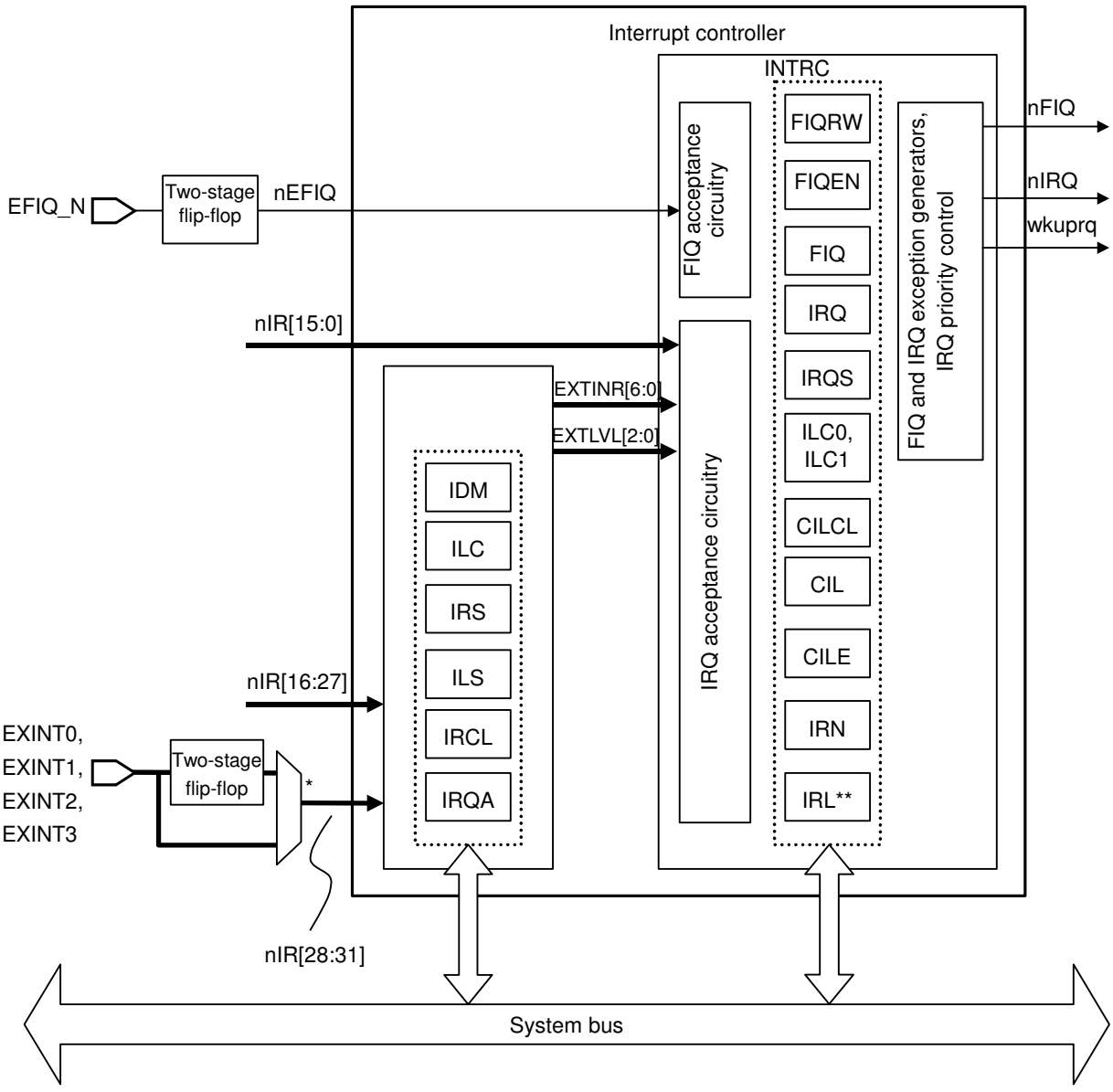
The following is an overview of main features of the interrupt controller:

Features

- One fast interrupt (FIQ) source (external)
- 23 interrupt (IRQ) sources (external and internal)
- Independent masking for each FIQ and IRQ source
- Independent interrupt priority level settings for each IRQ source
- Priority control blocking IRQ requests with priority levels at or below those for interrupt requests currently being processed
- Choice of level or edge sensing for external IRQ sources EXINT0 to EXINT3 (nIR28 to nIR31).
- Conversion of external interrupt requests to wake-up requests for restarting the clock and thus waking the LSI from STANDBY mode

8.1.1 Components

Figure 8.1 shows the interrupt controller components.



* STANDBY mode bypasses this flip-flop.
** IRL is an internal register.

Figure 8.1 Interrupt Controller Components

8.1.2 Pin List

Pin Name	I/O	Description
EXINT0	I	Interrupt input 0
EXINT1	I	Interrupt input 1
EXINT2	I	Interrupt input 2
EXINT3	I	Interrupt input 3
EFIQ_N	I	FIQ input (negative logic)

8.1.3 Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0x78000000	IRQ register	IRQ	R	32	0x00000000
0x78000004	Software interrupt register	IRQS	R/W	32	0x00000000
0x78000008	FIQ register	FIQ	R	32	0x00000000
0x7800000C	FIQRAW register	FIQRAW	R	32	Reflects EFIQ_N interrupt input
0x78000010	FIQ enable register	FIQEN	R/W	32	0x00000000
0x78000014	IRQ number register	IRN	R	32	0x00000000
0x78000018	Current interrupt level register	CIL	R/W	32	0x00000000
0x7800001C	(Reserved)				
0x78000020	Interrupt level control register 0	ILC0	R/W	32	0x00000000
0x78000024	Interrupt level control register 1	ILC1	R/W	32	0x00000000
0x78000028	Current interrupt level clear register	CILCL	W	32	—
0x7800002C	Current interrupt level encode register	CILE	R	32	0x00000000
0x7BF00000	(Reserved)				
0x7BF00004	IRQ clear register	IRCL	W	32	—
0x7BF00010	IRQA register	IRQA	R/W	32	0x00000000
0x7BF00014	IRQ detection mode setting register	IDM	R/W	32	0x00000000
0x7BF00018	Interrupt level control register	ILC	R/W	32	0x00000000

8.2 Interrupt Sources

8.2.1 External Fast Interrupt (EFIQ_N)

The external fast interrupt request (EFIQ_N) pin is a high-priority interrupt request normally assigned to a single, time critical source. If FIQEN, bit 0 in the FIQEN register, does not mask the interrupt request, the interrupt controller asserts the fast interrupt request (nFIQ) signal to the CPU in response to detecting an incoming interrupt at the EFIQ_N input pin of this LSI.

8.2.2 External Interrupts (EXINT[n])

There are four external interrupt request inputs available: EXINT[0] to EXINT[3] (nIR28 to nIR31). Each has its own register settings for selecting edge or level detection and specifying polarity.

8.2.3 Internal Interrupts (IRQn)

There are internal interrupts from the following functional blocks. (See Table in Section 8.2.4 "Interrupt Source List.")

- System timer
- Watchdog timer
- Watchdog timer interval timer operation
- General-purpose I/O ports (GPIOA and GPIOB)
- Software interrupt requests
- UART
- Serial I/O (SIO)
- Analog-to-digital converter
- PWM outputs 0 and 1
- Timers 0 to 5
- DMA channels 0 and 1

8.2.4 Interrupt Source List

Interrupt Source Number	Interrupt Source	Notes
nFIQ	nFIQ	External input (EFIQ_N)
nIR0	System timer	
nIR1	Watchdog timer	
nIR2	Watchdog timer interval timer operation	
nIR3	(unused)	
nIR4	GPIOA	
nIR5	GPIOB	
nIR6	(unused)	
nIR7	(unused)	
nIR8	Software interrupt requests	
nIR9	UART	
nIR10	SIO	
nIR11	AD	
nIR12	PWM output 0	
nIR13	PWM output 1	
nIR14	(unused)	
nIR15	(unused)	
nIR16	Timer 0	
nIR17	Timer 1	
nIR18	Timer 2	
nIR19	Timer 3	
nIR20	Timer 4	
nIR21	Timer 5	
nIR22	(unused)	
nIR23	(unused)	
nIR24	DMA channel 0	
nIR25	DMA channel 1	
nIR26	(unused)	
nIR27	(unused)	
nIR28	External interrupt 0	External input, choice of edge or level sensing
nIR29	External interrupt 1	External input, choice of edge or level sensing
nIR30	External interrupt 2	External input, choice of edge or level sensing
nIR31	External interrupt 3	External input, choice of edge or level sensing

8.3 Interrupt Levels

Each IRQ source has its own interrupt level setting. The higher the numerical value, the higher the priority. A setting of zero, on the other hand, masks interrupts from that source.

Setting and Priority

Interrupt level setting	Priority
7	High
↕	↕
1	Low
0	Interrupts masked

8.4 Register Descriptions

8.4.1 IRQ Register (IRQ)

A "1" in bit n indicates a pending (unmasked) interrupt request from the corresponding IRQ source (nIRn). The CPU has only read access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures.

Bit Descriptions

- **IRQ[15:0]** (bits 0 to 15):
  A "1" in bit n indicates a pending (unmasked) interrupt request from the corresponding IRQ source (nIRn).

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

The following Table summarizes how, for interrupt source number n, the bit IRQ[n] contents depend on the nIRn source and the interrupt level setting from the interrupt level control registers (ILC0 and ILC1).

nIRn Source	Interrupt Level Setting (from ILCx)	Bit IRQ[n]
X	0	0
1	Nonzero (1 to 7)	0
0	Nonzero (1 to 7)	1

8.4.2 Software Interrupt Register (IRQS)

Writing "1" to bit 1 in this register generates a software interrupt request, which the interrupt controller maps to nIR8. This interrupt request has the interrupt source number 8 and the interrupt level setting from interrupt level control register 1.
Writing "0" to this bit cancels the interrupt request.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQS	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures. Always write "0" to these bits.

Bit Descriptions

- **IRQS** (bit 1):
  This bit controls the software interrupt request signal.

IRQS	Description
0	Negate software interrupt request
1	Assert software interrupt request

8.4.3 FIQ Register (FIQ)

A "1" in bit 0 indicates a pending fast interrupt request (FIQ) from the external fast interrupt request (EFIQ_N) pin.
The CPU has only read access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIQ	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures.

Bit Descriptions

- **FIQ** (bit 0):
  This bit indicates a pending fast interrupt request (FIQ) from the external fast interrupt request (EFIQ_N) pin.

FIQ	Description
0	No FIQ request pending
1	FIQ request pending

The following Table summarizes how the bit FIQ[0] contents depend on the FIQRAW[0] bit in the FIQRAW register and FIQEN, bit 0 in the FIQEN register.

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

8.4.4 FIQRAW Register (FIQRAW)

A "1" in bit 0 indicates a raw fast interrupt request (FIQ) from the external fast interrupt request (EFIQ_N) pin. Here raw means not masked by FIQEN, bit 0 in the FIQEN register.
The CPU has only read access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIQRAW	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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	(Note 2)

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

Notes

1. Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures.
2. The initial value reflects the EFIQ_N interrupt input.

Bit Descriptions

- **FIQRAW** (bit 0):
  This bit reflects the EFIQ_N interrupt input.

FIQRAW	Description
0	No FIQ request pending
1	FIQ request pending

8.4.5 FIQ Enable Register (FIQEN)

Bit 0 in this register controls masking of the external fast interrupt request (EFIQ_N) pin input.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIQEN	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures. Always write "0" to these bits.

Bit Descriptions

- **FIQEN** (bit 0):
  This bit controls masking of the external fast interrupt request (EFIQ_N) pin input.

FIQEN	Description
0	Disable FIQ requests
1	Enable FIQ requests

8.4.6 IRQ Number Register (IRN)

This register gives the interrupt source number for the IRQ request with the highest priority.
Reading this register clears it to zero and sets the current interrupt level (CIL) register bit corresponding to the interrupt level to "1", masking pending interrupt requests at or below that level.
When an interrupt request with a higher interrupt level subsequently arrives, the interrupt controller writes its interrupt source number to this register and asserts the interrupt request (nIRQ) signal to the CPU.
The CPU has only read access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRN	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures.

Bit Descriptions

- **IRN[6:0]** (bits 0 to 6):
  These bits give the interrupt source number for the IRQ request with the highest priority. For the mapping of interrupt sources to source numbers, see Section 8.2.4 "Interrupt Source List."

8.4.7 Current Interrupt Level Register (CIL)

This register indicates the interrupt levels for interrupts currently being processed--that is, whose interrupt source numbers have been read from the IRN register.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIL	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures. Always write "0" to these bits.

Bit Descriptions

- **CIL[7:1]** (bits 1 to 7):
  A "1" in bit n indicates that a level n interrupt request is currently being processed.

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

The interrupt controller masks pending interrupt requests at or below the level corresponding to the highest "1" bit in this set.

CIL[n] goes to "1" when the program reads the interrupt source number for a level n interrupt request from the IRN register.

CIL[n] returns to "0" when the program writes "1" to it. Alternatively, writing to the current interrupt level clear (CILCL) register clears the highest "1" bit in this set.

An interrupt handler must use one of these two writes to reset the highest "1" bit to "0" before returning.

8.4.8 Interrupt Level Control Register 0 (ILC0)

This register specifies the 3-bit interrupt levels for IRQ request sources nIR0 to nIR7 in four groups. The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ILC0	– *	– *	– *	– *	– *		ILR6		– *	– *	– *	– *	– *		ILR4	
After a 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
	– *	– *	– *	– *	– *	– *	– *	– *	– *		ILR1		– *		ILR0	
	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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures. Always write "0" to these bits.

Bit Descriptions

- **ILR0** (bits 0 to 2), **ILR1** (bits 4 to 6), **ILR4** (bits 16 to 18), **ILR6** (bits 24 to 26):
  ILRn specifies the 3-bit interrupt level for IRQ request source nIRn and any others in its group.

  The higher the numerical value, the higher the priority. A setting of zero, on the other hand, masks interrupts from that source.

ILR**			Interrupt Level (Priority)
2	**1**	**0**	
1	1	1	111B = 7 (high priority)
1	1	0	↕
1	0	1	
1	0	0	
0	1	1	
0	1	0	
0	0	1	001B = 1 (low priority)
0	0	0	Interrupts masked

The following Table summarizes the mapping of nIR0 to nIR7 to the four groups and the four fields in this register.

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 Interrupt Level Control Register 1 (ILC1)

This specifies the 3-bit interrupt levels for IRQ request sources nIR8 to nIR15.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ILC1	– *		ILR15		– *		ILR14		– *		ILR13		– *		ILR12	
After a 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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures. Always write "0" to these bits.

Bit Descriptions

- **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):
  ILRn specifies the 3-bit interrupt level for IRQ request source nIRn.

  The higher the numerical value, the higher the priority. A setting of zero, on the other hand, masks interrupts from that source.

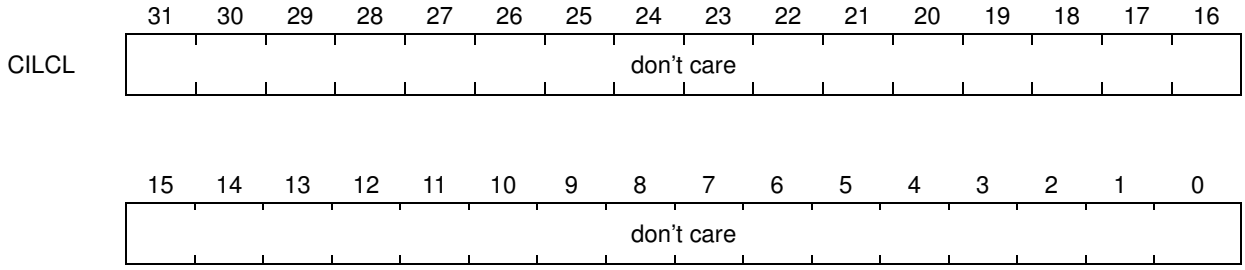
ILR8 to 15			Interrupt Level (Priority)
2	**1**	**0**	
1	1	1	111B = 7 (high priority)
1	1	0	↑
1	0	1	↑
1	0	0	↑
0	1	1	↑
0	1	0	↑
0	0	1	001B = 1 (low priority)
0	0	0	Interrupts masked

The following Table summarizes the mappings of nIR0 to nIR7 to the fields in this register.

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 Interrupt Level Clear Register (CILCL)

Writing to this register clears the highest "1" bit in the current interrupt level (CIL) register, indicating to the interrupt controller's priority judgment circuitry that processing of the current interrupt is complete.
The data written does not matter.
The CPU has only write access to this register.



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

Note

An interrupt handler, before returning, must reset the highest "1" bit in CIL to "0" either by writing to this register or by writing a "1" to the corresponding CIL bit.

8.4.11 Current Interrupt Level Encode Register (CILE)

This register gives, as a binary value, the bit position for the highest "1" bit in the current interrupt level (CIL) register and thus the interrupt level for the current interrupt request.
The CPU has only read access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIL	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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: 32 bits

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures.

Bit Descriptions

- **CILE[2:0]** (bits 0 to 2):
  These bits give the bit position and thus the interrupt level.

CIL[7:1]	CILE[2:0]	Interrupt Level (from CILE[2:0])
0000000	000	No interrupts pending
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)

Writing an interrupt source number (28 to 31) to this register resets the corresponding interrupt request bit to "0", but only if the specified external interrupt (nIR28 to nIR31) uses edge detection as the trigger. Such writes are ignored for level detection.
The CPU has only write access to this register.

IRCL

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

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: 32 bits

Notes
– *: Always write "0" to these bits.

Bit Descriptions

- **IRCL[6:0]** (bits 0 to 6):
  These bits correspond to the interrupt source number for external interrupts (nIR28 to nIR31). For edge triggered interrupts, writing an interrupt source number to these bits will de-assert the corresponding interrupt.

8.4.13 IRQA Register (IRQA)

This register indicates the status of a pending interrupt request. Reading this register returns a "1" for a corresponding pending interrupt. Writing "1" to a bit resets it to "0," clearing the corresponding interrupt request, but only if that external interrupt (nIR28 to nIR31) uses edge detection as the trigger. Such writes are ignored for level detection as are writes to the IRQ clear (IRCL) register. Also, writes to bits 0 to 11 of this register are ignored.
The CPU has read/write access to this register, subject to write restrictions noted below.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQA	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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: 32 bits

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures. Always write "0" to these bits.

Bit Descriptions

- **IRQ[31:16]** (bits 0 to 15):
  When reading, these bits return the status of pending interrupts. Reading a "1" in bit n indicates a pending interrupt request (unmasked) from the corresponding IRQ source. The below table summarizes this.

IRQ[31:16]	Description
0	No interrupt request pending
1	Interrupt request pending

When writing, writing a "1" to either of bits 12 to 15 resets it to "0," clearing the corresponding interrupt request, but only if that external interrupt (nIR28 to nIR31) uses edge detection as the trigger. Such writes are ignored for level detection as are writes to the IRQ clear (IRCL) register. Also, writes to bits 0 to 11 of this register are ignored.
The following table summarizes the effects of a write to bits 12-15 of this register.

IRQ[31:16]	Description
0	No effect
1	Clear interrupt request bit

The following table shows the condition that each IRQ[n] (n = 16 to 31) bit is set, if the interrupt uses level detection as the trigger. Interrupt Request State in the table is interrupt request signal from each internal device or external interrupt pin. Interrupt Level Setting is a value of interrupt level in the ILC1 register.

Interrupt Request State	Interrupt Level Setting	IRQ[n]
"H"	—	"0"
"L"	0	"0"
"L"	Nonzero (1 to 7)	"1"

The following table shows the condition that each IRQ[n] (n = 28 to 31) bit is set or cleared, if the interrupt uses edge detection as the trigger. The IRQA register provides two ways to clear the corresponding bit for an external interrupt (IRQ[28] to IRQ[31]) that is using edge detection as the trigger.

Event	Interrupt Level Setting	IRQ[n]
Writing "1" to IRQ[n]	—	Clear to "0"
Writing corresponding interrupt number in IRCL register	—	Clear to "0"
Rising edge in interrupt request signal	—	Not set ("0")
Falling edge in interrupt request signal*	0	Not set ("0")
	Nonzero (1 to 7)	Set to "1"

*IRQ[n] goes to "1" only if the corresponding interrupt level is nonzero.

8.4.14 IRQ Detection Mode Setting Register (IDM)

This register specifies the interrupt detection modes (level or edge detection) and polarities (negative or positive logic) for interrupt request pairs from nIR28 to nIR31.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDM	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
	IDMP30	IDM30	IDMP28	IDM28	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
	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. When switching triggers to edge detection, write to either the IRQ clear (IRCL) register or the IRQ register A (IRQA) to initialize the edge detection circuitry before disabling masking.
2. Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures. Always write "0" to these bits.

Bit Descriptions

- **IDM28** (bit 12), **IDMP28** (bit 13), **IDM30** (bit 14), **IDMP30** (bit 15):
  IDMn (n = 28 or 30) specifies the detection mode for nIRn and nIRn+1; IDMPn, the polarity.

		Polarity (IDMPn)	
Detection Mode (IDMn)		**0**	**1**
Level detection	0	Low level input	High level input
Edge detection	1	Falling edge	Rising edge

Notes

If the External Interrupt (EXINT) is going to be used for exiting STANDBY mode, it must be configured for LEVEL detection.

The above settings apply to two interrupt requests each, nIRn and nIRn+1.

Interrupt Sources	Bits Specifying Detection Mode and Polarity
IRQ28	IDM28
IRQ29	IDMP28
IRQ30	IDM30
IRQ31	IDMP30

8.4.15 Interrupt Level Control Register (ILC)

This register specifies the 3-bit interrupt priority levels for interrupt request pairs from nIR16 to nIR31. The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ILC	– *		ILC30		– *		ILC28		– *		ILC26		– *		ILC24	
After a 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	
	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

Notes

Bits labeled "– *" return "0" for reads, but we recommend that the program not assume "0" and mask them or adopt other "don't care" measures. Always write "0" to these bits.

Bit Descriptions

- **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): ILCn specifies the 3-bit interrupt priority level for IRQn and IRQn+1.

  The higher the numerical value, the higher the priority. A setting of zero, on the other hand, masks interrupts from that source.

ILC**			Interrupt Level (Priority)
2	**1**	**0**	
1	1	1	111B = 7 (high priority)
1	1	0	↑
1	0	1	
1	0	0	
0	1	1	
0	1	0	↓
0	0	1	001B = 1 (low priority)
0	0	0	Interrupts masked

The above settings apply to two interrupt requests each, IRQn and IRQn+1.

Interrupt Sources	Interrupt Priority Level
IRQ16	ILC16
IRQ17	
IRQ18	ILC18
IRQ19	
IRQ20	ILC20
IRQ21	
IRQ22	ILC22
IRQ23	

Interrupt Sources	Interrupt Priority Level
IRQ24	ILC24
IRQ25	
IRQ26	ILC26
IRQ27	
IRQ28	ILC28
IRQ29	
IRQ30	ILC30
IRQ31	

8.4.16 Register Settings for Interrupt Sources

The following Table summarizes the register settings for configuring interrupt requests.

Interrupt Source Number	Interrupt Source	Interrupt Level		Detection Mode	
		Register	Bit	Register	Bit
nFIQ	nFIQ	—	—	—	—
nIR0	System timer	ILC0	ILR0	—	—
nIR1	Watchdog timer		ILR1		
nIR2	Watchdog timer interval timer operation				
nIR3	(unused)				
nIR4	GPIOA	ILC0	ILR4	—	—
nIR5	GPIOB				
nIR6	(unused)		ILR6		
nIR7	(unused)				
nIR8	Software interrupt requests	ILC1	ILR8		
nIR9	UART		ILR9		
nIR10	SIO		ILR10		
nIR11	AD		ILR11		
nIR12	PWM output 0		ILR12		
nIR13	PWM output 1		ILR13		
nIR14	(unused)		ILR14		
nIR15	(unused)		ILR15		
nIR16	Timer 0	ILC	ILC16		
nIR17	Timer 1				
nIR18	Timer 2		ILC18		
nIR19	Timer 3				
nIR20	Timer 4		ILC20		
nIR21	Timer 5				
nIR22	(unused)		ILC22		
nIR23	(unused)				
nIR24	DMA channel 0		ILC24		
nIR25	DMA channel 1				
nIR26	(unused)		ILC26		
nIR27	(unused)				
nIR28	External interrupt 0		ILC28	IDM	IDM28, IDMP28
nIR29	External interrupt 1				
nIR30	External interrupt 2		ILC30		IDM30, IDMP30
nIR31	External interrupt 3				

8.5 Description of Operation

8.5.1 External Fast Interrupt (EFIQ_N)

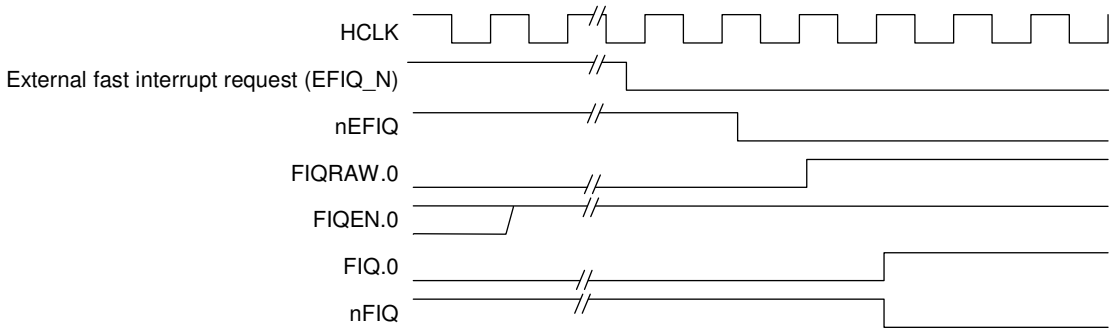
The external fast interrupt request (EFIQ_N) pin is a high-priority interrupt request normally assigned to a single, time critical source. If FIQEN, bit 0 in the FIQEN register, does not mask the interrupt request, the interrupt controller asserts the fast interrupt request (nFIQ) signal to the CPU in response to a detected interrupt trigger.

External Fast Interrupt (EFIQ_N) Sequence

0. Remove FIQ interrupt mask
   Setting F, bit 6 in the CPU's current program status (CPSR) register, to "0" and FIQEN, bit 0 in the FIQEN register, to "1" disables masking.

1. Wait for interrupt
   External fast interrupt request input from the ML674000 EFIQ_N pin sets bit 0 in the FIQRAW register to "1."

2. Relay exception request to CPU
   The interrupt controller sets bit 0 in the FIQ register to "1" and asserts the fast interrupt request (nFIQ) signal to the CPU.



3. Accept request
   If F, bit 6 in the CPU's current program status (CPSR) register, enables FIQ exceptions, the CPU saves the address of the next instruction in r14_fiq, saves the CPSR contents in SPSR_fiq, and sets the CPSR F and I bits to "1" to block acceptance of both FIQ and IRQ exceptions by the CPU.

4. Process interrupt
   The FIQ exception handler must negate the FIQ request from the source before returning.

5. Return from interrupt
   The FIQ exception handler terminates by executing a return from interrupt instruction, which restores the instruction address and CPSR contents from r14_fiq and SPSR_fiq.

Note
For further details on CPU processing of FIQ exceptions, refer to the ARM7TDMI data sheet.

8.5.2 External and Internal Interrupts (IRQn)

These are regular interrupt requests with lower priority than the fast interrupt request (FIQ). The interrupt controller assigns priority of execution to simultaneous interrupt requests by comparing their interrupt levels. If a new interrupt request has a higher interrupt level than the one currently being processed, the interrupt controller asserts the interrupt request (nIRQ) signal to the CPU.
The IRQ exception handler reads the interrupt source number and level from interrupt controller registers.

Assigning Priority to Interrupt Requests

1. The higher the numerical value, the higher the priority.

2. If two interrupt requests have the same interrupt level, priority goes to the one with the higher interrupt source number.

3. Reading the IRN register in response to an IRQ exception masks interrupt requests at or below the new interrupt level.

External/Internal Interrupt (IRQn) Sequence

0. Specify interrupt level (software)
   Setting I, bit 7 in the CPU's current program status (CPSR) register, to "0" and specifying a nonzero interrupt level for the interrupt disables masking. An interrupt level of zero masks interrupts.

Note

Sources 28 to 31 require an additional preliminary step: specifying the detection mode and polarity in the IRQ detection mode setting register (IDM). When switching triggers to edge detection, write to either the IRQ clear (IRCL) register or the IRQ register A (IRQA) to initialize the edge detection circuitry before disabling masking.

1. Wait for interrupt (hardware)
   If the interrupt request is an external interrupt (nIR28 to nIR31) using edge detection as the trigger, the interrupt controller sets the corresponding bit in the IRQ register A (IRQA) to "1".

2. Relay exception request to CPU (hardware)
   If the interrupt request has an interrupt level higher than the contents of the current interrupt level encode (CILE) register, which gives the bit position for the highest "1" bit in the current interrupt level (CIL) register, the interrupt controller writes the highest interrupt number for interrupt requests at that higher interrupt level to the IRQ number (IRN) register and asserts the interrupt (nIRQ) signal to the CPU.
   If the interrupt level is less than or equal to that in CIL (and CILE), however, there is no IRQ exception, and IRN goes to zero.

3. Accept request (hardware)
   If I, bit 7 in the CPU's current program status (CPSR) register, enables IRQ exceptions, the CPU saves the address of the next instruction in the link (R14_irq) register, saves the CPSR contents in the program status (SPSR_irq) register, and sets I, bit 6 in the CPU's current program status (CPSR) register, to "1" to block acceptance of IRQ exceptions by the CPU. Control then passes to the IRQ exception handler.

4. Process interrupt (software & hardware)
   The IRQ exception handler (software) reads the interrupt source number from the IRQ number (IRN) register and branches to the corresponding interrupt handler.
   The interrupt controller (hardware) clears the IRN register to zero, sets the current interrupt level (CIL) register bit corresponding to the interrupt level to "1," masking pending interrupt requests at or below that level, negates the interrupt request (nIRQ) signal to the CPU, and writes that interrupt level as a binary value to the current interrupt level (CIL) register.

The interrupt handler (software), before returning, resets the corresponding bit in the IRQ register A (IRQA) to "0" if the external interrupt (nIR28 to nIR31) uses edge detection as the trigger.

5. Return from interrupt (software)
   Writing to the current interrupt level clear (CILCL) register resets the highest "1" bit in CIL, the one indicating the interrupt level currently being processed, to "0." The data written does not matter.
   The interrupt handler terminates by executing a return from interrupt instruction, which restores the instruction address and CPSR contents from the link (R14_irq) and program status (SPSR_irq) registers.

Note

For further details on CPU processing of IRQ exceptions, refer to the ARM7TDMI data sheet.

8.5.3 Nested Interrupts and Re-Entrant Interrupt Service Routines

Setting I, bit 7 in the CPU's current program status (CPSR) register, to "0" disables masking, allowing an interrupt request with an interrupt level higher than that for the one currently being processed to take control, and thus facilitating interrupt nesting.
Before an interrupt handler enables interrupt nesting, however, it must first save to the stack the contents of the link (R14_irq) and program status (SPSR_irq) registers because the CPU hardware overwrites them when it accepts such a nested IRQ exception.
If the interrupt handler calls subroutines, it must also protect the subroutine's return address in R14_irq from the overwrite inherent in accepting such a nested IRQ exception by shifting to system mode. System mode uses the same register bank as user mode, but allows the same access as IRQ mode to status registers and the like. Note that an interrupt handler running in this (or user) mode must save to the stack the contents of the link (R14_irq) register and any work registers that it uses.

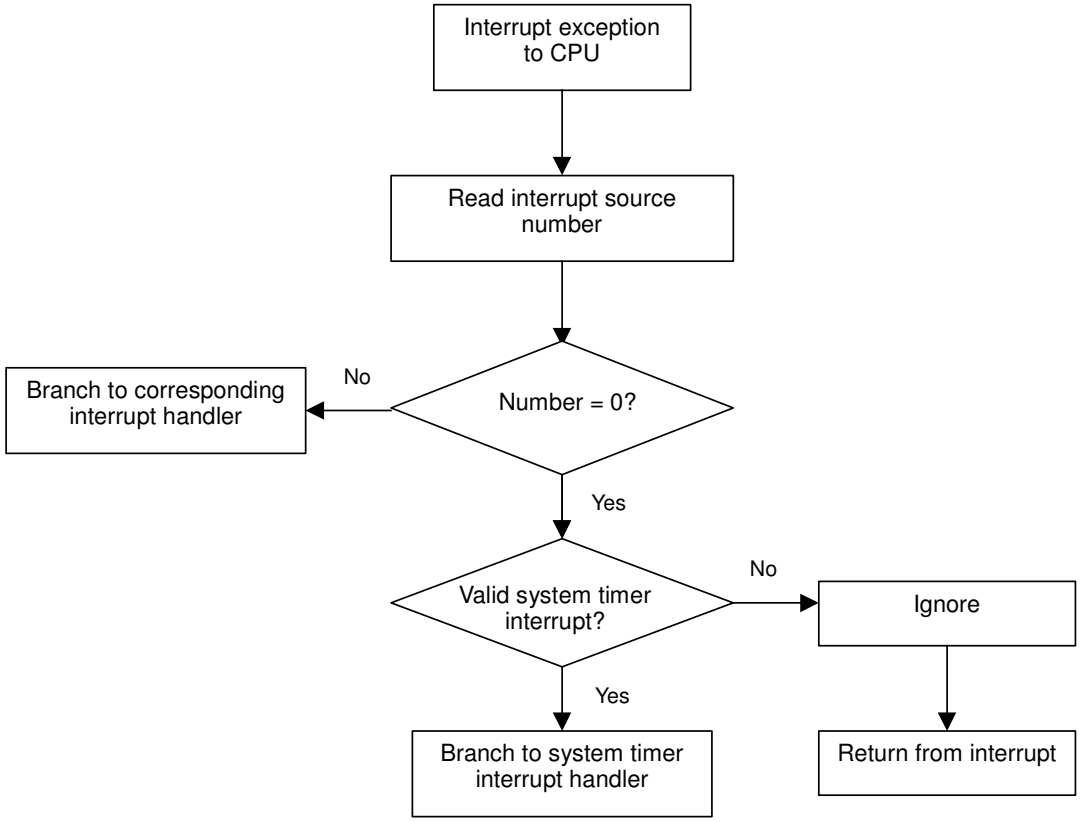
The following outlines the procedure.

1. Save contents of R14_irq, SPSR_irq, and any necessary work registers.
2. Read IRN.
3. Switch to system mode and enable IRQ exceptions.
4. Save contents of system (user) mode link register and any necessary work registers.
5. Execute body of interrupt handler.
6. Restore contents of system (user) mode registers.
7. Disable IRQ exceptions and switch back to IRQ mode.
8. Clear corresponding bit in CIL register.
9. Restore contents of work registers, SPSR_irq, and R14_irq.
10. Return from interrupt.

8.5.4 Important Notes on Interrupts

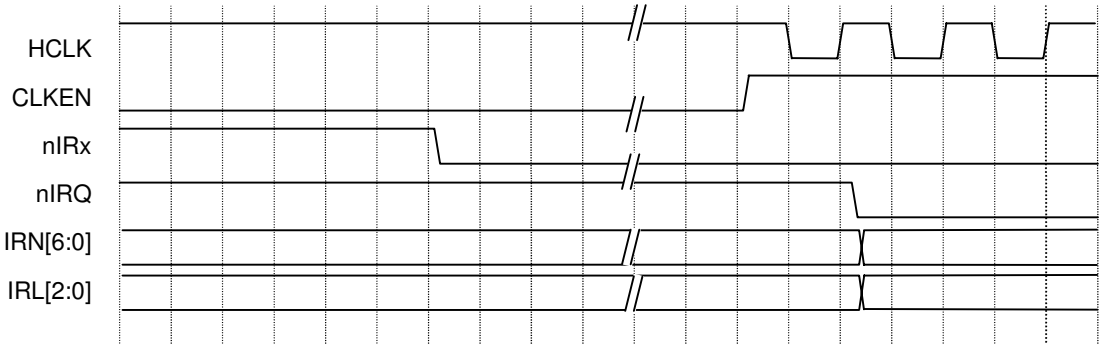
- Reading IRN
  Reading the IRN register clears it to zero.

- Invalid Interrupts
  Loss of the interrupt trigger after the interrupt controller asserts the interrupt request (nIRQ) signal to the CPU sometimes causes the read from the IRQ number (IRN) register to return 0, the interrupt source number assigned to the system timer interrupt. The IRQ exception handler must therefore cross-check the system timer interrupt status and branch to the corresponding interrupt handler only if the interrupt request is valid. Otherwise, it must ignore the invalid interrupt and simply return.

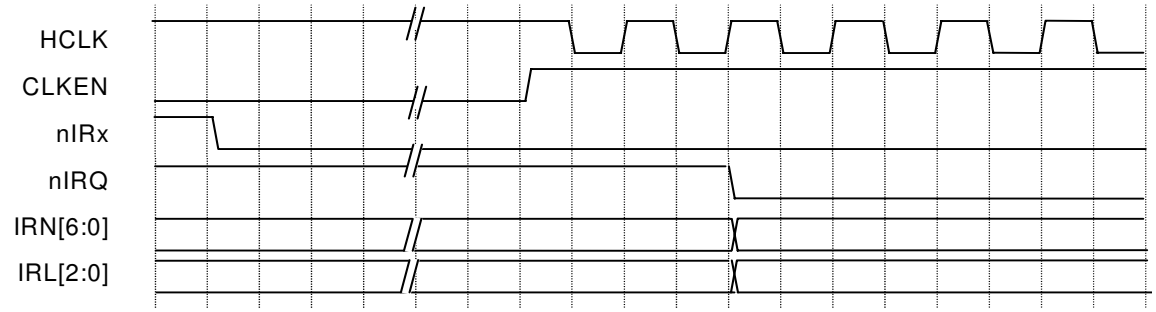


8.5.5 Waking from HALT and STANDBY Modes

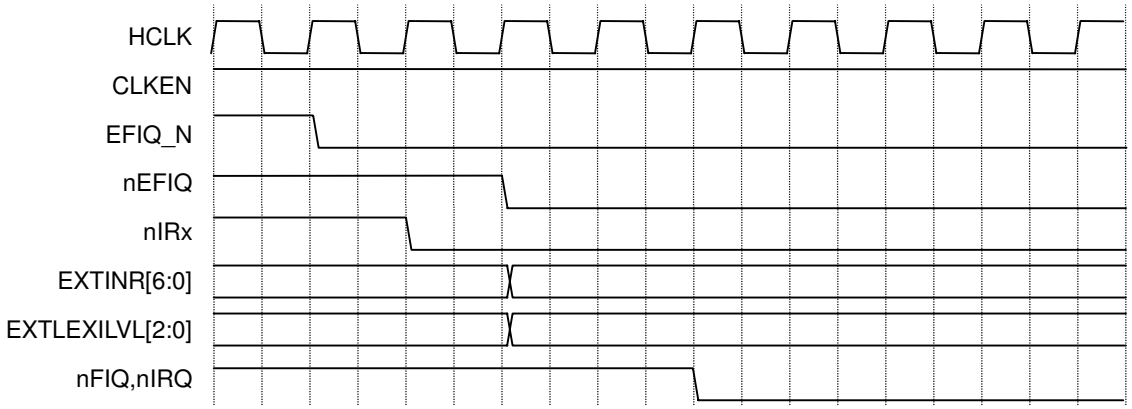
The following Figures represent timing charts for waking the CPU from HALT or STANDBY mode with interrupt requests.



(a) HALT/STANDBY/STOP → RUN for interrupt requests nIR0 to nIR15 (asynchronous clock)



(b) HALT/STANDBY/STOP → RUN for interrupt requests nIR16 to nIR31 (asynchronous clock)



(c) RUN (synchronous clock)

8.5.6 Error Response

Accessing addresses 0x78000030 to 0x780FFFFF in the region assigned to the interrupt controller (INTRC) produces a data abort exception.

8.5.7 Interrupt Response Times

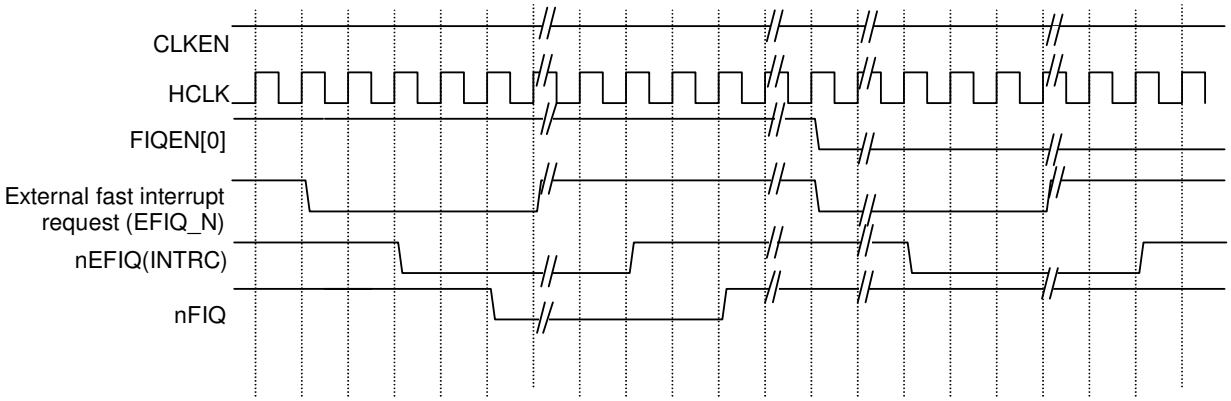
Fast interrupt (nFIQ)	4 clock cycles from EFIQ_N external interrupt input to CPU FIQ input
Interrupt request sources (nIR0 to nIR15)	2 clock cycles from IRQ request to CPU IRQ input
Interrupt request sources (nIR16 to nIR27)	3 clock cycles from IRQ request to CPU IRQ input
Interrupt request sources (nIR28 to nIR31)	5 clock cycles from EXINTx external interrupt input to CPU IRQ input

Note

For the delays between the IRQ or FIQ exception request to the CPU and the actual start of the corresponding exception handler, refer to the ARM7TDMI data sheet.

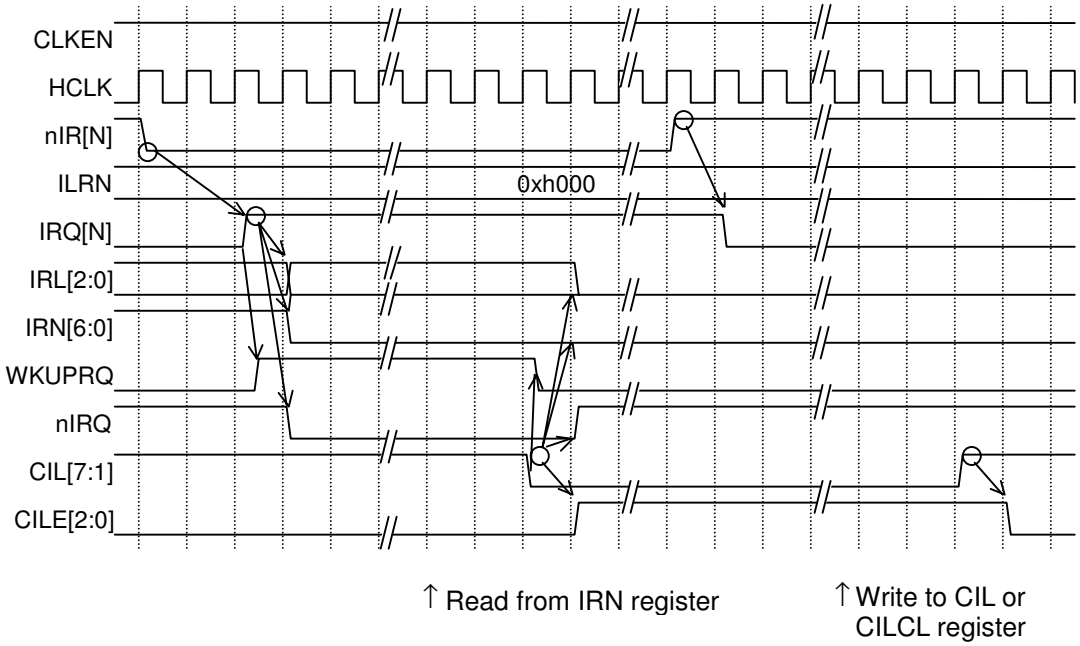
8.6 Interrupt Acceptance Timing Charts

8.6.1 FIQ Interrupt Timing Chart

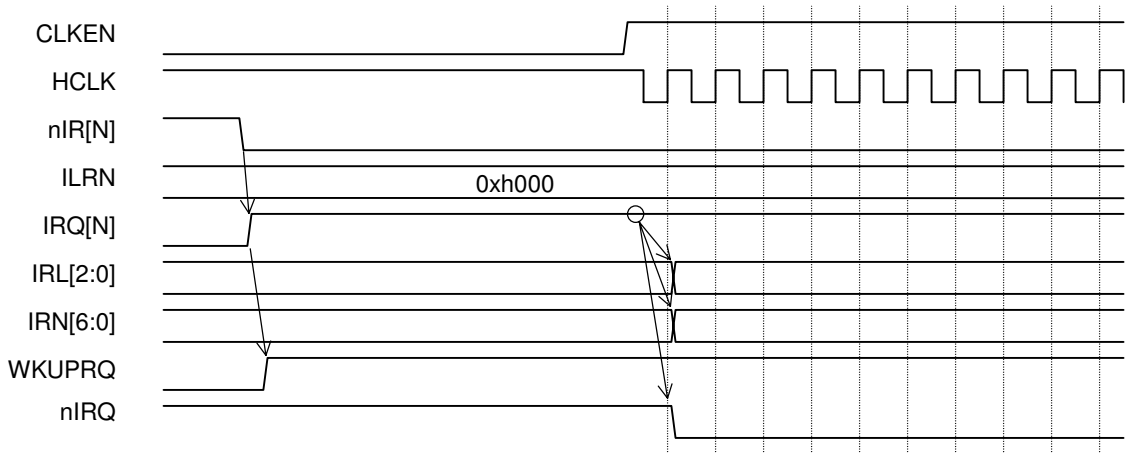


Clock operational

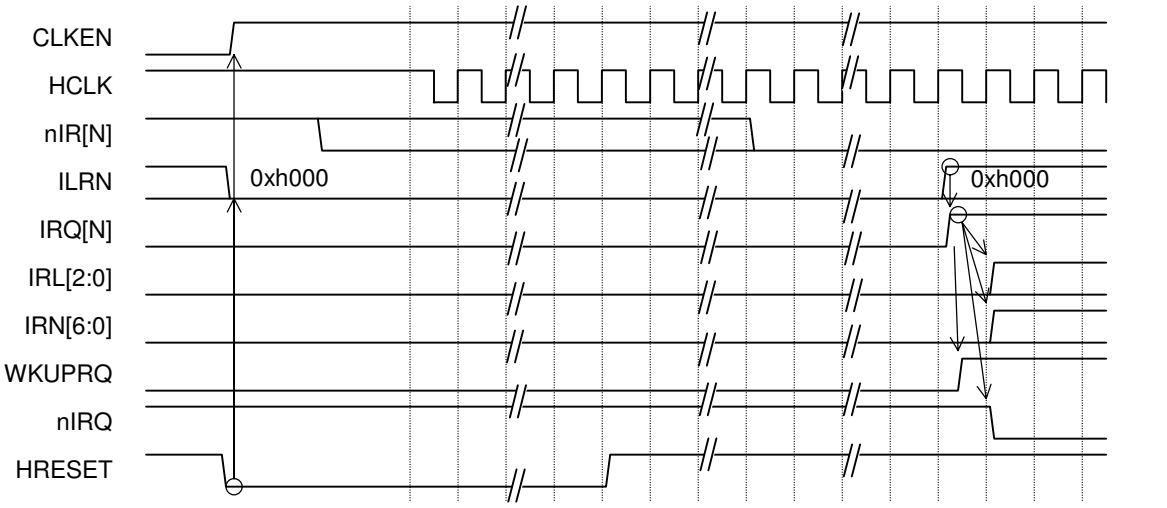
8.6.2 IRQ Interrupt Timing Chart (nIR0 to nIR15)



Clock operational

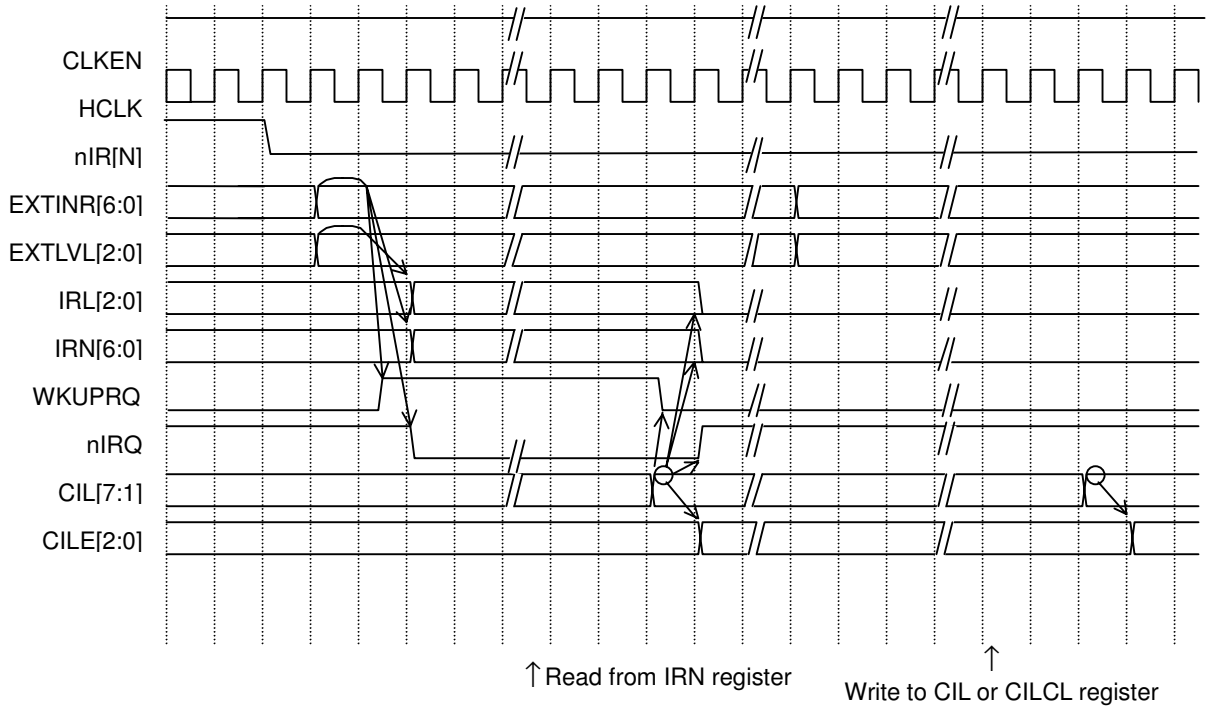


STOP → RUN

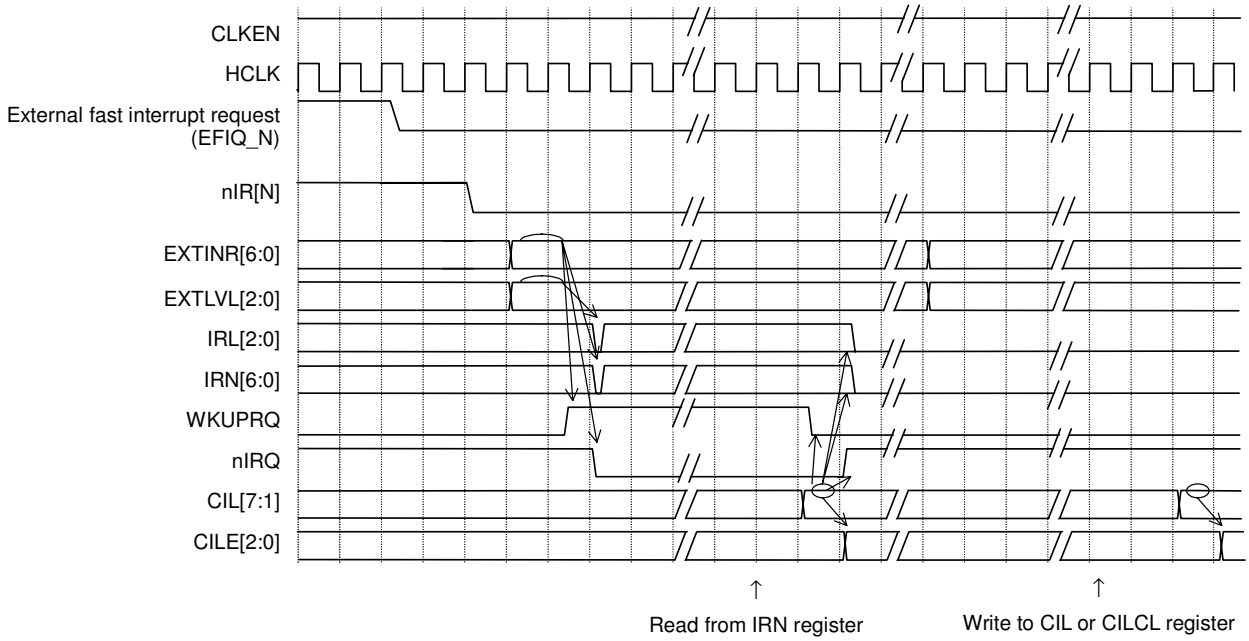


Sleep → RUN

8.6.3 IRQ Interrupt Timing Chart (nIR16 to nIR31)



Clock operational (nIR16 to nIR27)



Clock operational (nIR28 to nIR31)

Chapter 9

Built-In Memory

Chapter 9 Built-In Memory

9.1 Overview

This LSI contains 8 kilobytes (2K × 32 bits) of built-in SRAM at addresses 0x10000000 to 0x10001FFF.

9.2 Built-In Memory

After a reset, bank 0 (0x00000000 to 0x07FFFFFF) is mapped to an external ROM for use as the boot device. Writing 0x000C to the remap control (RMPCON) register remaps bank 0 to built-in RAM.
The built-in RAM has the following features.

- Bus width: 32 bits, with 8-, 16-, and 32-bit read/write access
- Memory WAIT cycles: 0 for reads; 1 for writes
- Bank 5 mirrors the internal SRAM.

Chapter 10

External Memory Controller

Chapter 10 External Memory Controller

10.1 Overview

The external memory controller built into this LSI is for connecting DRAM, ROM, SRAM, I/O devices, and other external devices to this LSI. Supported devices include the following.

- Asynchronous ROM
- Asynchronous SRAM
- Flash memory
- I/O devices
- SDRAM
- EDO DRAM

10.1.1 Pin List

Table 10.1 lists the pins for connecting devices to this controller.

Table 10.1

Pin Name	I/O	Function	Secondary Function
XA[23:19]	O	External address bus	PIOA[14:10]
XA[18:0]	O	External address bus	–
XD[15:0]	I/O	External data bus	–
XROMCS_N	O	External ROM chip select	–
XRAMCS_N	O	External RAM chip select	–
XIOCS_N[0]	O	I/O bank 0 chip select	–
XIOCS_N[1]	O	I/O bank 1 chip select	–
XOE_N	O	Output enable	–
XWE_N	O	Write enable	–
XBS_N[1:0]	O	External bus (XBUS) byte select	–
XBWE_N[0]	O	Write enable (LSB)	–
XBWE_N[1]	O	Write enable (MSB)	–
XSDCS_N	O	SDRAM chip select	PIOB[12]
XSDCLK	O	SDRAM clock	PIOB[11]
XSDCKE	O	SDRAM clock enable	PIOB[13]
XCAS_N	O	SDRAM column address strobe	PIOB[9]
XRAS_N	O	Row address strobe	PIOB[10]
XDQM[1]/XCAS_N[1]	O	Data I/O mask for SDRAM or MSB column address strobe for EDO DRAM	PIOB[14]
XDQM[0]/XCAS_N[0]	O	Data I/O mask for SDRAM or LSB column address strobe for EDO DRAM	PIOB[15]
XWAIT	I	Memory wait indicator	PIOB[8]
XWR	O	External bus (XBUS) transfer direction	PIOA[15]

10.1.2 Register List

Address	Name	Abbreviation	R/W	Size	Initial Setting
0x78100000	Bus width control register	BWC	R/W	32	0x00000008
0x78100004	External ROM access control register	ROMAC	R/W	32	0x00000007
0x78100008	External RAM access control register	RAMAC	R/W	32	0x00000007
0x7810000C	External I/O bank 0 access control register	IO0AC	R/W	32	0x00000007
0x78100010	External I/O bank 1 access control register	IO1AC	R/W	32	0x00000007
0x78180000	DRAM bus width control register	DBWC	R/W	32	0x00000000
0x78180004	DRAM control register	DRMC	R/W	32	0x00000000
0x78180008	DRAM characteristics control register	DRPC	R/W	32	0x00000000
0x7818000C	SDRAM mode register	SDMD	R/W	32	0x00000001
0x78180010	DRAM command register	DCMD	R/W	32	0x00000000
0x78180014	DRAM refresh cycle control register 0	RFSH0	R/W	32	0x00000000
0x78180018	DRAM power down control register	RDWC	W	32	0x00000003
0x7818001C	DRAM refresh cycle control register 1	RFSH1	R/W	32	0x00000000

10.2 Register Descriptions

10.2.1 Bus Width Control Register (BWC)

This register specifies the external data bus widths for four banks/regions.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BWC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **ROMBW[1:0]** (bits 2 and 3):
  These bits specify the bus width for the ROM region.

ROMBW[1:0]		Description
3	2	
0	0	Not physically present (Access to bank/region disabled. Access from CPU produces abort response.)
0	1	(reserved)
1	0	16 bits
1	1	(reserved)

Note: The only bus width supported is 16 bits. Operation is not guaranteed for a setting labeled "reserved."

- **RAMBW[1:0]** (bits 4 and 5):
These bits specify the bus width for the SRAM region.

RAMBW[1:0]		Description
5	4	
0	0	Not physically present (Access to bank/region disabled. Access from CPU produces abort response. Access from DMA controller produces error response.)
0	1	(reserved)
1	0	16 bits
1	1	(reserved)

Note: The only bus width supported is 16 bits. Operation is not guaranteed for a setting labeled "reserved."

- **IO0BW[1:0]** (bits 6 and 7):
These bits specify the bus width for I/O bank 0.

IO0BW[1:0]		Description
7	6	
0	0	Not physically present (Access to bank/region disabled. Access from DMA controller produces error response.)
0	1	8 bits
1	0	16 bits
1	1	(reserved)

Note: Operation is not guaranteed for a setting labeled "reserved."

- **IO1BW[1:0]** (bits 8 and 9):
These bits specify the bus width for I/O bank 1.

IO1BW[1:0]		Description
9	8	
0	0	Not physically present (Access to bank/region disabled. Access from DMA controller produces error response.)
0	1	8 bits
1	0	16 bits
1	1	(reserved)

Note: Operation is not guaranteed for a setting labeled "reserved."

10.2.2 External ROM Access Control Register (ROMAC)

This register controls ROM access timing.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ROMAC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	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

Notes

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
When switching operating frequencies, adjust the contents of this register at the lower frequency--that is, AFTER a change from high speed to low speed and BEFORE one from low speed to high speed.

Bit Descriptions

- **ROMTYPE[2:0]** (bits 0 to 2):
  These bits specify the ROM access timing.

ROMTYPE[2:0]			OE/WE pulse width (in clock cycles)	Read off timing[1]	Notes
2	1	0			
0	0	0	1	0	
0	0	1	2	0	
0	1	0	3	2	
0	1	1	4	2	
1	0	0	(reserved)		Operation is not guaranteed for a setting labeled "reserved."
1	0	1	(reserved)		Operation is not guaranteed for a setting labeled "reserved."
1	1	0	(reserved)		Operation is not guaranteed for a setting labeled "reserved."
1	1	1	8	4	

1: Read off time is the interval measured from, the time when Output Enable (OE) becomes inactive in one memory cycle to the time it becomes active in the next memory cycle.

10.2.3 External SRAM Access Control Register (RAMAC)

This register controls SRAM access timing.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RAMAC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	RAMTYPE[2:0]		
	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

Notes

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
When switching operating frequencies, adjust the contents of this register at the lower frequency--that is, AFTER a change from high speed to low speed and BEFORE one from low speed to high speed.

Bit Descriptions

- **RAMTYPE[2:0]** (bits 0 to 2):
  These bits specify the SRAM access timing.

RAMTYPE[2:0]			OE/WE pulse width (in clock cycles)	Read off timing[1]	Notes
2	1	0			
0	0	0	1	0	
0	0	1	2	0	
0	1	0	3	2	
0	1	1	4	2	
1	0	0	(reserved)		Operation is not guaranteed for a setting labeled "reserved."
1	0	1	(reserved)		Operation is not guaranteed for a setting labeled "reserved."
1	1	0	(reserved)		Operation is not guaranteed for a setting labeled "reserved."
1	1	1	8	4 (3)[2]	

1: Read off time is the interval measured from, the time when Output Enable (OE) becomes inactive in one memory cycle to the time it becomes active in the next memory cycle.
2: Read off timing is 3 in case of Accessing by DMA

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

This register controls the access timing for I/O bank 0.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IO0AC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
IO0AC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	IO0TYPE[2:0]		
After a reset	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

Notes

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

When switching operating frequencies, adjust the contents of this register at the lower frequency--that is, AFTER a change from high speed to low speed and BEFORE one from low speed to high speed.

Bit Descriptions

- **IO0TYPE[2:0]** (bits 0 to 2):
  These bits specify the access timing for I/O bank 0.

IO0TYPE[2:0]			Pulse width (in clock cycles)			Notes
2	**1**	**0**	**(1)**	**(2)**	**(3)**	
0	0	0	1	1	1	
0	0	1	1	4	2	
0	1	0	(reserved)			Operation is not guaranteed for a setting labeled "reserved."
0	1	1	2	8	4	
1	0	0	2	12	6	
1	0	1	2	16	7	
1	1	0	(reserved)			Operation is not guaranteed for a setting labeled "reserved."
1	1	1	4	24	10	

(1) Address setup (in clock cycles)
(2) OE/WE pulse width (in clock cycles)
(3) Read off timing or write data hold (in clock cycles)

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

This register controls the access timing for I/O bank 1.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IO1AC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	IO1TYPE[2:0]		
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

Notes

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

When switching operating frequencies, adjust the contents of this register at the lower frequency--that is, AFTER a change from high speed to low speed and BEFORE one from low speed to high speed.

Bit Descriptions

- **IO1TYPE[2:0]** (bits 0 to 2):
  These bits specify the access timing for I/O bank 1.

IO1TYPE[2:0]			Pulse width (in clock cycles)			Notes
2	**1**	**0**	**(1)**	**(2)**	**(3)**	
0	0	0	1	1	1	
0	0	1	1	4	2	
0	1	0	(reserved)			Operation is not guaranteed for a setting labeled "reserved."
0	1	1	2	8	4	
1	0	0	2	12	6	
1	0	1	2	16	7	
1	1	0	(reserved)			Operation is not guaranteed for a setting labeled "reserved."
1	1	1	4	24	10	

(1) Address setup (in clock cycles)
(2) OE/WE pulse width (in clock cycles)
(3) Read off timing or write data hold (in clock cycles)

10.2.6 DRAM Bus Width Control Register (DBWC)

This register specifies the bus width for the DRAM region.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBWC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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	1	1

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

Notes
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
The contents of this register do not affect the SDRAM clock output XSDCLK.

Bit Descriptions

- BWDRAM (bits 0 and 1): These bits specify the bus width for the DRAM region.

BWDRAM[1:0]		**Description**
1	**0**	
0	0	Not physically present (Access produces error response.)
0	1	8 bits (only possible for EDO DRAM)
1	0	16 bits
1	1	(reserved)

10.2.7 DRAM Control Register (DRMC)

This register specifies the DRAM type, access timing, and other settings.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRMC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
DRMC	– *	– *	– *	– *	– *	– *	– *	– *	RFRSH	PDWN	– *	PRELA	– *	ARCH	AMUX[1:0]	
After a reset	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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **AMUX[1:0]** (bits 0 and 1):
  These bits specify the column length which is the dividing line for address multiplexing.

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

- **ARCH** (bit 2):
  This bit specifies the DRAM type.

ARCH	Description
0	SDRAM
1	EDO-DRAM

- **PRELAT** (bit 4):
  This bit specifies the SDRAM precharge latency.

PRELAT	Description
0	Fixed at 2 clock cycles
1	Use CAS latency setting

- **PDWN** (bit 6):
  This bit controls automatic shifting to SDRAM power down mode.

PDWN	Description
0	Disable automatic shift
1	Enable automatic shift

- RFRSH (bit 7): This bit controls distributed CAS before RAS (CBR) refresh operation.

RFRSH	Description
0	Stop distributed CBR refresh
1	Enable distributed CBR refresh

10.2.8 DRAM Characteristics Control Register (DRPC)

This register specifies the DRAM access timing parameters in clock cycles.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRPC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	DRAMSPEC[3:0]			
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Note
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **DRAMSPEC[3:0]** (bits 0 to 3): These bits specify the DRAM access timing parameters in clock cycles.

DRAMSPEC[3:0]				SDRAM operation				EDO DRAM operation				
3	2	1	0	tRCD	tRAS	tRP	tDPL	tRAH tCAS	tRCD	tCAC tOEZ	tRP	
0	0	0	0	1	2	1	1	1	2	1	1	High speed (low frequency)
0	0	0	1	1	3	1	1	1	2	1	2	
0	0	1	0	2	3	2	1	1	3	1	2	
0	0	1	1	2	4	2	1	1	3	1	3	
0	1	0	0	2	4	2	2	1	3	2	3	
0	1	0	1	2	5	2	1	1	4	2	4	
0	1	1	0	2	5	2	2	1	5	2	5	
0	1	1	1	2	5	3	1	2	4	2	4	
1	0	0	0	3	5	3	2	2	5	2	5	
1	0	0	1	3	6	3	2	2	6	2	6	Low speed (high frequency)
1	0	1	0	(reserved)				3	8	3	7	
1	0	1	1	(reserved)				(reserved)				
1	1	0	0	(reserved)				(reserved)				
1	1	0	1	(reserved)				(reserved)				
1	1	1	0	(reserved)				(reserved)				
1	1	1	1	(reserved)				(reserved)				

Notes
1. Operation is not guaranteed for a setting labeled "reserved."
2. Choose a setting with a combination satisfying the access timing parameters (tRCD, tRAS, tRP, etc.) appearing in the DRAM data sheet.

10.2.9 SDRAM Mode Register (SDMD)

This register specifies the SDRAM CAS latency.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMD	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
	– *	– *	– *	– *	– *	– *	– *	– *	MODEWR	– *	– *	– *	– *	– *	– *	LTMODE
	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

Notes

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
Writing "1" to MODEWR in this register produces an SDRAM setting cycle. MODEWR always returns "0" for reads.

Bit Descriptions

- **LTMODE** (bit 0):
  This bit specifies the SDRAM CAS latency in clock cycles.

LTMODE	Description
0	2
1	3

- **MODEWR** (bit 7):
  Writing "1" to this bit produces an SDRAM setting cycle. It always returns "0" for reads.

MODEWR	Description
0	Ignore new settings
1	Use new settings

This setting cycle has the following format.

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

* Axx are the SDRAM input address signals.

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

Note: LTMODE is the only field that varies--and then only by a single bit.

10.2.10 DRAM Command Register (DCMD)

Writing to this register executes the specified command (SDRAM) or produces the corresponding access (EDO DRAM).
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCMD	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **DRCMD** (bits 0 to 2):
  These bits specify the DRAM command to execute.

DRCMD[2:0]			SDRAM Operation	EDO DRAM Operation
2	1	0		
0	0	0	No Operation	No Operation
0	0	1	(reserved)	(reserved)
0	1	0	(reserved)	(reserved)
0	1	1	(reserved)	(reserved)
1	0	0	PALL (precharge all SDRAM banks)	EDO DRAM precharge
1	0	1	REF (SDRAM CBR refresh)	EDO DRAM CBR refresh
1	1	0	SELF (start SDRAM self refresh)	Start EDO DRAM self refresh
1	1	1	SREX (end SDRAM self refresh)	End EDO DRAM self refresh

Notes
1. Operation is not guaranteed for a setting labeled "reserved."
2. During self refresh operation, operation is not guaranteed for commands other than the end self refresh command.
3. The SDRAM clock output SDCLK
   * Stops (Low level) for EDO DRAM operation (ARCH bit in DRMC register = 1)
   * Stops (Low level) after SELF command
   * Resumes after SREX command
4. The hardware blocks access to this register until the specified command finishes execution.
5. According to JEDEC standards, a DRAM chip requires 8 cycles of CBR to get properly initialized. Thus, when initializing a JEDEC standard DRAM, the application software must write '0x05', eight times to this register to initialize CBR for DRAM.

10.2.11 DRAM Refresh Cycle Control Register 0 (RFSH0)

This register specifies the DRAM refresh period relative to that specified in the RFSH1 register: double or same.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RFSH0	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	RCCON
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

Notes

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

DRAM controller operation is not guaranteed after writes to this register when there is no DRAM physically present (DBWC = 0).

Bit Descriptions

- **RCCON** (bit 0):
  This bit specifies the DRAM refresh period relative to that specified in the RFSH1 register.

RCCON	Description
0	Refresh using twice clock period specified in RFSH1 register
1	Refresh using the clock period specified in RFSH1 register

10.2.12 DRAM Refresh Cycle Control Register 1 (RFSH1)

This register specifies the divider for deriving the DRAM refresh period from the CCLK period.
The program has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RFSH1	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
RFSH1	– *	– *	– *	– *	– *	RFSEL1										
After a reset	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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **RFSEL1** (bits 0 to 10):
  These bits specify the divider for deriving the DRAM refresh period from the CCLK frequency using the following formula.

  refresh clock period = CCLK / RFSEL1[10:0]

Note
The RFSH1 register should be set before modifying CCLK divider settings.

The following table lists the results for typical combinations of the RFSEL1 setting and CCLK frequency.
The RFSEL1 setting must be within the range shown in this Table: 0x00000008 to 0x00000406. Choose it to produce a refresh frequency of at least 32 kHz or 64 kHz.

CCLK(MHz)		0.56	2	4	8	10	20	25	33
RFSEL1 setting	Divisor	kHz	kHz	kHz	kHz	kHz	kHz	kHz	kHz
0008	8	70.00	250.00	500.00	1000.00	1250.00	2500.00	3125.00	4125.00
0010	16	35.00	125.00	250.00	500.00	625.00	1250.00	1562.50	2062.50
001E	30	18.67	66.67	133.33	266.67	333.33	666.67	833.33	1100.00
003E	62	9.03	32.26	64.52	129.03	161.29	322.58	403.23	532.26
007C	124	4.52	16.13	32.26	64.52	80.65	161.29	201.61	266.13
009C	156	3.59	12.82	25.64	51.28	64.10	128.21	160.26	211.54
00FA	250	2.24	8.00	16.00	32.00	40.00	80.00	100.00	132.00
0138	312	1.79	6.41	12.82	25.64	32.05	64.10	80.13	105.77
0186	390	1.44	5.13	10.26	20.51	25.64	51.28	64.10	84.62
0202	514	1.09	3.89	7.78	15.56	19.46	38.91	48.64	64.20
0270	624	0.90	3.21	6.41	12.82	16.03	32.05	40.06	52.88
030C	780	0.72	2.56	5.13	10.26	12.82	25.64	32.05	42.31
0406	1030	0.54	1.94	3.88	7.77	9.71	19.42	24.27	32.04

The shading indicates recommended settings.

Note: The user application system must modify the contents of this register each time that it changes the CCLK frequency with the clock gear.

10.2.13 DRAM Power Down Control Register (RDWC)

This register specifies the number of idle cycles to wait before shifting DRAM to power down mode. The program has only write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDWC	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *: These bits are reserved for future expansion. Writes to them are ignored.

Bit Descriptions

- **PDCNT** (bits 0 to 3):
  This number plus one specifies the number of idle cycles to wait before shifting DRAM to power down mode.

PDCNT				Description
3	2	1	0	
0	0	0	0	Shift DRAM to power down mode after 1 or more cycles
0	0	0	1	Shift DRAM to power down mode after 2 or more cycles
0	0	1	0	Shift DRAM to power down mode after 3 or more cycles
0	0	1	1	Shift DRAM to power down mode after 4 or more cycles
0	1	0	0	Shift DRAM to power down mode after 5 or more cycles
0	1	0	1	Shift DRAM to power down mode after 6 or more cycles
0	1	1	0	Shift DRAM to power down mode after 7 or more cycles
0	1	1	1	Shift DRAM to power down mode after 8 or more cycles
1	0	0	0	Shift DRAM to power down mode after 9 or more cycles
1	0	0	1	Shift DRAM to power down mode after 10 or more cycles
1	0	1	0	Shift DRAM to power down mode after 11 or more cycles
1	0	1	1	Shift DRAM to power down mode after 12 or more cycles
1	1	0	0	Shift DRAM to power down mode after 13 or more cycles
1	1	0	1	Shift DRAM to power down mode after 14 or more cycles
1	1	1	0	Shift DRAM to power down mode after 15 or more cycles
1	1	1	1	Shift DRAM to power down mode after 16 or more cycles

10.3 Operational Description

10.3.1 Bus Width

The bus width control (BWC) register offers two bus width settings (0 and 16) for the external ROM and SRAM banks/regions and three (0, 8, and 16) for the external I/O banks.

Note:
A bus width setting of 0 means "not present."

After a reset, the contents (0x0008) specify that only ROM is present, accessible over a 16-bit bus.
All banks/regions support 16-bit access.

The table shows which addressable regions support 8-bit access as well. Operation is not guaranteed for a bank/region with an X in the 8-bit column.

Bank/Region	Bus Width	
	8 bits	16 bits
ROM	X	O
RAM	X	O
IO0	O	O
IO1	O	O
SDRAM	X	O
EDO-DRAM	O	O

If the data size differs from the bus width, access uses the bus width. A word (32-bit) read over a 16-bit bus, for example, becomes two reads: first the lower 16 bits and then the upper ones. As a result, this 32-bit access produces two XOE_N or XWE_N output signals. For further details, see the timing charts in Figure 10.1, Figure 10.2, etc.

10.3.2 ROM/SRAM Control

1. To access SRAM, set the SRAM bus width to 16 bits in the BWC register.
2. Specify the XOE_N/XWE_N pulse width in the ROMAC or RAMAC register.

After a reset, the ROM bus width is 16 bits.

10.3.3 I/O Banks Control

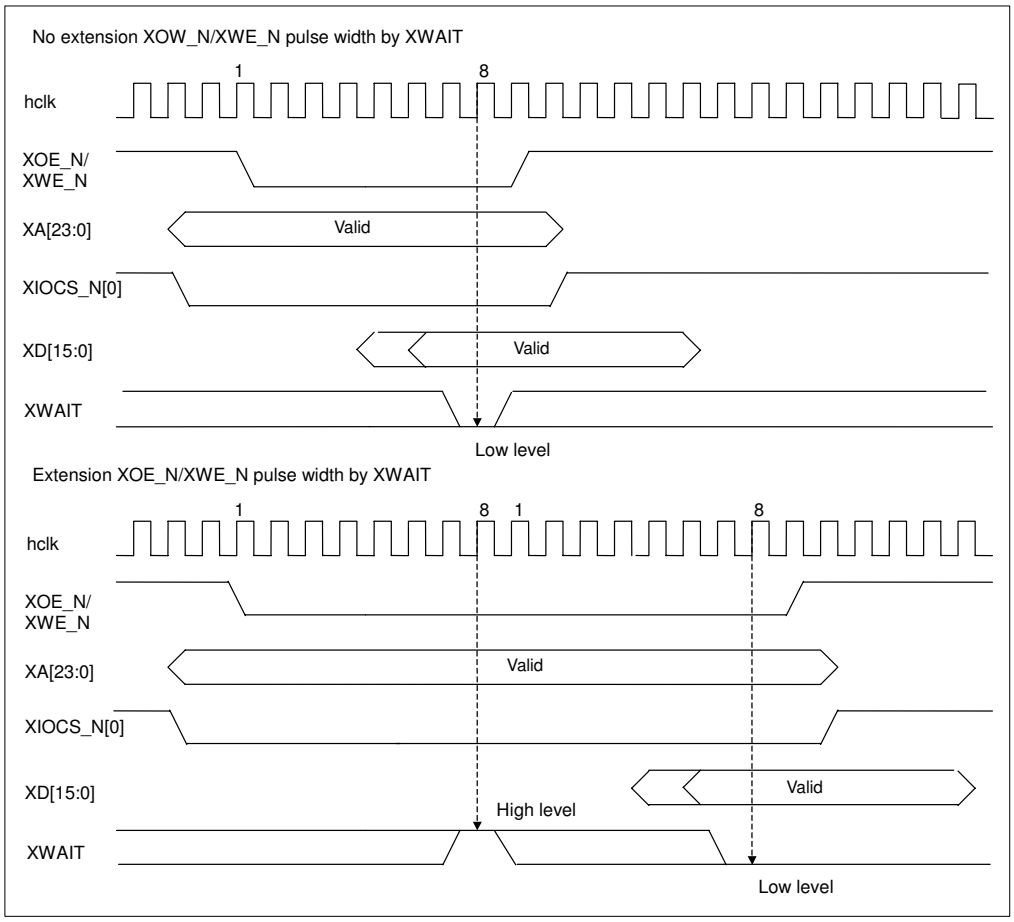
1. To access I/O Banks, set the I/O Banks bus width to 8 or 16 bits in the BWC register.
2. Specify the XOE_N/XWE_N pulse width in the IO0AC or IO1AC register.

- XWR

Indicates the direction of data transfer on XD (Data Bus) for I/O Banks.
Low : read (from the external I/O device to the chip)
High : write (from the chip to the external I/O device)

- XWAIT (Only for I/O Bank 0)

XWAIT input is for extending the I/O device access time. XWAIT is sampled one system clock before XOE_N/XWE_N is deasserted. If XWAIT is at high level at sampling time, the XOE_N/XWE_N pulse width will be extended for additional time based on parameters specified in the IO0AC. When XWAIT is at low level at sampling time, XOE_N/XWE_N pulse width is not extended.



10.3.4 DRAM Control

DRAM Controller disable by Mode selection pins

For the DRAM controller to be operational, the Mode Selection Pins of the MCU should be configured so as to enable the DRAM controller. The Mode Selection Pin configuration is described in Chapter 4 --"Chip Configuration"--, of this manual.
When the Mode Select Pins of the MCU are configured so as to disable the DRAM controller, the DRAM controller registers will become inaccessible as well.

DRAM initialization by software

DRAM must be initialized when the power is first applied with a sequence similar to the following.

1. Wait at least 200 µs after the end of the reset sequence before enabling DRAM access by writing a nonzero bus width to the DBWC register.

   This interval is actually only necessary when the power is first applied, but taking this approach simplifies the program.

2. Specify the DRAM access timing parameters in clock cycles in the DRPC register.

3. Write 0x00000004 to the DCMD register to precharge the DRAM (all banks for SDRAM).

4. Write 0x00000005 to the DCMD register eight times to produce CAS before RAS (CBR) refresh operations.

5. Specify the DRAM type (SDRAM or EDO DRAM) and the column length for address multiplexing in the DRMC register. For SDRAM, also specify the SDRAM precharge latency and whether to automatically shift SDRAM to power down mode.

6. For SDRAM, specify the SDRAM CAS latency in the SDMD register.

7. Specify the DRAM refresh periods in the RFSH0 and RFSH1 registers.

8. Specify the number of idle cycles to wait before shifting DRAM to power down mode in the RDWC register.

9. Write DRAM commands to the DCMD register: precharge, CBR refresh, start/end self refresh, etc.

Address Outputs and Bus Width

The following Table summarizes the relationships between address outputs (XA[*]) and the DRAM address pins (A[*]).

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

Notes

Ha stands for host address, the address used by the program.

1. The BWDRAM bit in the DBWC register specifies the data bus width and thus the mapping of the XA[*] signals to the DRAM address A[*] pins regardless of the column length. SDRAM cannot use an 8-bit bus.
2. The AMUX bits in the DRMC register specify the column length, the dividing line (8 to 10 bits) for address multiplexing of the row and column address outputs to the XA pins. DRAM cannot use column lengths of 11 and higher.
3. This bit represents a column address bit during EDO DRAM column output. During SDRAM operation, however, it specifies auto precharge, going to "0" during column output to disable auto precharge during READ and WRITE commands.
4. For SDRAM ACT, READ, and WRITE commands, the XA[18:12] outputs have the same value, the bank number, during both row and column output.

Distributed CAS before RAS (CBR) Refresh

The DRAM controller provides a facility for automatically performing distributed CBR refresh operations at regular intervals.

- Specify the distributed refresh periods in the RFSH0 and RFSH1 registers.
- Start and stop operation by changing the RFRSH bit in the DRMC register.

Note that manual refresh operation is available at any time by writing the CBR refresh command to the DCMD register.

Self Refresh

SDRAM self refresh operation is used, for example, as part of power management.
Start and stop operation by writing the corresponding commands to the DRCMD bits in the DCMD register: start SDRAM self refresh and end SDRAM self refresh. The start command stops the SDRAM clock, fixing the SDCLK output at High level; the stop command restarts output.
Operation is not guaranteed if the program writes any other command to the DCMD register between the start and end commands.
During Self Refresh, changes to SDMD register or other SDRAM settings will have unpredictable results.

SDRAM Commands

The DRAM controller provides the following SDRAM command outputs.

Command	Abbreviation	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	OPECODE		
CBR (auto) refresh	REFH	H	H	L	L	L	H	H	X	X	X
Start self refresh	SELF	H	L	L	L	L	H	H	X	X	X
End self refresh	SREX	L	H	L	H	H	H	H	X	X	X
		L	H	H	H	H	H	H	X	X	X
Precharge all banks	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
Deselect device	DESL	H	H	H	H	H	X	H	X	X	X
Power down *1	–	X	L	H	H	H	X	H	X	X	X
Continue self refresh *1	–	L	L	L	L	L	X	H	X	X	X

H: High level, L: Low level, X: Don't care level High or Low

BA: Bank number, RA: Row address, CA: Column address

Notes

1. These represent pseudo-commands defined for purposes of explication.
2. Unnecessary byte lanes are all at High level (masked state).

SDRAM Power Down Mode

The DRAM controller drives the clock enable signal (CKE) output at Low level to shift SDRAM to power down mode, conserving power, when the number of idle cycles specified in the RDWC register have elapsed and there are no DRAM access requests--that is, no DRAM space access requests (or error responses either)[*1] no DRAM register space access requests, and no distributed CBR refresh requests.

The DRAM controllers switches back if any of the above conditions are not met.

Switching back carries with it a penalty: an additional memory access overhead of three clock cycles.

Signal outputs in power down mode
- SDRAM clock signal (SDCLK) output: Stopped (fixed at Low level)
- Clock enable signal (CKE) output: Disabled (fixed at Low level)

Notes

1. After a power on reset, do not shift SDRAM to power down mode until the SDRAM initialization sequence is complete.
2. EDO DRAM does not provide an equivalent to SDRAM power down mode.

10.3.5 Access Timing Parameters for DRAM

The following Table lists DRAM device speeds compatible with various operating frequencies.
Figuring out the necessary DRAMSPEC bit settings in the DRPC register is left as an exercise for the reader.

Operating Frequency (MHz)	SDRAM					EDO-DRAM (Trac [ns])				
	PC133	PC125	PC100	PC33	PC66	50	60	70	80	100
33.3	O	O	O	O	O	O	O	O	O	O
30.0	O	O	O	O	O	O	O	O	O	O
16.7	O	O	O	O	O	O	O	O	O	O
15.0	O	O	O	O	O	O	O	O	O	O
8.4	O	O	O	O	O	O	O	O	O	O
7.5	O	O	O	O	O	O	O	O	O	O

The DRAMSPEC bits in the DRPC register specify a combination of access timing parameters (tRCD, tRAS, tRP, etc.) as shown in the table under the register description. There is no way to specify these parameters individually.

Minimum Operating Frequencies

SDRAM: 2.56 MHz
EDO DRAM: 6.4 MHz

SDRAM Access Timing Parameters and Operating Frequency

SDRAM Type (PCxx)	Access Parameters (ns)				Access Parameters (clock cycles) at 16.7 MHz				Access Parameters (clock cycles) at 33.3 MHz			
	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
133	20	45	20	10	1	2	1	1	1	2	1	1
133	20	45	20	15	1	2	1	1	1	2	1	1
125	20	48	20	8	1	2	1	1	1	2	1	1
125	20	48	20	10	1	2	1	1	1	2	1	1
125	20	50	30	8	1	2	1	1	1	2	1	1
125	24	48	24	10	1	2	1	1	1	2	1	1
100	20	50	20	10	1	2	1	1	1	2	1	1
100	20	50	20	15	1	2	1	1	1	2	1	1
100	20	50	20	20	1	2	1	1	1	2	1	1
100	30	50	30	15	1	2	1	1	1	2	1	1
100	30	60	30	10	1	2	1	1	1	2	1	1
100	30	60	30	15	1	2	1	1	1	2	1	1
83	35	70	45	24	1	2	1	1	2	3	2	1
66	30	60	30	15	1	2	1	1	1	2	1	1
66	30	70	30	15	1	2	1	1	1	3	1	1

Note: The DRAM controller assumes tRC = tRAS + tRP, so also check tRC if the SDRAM specifies tRC > tRAS + tRP.

EDO DRAM Access Timing Parameters and Operating Frequency

tRAC (ns)	Access Parameters (ns)				Access Parameters (clock cycles) at 16.7 MHz				Access Parameters (clock cycles) at 33.3 MHz			
	tRAH tCAS	tRCD	tCAC tOEZ	tRP	tRAH tCAS	tRCD	tCAC tOEZ	tRP	tRAH tCAS	tRCD	tCAC tOEZ	tRP
50	7	37	13	30	1	2	1	1	1	2	1	1
50	8	35	15	30	1	2	1	1	1	2	1	1
50	8	37	13	30	1	2	1	1	1	2	1	1
50	10	36	20	40	1	2	1	1	1	2	1	2
50	15	45	15	40	1	2	1	1	1	2	1	2
60	10	40	20	40	1	2	1	1	1	2	1	2
60	10	45	15	40	1	2	1	1	1	2	1	2
60	20	50	20	50	1	2	1	1	1	2	1	2
70	10	50	20	40	1	2	1	1	1	2	1	2
70	10	50	20	50	1	2	1	1	1	2	1	2
70	13	50	20	50	1	2	1	1	1	2	1	2
70	20	60	20	60	1	2	1	1	1	2	1	2
80	10	60	15	60	1	2	1	1	1	2	1	2
100	25	75	25	70	1	2	1	2	1	3	1	3

10.3.6 Read off time Control

The External Memory Controller automatically inserts, as necessary, read off time to avoid collisions between XD data from successive accesses to external ROM, SRAM, I/O or DRAM.

The following Table summarizes the insertion read off time for all combinations of access to external ROM, SRAM, I/O or DRAM.

Current Cycle			Following Cycle											
			ARM				DMA		ARM & DMA					
			ROM		RAM		RAM		IO		SDRAM		EDO-DRAM	
			R	W	R	W	R	W	R	W	R	W	R	W
ARM	ROM	Read		O	O	O	O	O	O	O	O	O	O	O
		Write												
	RAM	Read	O	O		O	O	O	O	O	O	O	O	O
		Write												
	IO	Read	O	O	O	O	O	O		O	O	O	O	O
		Write												
DMA	RAM	Read	O	O	O	O		O	O	O	O	O	O	O
		Write												
	IO	Read	O	O	O	O	O	O		O	O	O	O	O
		Write												
ARM & DMA	SDRAM	Read											-	-
		Write											-	-
	EDO-DRAM	Read	O	O	O	O	O	O	O	O	-	-	O	O
		Write									-	-		

Note: The read off time for ROM, RAM, IO is specified by ROMAC RAMAC IO0AC IO1AC registers. And that time for EDO-DRAM is specified by tOEZ of DRAMSPEC register. There is no specified Read off time for SDRAM.

10.4 Access Timing

10.4.1 Accessing External Devices

External ROM/RAM Access

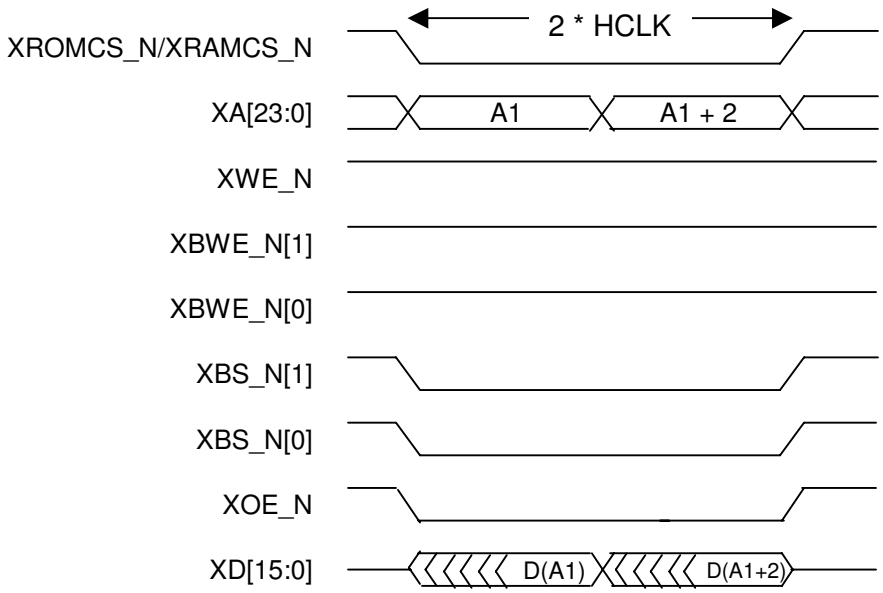


Figure 10.1 External ROM/RAM, Word Read, OE/WE Pulse Width 1 (Minimum Setting)

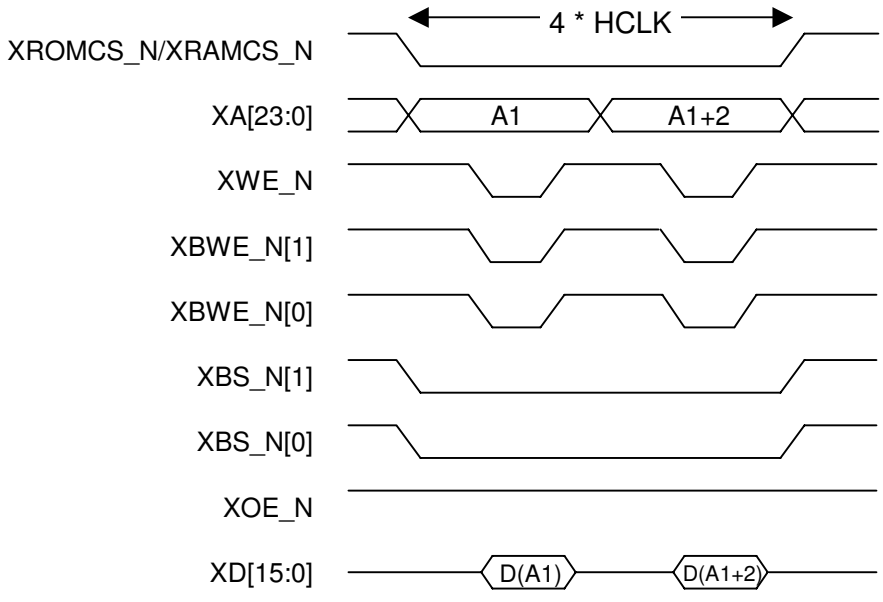


Figure 10.2 External ROM/RAM, Word Write, OE/WE Pulse Width 1 (Minimum Setting)

External I/O Bank Access

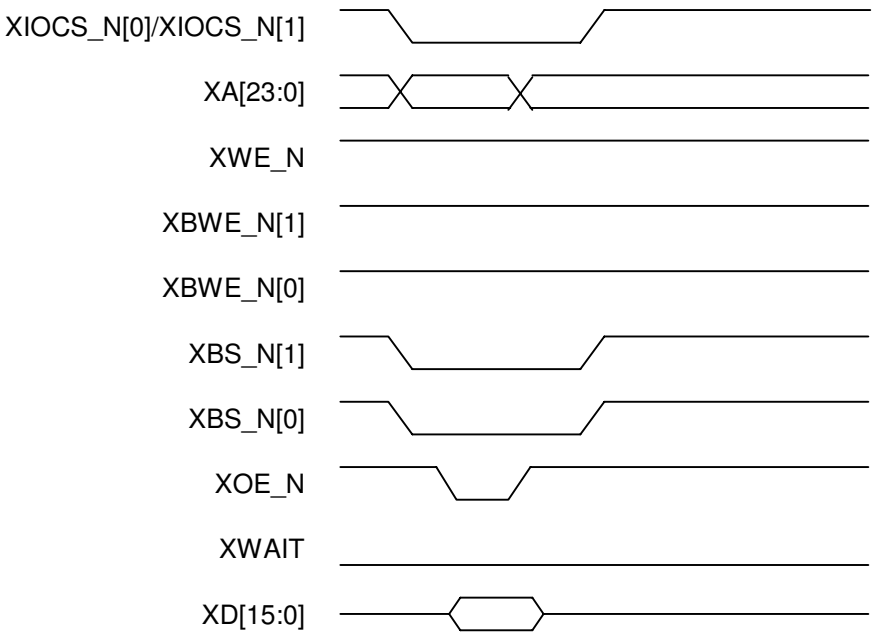


Figure 10.3 External I/O Bank, 16-Bit External Bus, Halfword Read, OE/WE Pulse Width 1 (Minimum Setting) without XWAIT Memory Wait Cycles

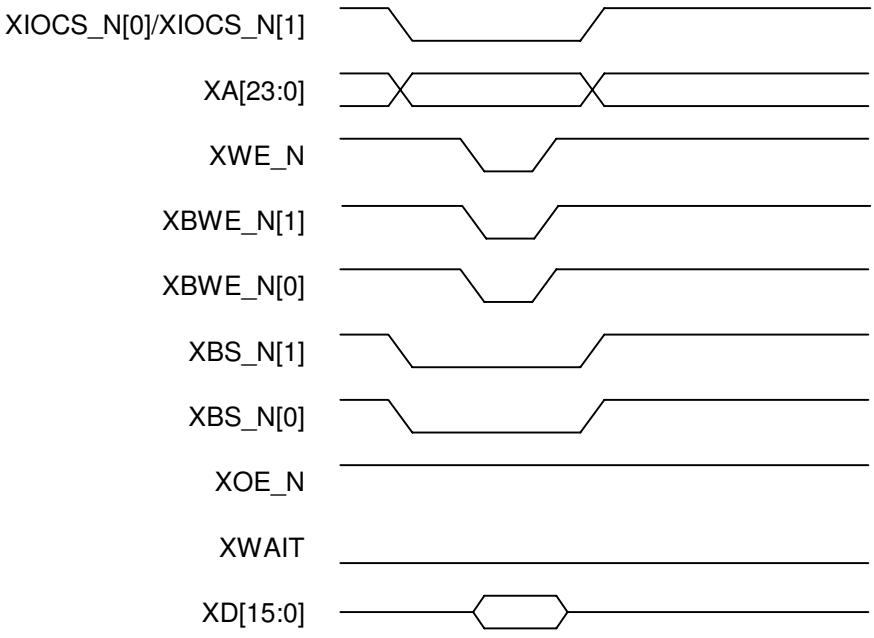


Figure 10.4 External I/O Bank, 16-Bit External Bus, Halfword Write, OE/WE Pulse Width 1 (Minimum Setting) without XWAIT Memory Wait Cycles

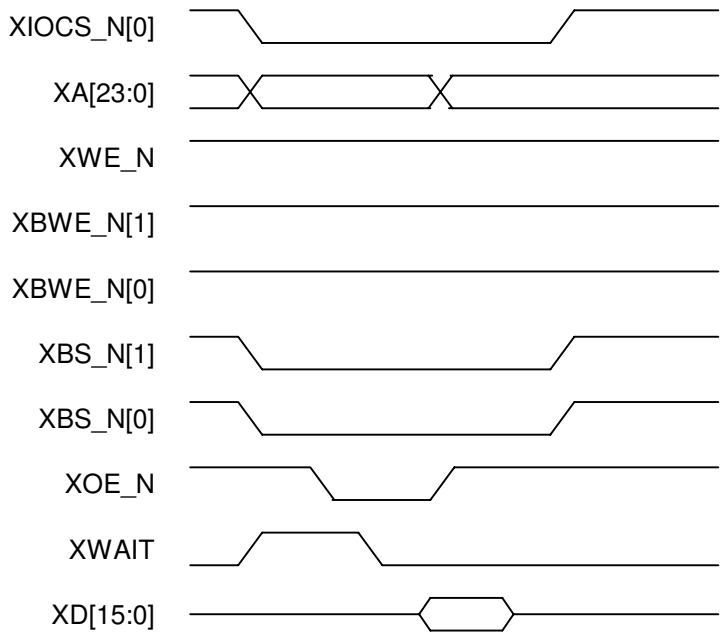


Figure 10.5 External I/O Bank, 16-Bit External Bus, Halfword Read, OE/WE Pulse Width 1 (Minimum Setting) with XWAIT Memory Wait Cycles

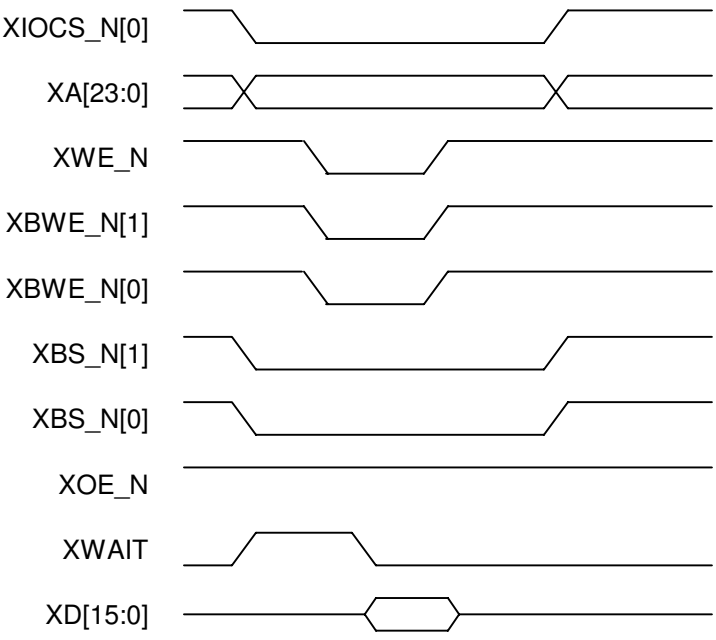


Figure 10.6 External I/O Bank, 16-Bit External Bus, Halfword Write, OE/WE Pulse Width 1 (Minimum Setting) with XWAIT Memory Wait Cycles

EDO DRAM Access

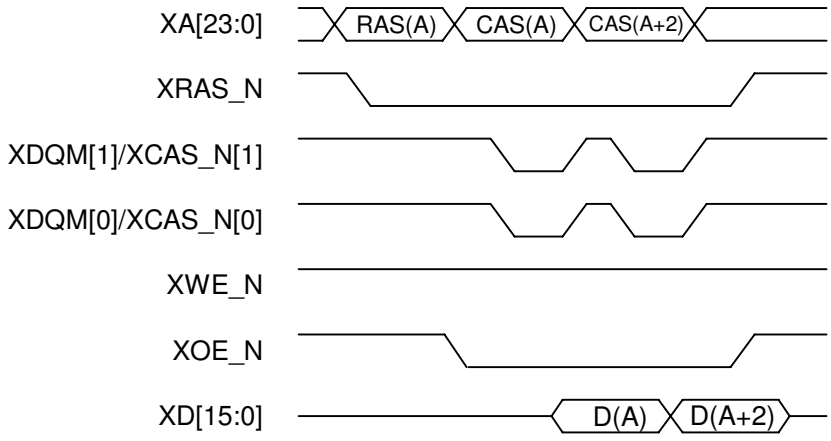


Figure 10.7 EDO DRAM, Word Read

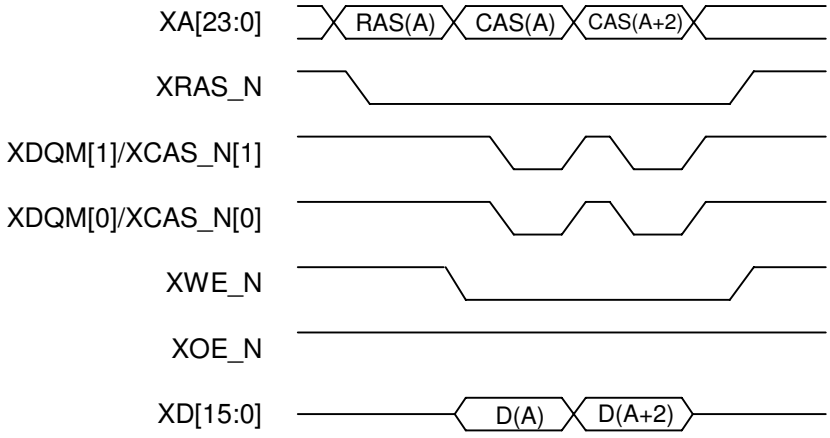


Figure 10.8 EDO DRAM, Word Write

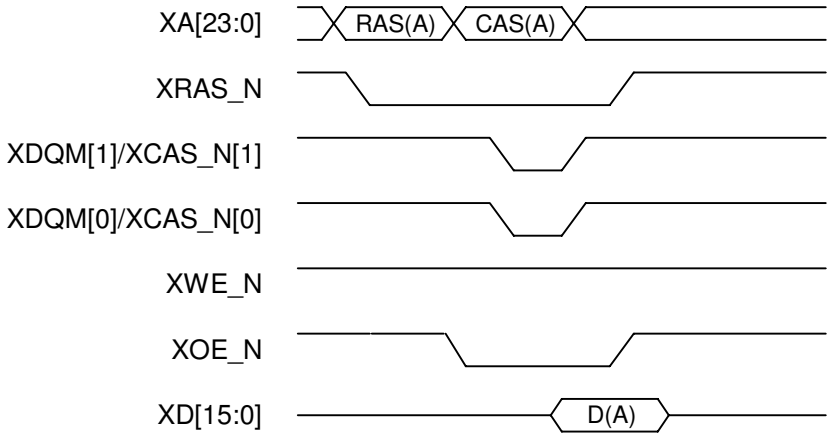


Figure 10.9 EDO DRAM, Halfword Read

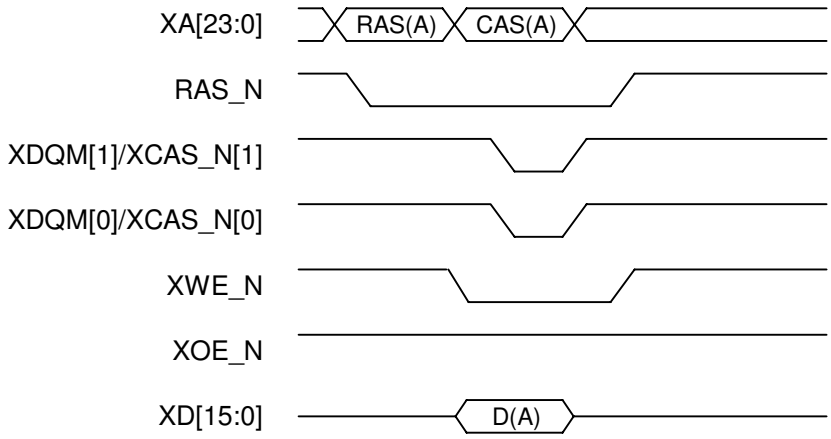


Figure 10.10 EDO DRAM, Halfword Write

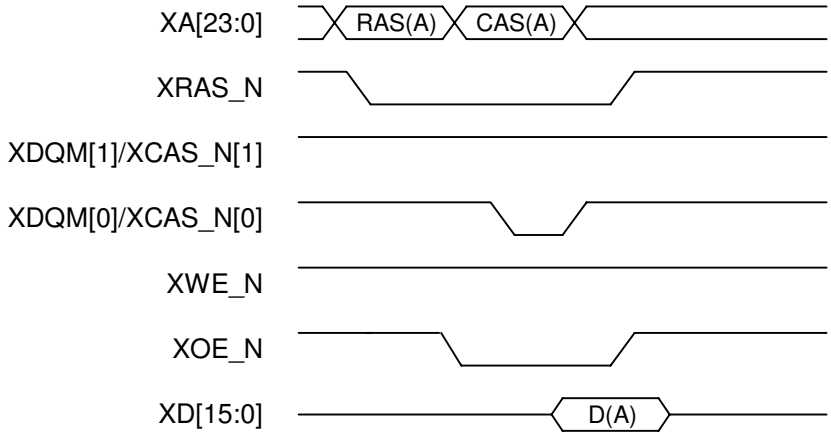


Figure 10.11 EDO DRAM, Byte (LSB) Read

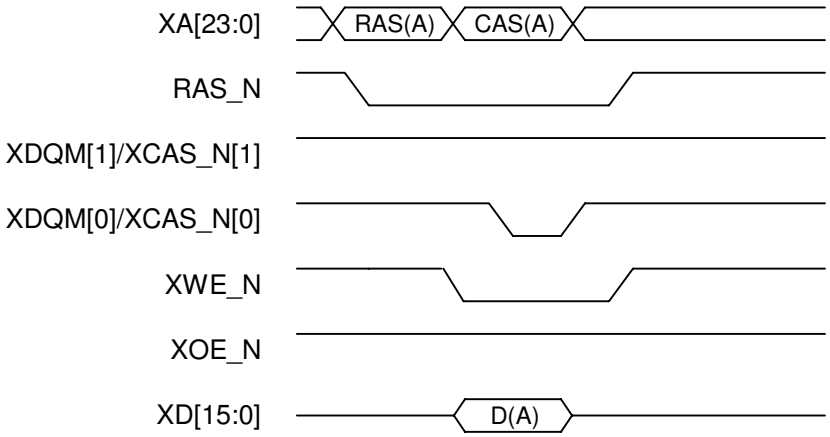


Figure 10.12 EDO DRAM, Byte (LSB) Write

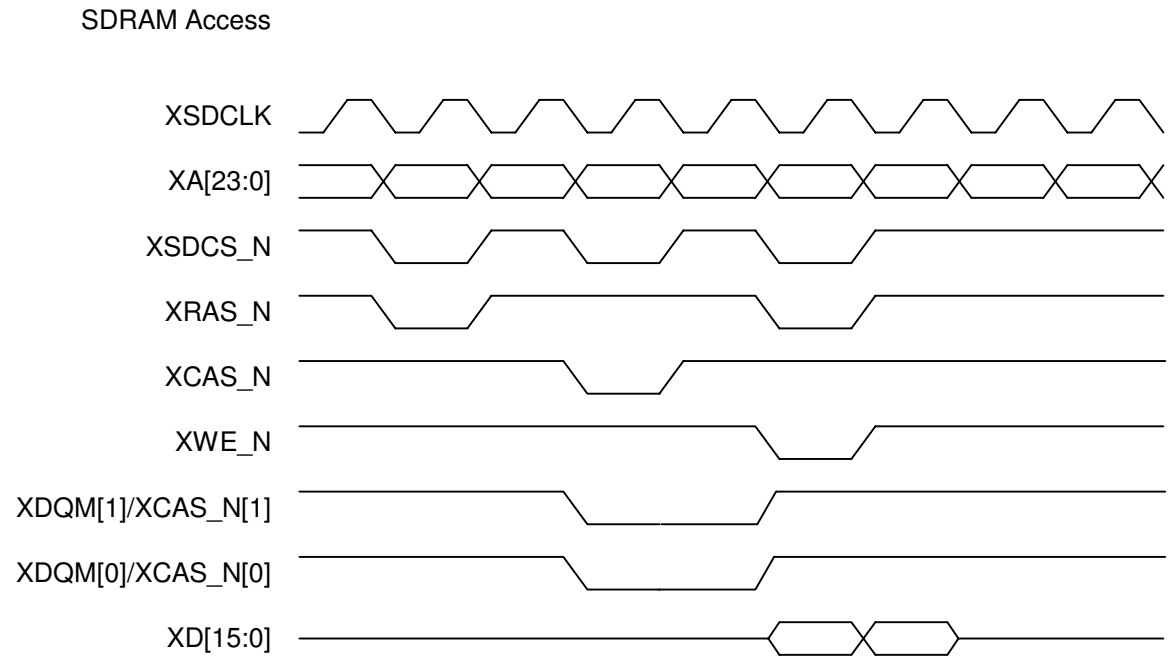


Figure 10.13 SDRAM, Word Read

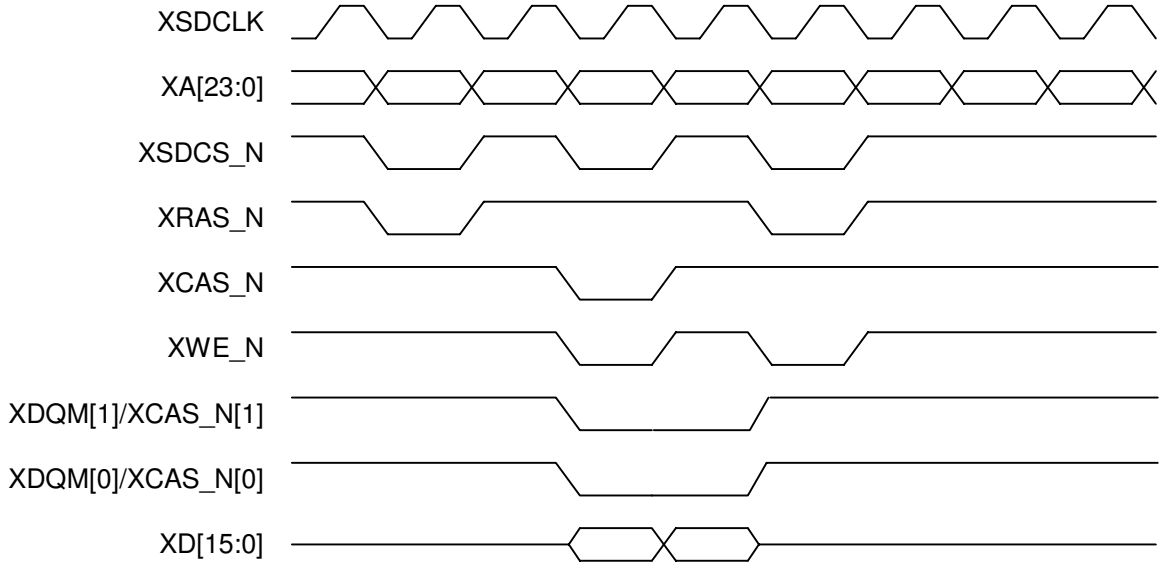


Figure 10.14 SDRAM, Word Write

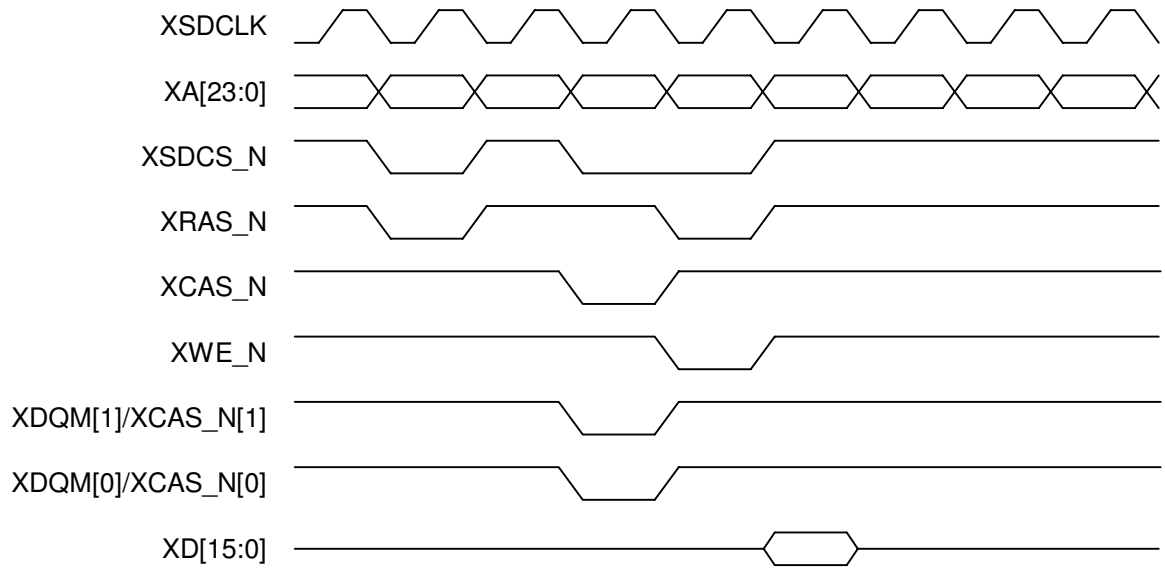


Figure 10.15 SDRAM, Halfword Read

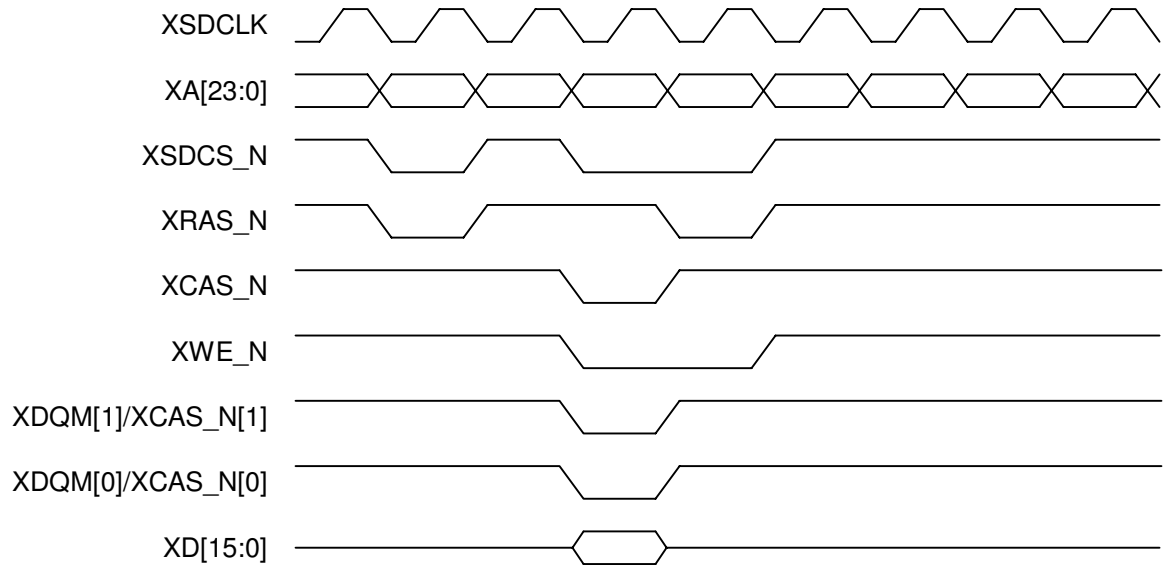


Figure 10.16 SDRAM, Halfword Write

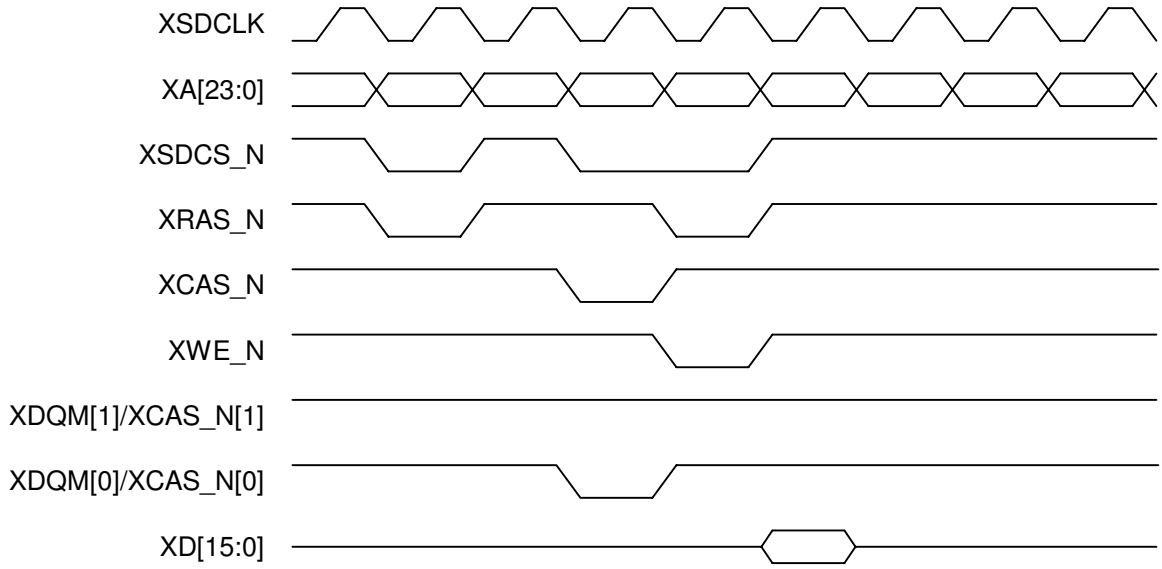


Figure 10.17 SDRAM, Byte (LSB) Read

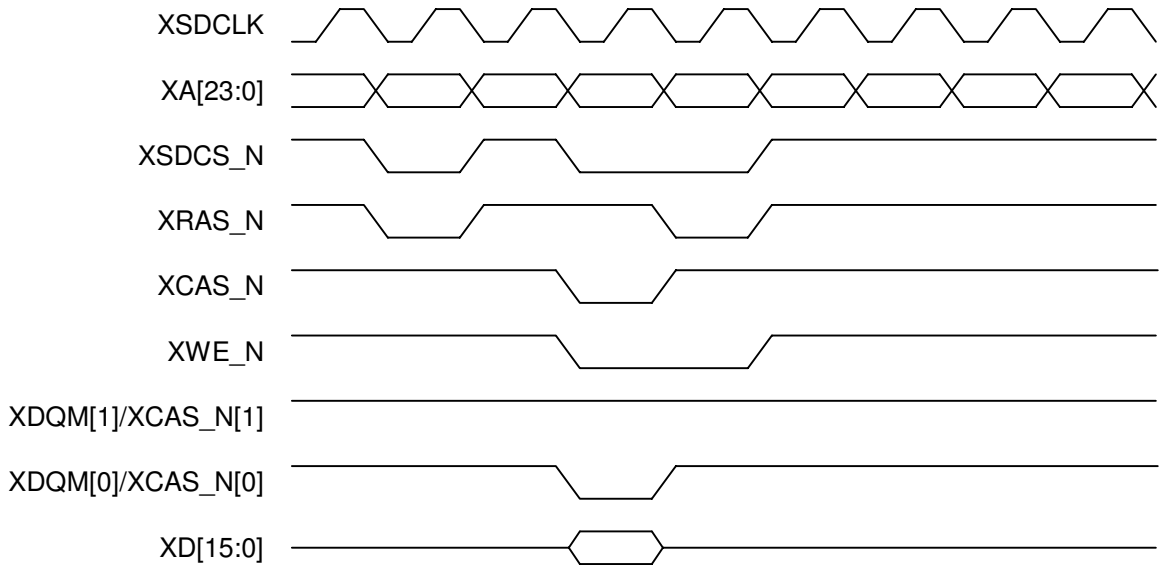


Figure 10.18 SDRAM, Byte (LSB) Write

10.5 DRAM Power Management

Using STANDBY or HALT mode requires program manipulation of the DRAM controller. For further details, see Chapter 7 "Power Management."

10.6 Sample External Memory Connections

The following pages give connection examples for each device type.
Devices differ as to signal names and other points, so refer also to the data sheet for the actual intended device.
As the number of devices increases, insert buffer elements and other components to ensure adequate drive power, load capacity, etc.
We recommend pull-up resistors on the XD pins.

Pin Name	I/O	function	ROM	SRAM	IO (Bank0)	IO (Bank1)	SDRAM	EDO-DRAM
XA[23:19]	O	External address bus	O *3	O *3	O *3	O *3		
XA[18:0]	O	External address bus	O	O	O	O	O	O
XD[15:0]	I/O	External data bus	O	O	O	O	O	O
XROMCS_N	O	External ROM chip select	O					
XRAMCS_N	O	External RAM chip select		O				
XIOCS_N[0]	O	I/O bank 0 chip select			O			
XIOCS_N[1]	O	I/O bank 1 chip select				O		
XOE_N	O	Output enable	O	O	O	O		O
XWE_N	O	Write enable	O *1	O *1	O *1	O *1	O	O
XBS_N[1:0]	O	External bus byte select	O *1	O *1	O *1	O *1		
XBWE_N[0]	O	Write enable (LSB)	O *2	O *2	O *2	O *2		
XBWE_N[1]	O	Write enable (MSB)	O *2	O *2	O *2	O *2		
XSDCS_N	O	SDRAM chip select					O	
XSDCLK	O	SDRAM clock					O	
XSDCKE	O	SDRAM clock enable					O	
XCAS_N	O	SDRAM column address strobe					O	
XRAS_N	O	Row address strobe					O	O
XDQM[1]/ XCAS_N[1]	O	Data I/O mask for SDRAM or MSB column address strobe for EDO DRAM					O	O
XDQM[0]/ XCAS_N[0]	O	Data I/O mask for SDRAM or LSB column address strobe for EDO DRAM					O	O
XWAIT	I	Memory wait indicator			O			
XWR	O	External bus data transfer direction			O *4	O *4		

Notes

*1. For devices using byte select signals for byte access

*2. For devices using byte write enable signals for byte access

*3. There is no XA[23:19] output during the boot sequence, so user application systems with devices needing these outputs must provide an external mechanism for fixing these pins at Low level until the secondary (XA) function takes effect.

*4. Use as necessary.

10.6.1 Connecting ROM

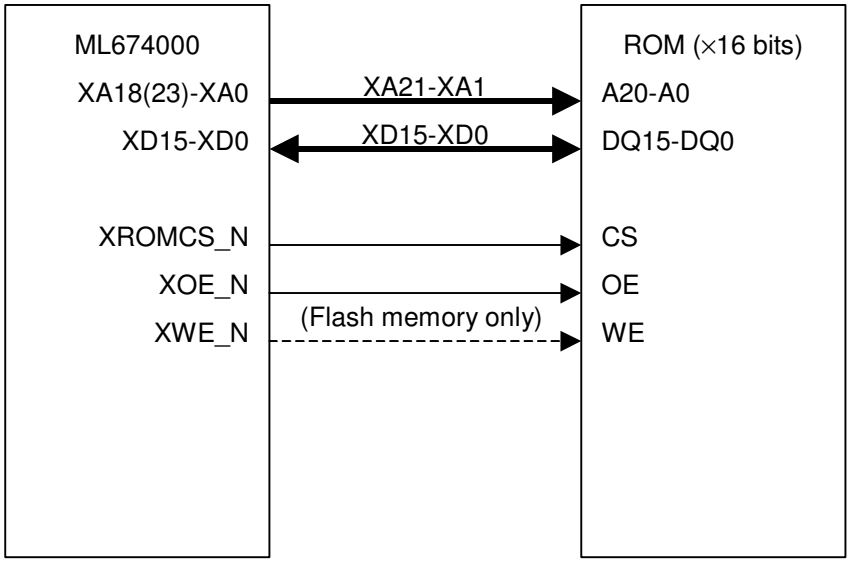


Figure 10.19 Connecting 2M × 16-Bit ROM (4 MB)

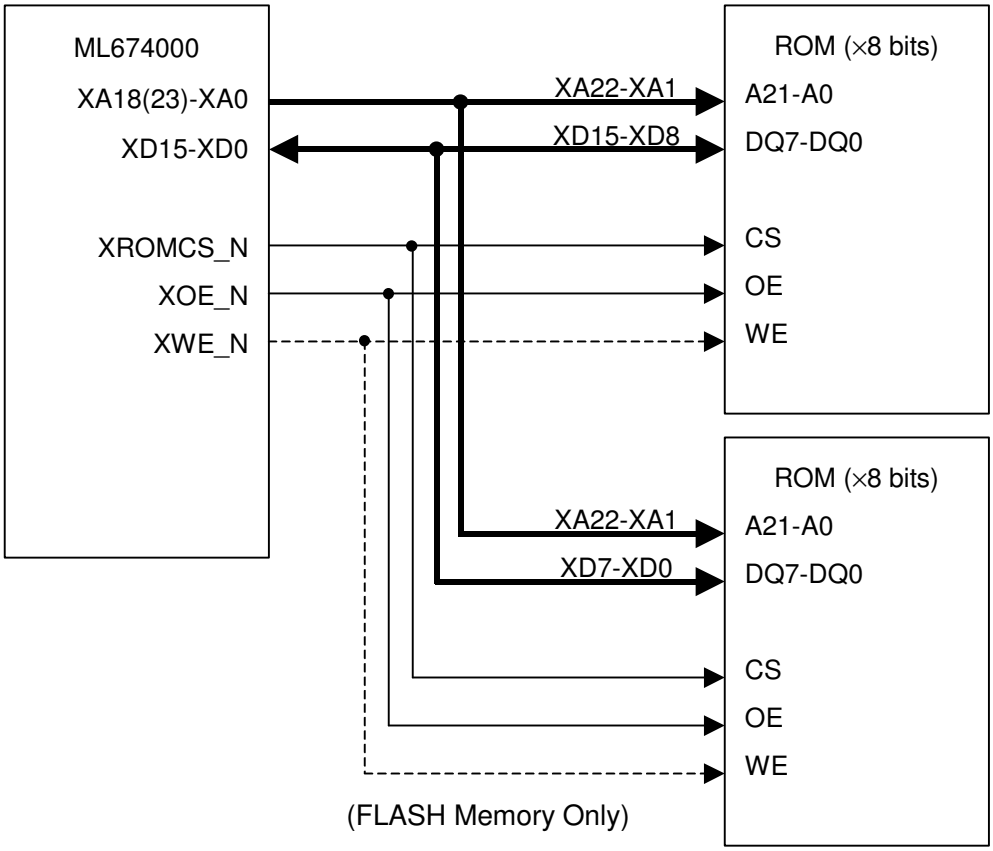


Figure 10.20 Connecting Two 4M × 8-Bit ROMs (4 MB)

10.6.2 Connecting SRAM

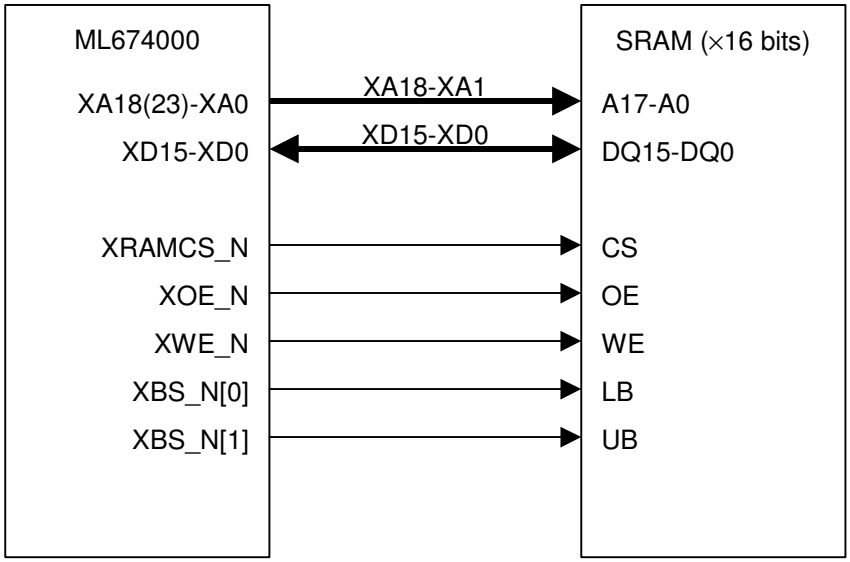


Figure 10.21 Connecting 256K × 16-Bit SRAM (512 KB) Using Byte Select Signals

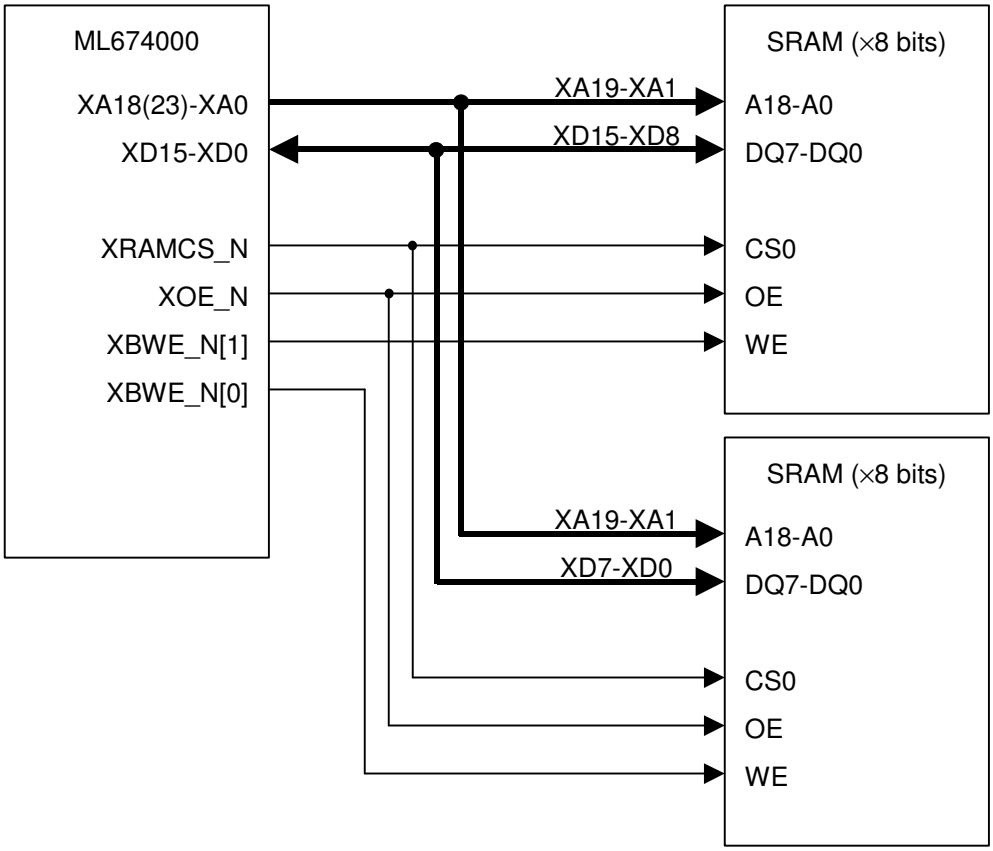


Figure 10.22 Connecting Two 512K × 8-Bit SRAMs (512 KB)

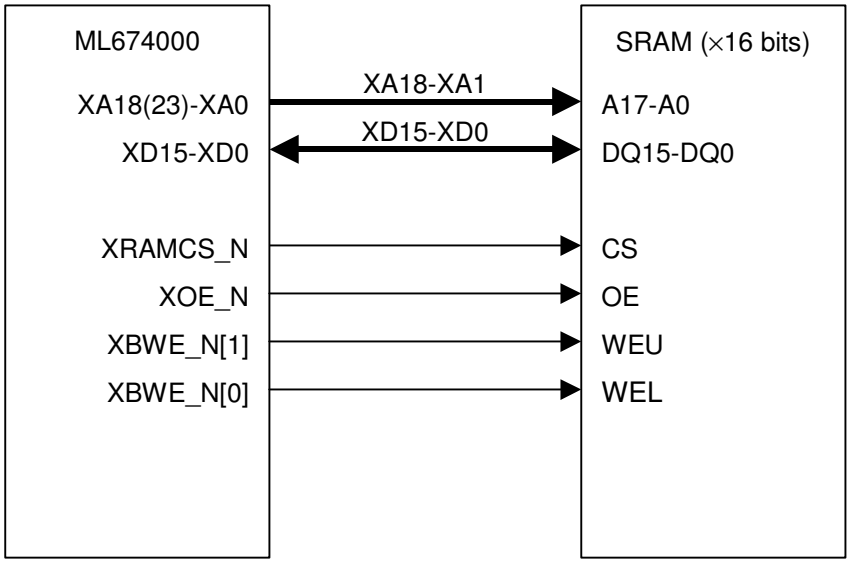


Figure 10.23 Connecting 256K × 16-Bit SRAM (512 KB) Using Byte Write Enable Signals

10.6.3 Connecting EDO DRAM

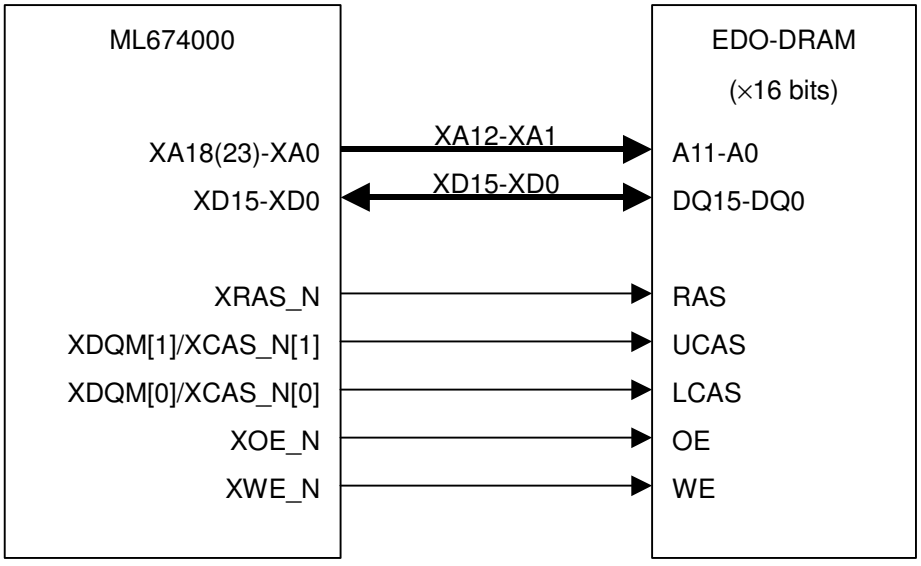


Figure 10.24 Connecting 1M × 16-Bit EDO DRAM (2 MB)

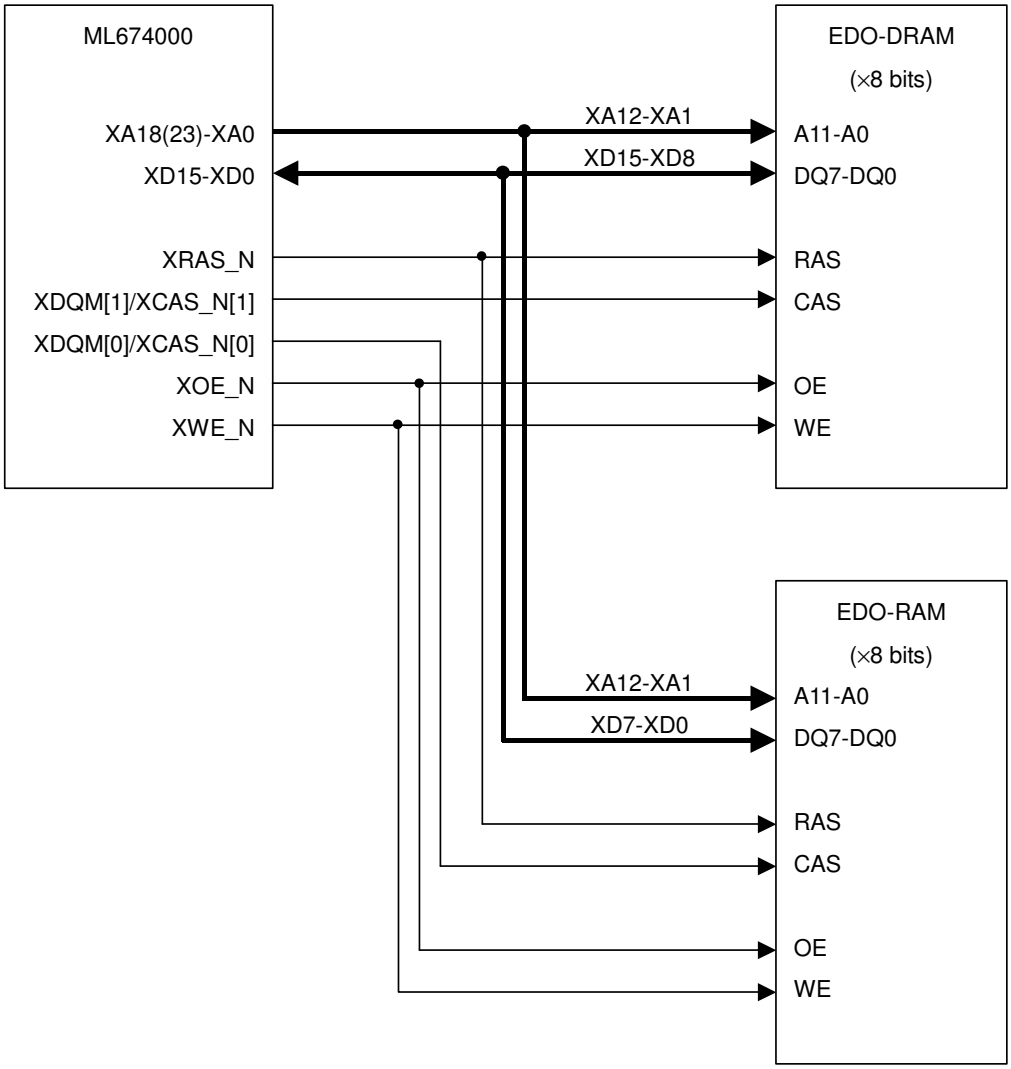


Figure 10.25 Connecting Two 2M × 8-Bit EDO DRAMs (2 MB)

10.6.4 Connecting SDRAM

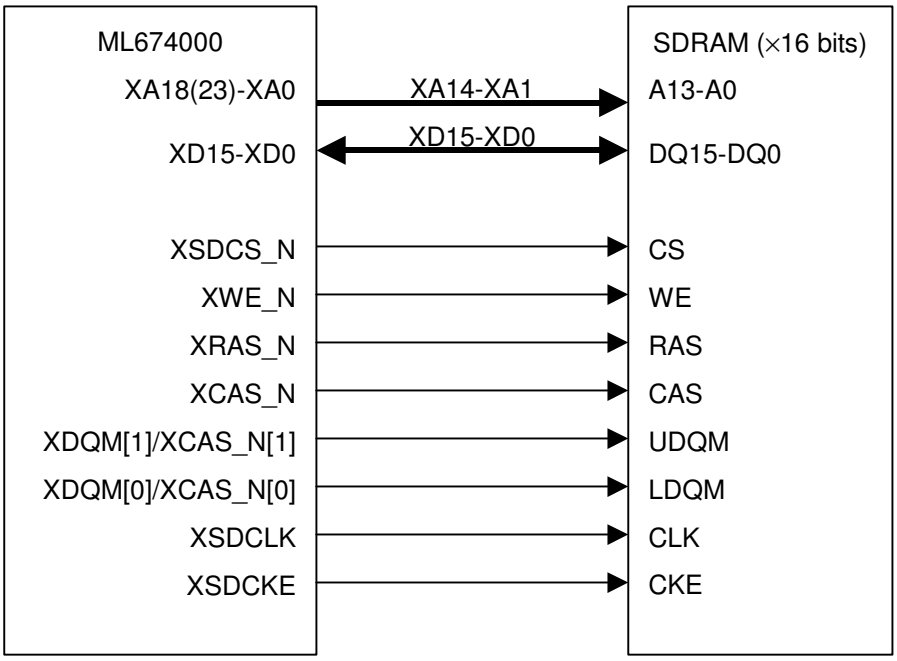


Figure 10.26 Connecting 1M × 16-Bit SDRAM (8 MB)

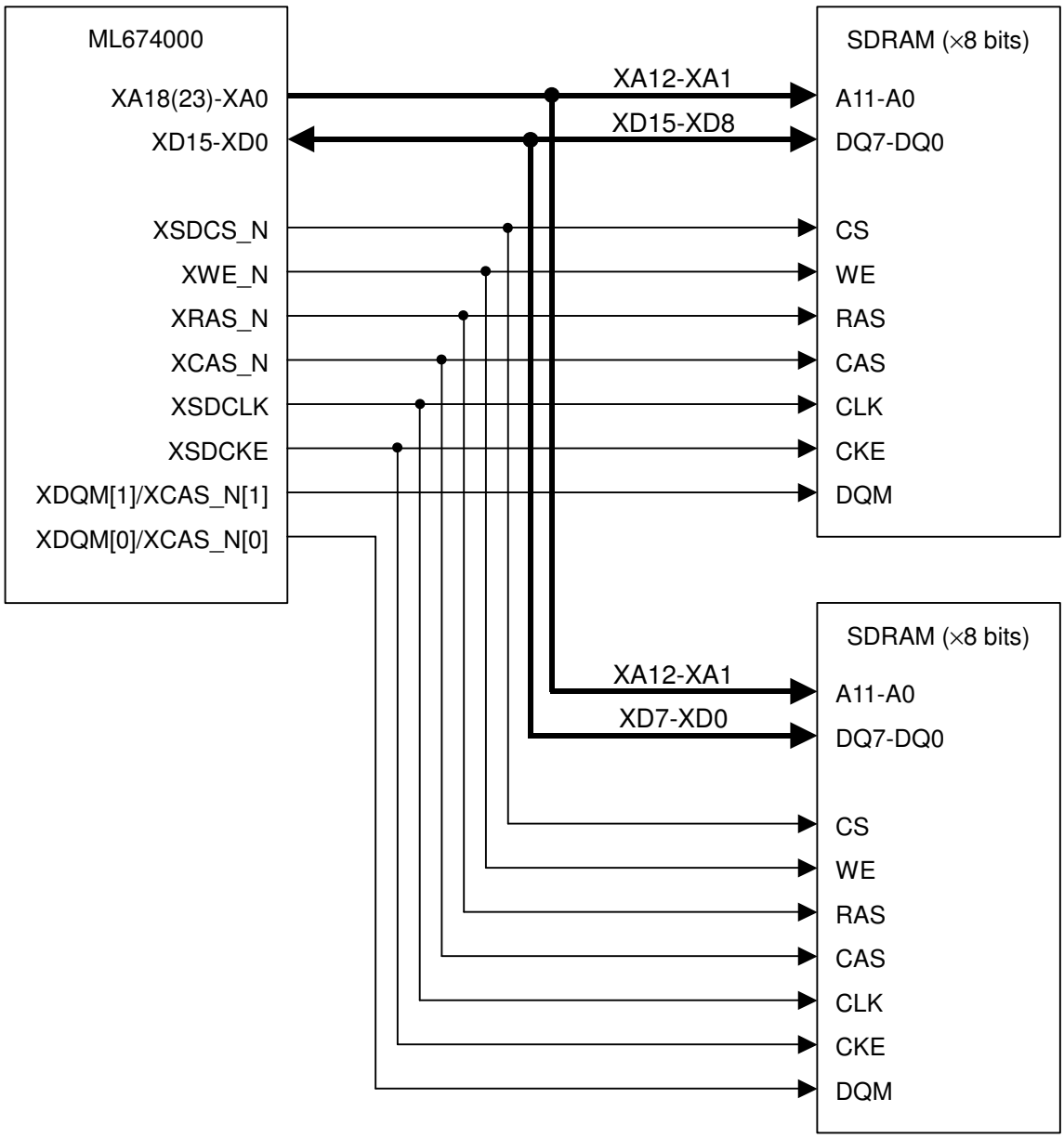


Figure 10.27 Connecting Two × 1M × 8-Bit SDRAMs (4 MB)

Chapter 11

Direct Memory Access Controller (DMAC)

Chapter 11 Direct Memory Access Controller (DMAC)

11.1 Overview

This 2-channel direct memory access controller (DMAC) transfers data directly between memory and memory, between an I/O device and memory, and between I/O devices, reducing the CPU load and thus boosting overall LSI operational efficiency.

Features

- Number of channels: 2
- Channel Priority
  - Fixed mode: Channel priority never changes (channel 0 > channel 1).
  - Round robin mode: Last channel used has lower priority.
- Maximum number of transfers: 65,536
- Transfer sizes: byte, halfword, and word (8, 16, and 32 bits)
- Dual address access: A read from the transfer source is independent from a write to the transfer destination.
- Bus Access
  - Cycle stealing mode: The DMA controller surrenders bus access after each individual DMA transfer.
  - Burst mode: The DMA controller does not surrender bus access until the specified number of transfers are complete.
- DMA Transfer Requests
  - Software request mode: The DMA controller generates transfer requests until the specified number of transfers are complete.
  - External request mode: The DMA controller generates transfer requests at rising edges in the external request signals (DREQ0 or DREQ1) for the channel.
- Interrupt requests: The DMA controller sends interrupt requests to the CPU when the specified number of transfers are complete or there is an error (nonexistent address specified for transfer destination or transfer source). There are separate interrupt request signals for each channel, and these can be masked by channel.

11.1.1 Components

Figure 11.1 shows DMA controller components; Figure 11.2, DMAC peripheral components.

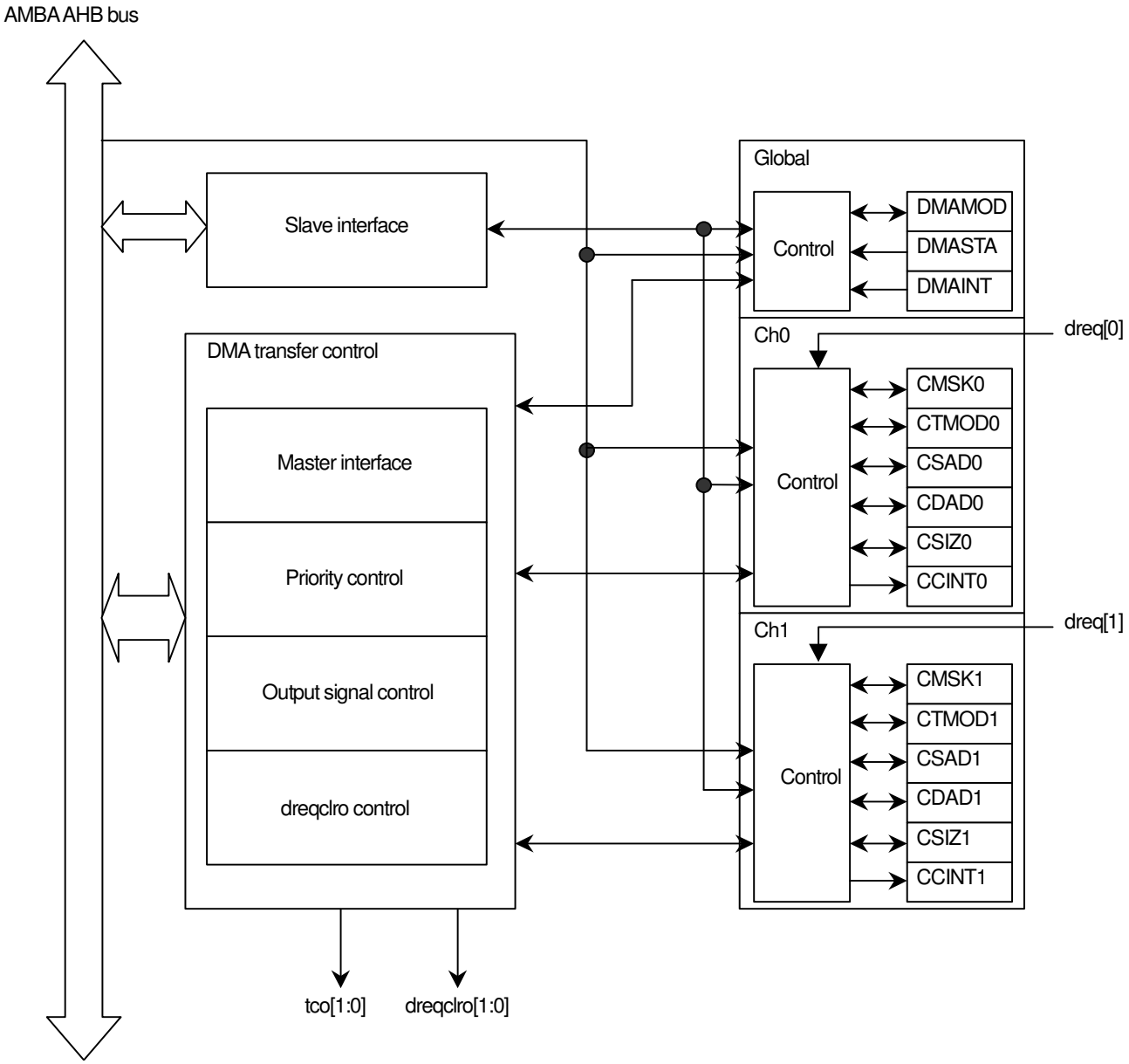


Figure 11.1 DMA Controller Block Diagram

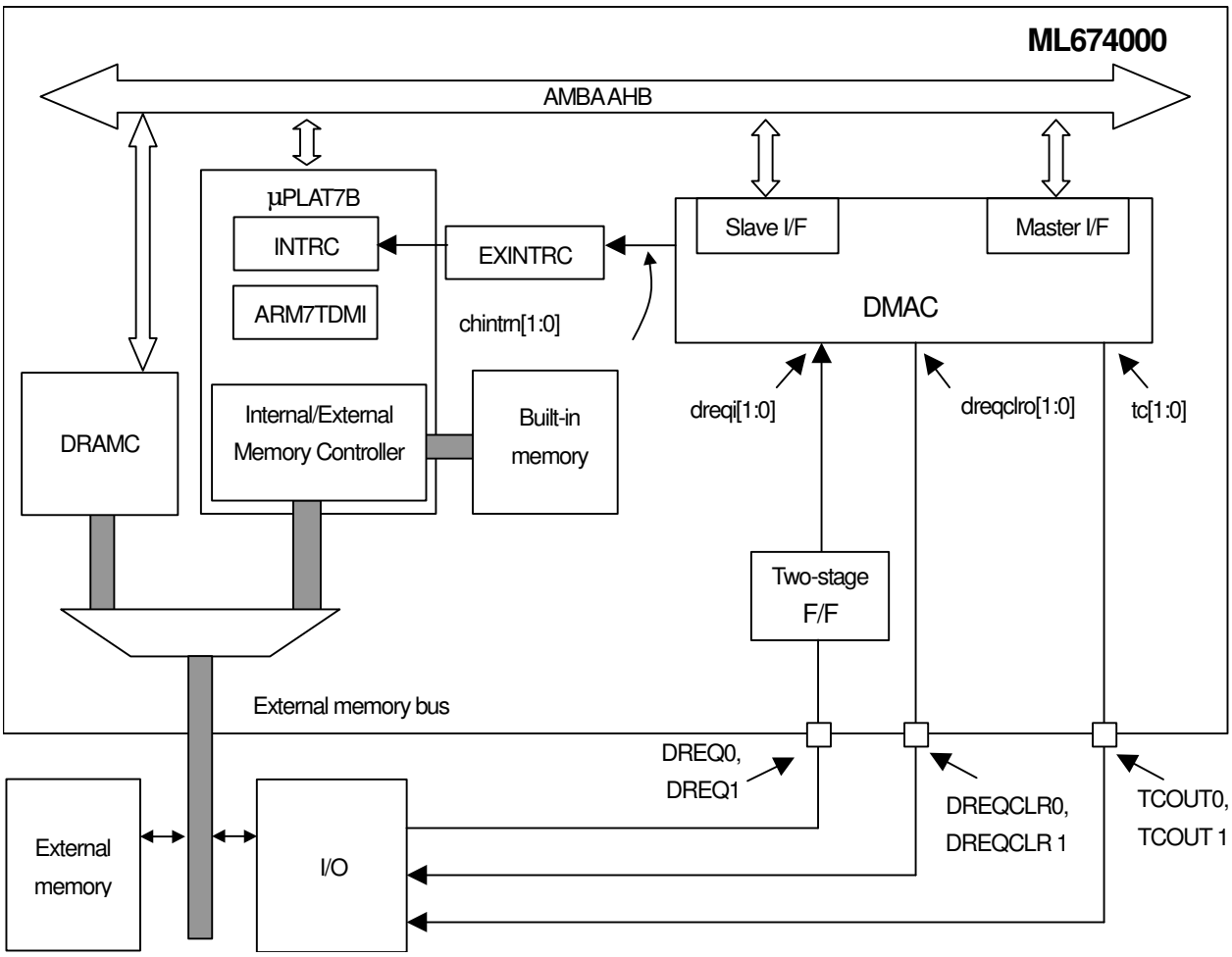


Figure 11.2 DMA Controller Peripheral Block Diagram

AHB Interface

The DMA controller features a 32-bit connection to the AMBA AHB bus.
The DMA controller provides both slave and master interfaces.

- Slave interface: The program uses this to access DMA controller registers (both global and channel).
- Master interface: The DMA controller uses this for DMA transfers.

11.1.2 Pin List

Pin Name	I/O	Description
DREQ0	I	DMA request signal (channel 0)
DREQCLR0	O	DREQ clear request signal (channel 0)
DREQ1	I	DMA request signal (channel 1)
DREQCLR1	O	DREQ clear request signal (channel 1)
TCOUT0	O	DMA transfer complete signal (channel 0)
TCOUT1	O	DMA transfer complete signal (channel 1)

11.1.3 Register List

CH	Address [H]	Name	Abbreviation	R/W	Size	Initial Value [H]
Global	0x7BE00000	DMA mode register	DMAMOD	R/W	32	0x00000000
	0x7BE00004	DMA status register	DMASTA	R	32	0x00000000
	0x7BE00008	DMA transfer complete status register	DMAINT	R	32	0x00000000
Channel 0	0x7BE00100	DMA channel mask register	DMACMSK0	R/W	32	0x00000001
	0x7BE00104	DMA transfer mode register	DMACTMOD0	R/W	32	0x00000040
	0x7BE00108	DMA transfer source address register	DMACSAD0	R/W	32	0x00000000
	0x7BE0010C	DMA transfer destination address register	DMACDAD0	R/W	32	0x00000000
	0x7BE00110	DMA transfer count register	DMACSIZ0	R/W	32	0x00000000
	0x7BE00114	DMA transfer complete status clear register	DMACCINT0	W	32	—
Channel 1	0x7BE00200	DMA channel mask register	DMACMSK1	R/W	32	0x00000001
	0x7BE00204	DMA transfer mode register	DMACTMOD1	R/W	32	0x00000040
	0x7BE00208	DMA transfer source address register	DMACSAD1	R/W	32	0x00000000
	0x7BE0020C	DMA transfer destination address register	DMACDAD1	R/W	32	0x00000000
	0x7BE00210	DMA transfer count register	DMACSIZ1	R/W	32	0x00000000
	0x7BE00214	DMA transfer complete status clear register	DMACCINT1	W	32	—

Note: These registers require word (32-bit) access. Operation is not guaranteed for byte (8-bit) or halfword (16-bit) access.

11.2 Register Descriptions

11.2.1 DMA Mode Register (DMAMOD)

This register is for specifying DMA channel priority.
The program has read/write access to this register.
The contents after a reset are 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAMOD	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **PRI** (bit 0):
  This bit specifies DMA channel priority.

PRI	Description
0	Fixed (channel 0 > channel 1)
1	Round robin (Last channel used has lower priority)

11.2.2 DMA Status Register (DMASTA)

This register gives the transfer status for each DMA channel. A "1" in a bit indicates that the transfer count register (DMACSIZ) for the channel contains a nonzero value. The bit returns to "0" when the specified number of transfers are complete (normal termination) or an error condition forces an abnormal termination.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASTA	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	STA1	STA0
	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
– *: These bits are reserved for future expansion. They return "0" for reads.

Bit Descriptions

- **STA0** (bit 0):
  This bit gives the transfer status for DMA channel 0.

STA0	Description
0	Idle (no data to transfer)
1	Busy (transferring data)

- **STA1** (bit 1):
  This bit gives the transfer status for DMA channel 1.

STA1	Description
0	Idle (no data to transfer)
1	Busy (transferring data)

11.2.3 DMA Transfer Complete Status Register (DMAINT)

This register gives the source for an interrupt request indicating the end of a DMA transfer: the channel number, the termination reason (normal or error), and the cycle (read from transfer source or write to transfer destination). Writing anything to an interrupt clear register (DMACCINT0 or DMACCINT1) resets the IREQ, ISTA, and ISTP bits for the corresponding DMA channel to "0."
The program has only read access to this register.
The contents after a reset are 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAINT	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	ISTP1	ISTP0
After a 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
	– *	– *	– *	– *	– *	– *	ISTA1	ISTA0	– *	– *	– *	– *	– *	– *	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

Notes
– *: These bits are reserved for future expansion. They return "0" for reads.
These status bits remain "1" until the program explicitly resets them to "0."

Bit Descriptions

- **IREQ0** (bit 0):
  This bit gives the transfer complete status for DMA channel 0.

IREQ0	Description
0	No request. DMA transfer is not complete or has not started
1	Request pending. DMA transfer is complete

- **IREQ1** (bit 1):
  This bit gives the transfer complete status for DMA channel 1.

IREQ1	Description
0	No request. DMA transfer is not complete or has not started
1	Request pending. DMA transfer is complete

- **ISTA0** (bit 8):
  This bit gives the termination reason for DMA channel 0.

ISTA0	Description
0	Normal termination
1	Error

- **ISTA1** (bit 9):
  This bit gives the termination reason for DMA channel 1.

ISTA1	Description
0	Normal termination
1	Error

- **ISTP0** (bit 16):
  This bit gives the cycle in which the DMA channel 0 error occurred. It is only valid when ISTA0 = 1.

ISTP0	Description
0	Read from transfer source
1	Write to transfer destination

- **ISTP1** (bit 17):
  This bit gives the cycle in which the DMA channel 1 error occurred. It is only valid when ISTA1 = 1.

ISTP1	Description
0	Read from transfer source
1	Write to transfer destination

11.2.4 DMA Channel Mask Registers (DMACMSK0 and DMACMSK1)

These registers control masking, which blocks transfers on the corresponding channel.
The program has read/write access to these registers.
The contents after a reset are 0x00000001, masking all channels.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACMSK0 to 1	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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)
Access: R/W
Access size: 32 bits

Note
– *: These bits are reserved for future expansion. They return "0" for reads.

Bit Descriptions

- **MSK** (bit 0):
  This bit specifies the channel mask.

MSK	Description
0	Remove mask to enable or restart DMA channel operation
1	Mask (stop) DMA channel operation

11.2.5 DMA Transfer Mode Registers (DMACTMOD0 and DMACTMOD1)

These registers specify the DMA transfer mode for the corresponding DMA channel: request source, transfer size, device types, and whether there is an interrupt when the specified number of transfers are complete.
The program has read/write access to this register.
The contents after a reset are 0x00000040.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACTMOD0 to 1	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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)
Access: R/W
Access size: 32 bits

Notes
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored. Certain restrictions apply. For further details, see Section 8.3.6 "Important Usage Notes."

Bit Descriptions

- **ARQ** (bit 0):
  This bit specifies the transfer request source.

ARQ	Description
0	External input (DREQ)
1	Software request mode

- **TSIZ** (bits 2 and 1):
  This field specifies the transfer size.

TSIZ		Description
2	1	
0	0	Byte (8 bits)
0	1	Halfword (16 bits)
1	0	Word (32 bits)
1	1	Setting not allowed

- **SDP** (bit 3):
  This bit specifies the device type for the transfer source.

SDP	Description
0	Fixed address device. The DMA controller always reads from the same address.
1	Incremental address device. The DMA controller increments the address by the transfer size in bytes after each successful read from the transfer source.

- **DDP** (bit 4):
  This bit specifies the device type for the transfer destination.

DDP	Description
0	Fixed address device. The DMA controller always writes to the same address.
1	Incremental address device. The DMA controller increments the address by the transfer size in bytes after each successful write to the transfer destination.

- **BRQ** (bit 5):
  This bit specifies the bus request mode.

BRQ	Description
0	Burst mode. The DMA controller does not surrender bus access until the specified number of transfers are complete.
1	Cycle stealing mode. The DMA controller surrenders bus access after each individual DMA transfer.

- **IMK** (bit 6):
  This bit specifies the interrupt mask.

IMK	Description
0	Remove mask to enable interrupt requests
1	Mask (stop) interrupt requests

11.2.6 DMA Transfer Source Address Registers (DMACSAD0 and DMACSAD1)

These registers specify the transfer source address for the corresponding DMA channel. The DMA controller starts reading data for the DMA transfer from the address in this register.
The DMA controller increments the address by the transfer size in bytes after each successful read if the transfer source is an incremental address device. For a fixed address device, however, the address does not change.
The program has read/write access to these registers.
The contents after a reset are 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACSAD0 to 1	CSAD[31:16]															
After a 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)
Access: R/W
Access size: 32 bits

Note

– *: The DMA controller increments the source address (DMACSAD0 or DMACSAD1) by the transfer size in bytes if the corresponding device type so specifies. The hardware enforces alignment by internally ignoring the lowest address bits as appropriate for the transfer size, but reading an address register returns those bits exactly as written.

Example: Writing address of 0x10000011

Word (32-bit) transfer: The DMA controller internally forces the lowest 2 bits to "00" for an effective address of 0x10000000. Reading the register, however, returns 0x10000011.

Halfword (16-bit) transfer: The DMA controller internally forces the lowest bit to "0" for an effective address of 0x10000010. Reading the register, however, returns 0x10000011.

11.2.7 DMA Transfer Destination Address Registers (DMACDAD0 and DMACDAD1)

These registers specify the transfer destination address for the corresponding DMA channel. The DMA controller starts writing data for the DMA transfer to the address in this register.
The DMA controller increments the address by the transfer size in bytes after each successful write if the transfer destination is an incremental address device. For a fixed address device, however, the address does not change.
The program has read/write access to these registers.
The contents after a reset are 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACDAD0 to 1	CDAD[31:16]															
After a 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)
Access: R/W
Access size: 32 bits

Notes

– *: The DMA controller increments the destination address (DMACDAD0 or DMACDAD1) by the transfer size in bytes if the corresponding device type so specifies. The hardware enforces alignment by internally ignoring the lowest address bits as appropriate for the transfer size, but reading an address register returns those bits exactly as written.

Example: Writing address of 0x10000011

Word (32-bit) transfer:	The DMA controller internally forces the lowest 2 bits to "00" for an effective address of 0x10000000. Reading the register, however, returns 0x10000011.
Halfword (16-bit) transfer:	The DMA controller internally forces the lowest bit to "0" for an effective address of 0x10000010. Reading the register, however, returns 0x10000011.

11.2.8 DMA Transfer Count Registers (DMACSIZ0 and DMACSIZ1)

These registers specify the transfer count for the corresponding DMA channel. The DMA controller decrements this count after each successful read from the transfer source.
Note that this count represents the number of transfers, not the total size of the transfer. The maximum possible setting for this register is "0x00010000" (65,536).
The program has read/write access to this register.
The contents after a reset are 0x00000000.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMACSIZ0 to 1	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	CSIZ16
After a 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)
Access: R/W
Access size: 32 bits

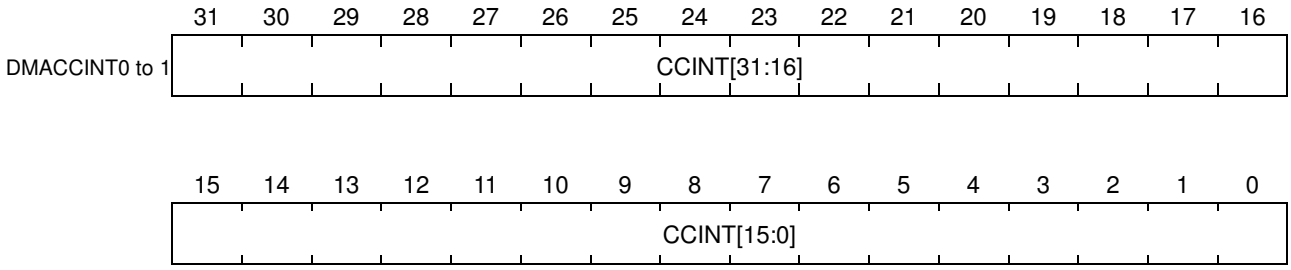
Notes

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

The maximum possible setting for this register is "0x00010000" (65,536). Operation is not guaranteed for higher values.

11.2.9 DMA Transfer Complete Status Clear Registers (DMACCINT0 and DMACCINT1)

These registers are for clearing the DMA transfer complete status for the corresponding DMA channel. Writing a 32-bit value to one resets the following bits in the corresponding DMAINT0 or DMAINT1 register: DMA transfer complete (IREQ0 or IREQ1), termination reason (ISTA0 or ISTA1), and abnormal termination cycle (ISTP0 or ISTP1).
The program has only write access to these registers.



Address: 0x7BE00114 (CH0), 0x7BE00214 (CH1)
Access: W
Access size: 32 bits

11.3 Operational Description

A transfer request starts DMA transfers. The DMA controller automatically stops when the specified number of transfers are complete. The DMA controller utilizes Dual Address Mode transfer. This type of transfer involves two memory or I/O cycles. The data being transferred is first read from a source address and subsequently written to a destination address in the next cycle. The device will initially signal a request to the DMA controller, e.g. by asserting the DREQ signal. In response, the DMA controller will grant the device or memory access to the bus by causing the OUTPUT_ENABLE or WRITE_ENABLE and CS strobes of the device or memory to be asserted thus giving the device or memory access to the bus.

11.3.1 DMA Transfer Modes

BRQ, bit 5 in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1), offers a choice of two DMA transfer modes for the corresponding DMA channel.

1. Cycle stealing mode: The DMA controller surrenders bus access after each individual DMA transfer and waits for another transfer request before acquiring bus access for the next transfer. This start and stop process repeats until the specified number of transfers are complete or there is an error.

2. Burst mode: The DMA controller does not surrender bus access until the specified number of transfers are complete or there is an error.

11.3.2 DMA Request Sources

ARQ, bit 0 in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1), offers a choice of two sources for DMA transfer requests: external and internal.

1. External input (DREQ):
   The trigger here is a rising edge in the input signal DREQ. The external source must then negate (falling edge) DREQ for each individual transfer in cycle stealing mode. Burst mode ignores this input until the specified number of transfers are complete.

   The output signal DREQCLR indicates when the DMA controller is ready for such DREQ edges. The external source must assert DREQ when DREQCLR is at Low level and negate DREQ when DREQCLR is at High level.

   Note that the DREQCLR timing differs between cycle stealing and burst modes. For cycle stealing mode, DREQCLR goes to High level after each individual transfer (byte, halfword, or word), so the external source must wait for TCOUT, the final transfer start signal, to also go to High level before starting any cleanup operations. For burst mode, DREQCLR and TCOUT go to High level simultaneously after the specified number of transfers are complete.

   Negating DREQ does not cancel a DMA transfer already in progress. The DMA controller retains bus access as described above. DREQCLR and TCOUT go to High level regardless of whether DREQ is already negated. We recommend that the external source wait for DREQCLR to go to High level before negating DREQ.

2. Software request mode:
   For memory-to-memory transfers and transfers between memory and I/O modules that cannot generate DREQ transfer requests, the program sets a bit to have the DMA controller generate internal requests until the specified number of transfers are complete. As long as this bit remains set, the DMA controller automatically performs DMA transfers each time that it is started.

11.3.3 Starting a DMA Transfer

The following is the procedure for starting a DMA transfer.

1. Set the IMK bit in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) for the channel to "1" to mask channel operation. (This step is not necessary if the channel is currently masked.)

2. Write to the DMA interrupt clear register (DMACCINT0 or DMACCINT1) for the channel to clear the status bits ready for a new transfer. (This step is not necessary if there is no interrupt request pending from the last DMA transfer.)

3. Configure the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) for the channel.

4. Load the DMA transfer source address register (DMACSAD0 or DMACSAD1) for the channel.

5. Load the DMA transfer destination address register (DMACDAD0 or DMACDAD1) for the channel.

6. Load the DMA transfer count register (DMACSIZ0 or DMACSIZ1) for the channel.

7. Set the IMK bit in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) for the channel to "0" to release the mask.

Note: DMA transfers do not start, however, if the DMA transfer complete status is not clear (step 2).

11.3.4 Ending a DMA Transfer

DMA transfers end in one of three ways.

1. Normal termination:
   The DMA transfer automatically ends when the number of transfers specified in the DMA transfer count register (DMACSIZ0 or DMACSIZ1) for the DMA channel are complete. The DMA controller sets IREQ0 or IREQ1 in the DMA transfer complete interrupt status register (DMAINT) to "1," but does not change the ISTA0 and ISTA1 bits. If the IMK bit in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) is "0," the DMA controller generates an interrupt request.

2. Abnormal termination:
   An error response during the cycle reading from the transfer source or the one writing to the transfer destination immediately terminates the DMA transfer. The DMA controller sets IREQ0 or IREQ1 in the DMA transfer complete interrupt status register (DMAINT) to "1," sets ISTA0 or ISTA1 to "1," and indicates which cycle triggered the error in ISTP0 or ISTP1. If the IMK bit in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) is "0," the DMA controller generates an interrupt request.

3. Forced termination:
   Setting the IMK bit in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) to "1" during a DMA transfer suspends operation after the write cycle. There is no interrupt. The program can restart operation by releasing the mask.

The program uses an interrupt handler or polling to determine termination and branches according to the DMA transfer complete status.

Normal termination

1. The program sets the IMK bit in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) to "1" to mask channel operation.

2. The program writes to the DMA interrupt clear register (DMACCINT0 or DMACCINT1) for the channel to clear the status bits.

Abnormal termination

1. The program sets the IMK bit in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) to "1" to mask channel operation.

2. The program reads the status bits (IREQ, ISTA, and ISTP) from the DMA transfer complete status register (DMAINT).

3. The program writes to the DMA interrupt clear register (DMACCINT0 or DMACCINT1) for the channel to clear the status bits.

4. The program processes the error in accordance with the needs of the user application system.

Table 11.1 summarizes register contents during and after a transfer.

Table 11.1 Register Contents during and after Transfer

Register		1. During operation	2. After normal termination	3. After abnormal termination	4. After forced termination
Status register (DMASTA)		1 = Busy (transferring data)	0 = Idle (no data to transfer)	0 = Idle (no data to transfer)	1 = Busy (transferring data)
Interrupt status register (DMAINT)	IREQ	0 = no effect	1 (*1)	1 (*1)	0 = no effect
	ISTA	0 = no effect	0 = normal termination	1 = abnormal termination	0 = no effect
	ISTP	0 = no effect	0 (initial state)	0 = error from transfer source 1 = error from transfer destination	0 = no effect
Transfer address registers (DMACSAD0 or DMACSAD1, DMACDAD0 or DMACDAD1)		Transferring Address	Address following that for final transfer	Address for error	Address following that at which transfer was interrupted
Transfer count register (DMACSIZ0 or DMACSIZ1)		Number of transfers remaining	0	(*2)	Number of transfers remaining

Notes

*1. The IREQ bit goes to "1" regardless of the interrupt mask (IMK) setting in the corresponding transfer mode register.

*2. The contents depend on which cycle triggered the error because the DMA controller decrements the transfer count register (DMACSIZ) only after a successful read from the transfer source.

11.3.5 DMA Channel Priority

The PRI bit in the DMA mode register (DMAMOD) specifies the strategy for assigning priority when there are simultaneous transfer requests for both channels 0 and 1: fixed priority or round robin.

- Fixed mode: Channel priority never changes (channel 0 > channel 1)
- Round robin mode: Last channel used has lower priority

Initial state	CH0 > CH1	
	↓	Transfer over channel 0
Priority after channel 0 transfer	CH0 < CH1	
	↓	Transfer over channel 1
Priority after channel 1 transfer	CH0 > CH1	
	↓	Transfer over channel 0
Priority after channel 0 transfer	CH0 < CH1	
	↓	Transfer over channel 1
Priority after channel 1 transfer	CH0 > CH1	

Figure 11.3 Round Robin Operation

The DMA controller determines the channel priority when it acquires bus access from the CPU. A burst mode transfer over the lower priority channel therefore continues through to completion even if there is a transfer request on the other channel.

11.3.6 Important Usage Notes

1. When setting up a DMA transfer, always take into consideration bus width, address alignment, and any other access restrictions in effect for the source and destination devices.

2. There can be no DMA transfers in HALT mode because the CPU is not available to grant bus access.

3. The DMA controller increments the source (DMACSAD0 or DMACSAD1) and destination (DMACDAD0 or DMACDAD1) addresses by the transfer size in bytes if the corresponding device type so specifies. The hardware enforces alignment by internally ignoring the lowest address bits as appropriate for the transfer size, but reading an address register returns those bits exactly as written.

   Example: Writing address of 0x10000011

   Word (32-bit) transfer: The DMA controller internally forces the lowest 2 bits to "00" for an effective address of 0x10000000. Reading the register, however, returns 0x10000011.

   Halfword (16-bit) transfer: The DMA controller internally forces the lowest bit to "0" for an effective address of 0x10000010. Reading the register, however, returns 0x10000011.

4. DMA transfers are not possible between certain combinations of source and destination devices. The following table lists the choices available for the transfer size (TSIZ) field in the DMA transfer mode register (DMAMOD0 or DMAMOD1) for the channel.

		Transfer destination	Internal device (internal SRAM)	External device			
			Incremental address device	Incremental address device		Fixed address device	
Transfer source		Bus width	32bit	8bit	16bit	8bit	16bit
Internal device (internal SRAM)	Incremental address device	32 bit	W/H/B	W/H/B	W/H/B	B	H
External device	Incremental address device	8 bit	W/H/B	W/H/B	W/H/B	B	H
		16 bit	W/H/B	W/H/B	W/H/B	B	H
	Fixed address device	8 bit	B	B	B	B	X
		16 bit	H	H	H	X	H

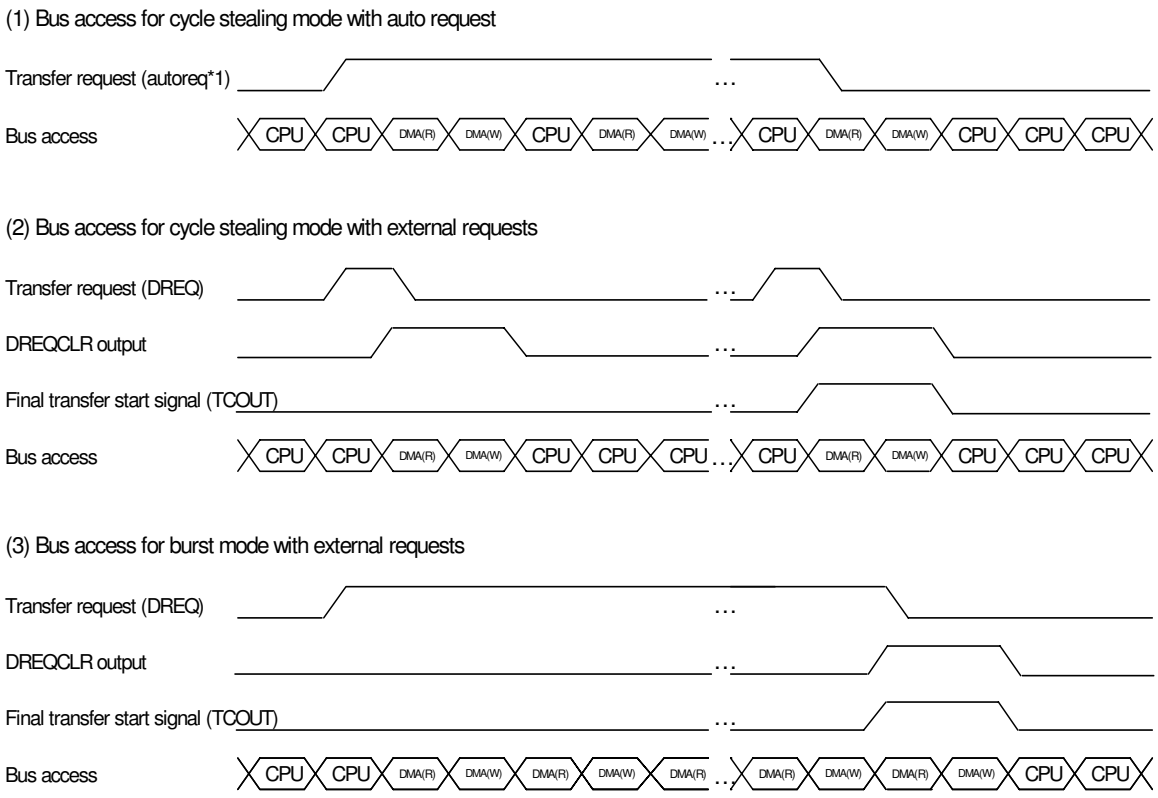
W: Word (32-bit) transfer
H: Halfword (16-bit) transfer
B: Byte (8-bit) transfer
X: Invalid combination

5. DMA restriction to External ROM area
   DMA controller can not access external ROM area/BANK25, because external ROM area is designed as the ARM processor execute program efficiently. Remapped the external ROM area to BANK0 is not support DMA controller access as same as BANK25. It should not be set BANK25(C800_0000~CFFF_FFFF) address nor BANK0(0000_0000~07FF_FFFF) address, when the external ROM area remapped to BANK0, to DMACSAD0/1 and DMACDAD0/1.

11.4 DMA Transfer Timing

11.4.1 Starting a Transfer

At least five clock cycles elapse between a transfer request and the actual start of the DMA transfer.



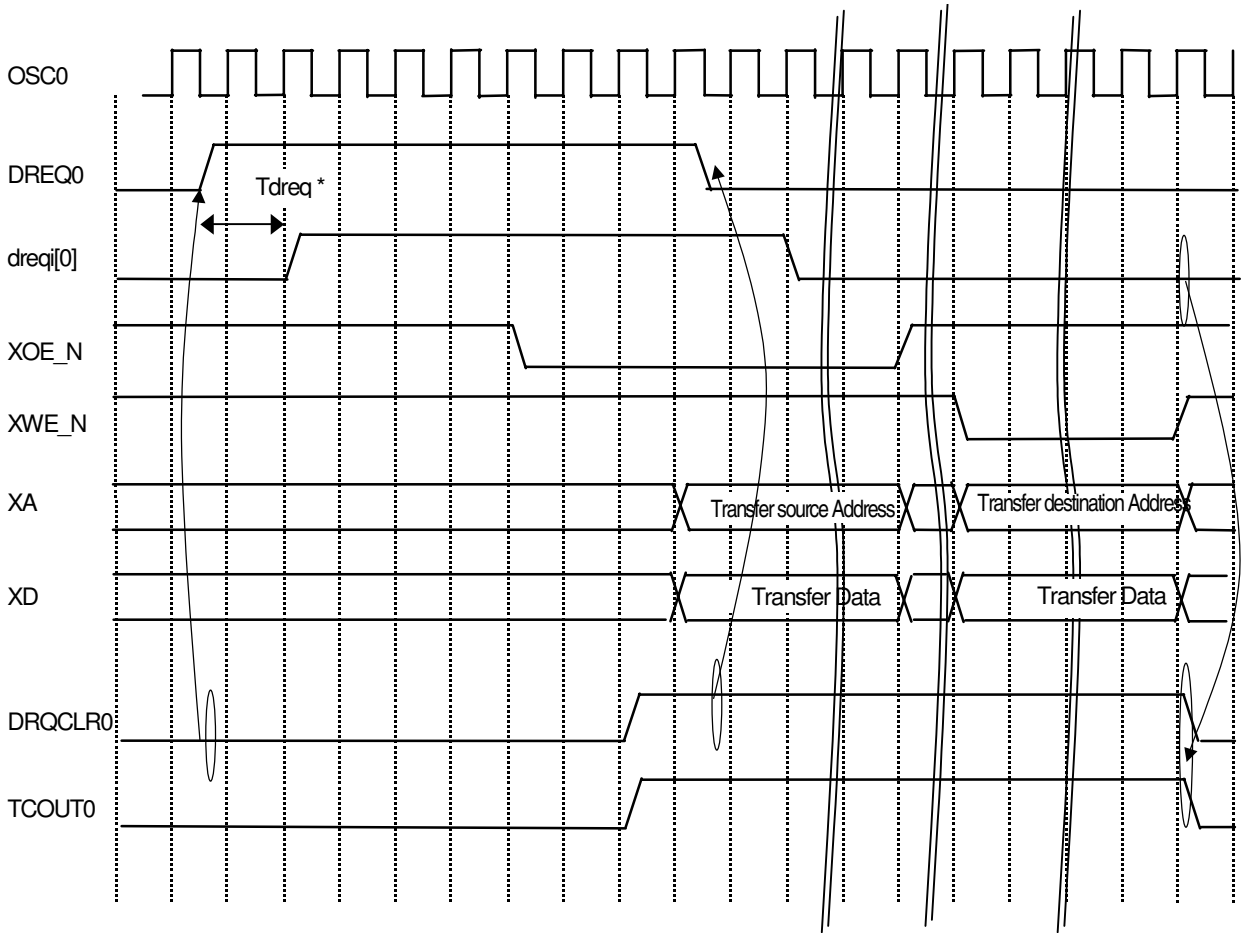
*1. Setting a bit in the DMA transfer mode register (DMACTMOD0 or DMACTMOD1) for the channel automatically generates this internal transfer request signal.

*2. The DMA controller uses dual address mode.

Figure 11.1 Transfer Start Timing

11.4.2 Transfer Timing

Figure 11.2 shows the timing for transferring data from an I/O device on the external bus to memory using cycle stealing mode and DREQ triggers.



* T_{dreq}, the delay between the external request (DREQ) and acceptance of the internal one (dreqi), is a maximum of two clock cycles.

Figure 11.2 Timing for Transferring Single Word from External I/O Device to Memory Using Cycle Stealing Mode and DREQ Triggers

Chapter 12

GPIO

Chapter 12 GPIO

12.1 Overview

The two 16-bit GPIO (PIOx) have the following features.

- Ports: Two
- The I/O direction is specified at the individual pin level, with input the default configuration immediately after a reset.
- Edge detection triggers for interrupts, interrupt masks, and interrupt modes settings are specified at the port level.

12.1.1 Components

Figure 12.1 gives the components for a 16-bit GPIO (PIOx). Only pin 0 is shown in detail because all pins have exactly the same structure.

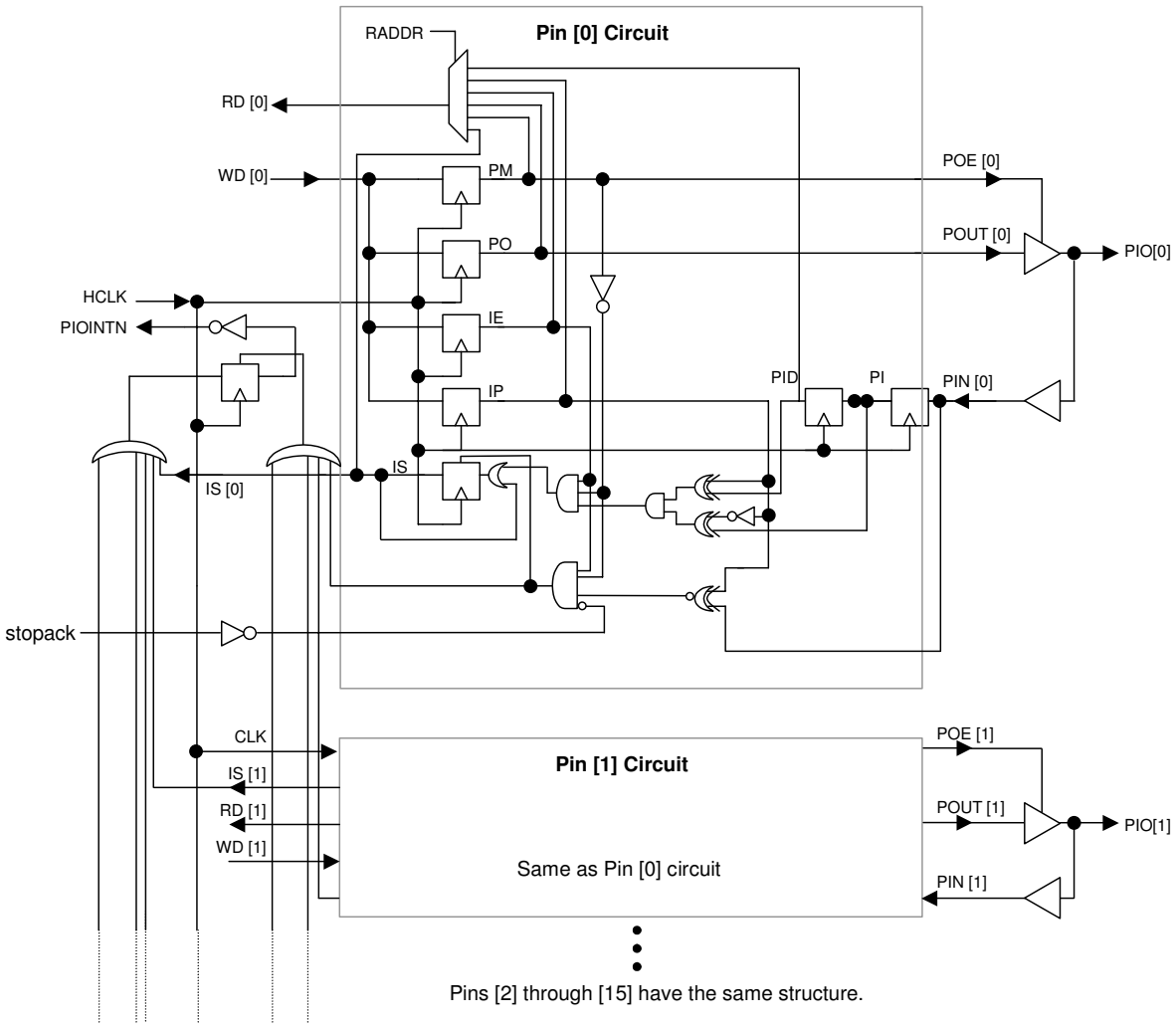


Figure 12.1 PIOx Block Diagram

12.1.2 Pin List

Pin Name	I/O	Function	
		Primary Function	Secondary Function (Pin Name)
PIOA[0]	I/O	General-purpose port A bit 0	SIN
PIOA[1]	I/O	General-purpose port A bit 1	SOUT
PIOA[2]	I/O	General-purpose port A bit 2	CTS
PIOA[3]	I/O	General-purpose port A bit 3	DSR
PIOA[4]	I/O	General-purpose port A bit 4	DCD
PIOA[5]	I/O	General-purpose port A bit 5	DTR
PIOA[6]	I/O	General-purpose port A bit 6	RTS
PIOA[7]	I/O	General-purpose port A bit 7	RI
PIOA[8]	I/O	General-purpose port A bit 8	STXD
PIOA[9]	I/O	General-purpose port A bit 9	SRXD
PIOA[10]	I/O	General-purpose port A bit 10	XA[19]
PIOA[11]	I/O	General-purpose port A bit 11	XA[20]
PIOA[12]	I/O	General-purpose port A bit 12	XA[21]
PIOA[13]	I/O	General-purpose port A bit 13	XA[22]
PIOA[14]	I/O	General-purpose port A bit 14	XA[23]
PIOA[15]	I/O	General-purpose port A bit 15	XWR
PIOB[0]	I/O	General-purpose port B bit 0	DREQ0
PIOB[1]	I/O	General-purpose port B bit 1	DREQCLR0
PIOB[2]	I/O	General-purpose port B bit 2	DREQ1
PIOB[3]	I/O	General-purpose port B bit 3	DREQCLR1
PIOB[4]	I/O	General-purpose port B bit 4	TCOUT0
PIOB[5]	I/O	General-purpose port B bit 5	TCOUT1
PIOB[6]	I/O	General-purpose port B bit 6	PWMOUT0
PIOB[7]	I/O	General-purpose port B bit 7	PWMOUT1
PIOB[8]	I/O	General-purpose port B bit 8	XWAIT
PIOB[9]	I/O	General-purpose port B bit 9	XCAS_N
PIOB[10]	I/O	General-purpose port B bit 10	XRAS_N
PIOB[11]	I/O	General-purpose port B bit 11	XSDCLK
PIOB[12]	I/O	General-purpose port B bit 12	XSDCS_N
PIOB[13]	I/O	General-purpose port B bit 13	XSDCKE
PIOB[14]	I/O	General-purpose port B bit 14	XDQM[1]/XCAS_N[1]
PIOB[15]	I/O	General-purpose port B bit 15	XDQM[0]/XCAS_N[0]

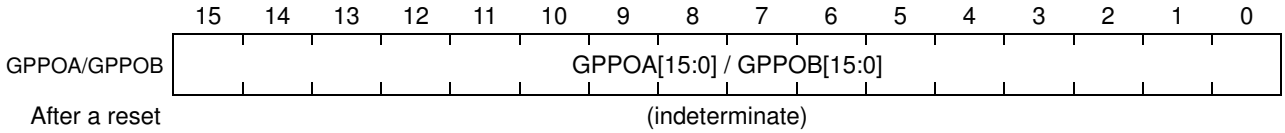
12.1.3 Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0xB7A00000	Port A output register	GPPOA	R/W	16	Indeterminate
0xB7A00004	Port A input register	GPPIA	R	16	Reflects pin states
0xB7A00008	Port A mode register	GPPMA	R/W	16	0x0000
0xB7A0000C	Port A interrupt enable register	GPIEA	R/W	16	0x0000
0xB7A00010	Port A interrupt polarity register	GPIPA	R/W	16	0x0000
0xB7A00014	Port A interrupt status register	GPISA	R/W	16	0x0000
0xB7A00020	Port B output register	GPPOB	R/W	16	Indeterminate
0xB7A00024	Port B input register	GPPIB	R	16	Reflects pin states
0xB7A00028	Port B mode register	GPPMB	R/W	16	0x0000
0xB7A0002C	Port B interrupt enable register	GPIEB	R/W	16	0x0000
0xB7A00030	Port B interrupt polarity register	GPIPB	R/W	16	0x0000
0xB7A00034	Port B interrupt status register	GPISB	R/W	16	0x0000
0xB7000000	Port function select register	GPCTL	R/W	16	0x0000

12.2 Register Descriptions

12.2.1 Port Output Registers (GPPOA and GPPOB)

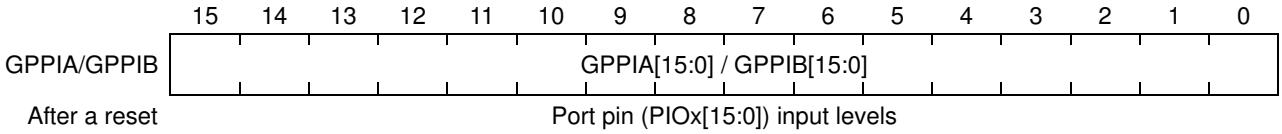
These registers specify the output values for the corresponding port pins (PIOx[15:0]). However, a pin's output will only track the specified output level if the GPCTL register has been configured for its primary function and the corresponding port mode (GPPMx) register has been configured for output.
The CPU has read/write access to these registers.
The register contents after a reset are indeterminate.



Address: 0xB7A00000 (GPPOA), 0xB7A00020 (GPPOB)
Access: R/W
Access size: 16 bits

12.2.2 Port Input Registers (GPPIA and GPPIB)

These registers reflect the input levels for the corresponding port pins (PIOx[15:0]). If a pin is configured for output, however, the corresponding bit reflects its output level.
The CPU has only read access to these registers.
The register contents after a reset reflect the port pin (PIOx[15:0]) input levels.



Address: 0xB7A00004 (GPPIA), 0xB7A00024 (GPPIB)
Access: R
Access size: 16 bits

12.2.3 Port Mode Registers (GPPMA and GPPMB)

These registers specify the I/O directions for the corresponding port pins (PIOx[15:0]) at the individual pin level.
The CPU has read/write access to these registers.
The register contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPPMA/GPPMB	GPPMA[15:0] / GPPMB[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7A00008 (GPPMA), 0xB7A00028 (GPPMB)
Access: R/W
Access size: 16 bits

Bit Descriptions

- **GPPMA[15:0]/GPPMB[15:0]** (bits 0 to 15):
  These bits specify the I/O directions for the corresponding port pins.

GPPMA[15:0] / GPPMB[15:0]	Description
0	Input
1	Output

12.2.4 Port Interrupt Enable Registers (GPIEA and GPIEB)

These registers enable or disable interrupts detected from a corresponding PIOx pin. The detection of interrupts through any of the PIOx pins requires the particular pin to be configured (GPPMx register) as input. In addition, the input edge to be detected at the PIOx pin can be specified in the GPIPx register.
There are no such interrupt requests for pins configured for output. When the port is set as output, the interface ignores the corresponding bits in these registers.
The CPU has read/write access to these registers. The register contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIEA/GPIEB	GPIEA[15:0] / GPIEB[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7A0000C (GPIEA), 0xB7A0002C (GPIEB)
Access: R/W
Access size: 16 bits

Note

PIOx pin input restarts the clock, releasing the LSI from STANDBY mode, if the pin is configured for input and interrupts are enabled. The outside source must, however, maintain the input signal at the specified level until clock supply resumes. For further details on starting and stopping the clock signal, see Chapter 7 "Power Management."

Bit Descriptions

- **GPIEA[15:0]/GPIEB[15:0]** (bits 0 to 15):
  These bits control interrupt requests from the corresponding port pins (PIOx[15:0]).

GPIEA[15:0] / GPIEB[15:0]	Description
0	Disable interrupts
1	Enable interrupts

12.2.5 Port Interrupt Polarity Register (GPIPA and GPIPB)

These registers specify the edge triggers for interrupt requests from the corresponding port pins (PIOx[15:0]).
Detection of an edge with the specified polarity in the input from a PIOx pin configured (GPPMx register) for input with interrupts enabled (GPIEx register) triggers an interrupt request.
The CPU has read/write access to these registers.
The register contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIPA/GPIPB	GPIPA[15:0] / GPIPB[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7A00010 (GPIPA), 0xB7A00030 (GPIPB)
Access: R/W
Access size: 16 bits

Bit Descriptions

- **GPIPA[15:0]/GPIPB[15:0]** (bits 0 to 15):
  These bits specify edge trigger polarity at the individual pin level.

GPIPA[15:0] / GPIPB[15:0]	Description
0	Falling edge of input signal triggers interrupt request
1	Rising edge of input signal triggers interrupt request

12.2.6 Port Interrupt Status Registers (GPISA and GPISB)

These registers contain flags indicating the sources for pending interrupt requests.
Writing "1" to a bit resets it to "0." Writes of "0" are ignored.
The register contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPISA/GPISB	GPISA[15:0] / GPISB[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7A00014 (GPISA), 0xB7A00034 (GPISB)
Access: R/W
Access size: 16 bits

Bit Descriptions

- **GPISA[15:0]/GPISB[15:0]** (bits 0 to 15):
  A "1" in a bit indicates an interrupt request pending from the corresponding pin.

GPISA[15:0] / GPISB[15:0]	Description
0	No interrupt pending
1	Interrupt request pending

GPISA bit 0, for example, goes to one
(a) at a rising edge for the combination GPIPA[0] = "1" plus GPPMA[0] = "0" and GPIEA[0] = "1"
(b) at a falling edge for the combination GPIPA[0] = "0" plus GPPMA[0] = "0" and GPIEA[0] = "1"

12.2.7 Port Function Select Register (GPCTL)

This register configures groups of PIOA[15:0]/PIOB[8:0] pins for their primary and secondary functions. The CPU has read/write access to this register.
The register contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCTL	– *	– *	– *	– *	– *	– *	– *	– *	GPCTL7	GPCTL6	GPCTL5	GPCTL4	GPCTL3	GPCTL2	GPCTL1	GPCTL0
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7000000
Access: R/W
Access size: 16 bits

Note
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **GPCTL0** (bit 0):
  This bit controls the function of pins PIOA[7:0]. Their secondary function is as a UART interface.

GPCTL0 = "0" (primary function)		GPCTL0 = "1" (secondary function)	
Function	**In/Out**	**Function**	**In/Out**
PIOA[0]	In/Out	SIN	Input
PIOA[1]	In/Out	SOUT	Output
PIOA[2]	In/Out	CTS	Input
PIOA[3]	In/Out	DSR	Input
PIOA[4]	In/Out	DCD	Input
PIOA[5]	In/Out	DTR	Output
PIOA[6]	In/Out	RTS	Output
PIOA[7]	In/Out	RI	Input

- **GPCTL1** (bit 1):
  This bit controls the function of pins PIOA[9:8]. Their secondary function is as serial interface.

GPCTL1 = "0" (primary function)		GPCTL1 = "1" (secondary function)	
Function	**In/Out**	**Function**	**In/Out**
PIOA[8]	In/Out	STXD	Output
PIOA[9]	In/Out	SRXD	Input

- **GPCTL2** (bit 2):
  This bit controls the function of pins PIOA[14:10]. Their secondary function is as an external bus.

GPCTL2 = "0" (primary function)		GPCTL2 = "1" (secondary function)	
Function	**In/Out**	**Function**	**In/Out**
PIOA[10]	In/Out	XA[19]	Output
PIOA[11]	In/Out	XA[20]	Output
PIOA[12]	In/Out	XA[21]	Output
PIOA[13]	In/Out	XA[22]	Output
PIOA[14]	In/Out	XA[23]	Output

- GPCTL3 (bit 3): This bit controls the function of pins PIOB[4] and PIOB[1:0]. Their secondary function is as DMA channel 0.

GPCTL3 = "0" (primary function)		GPCTL3 = "1" (secondary function)	
Function	**In/Out**	**Function**	**In/Out**
PIOB[0]	In/Out	DREQ0	Input
PIOB[1]	In/Out	DREQCLR0	Output
PIOB[4]	In/Out	TCOUT0	Output

- **GPCTL4** (bit 4):
This bit controls the function of pins PIOB[5] and PIOB[3:2]. Their secondary function is as DMA channel 1.

GPCTL4 = "0" (primary function)		GPCTL4 = "1" (secondary function)	
Function	**In/Out**	**Function**	**In/Out**
PIOB[2]	In/Out	DREQ1	Input
PIOB[3]	In/Out	DREQCLR1	Output
PIOB[5]	In/Out	TCOUT1	Output

- **GPCTL5** (bit 5):
This bit controls the function of pins PIOB[7:6]. Their secondary function is as PWM outputs.

GPCTL5 = "0" (primary function)		GPCTL5 = "1" (secondary function)	
Function	**In/Out**	**Function**	**In/Out**
PIOB[6]	In/Out	PWMOUT0	Output
PIOB[7]	In/Out	PWMOUT1	Output

- **GPCTL6** (bit 6):
This bit controls the function of pin PIOB[8]. Its secondary function is as external bus WAIT input. GPCTL6 = "0" (primary function)

GPCTL6 = "0" (primary function)		GPCTL6 = "1" (secondary function)	
Function	**In/Out**	**Function**	**In/Out**
PIOB[8]	In/Out	XWAIT	Input

- **GPCTL7** (bit 7):
This bit controls the function of pin PIOA[15]. Its secondary function is as external bus data direction.

GPCTL7 = "0" (primary function)		GPCTL7 = "1" (secondary function)	
Function	**In/Out**	**Function**	**In/Out**
PIOA[15]	In/Out	XWR	Output

12.3 Description of Operation

The PIOx ports perform I/O operations with their pins via their port output (GPPOx) and port input (GPPIx) registers.
Three sets of registers provide control at the individual pin level.

- The port mode (GPPMx) registers specifying the pin I/O directions
- The port interrupt enable (GPIEx) registers controlling interrupt requests
- The port interrupt polarity (GPIPx) registers specifying the edge trigger polarity for interrupts

12.3.1 Interrupt Requests

Figure 12.2 illustrates interrupt operation for PIOA pin 0. All pins operate exactly the same way.
If the pin is configured for input (GPPMA[0] = "0"), the status register bit (GPISA[0]) goes to "1" when the input monitor bit (PIOA[0]) changes to the specified value (GPIPA[0]). The transition triggers an interrupt request (PIOINTN = "0") if interrupts are enabled (GPIEA[0] = "1").

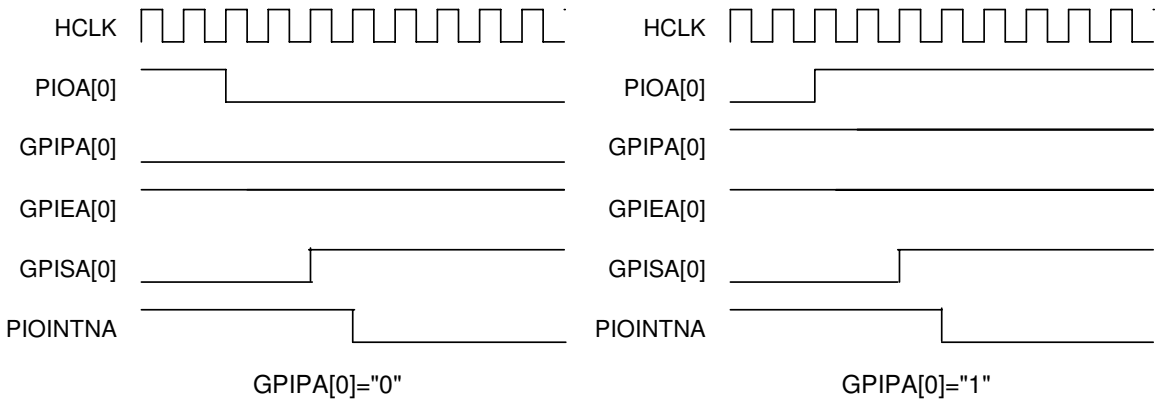


Figure 12.2 Interrupt Operation

12.3.2 Primary/Secondary function configuration

PIOA[15:0] and PIOB[15:0] have secondary functions.
PIOB[15:9] are configured by MODE selection pin for DRAM access function as explained in Chapter 4 "Chip Configuration" of this manual. By configuring the Mode Selection pins so as to disables the DRAM controller, the processor will be configured to use the pins PIOB[15:9] in their primary function, thus as GPIOs. By setting the Mode selection pins in a configuration to enable the DRAM controller, the This MCU will be configured to use the pins PIOB[15:9] in their secondary function which is as control signals for the DRAM bank.
PIOA[15:0] and PIOB[8:0] are configured by setting GPCTL register for secondary functions as described in this chapter of this manual.

Chapter 13

Watchdog Timer (WDT)

Chapter 13 Watchdog Timer (WDT)

13.1 Overview

The 16-bit watchdog timer monitors whether the program has run out of control. Alternatively, it can be used as an interval timer.
Counter overflow produces an interrupt request or, for the watchdog timer configuration, a forced system reset.

13.1.1 Components

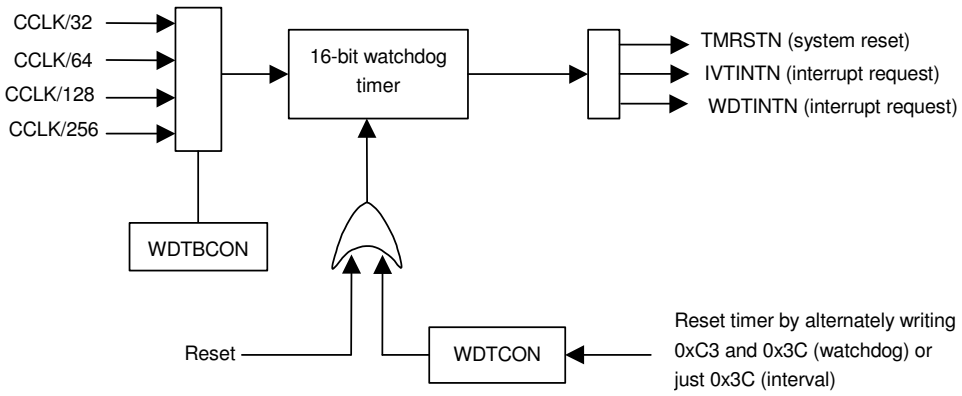


Figure 13.1 Watchdog Timer Components

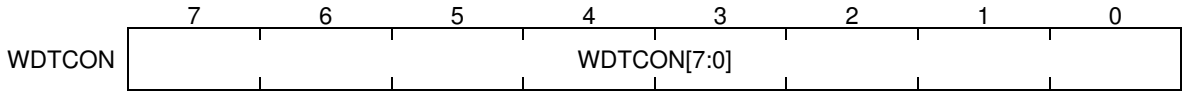
13.1.2 Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0xB7E00000	Watchdog timer control register	WDTCON	W	8	—
0xB7E00004	Time base counter control register	WDTBCON	R/W	8	0x00
0xB7E00014	Status register	WDSTAT	R/W	8	0x00

13.2 Register Descriptions

13.2.1 Watchdog Timer Control Register (WDTCON)

This register is for resetting the watchdog timer to zero.
The CPU has only write access to this register
To start the timer after a system reset, write 0x3C to this register. From that point onward, the program must reset the counter to zero at regular intervals by writing to this register: 0xC3 and 0x3C alternately for watchdog timer operation or just 0x3C for interval timer operation.



Address: 0xB7E00000
Access: W
Access size: 8 bits

13.2.2 Time Base Counter Control Register (WDTBCON)

This register controls watchdog timer operation, specifying such things as the operating clock frequency, the operation mode (interval timer or watchdog timer), and whether counter overflow produces an interrupt request or a system reset (watchdog timer operation only).
The CPU has read/write access to this register.
A write protection function shields the contents from random writes. The program must first write 0x5A and then the new value.
The register contents after a reset are 0x00.

	7	6	5	4	3	2	1	0
WDTBCON	WDHLT	OFINTM	– *	ITEN	ITM	– *	WDCLK[1:0]	
After a	0	0	0	0	0	0	0	0

Address: 0xB7E00004
Access: R/W
Access size: 8 bits

Note
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **WDCLK[1:0]** (bits 0 and 1):
  These bits specify the frequency divisor for deriving the operating clock from CCLK.

WDCLK[1:0]		Description
1	0	
0	0	CCLK/32
0	1	CCLK/64
1	0	CCLK/128
1	1	CCLK/256

- **ITM** (bit 3):
  This bit specifies the operation mode: watchdog timer or interval timer.

ITM	Description
0	Watchdog timer
1	Interval timer

- **ITEN** (bit 4):
  This bit controls interval timer (ITM = 1) operation.

ITEN	Description
0	Stop
1	Start/Enable

- **OFINTMODE** (bit 6):
  This bit specifies the action after counter overflow (watchdog operation only).

OFINTMODE	Description
0	Interrupt request
1	System reset

- **WDHLT** (bit 7):
  This bit controls watchdog timer/interval timer operation.

WDHLT	Description
0	Start/Enable
1	Stop

13.2.3 Status Register (WDSTAT)

This register gives the status of the watchdog timer interrupt and system reset requests.
The CPU has read/write access to this register.
The register contents after a reset are 0x00.

	7	6	5	4	3	2	1	0
WDSTAT	– *	– *	IVTIST	WDTIST	– *	– *	– *	RSTSTA
After a	0	0	0	0	0	0	0	0

Address: 0xB7E00014
Access: R/W
Access size: 8 bits

Note
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **RSTSTATUS** (bit 0):
  This bit gives the reset source. Once this bit goes to "1," it does not return to "0" until there is RESET_N input.

RSTSTATUS	Description
0	Power on reset
1	Watchdog timer reset

- **WDTIST** (bit 4):
  A "1" in this bit indicates an interrupt request from the watchdog timer.
  Writing "1" to this bit resets it to "0."

  Note: Failure to clear this bit after a interrupt/reset request produces a series of such requests. Clearing this bit requires up to 20 HCLK cycles, so always wait at least 20 HCLK cycles before clearing the interrupt controller.

WDTST	Description
0	No watchdog timer interrupt request pending
1	Watchdog timer interrupt request pending

- **IVTIST** (bit 5):
  A "1" in this bit indicates an interrupt request from the interval timer.
  Writing "1" to this bit resets it to "0."

  Note: Failure to clear this bit after a interrupt/reset request produces a series of such requests. Clearing this bit requires up to 20 HCLK cycles, so always wait at least 20 HCLK cycles before clearing the interrupt controller.

IVTST	Description
0	No interval timer interrupt request pending
1	Interval timer interrupt request pending

13.3 Description of Operation

The watchdog timer allows the programmer to detect when the program has run out of control by having counter overflow generate an interrupt or system reset request.
If the design does not call for a watchdog timer, this timer is available for use as an interval timer.

13.3.1 Operation Modes

The ITM bit in the WDTBCON register specifies the operation mode: watchdog timer or interval timer.
The WDCLK[1:0] bits in the WDTBCON register specify the frequency divisor for deriving the operating clock from CCLK.

13.3.2 Interval Timer Operation

Setting both the ITM and ITEN bits in the WDTBCON register to "1" produces interval timer operation. Writing 0x3C to the WDTCON register then starts counting up from zero. Additional writes of the same value reset the counter to zero. Failure to reset the counter in a timely fashion produces overflow and an interrupt request.
This configuration does not generate system reset signals.

13.3.3 Watchdog Timer Operation

Setting the ITM bit in the WDTBCON register to "0" produces watchdog timer operation. Writing 0x3C to the WDTCON register then starts counting up from zero. After that, alternately writing 0xC3 and 0x3C resets the timer to zero. Failure to reset the counter in a timely fashion produces overflow and an interrupt or system reset request.
In a typical application, timer overflow should be avoided entirely except as an indication that the program has run out of control.

13.3.4 Starting Timer

Specifying the operation mode and other settings in the count operation (WDTBCON) register and then writing 0x3C to the watchdog timer control (WDTCON) register starts counter operation. From that point onward, the program must reset the counter to zero at regular intervals by writing to WDTCON: 0xC3 and 0x3C alternately for watchdog timer operation or just 0x3C for interval timer operation.
Failure to reset the counter in a timely fashion produces overflow and an interrupt or system reset request as specified by the OFINTMODE bit in the WDTBCON register. Note, however, that interval timer operation only produces interrupt requests.
Watchdog timer operation continues even after the CPU shifts to HALT mode. If this behavior is not desirable, set the ITM and ITEN bits in the WDTBCON register to "1" and "0," respectively, to switch to interval timer operation and stop the counter. When the CPU leaves HALT mode, write "0" to the ITM bit to resume watchdog timer operation and restart the counter.
STANDBY mode suspends watchdog timer operation.

Chapter 14

Timers

Chapter 14 Timers

14.1 Overview

This LSI features one 16-bit System timer and a Timer Counter block which includes six identical 16-bit timer counter channels. The System Timer is internal to the core block of the MCU and is ideal for handling system tasks. The Auto Reload Timer channels are each programmed independently to perform a wide variety of tasks. In addition, each Timer drives an internal interrupt signal, which can be programmed to generate processor interrupts.

The following list summarizes the main features of the System Timer and the Auto Reload Timers:

- System Timer
  - 16-bit counter
  - Interrupt request on overflow
  - Timer clock base as CCLK/16
  - Interval timer operation

- Auto Reload Timers
  - 16-bit counters for each timer
  - Interrupt request on compare match
  - Independent clock settings for each timer
  - Choice of one-shot or interval timer operation for each timer

14.1.1 Components

Figure 14.1 is a block diagram giving components for the system timer; Figure 14.2, one for an auto reload timer.

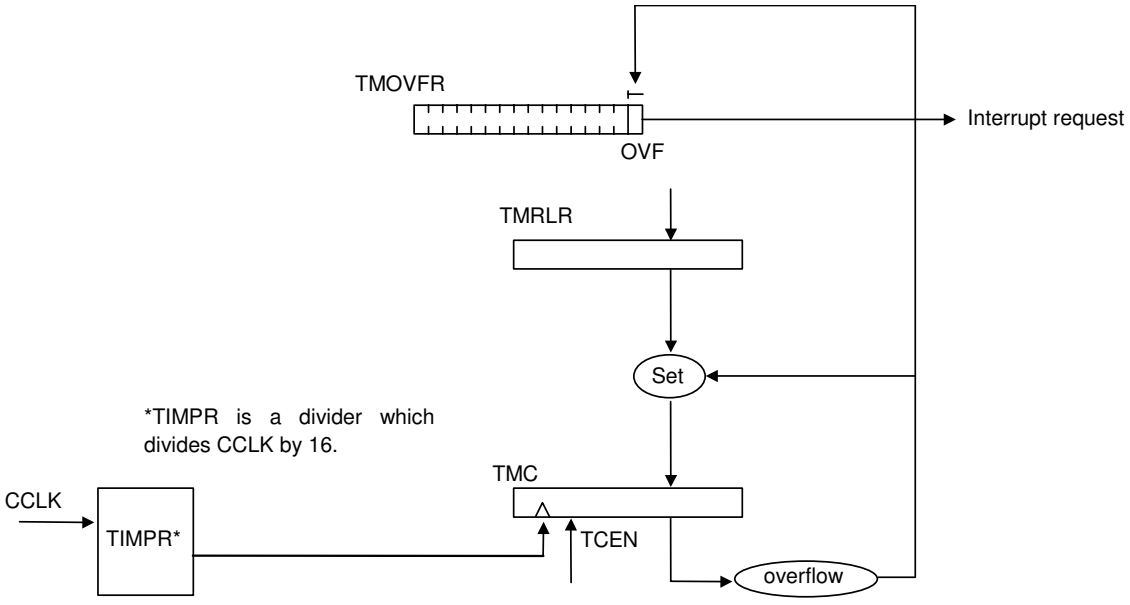


Figure 14.1 System Timer Components

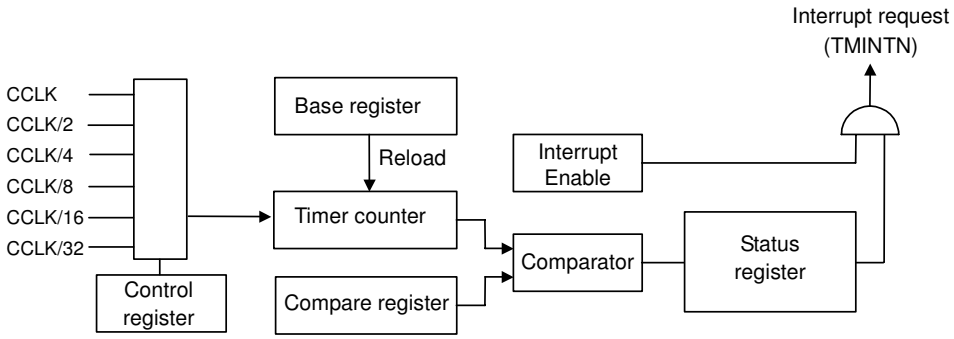


Figure 14.2 Auto Reload Timer Components

14.1.2 Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0xB8001004	System timer enable register	TMEN	R/W	32	0x00000000
0xB8001008	System timer reload register	TMRLR	R/W	32	0x00000000
0xB8001010	System counter overflow register	TMOVFR	R/W	32	0x00000000
0xB7F00000	Timer 0 control register	TIMECNTL0	R/W	16	0x0000
0xB7F00004	Timer 0 base register	TIMEBASE0	R/W	16	0x0000
0xB7F00008	Timer 0 counter register	TIMECNT0	R	16	0x0000
0xB7F0000C	Timer 0 compare register	TIMECMP0	R/W	16	0xFFFF
0xB7F00010	Timer 0 status register	TIMESTAT0	R/W	16	0x0000
0xB7F00020	Timer 1 control register	TIMECNTL1	R/W	16	0x0000
0xB7F00024	Timer 1 base register	TIMEBASE1	R/W	16	0x0000
0xB7F00028	Timer 1 counter register	TIMECNT1	R	16	0x0000
0xB7F0002C	Timer 1 compare register	TIMECMP1	R/W	16	0xFFFF
0xB7F00030	Timer 1 status register	TIMESTAT1	R/W	16	0x0000
0xB7F00040	Timer 2 control register	TIMECNTL2	R/W	16	0x0000
0xB7F00044	Timer 2 base register	TIMEBASE2	R/W	16	0x0000
0xB7F00048	Timer 2 counter register	TIMECNT2	R	16	0x0000
0xB7F0004C	Timer 2 compare register	TIMECMP2	R/W	16	0xFFFF
0xB7F00050	Timer 2 status register	TIMESTAT2	R/W	16	0x0000
0xB7F00060	Timer 3 control register	TIMECNTL3	R/W	16	0x0000
0xB7F00064	Timer 3 base register	TIMEBASE3	R/W	16	0x0000
0xB7F00068	Timer 3 counter register	TIMECNT3	R	16	0x0000
0xB7F0006C	Timer 3 compare register	TIMECMP3	R/W	16	0xFFFF
0xB7F00070	Timer 3 status register	TIMESTAT3	R/W	16	0x0000
0xB7F00080	Timer 4 control register	TIMECNTL4	R/W	16	0x0000
0xB7F00084	Timer 4 base register	TIMEBASE4	R/W	16	0x0000
0xB7F00088	Timer 4 counter register	TIMECNT4	R	16	0x0000
0xB7F0008C	Timer 4 compare register	TIMECMP4	R/W	16	0xFFFF
0xB7F00090	Timer 4 status register	TIMESTAT4	R/W	16	0x0000
0xB7F000A0	Timer 5 control register	TIMECNTL5	R/W	16	0x0000
0xB7F000A4	Timer 5 base register	TIMEBASE5	R/W	16	0x0000
0xB7F000A8	Timer 5 counter register	TIMECNT5	R	16	0x0000
0xB7F000AC	Timer 5 compare register	TIMECMP5	R/W	16	0xFFFF
0xB7F000B0	Timer 5 status register	TIMESTAT5	R/W	16	0x0000

14.2 Register Descriptions

14.2.1 System Timer Enable Register (TMEN)

This register starts and stops counting by the timer counter (TMC) register.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMEN	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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

Notes

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

If the timer's operating clock, CCLK with no frequency division, is slower than the bus clock (HCLK), leave the following interval between successive writes to this register. Otherwise, because such writes are synchronized with CCLK not HCLK, the data for the second write might arrive before the hardware has properly latched that for the first write.

$$n \times HCLK + 32 \times CCLK$$

where n = 16 × HCLK frequency / CCLK frequency

Bit Descriptions

- **TCEN** (bit 0):
  This bit controls system timer operation.

TCEN	Description
0	Stop system timer
1	Start system timer

14.2.2 System Timer Reload Register (TMRLR)

This register specifies the timer counter (TMC) reload value.
Writing to this register simultaneously writes the same value to the timer counter (TMC) register.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMRLR	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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:
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
Leave the following interval between successive writes to this register.

$$n \times HCLK + 79 \times CCLK$$

where $n = 16 \times$ HCLK frequency / CCLK frequency

14.2.3 System Timer Overflow Register (TMOVFR)

The OVF bit in this register goes to "1" when timer counter (TMC) overflow generates a system counter overflow interrupt request.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMOVFR	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **OVF** (bit 0):
  A "1" in this bit indicates counter overflow. Writing "1" to this bit resets it to "0." Writes of "0" are ignored.

OVF	Description
0	No overflow detected
1	Overflow detected

Note: Failure to clear this bit after a interrupt/reset request produces a series of such requests.

14.2.4 Timer Control Registers (TIMECNTL0 to TIMECNTL5)

These registers control timer operation, specifying the operating clock, enabling and disabling interrupts, starting and stopping the timer, and specifying the timer operation mode. The CPU has read/write access to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMECNTL0 to 5	– *	– *	– *	– *	– *	– *	– *	– *	CLKSEL[2:0]			IE	START	– *	– *	MODE
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7F00000 (CH0), 0xB7F00020 (CH1), 0xB7F00040 (CH2), 0xB7F00060 (CH3), 0xB7F00080 (CH4), 0xB7F000A0 (CH5)

Access: R/W

Access size: 16 bits

Note

CLKSEL should be set before START bit setting, if application need correct timer interval. Because timer start count with maximum 6 count by the clock that selected by CLKSEL before this register write.

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **MODE** (bit 0):
  This bit specifies the timer operation mode.

MODE	Description
0	Interval timer
1	One-shot timer

- **START** (bit 3):
  This bit controls timer operation.

START	Description
0	Stop timer
1	Start timer

- **IE** (bit 4):
  This bit controls interrupt requests.

IE	Description
0	Disable interrupts
1	Enable interrupts

- **CLKSEL** (bit 5 to 7):
  These bits specify the frequency divisor for deriving the operating clock from CCLK.

CLKSEL			Description
7	6	5	
0	0	0	CCLK
0	0	1	CCLK/2
0	1	0	CCLK/4
0	1	1	CCLK/8
1	0	0	CCLK/16
1	0	1	CCLK/32
1	1	0	(reserved)
1	1	1	(reserved)

14.2.5 Timer Base Registers (TIMEBASE0 to TIMEBASE5)

These registers specify the counter contents at the start of operation for the corresponding timer. Writing to one simultaneously writes the same value to the corresponding timer counter register (TIMECNT0 to TIMECNT5).
The CPU has read/write access to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEBASE0 to	TIMEBASE[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7F00004 (CH0), 0xB7F00024 (CH1), 0xB7F00044 (CH2), 0xB7F00064 (CH3), 0xB7F00084 (CH4), 0xB7F000A4 (CH5)

Access: R/W

Access size: 16 bits

14.2.6 Timer Counter Register (TIMECNT0 to TIMECNT5)

These registers represent the 16-bit up counters for the timers. Reading one returns the current counter contents.
The CPU has only read access to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMECNT0 to 5	TIMECNT[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7F00008 (CH0), 0xB7F00028 (CH1), 0xB7F00048 (CH2), 0xB7F00068 (CH3), 0xB7F00088 (CH4), 0xB7F000A8 (CH5)
Access: R
Access size: 16 bits

14.2.7 Timer Compare Registers (TIMECMP0 to TIMECMP5)

A match between the contents of a timer counter register (TIMECNTx) and those of the corresponding TIMECMPx register triggers an interrupt request and reloads TIMECNTx from the corresponding timer base register (TIMEBASEx).
The CPU has read/write access to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMECMP0 to 5	TIMECMP[15:0]															
After a reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Address: 0xB7F0000C (CH0), 0xB7F0002C (CH1), 0xB7F0004C (CH2), 0xB7F0006C (CH3), 0xB7F0008C (CH4), 0xB7F000AC (CH5)

Access: R/W

Access size: 16 bits

14.2.8 Timer Status Registers (TIMESTAT0 to TIMESTAT5)

The STATUS bit in this register goes to "1" to indicate a match between the counter and compare registers. The CPU has read/write access to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMESTAT0 to 5	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	STATUS
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7F00010 (CH0), 0xB7F00030 (CH1), 0xB7F00050 (CH2), 0xB7F00070 (CH3), 0xB7F00090 (CH4), 0xB7F000B0 (CH5)

Access: R/W

Access size: 16 bits

Note

– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **STATUS** (bit 0):
  This bit goes to "1" to indicate a match between the counter and compare registers.
  Writing "1" to this bit resets it to "0." Writes of "0" are ignored.

STATUS	Description
0	No match
1	Match

14.3 Description of Operation

14.3.1 System Timer

1. Reload timer operation

The System Timer operates independently from the six Auto Reload Timers. The three registers that control this timer are outlined in the Register List table on page 14-3.

The System Timer is organized around a 16-bit counter. The input time-base to the System Timer block is the CCLK. The CCLK input is divided by a factor of 16 and this forms the time-base for the counter. Once enabled by writing a "1" to the Timer Enable Register (TMEN), the counter counts up each cycle of the 'CCLK/16' input-signal starting from the initial value programmed in to the Timer Counter Register (TMC). Note that when the user program sets the Timer Reload Register (TMRLR), the hardware simultaneously copies the contents of TMRLR to the Timer Counter (TMC) register. The user program does not have direct access to the TMC register.

The value of the counter is incremented at each cycle of the clock. When the counter has reached the value of 0xFFFF, an overflow occurs and bit OVF of the System Timer Overflow Register (TMOVFR) is set to '1', thus asserting an interrupt request signal. In the next processor clock cycle, the timer counter is reloaded with the program specified value from TMRLR register, and continues counting up.

The time-base for System Timer is CCLK and as noted in Chapter 7, CCLK can be divided by setting the Clock Gear Control Register (CGBCNT0) as necessary. The user should note that changing CGBCNT0 register will affect the operation of other peripherals in this LSI.

To avoid timing conflicts in the System Timer block, the program must leave the following interval between successive writes to TMRLR register.

$$n \times HCLK + 79 \times CCLK$$

The System Timer operation can be stopped at anytime by writing a "0" to the System Timer Enable Register (TMEN).

2. Timer overflow interrupt

Timer counter overflow triggers an interrupt request and sets the OVF bit in the counter overflow register (TMOVFR) to "1."

Writing "1" to TMOVFR clears the interrupt request.

14.3.2 Auto Reload Timers

There are six auto reload timers, each with its own counter and settings for such parameters as operating clock frequency and operation mode.

1. One-shot operation

   A match between the counter (TIMECNTx) and compare (TIMECMPx) registers reloads TIMECNTx from the corresponding timer base register (TIMEBASEx) and resets the START bit to "0" to stop counting.
   The timer then generates an interrupt request if interrupts are enabled.
   Note, however, that the hardware does not automatically cancel this interrupt request. The interrupt handler must clear interrupt by writinga "1" to the STATUS bit of TIMESTATx which will in turn reset this bit to "0".

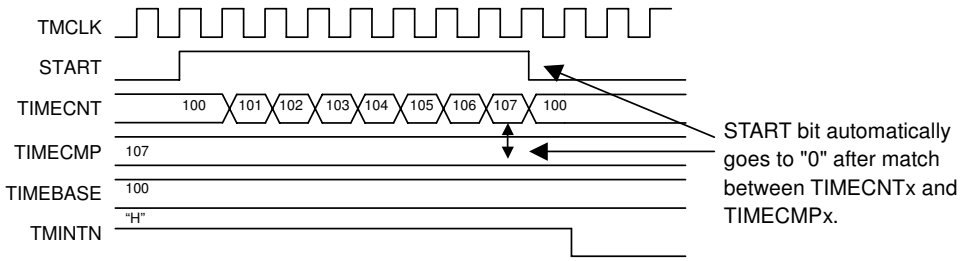


Figure 14.3 One-Shot Operation

2. Interval timer operation

   A match between TIMECNTx and TIMECMPx reloads TIMECNTx from TIMEBASEx, but does not reset the START bit to "0," so counting continues.
   The timer then generates an interrupt request if interrupts are enabled.
   Note, however, that the hardware does not automatically cancel this interrupt request. The interrupt handler must do so by writing a "1" to the STATUS bit of TIMESTATx which will in turn reset it to "0".

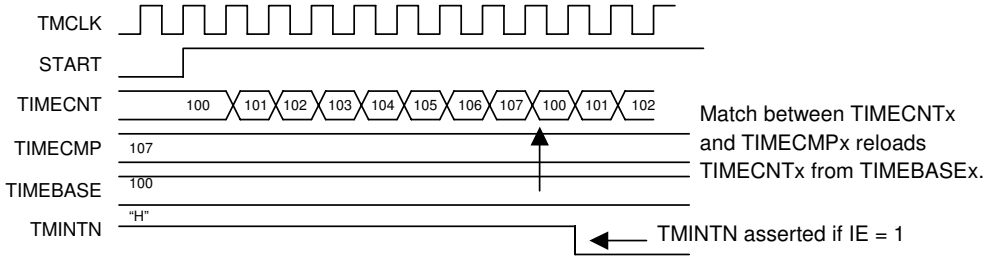


Figure 14.4 Interval Timer Operation

14.3.3 Specifying Clock and Starting Auto Reload Timers

The CLKSEL[2:0] bits in a timer control register (TIMECNTLx) specify, for the corresponding timer, the frequency divisor for deriving the operating clock from CCLK. The choices available are 1, 2, 4, 8, 16, and 32.

Writing "1" to the START bit in the same register starts the timer.
For one-shot operation, a match between the counter and compare registers automatically resets the START bit to "0" to stop counting. Interval timer operation does not clear START, so counting continues.

Notes:

1. Operation is not guaranteed for clock (CLKSEL[2:0]) settings "110" and "111."
2. Do not change the clock (CLKSEL[2:0]) setting while the timer is in operation.

Chapter 15

PWM Generator

Chapter 15 PWM Generator

15.1 Overview

This block provides two pulse width modulation (PWM) outputs, PWMOUT0 and PWMOUT1, that support changing the duty under program control with each cycle. These outputs have a resolution of 16 bits.

15.1.1 Components

The block diagram in Figure 15.1 illustrates the different components within the PWM Generator.

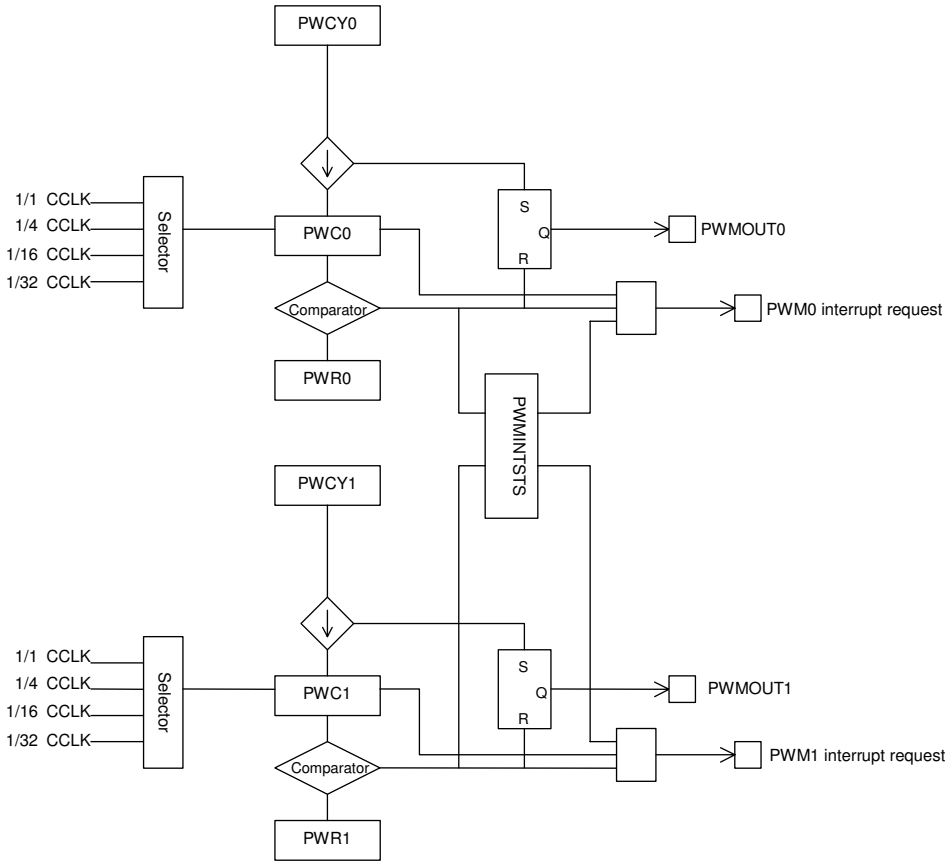


Figure 15.1 PWM Generator Components

15.1.2 Pin List

Pin Name	I/O	Description
PWMOUT0	O	PWM output 0. Secondary function for general-purpose port PIOB[6].
PWMOUT1	O	PWM output 1. Secondary function for general-purpose port PIOB[7].

15.1.3 Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0xB7D00000	PWM register 0	PWR0	R/W	16	0x0000
0xB7D00004	PWM period register 0	PWCY0	R/W	16	0x0000
0xB7D00008	PWM counter 0	PWC0	R/W	16	0x0000
0xB7D0000C	PWM control register 0	PWCON0	R/W	16	0x0000
0xB7D00020	PWM register 1	PWR1	R/W	16	0x0000
0xB7D00024	PWM period register 1	PWCY1	R/W	16	0x0000
0xB7D00028	PWM counter 1	PWC1	R/W	16	0x0000
0xB7D0002C	PWM control register 1	PWCON1	R/W	16	0x0000
0xB7D0003C	Interrupt status register	PWINTSTS	R/W	16	0x0000

15.2 Register Descriptions

15.2.1 PWM Registers (PWR0 and PWR1)

These registers specify the duty, the High level pulse width, for the corresponding PWM outputs.
Note that the setting must be less than that in the corresponding PWM period register (PWCYx), which specifies the PWM output frequency and thus the period. (See register description below.)

$$\text{Duty} = ((\text{PWM} - \text{PWCY} + 1) / (65536 - \text{PWCY})) \times 100\ (\%)$$

The CPU has read/write access to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWR0 to 1	PWR[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D00000 (CH0), 0xB7D00020 (CH1)
Access: R/W
Access size: 16 bits

15.2.2 PWM Period Registers (PWCY0 and PWCY1)

These registers specify the periods for the corresponding PWM outputs according to the following formula.

$$T_{pwmcyc} = (65536 - PWCYx) / ((\text{frequency divisor from PWCKx}) \times CCLK) \quad [sec]$$

where the PWCKx bits in the corresponding PWM control (PWCONx) register specify the frequency divider for CCLK.

For example, when CCLK is 16.67 MHz and PWCKx is set to "01" (divisor is 1/4), and PWCYx is set to 0xffff, the period for the corresponding PWM output can be calculated as below:

$$T_{pwmcyc} = (65536 - 65535) / (1/4 \times (16.67 \times 10^6)) \quad [sec]$$

$$= 239.952 \text{ [nsec]}$$

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWCY0 to 1	PWCY[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D00004 (CH0), 0xB7D00024 (CH1)
Access: R/W
Access size: 16 bits

Note

Note that the setting must be greater than that in the corresponding PWM register (PWRx), which specifies the duty.

15.2.3 PWM Counter Registers (PWC0 and PWC1)

These up counters automatically reload from the corresponding PWM period register (PWCYx) when they overflow.
The CPU has read/write access to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWC0 to 1	PWC[15:0]															
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D00008 (CH0), 0xB7D00028 (CH1)
Access: R/W
Access size: 16 bits

Note

Writing to PWCx simultaneously writes the same value to PWCYx.

15.2.4 PWM Control Registers (PWCON0 and PWCON1)

These registers start and stop the PWM counter, specify the operating clock, and specify the interrupt source.
The CPU has read/write access to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWCON0 to 1	– *	– *	– *	– *	– *	– *	– *	– *	PWC0OV/ PWC1OV	INTIE0/ INTIE1	– *	– *	– *	PWCK0/ PWCK1		PW0R/ PW1R
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D0000C (CH0), 0xB7D0002C (CH1)
Access: R/W
Access size: 16 bits

Note
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **PW0R/PW1R** (bit 0):
  This bit controls PWM counter operation.

PW0R/PW1R	Description
0	Stop PWCx counter
1	Start/enable PWCx counter

- **PWCK0/PWCK1** (bits 1 and 2):
  These bits specify the frequency divisor for deriving the PWCx operating clock from CCLK.

PWCK0/PWCK1		Description
2	1	
0	0	1/1CCLK
0	1	1/4CCLK
1	0	1/16CCLK
1	1	1/32CCLK

- **INTIE0/INTIE1** (bit 6):
  This bit controls interrupt requests.

INTIE0/INTIE1	Description
0	Disable interrupts
1	Enable interrupts

- **PWC0OV/PWC1OV** (bit 7):
  This bit specifies the interrupt request trigger and output timing.

PWC0OV/PWC1OV	Description
0	When PWCx = PWRx (at PWM output falling edge)
1	PWCx overflow (at PWM output rising edge)

15.2.5 PWM Interrupt Status Register (PWINTSTS)

This register gives the status of PWM output interrupt requests and provides bits for clearing them. The CPU has read/write access to this register.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWINTSTS	– *	– *	– *	– *	– *	– *	INT1S	INT0S	– *	– *	– *	– *	– *	– *	INT1CLR	INT0CLR
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB7D0003C
Access: R/W
Access size: 16 bits

Notes
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
Writes to bits 2 to 15 in this register are ignored.

Bit Descriptions

- **INT0CLR** (bit 0):
  Writing "1" to this bit clears the PWM output 0 interrupt request (INT0S).
  Writes of "0" are ignored. Reads always return "0."

- **INT1CLR** (bit 1):
  Writing "1" to this bit clears the PWM output 1 interrupt request (INT1S).
  Writes of "0" are ignored. Reads always return "0."

- **INT0S** (bit 8):
  This flag gives the interrupt request status for PWM output 0.

INT0S	Description
0	No interrupt pending
1	Interrupt pending

- **INT1S** (bit 9):
  This flag gives the interrupt request status for PWM output 1.

INT1S	Description
0	No interrupt pending
1	Interrupt pending

15.3 Description of Operation

15.3.1 PWM Operation

This block provides two output pins: PWMOUT0 and PWMOUT1.
To start PWM output, write "1" to PWxR which will in turn start the counter (PWCx) and set the output flip-flop automatically to "1" to drive the PWMOUTx pin output at High level.
When the PWCx contents match those of PWRx, the output flip-flop switches to "0" to drive the PWMOUTx pin output at Low level. If INTIEx is "1" and PWCxOV is "0," an interrupt request will signal this transition.
The hardware repeats the above two steps, producing PWMOUTx pin output whose duty is under program control, until the program resets PWxR to "0."

Notes

- Depending on the operating clock selected, the PWM output duty is sometimes less than specified for the first cycle immediately after starting output.
- Setting both PWCx and PWRx to 0x0000 produces an output duty (High level pulse width) of 1/65536. Increasing PWRx increases the output duty. The maximum value, 0xFFFF, produces an output duty of 65536/65536 or 100%.

  The PWM generator cannot produce 0/65536 (0%) duty.

15.3.2 Timing Examples

Figure 15.2 summarizes basic output operation. Figure 15.3 shows the results of changing PWM output timing.

Notes

- Write to PWC/PWCY/PWR under PWM stop state.
- Next PWM interrupt will be merged if the interrupt occurs before clearing the status bit of previous interrupt.

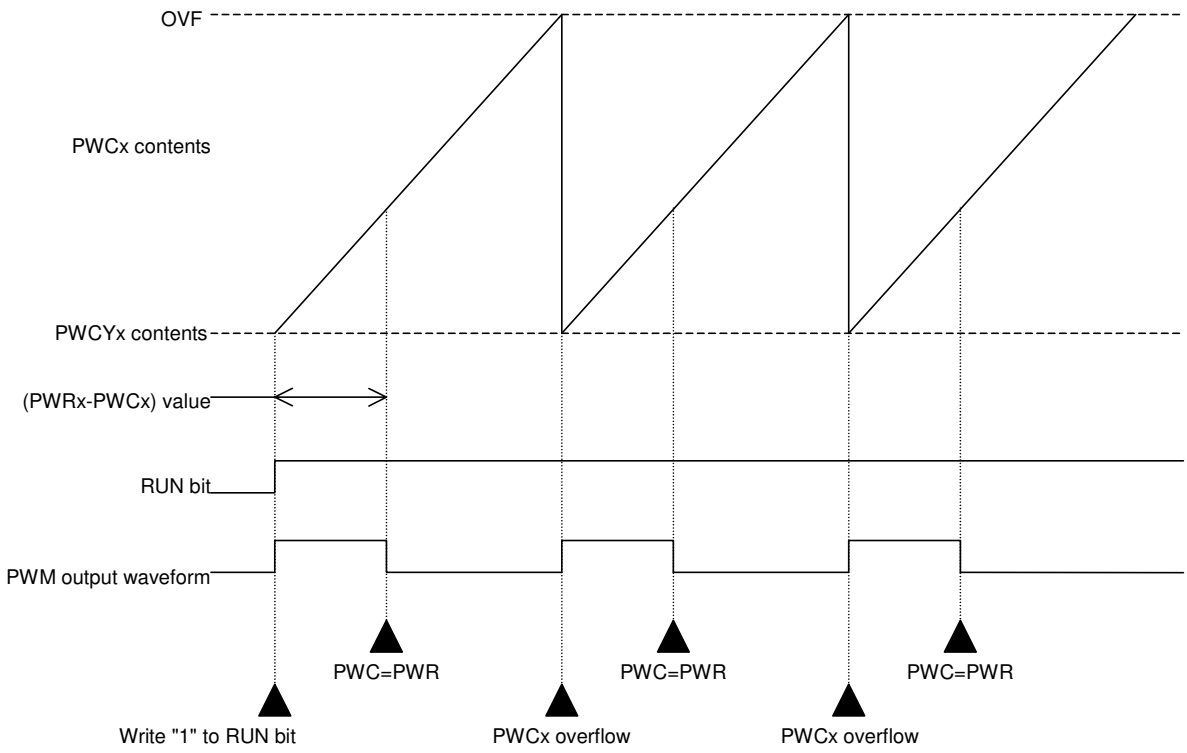


Figure 15.2 Basic Output Operation

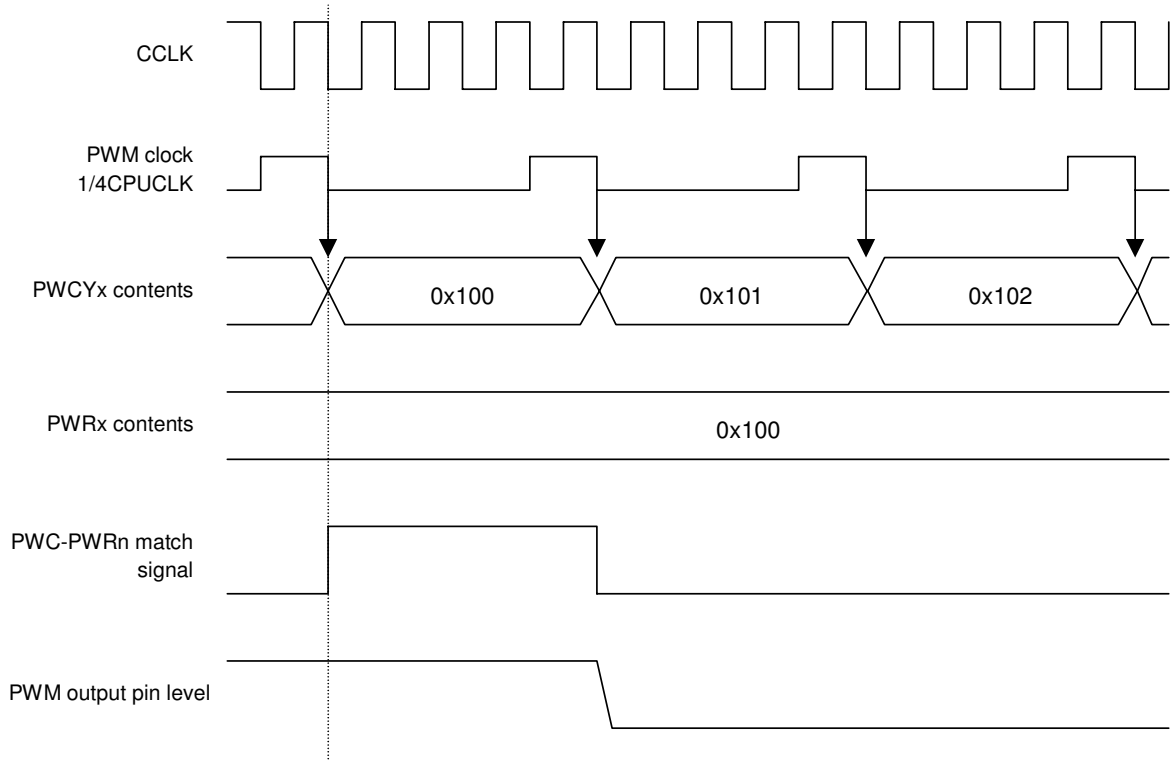


Figure 15.3 Changing PWM Output Timing

Chapter 16

SIO

Chapter 16 SIO

16.1 Overview

This serial port transfers data, synchronizing individual characters.
A dedicated baud rate generator provides program control over a baud rate clock that determines the transfer speed independently of the bus clock.
Frame parameters available include the character length, number of stop bits, and parity.

Features

- Full duplex asynchronous operation
- Sampling rate = baud rate × 16
- Data bits: 7 or 8
- Stop bits: 1 or 2
- Parity: even, odd, or none
- Error detection: parity, framing, and overrun errors
- Loopback function: ON/OFF control plus forced addition of parity, framing, and overrun errors
- Baud rate: Dedicated baud rate generator, with 8-bit counter, providing a baud rate clock that is independent of the bus clock. This internal baud rate clock stops in HALT mode.

16.1.1 Components

Figure 16.1 is a block diagram showing asynchronous serial interface components.

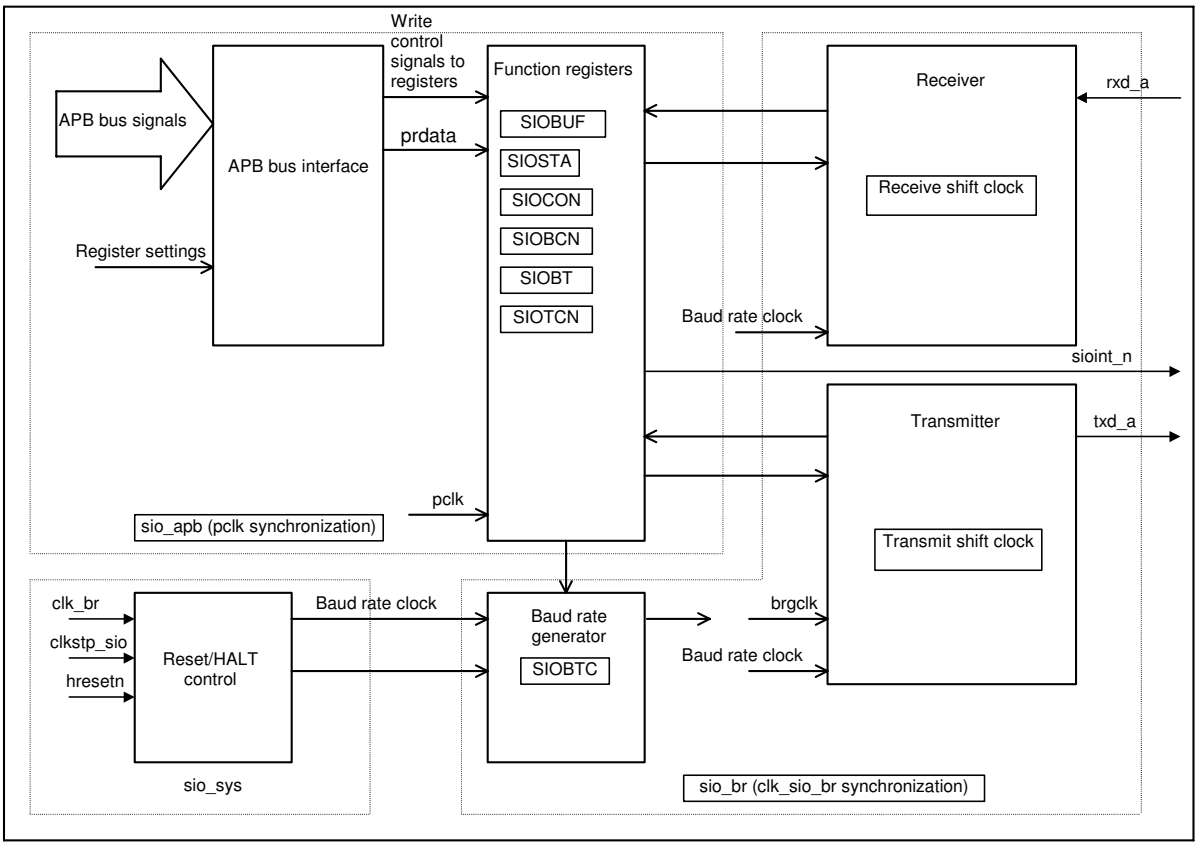


Figure 16.1 Block Components

16.1.2 Pin List

Pin Name	I/O	Description
STXD	O	SIO transmitter output. Secondary function for PIOA[8].
SRXD	I	SIO receiver input. Secondary function for PIOA[9].

16.1.3 Control Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0xB8002000	SIO transfer buffer register	SIOBUF	R/W	32	0x00000000
0xB8002004	SIO status register	SIOSTA	R/W	32	0x00000000
0xB8002008	SIO control register	SIOCON	R/W	32	0x00000000
0xB800200C	Baud rate control register	SIOBCN	R/W	32	0x00000000
0xB8002010	(reserved)		—	32	0x00000000
0xB8002014	Baud rate timer register	SIOBT	R/W	32	0x00000000
0xB8002018	SIO test control register	SIOTCN	R/W	32	0x00000000

16.2 Control Register Descriptions

16.2.1 Transfer Buffer Register (SIOBUF)

This register holds transfer data.
The CPU has read/write access to this register, but the same register has two separate functions. It functions as a receive buffer for reads and as a transmit buffer for writes.
When a receive operation ends, the interface transfers the contents of the receive shift register to the receive buffer and generates a receive interrupt request. The receive buffer contents remain valid until the end of the next receive operation.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOBUF	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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	– *	– *	– *	– *	– *	– *	– *	– *	SIOBUF[7:0]							
After a reset	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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

16.2.2 SIO Status Register (SIOSTA)

This register contains flags indicating asynchronous serial (SIO) interface transmitter and receiver ready status and ones indicating errors detected during receive operations.
The CPU has read/write access to this register.
The interface updates the lowest three bits at the end of each receive operation to indicate any errors detected. Once an error bit goes to "1" to indicate detection of a receive error, it remains "1" until the program writes "0" to it. It does not automatically return to "0" if the next receive operation is free of the corresponding error.
The program is also responsible for resetting the transmitter and receiver ready flags to "0" by writing "1" to them. Writes of "0" are ignored.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOSTA	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
After a 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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **FERR** (bit 0):
  A "1" in this bit indicates detection of a framing error. This error flag goes to "1" when a stop bit of "0" instead of "1" is detected which indicates a loss of frame synchronization.

FERR	Description
0	No framing error detected
1	Framing error detected

- **OERR** (bit 1):
  A "1" in this bit indicates detection of an overrun error. This bit goes to "1" if the CPU has not read the previously received byte from the receive buffer (SIOBUF) register when the interface transfers the next one there from the receive shift register, invalidating the previous data.

OERR	Description
0	No overrun error detected
1	Overrun error detected

- **PERR** (bit 2):
  A "1" in this bit indicates detection of a parity error. This bit goes to "1" if the parity bit received does not match the parity calculated for the data bits received.

PERR	Description
0	No parity error detected
1	Parity error detected

- **RVIRQ** (bit 4):
  This bit indicates the receiver ready status. This bit goes to "1" when the interface updates the contents of the receive buffer (SIOBUF) from the receive shift register.
  Branching to an interrupt handler or reading the received byte from SIOBUF does not automatically reset this bit to "0." The program must reset it to "0" by writing a "1" to it.
  Note that the interface always updates SIOBUF and sets this bit to "1" whenever a receive operation is complete regardless of any framing, overrun, or parity errors detected.

RVIRQ	Description
0	Receiver not ready
1	Receiver ready

- **TRIRQ** (bit 5):
  This bit indicates the transmitter ready status. This bit goes to "1" when the interface transfers data from the transmit buffer (SIOBUF) to the transmit shift register.
  Branching to an interrupt handler or writing transmit data to SIOBUF does not automatically reset this bit to "0." The program must reset it to "0" by writing "1" to it.

TRIRQ	Description
0	Transmitter not ready
1	Transmitter ready

16.2.3 SIO Control Register (SIOCON)

This register specifies the frame format for transfers over the asynchronous serial (SIO) interface. The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOCON	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
After a 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

Notes
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
Operation is not guaranteed when the program changes the contents of this register during a transmit or receive operation.

Bit Descriptions

- LN (bit 0): This bit specifies the number of data bits for SIO transfers.

LN	Description
0	8 data bits
1	7 data bits

- **PEN** (bit 1):
This bit controls the use of parity bits during SIO transfers. Setting this bit to "1" adds a parity bit to outgoing frames and checks parity for incoming ones.

PEN	Description
0	Disable parity
1	Enable parity

- **EVN** (bit 2):
This bit specifies the parity logic (odd or even) for SIO transfers. It is only valid, however, when PEN is "1."

EVN	Description
0	Odd parity
1	Even parity

- **TSTB** (bit 3):
This bit specifies the number of stop bits for SIO transfers.

TSTB	Description
0	2 stop bits
1	1 stop bit

16.2.4 Baud Rate Control Register (SIOBCN)

This register starts and stops the baud rate timer counter (SIOBTC).
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOBCN	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **BGRUN** (bit 4):
  This bit controls SIOBTC operation.

BGRUN	Description
0	Stop counter
1	Start counter

16.2.5 Baud Rate Timer Register (SIOBT)

This register holds the starting value that SIOBTC overflow automatically reloads into the baud rate timer counter (SIOBTC) register.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOBT	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

16.2.6 SIO test control Register (SIOTCN)

This register is for testing the SIO function.
The CPU has read/write access to this register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOTCN	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *
After a 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
– *	– *	– *	– *	– *	– *	– *	– *	LBTST	– *	– *	– *	– *	– *	WPERR	MFERR
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
– *: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **MFERR** (bit 0):
  Setting this bit to "1" during loopback testing (LBTST = "1") forces framing errors.

MFERR	Description
0	Skip framing errors
1	Add framing errors

- **MPERR** (bit 1):
  Setting this bit to "1" during loopback testing (LBTST = "1") forces parity errors.

MPERR	Description
0	Skip parity errors
1	Add parity errors

- **LBTST** (bit 7):
  Setting this bit to "1" internally connects the transmitter outputs to the receiver inputs for testing.

LBTST	Description
0	Disable loopback
1	Enable loopback

16.3 Description of Operation

Settings in the SIOCON register specify the frame format: character length, number of stop bits, and parity. Figure 16.2 gives the register settings for sample formats.

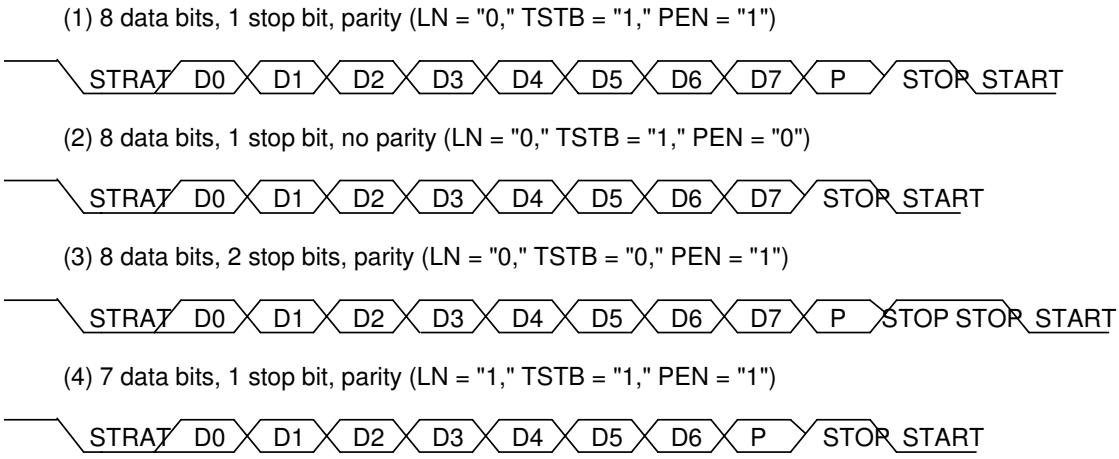


Figure 16.2 Sample Frame Formats

16.3.1 Transmitting Data

Writing 8-bit data to the transmit buffer (SIOBUF) register starts a transmit operation. Note that the interface does not transmit the top bit if the character length is 7 bits.
When the interface transfers the SIOBUF contents to the transmit shift register, it generates a transmitter ready interrupt request, and sets the TRIRQ bit in the SIOSTA register to "1" to indicate that SIOBUF is now empty, ready for the next write of transmit data to SIOBUF.
The interface then transmits the frame one bit at a time as specified by the SIOCON register settings: start bit, seven or eight data bits, optional parity bit, and one or two stop bits.

16.3.2 Receiving Data

When the interface detects the start bit for a new frame, it begins latching the data bits into the receive shift register as specified by the frame format in the SIOCON register.
When the interface detects a stop bit, it transfers the received data from the shift register to the receive buffer (SIOBUF) register, signals any errors detected by writing "1" to the corresponding bits (PERR, OERR, and FERR) in the SIOSTA register, sets the RVIRQ bit in the SIOSTA register to "1," and generates a receive ready interrupt request to indicate that the receive shift register is empty, ready to receive the next frame.

16.3.3 Generating Baud Rate Clock

Figure 16.3 gives a block diagram showing the following registers for the baud rate generator block.

- Baud rate timer counter (SIOBTC), with 8-bit counter
- Baud rate timer (SIOBT) register
- Baud rate control (SIOBCN) register

SIOBTC overflow generates the baud rate clock. It also reloads SIOBTC with the starting value from SIOBT so that counting continues.
Writing to SIOBT simultaneously writes the same value to SIOBTC.

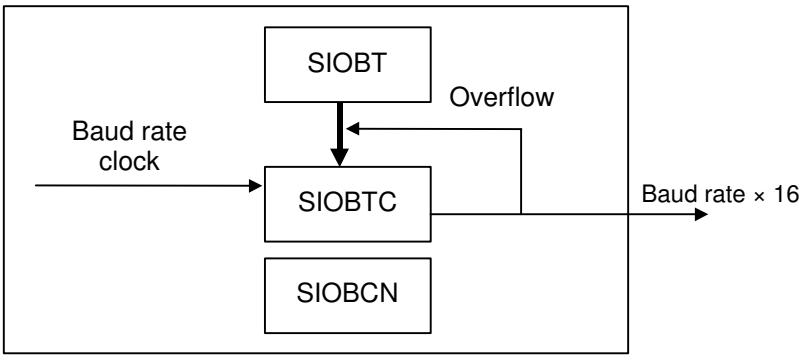


Figure 16.3 Baud Rate Generator Block Diagram

The following is the procedure for specifying the baud rate.

1. Calculate the starting value (D) from the desired baud rate frequency (B) and the baud rate clock frequency ($f_{(B)}$) using the following formula. Note, however, that D must be between 0 and 255.

$$D = 256 - \frac{f_{(B)}}{16 \times B} \qquad \left(B = f_{(B)} \times \frac{1}{256 - D} \times \frac{1}{16}\right)$$

B: Baud rate
$f_{(B)}$: Input counter clock frequency, CCLK (Hz)
D: Reload value (0 to 255)

2. Write the starting value (D) to SIOBT.

3. Set the BGRUN bit in the SIOBCN register to "1" to start counting by the SIOBTC register and thus baud rate clock output.
The interface is ready to transfer data once five baud rate clock cycles have elapsed.

16.3.4 Receive Interrupts

A receive ready interrupt request indicates that the interface has transferred data from the receive shift register to the receive buffer (SIOBUF) register, signaled any errors detected by writing "1" to the corresponding bits (PERR, OERR, and FERR) in the SIOSTA register, and set the RVIRQ bit in the SIOSTA register to "1."
This signal remains asserted until the program writes "1" to RVIRQ to negate the interrupt request. Assertion takes precedence over negation if there is any conflict.

16.3.5 Transmit Interrupts

A transmit ready interrupt request indicates that the interface has transferred data from the transmit buffer (SIOBUF) register to the transmit shift register and set the TRIRQ bit in the SIOSTA register to "1."

This signal remains asserted until the program writes "1" to TRIRQ to negate the interrupt request. Assertion takes precedence over negation if there is any conflict.

16.4 Important Usage Notes

1. Initialize the baud rate generator by writing to the baud rate timer (SIOBT) and baud rate control (SIOBCN) registers in that order. After the write to SIOBCN, the interface allows five baud rate clock cycles for the clock period to stabilize before starting clock output.

2. Operation is not guaranteed if the program changes the SIOCON register settings after transfers start.

3. The interrupt handler must respond and complete its processing within the interval for transferring a single frame.

4. The interrupt handler must clear the interrupt source flag before reading from or writing to the SIOBUF register.

5. Leave at least five baud rate clock cycles between successive writes to the SIOBT register.

6. Register access from the bus must use word access.

Chapter 17

UART with FIFO(16byte)

Chapter 17 UART with FIFO(16byte)

17.1 Overview

The UART built into the LSI is an asynchronous communication element (ACE) functionally equivalent to the industry standard 16550A with separate 16-byte queues for both transmitting and receiving.
This UART functions as an I/O interface, converting serial data from a modem or other peripheral equipment into parallel data and converting parallel data from the CPU into serial data.
After a reset, the UART registers function as an industry standard 16450.
During buffered (16550A) operation, each queue holds up to 16 bytes of data.
The receive queue provides three error bits for each byte of data.
The CPU can read the ACE status at any time. The status information available includes the type and status of transfer operations in progress, parity, overrun, framing, and other errors, and the status of break and other interrupts.

Features

- Full duplex operation
- Reporting functions for all states
- 16-byte queues for both transmitting and receiving
- Transmit, receive, and line status data set interrupts plus independent control over each queue
- Modem control signals CTS, DCD, DSR, DTR, RI, and RTS
- Programmable serial interface
  - Choice of 5, 6, 7, or 8 bits per character
  - Choice of odd, even, or no parity
  - 1, 1.5, or 2 stop bits

17.1.1 Components

Figure 17.1 shows the interface components.

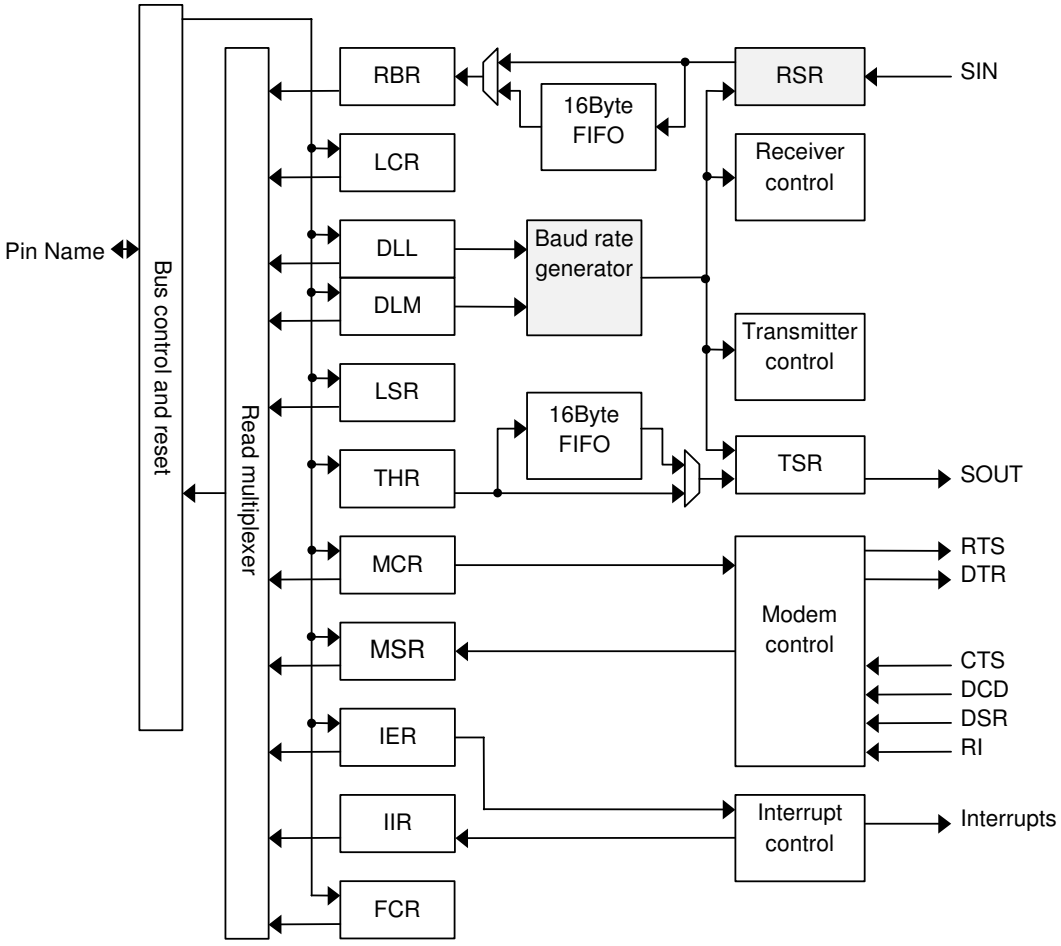


Figure 17.1 Asynchronous Serial Interface Components

17.1.2 Pins

Pin Name	I/O Direction	Description
SIN	I	UART serial data in. Secondary function for PIOA[0].
SOUT	O	UART serial data out. Secondary function for PIOA[1].
CTS	I	UART clear to send. Secondary function for PIOA[2].
DSR	I	UART data set ready. Secondary function for PIOA[3].
DCD	I	UART data carrier detect. Secondary function for PIOA[4].
DTR	O	UART data terminal ready. Secondary function for PIOA[5].
RTS	O	UART request to send. Secondary function for PIOA[6].
RI	I	UART ring indicator. Secondary function for PIOA[7].

Note: The interface does not support OUT1 and OUT2.

17.1.3 Register List

Address	Name	Symbol	R/W	Size	Initial Value
0xB7B00000	Receiver Buffer Register	UARTRBR	R	8	Indeterminate
0xB7B00000	Transmitter Holding Register	UARTTHR	W	8	Indeterminate
0xB7B00004	Interrupt Enable Register	UARTIER	R/W	8	0x00
0xB7B00008	Interrupt Identification Register	UARTIIR	R	8	0x01
0xB7B00008	FIFO Control Register	UARTFCR	W	8	0x00
0xB7B0000C	Line Control Register	UARTLCR	R/W	8	0x00
0xB7B00010	Modem Control Register	UARTMCR	R/W	8	0x00
0xB7B00014	Line Status Register	UARTLSR	R/W	8	0x60
0xB7B00018	Modem Status Register	UARTMSR	R/W	8	Contents depend on pin states
0xB7B0001C	Scratch Register	UARTSCR	R/W	8	Indeterminate
0xB7B00000	Divisor Latch (LSB)	UARTDLL	R/W	8	Indeterminate
0xB7B00004	Divisor Latch (MSB)	UARTDLM	R/W	8	Indeterminate

17.2 Register Descriptions

17.2.1 Receiver Buffer Register (UARTRBR)

This data register holds 5 to 8 bits of data depending on the character length.
Data transfers always start from the lowest bit (bit 0) in the serial data.
Double-buffering allows the software to read ACE data registers even while the UART is converting data between parallel and serial formats.
The CPU has only read access to this register.
The register contents after a reset are indeterminate.

	7	6	5	4	3	2	1	0
UARTRBR	RBR[7:0]							
After a reset	(Indeterminate)							

Address: 0xB7B00000
Access: R
Access size: 8 bits

17.2.2 Transmitter Holding Register (UARTTHR)

This data register holds 5 to 8 bits of data depending on the character length.
If the character length is less than 8 bits, the bits must be at the low end because data transfers always start from the lowest bit (bit 0).
Double-buffering allows the software to read ACE data registers even while the UART is converting data between parallel and serial formats.
The CPU has only write access to this register.

	7	6	5	4	3	2	1	0
UARTTHR	THR[7:0]							
After a reset	(Indeterminate)							

Address: 0xB7B00000
Access: W
Access size: 8 bits

17.2.3 Interrupt Enable Register (UARTIER)

This register is for independently enabling four serial interface interrupts.
The CPU has read/write access to this register.
The register contents after a reset are 0x00.

	7	6	5	4	3	2	1	0
UARTIER	IER[7:0]							
After a reset	0	0	0	0	0	0	0	0

Address: 0xB7B00004
Access: R/W
Access size: 8 bits

Bit Descriptions

- **IER[0]** (bit 0)
  This bit enables/disables received data available interrupts (plus character timeout interrupts during buffered operation).

IER[0]	Description
0	Disable received data available interrupts (plus character timeout interrupts during buffered operation)
1	Enable received data available interrupts (plus character timeout interrupts during buffered operation)

- **IER[1]** (bit 1)
  This bit enables/disables transmitter holding register empty interrupts.

IER[1]	Description
0	Disable transmitter holding register empty interrupts
1	Enable transmitter holding register empty interrupts

- **IER[2]** (bit 2)
  This bit enables/disables receiver line status interrupts.

IER[2]	Description
0	Disable receiver line status interrupts
1	Enable receiver line status interrupts

- **IER[3]** (bit 3)
  This bit enables/disables modem status interrupts.

IER[3]	Description
0	Disable modem status interrupts
1	Enable modem status interrupts

- **IER[7:4]** (bits 7 to 4)
  Unused. These return "0" for reads.

17.2.4 Interrupt Identification Register (UARTIIR)

This register indicates whether there is a prioritized interrupt pending and, if so, the source for that interrupt.
The IIR field gives the source for the interrupt with the highest priority. The hardware does not recognize other interrupts until the CPU processes the current interrupt.
The CPU has only read access to this register.
The register contents after a reset are 0x01.

	7	6	5	4	3	2	1	0
UARTIIR	IIR[7:0]							
After a reset	0	0	0	0	0	0	0	1

Address: 0xB7B00008
Access: R
Access size: 8 bits

Bit Descriptions

- **IIR[0]** (bit 0)
  This bit indicates whether there is an interrupt pending.

IIR[0]	Description
0	There is an interrupt pending.
1	There are no interrupts pending.

- **IIR[3:1]** (bits 3 to 1)
  These bits indicate the interrupt source.

Interrupt		Setting and Resetting Interrupt			Notes
IIR [3:1]	Priority level	Interrupt Flag	Interrupt Source	Resetting Interrupt	
011	1	Receiver Line Status	Overrun Error / Parity Error / Framing Error / Break Interrupt	Read LSR	
010	2	Received Data Available	Unbuffered (16450) operation: Receive data available Buffered operation: Trigger level reached	Read RBR (until queue below trigger level)	
110	2	Character Timeout Indication	There is at least one character in the receive queue, and there has been no data movement into or off the receive queue for the equivalent of four characters.	Read RBR	Buffered operation only
001	3	Transmitter Holding Register Empty (THRE)	Unbuffered (16450) operation: It is safe to write to THR. Buffered operation: Transmit queue is empty.	Read IIR or write to THR.	
000	4	Modem Status	CTS, DSR, RI, or DCD	Read MSR	

- **IIR[5:4]** (bits 5 to 4)
  Unused. These return "0" for reads.

- **IIR[7:6]** (bits 7 to 6)
  These bits indicate buffered operation.

IIR[7:6]		Description
7	6	
0	0	Unbuffered (16450) operation
0	1	Unused
1	0	Unused
1	1	Buffered operation

17.2.5 FIFO Control Register (UARTFCR)

This register enables buffered operation, clears the queues, and specifies the trigger level for the receive queue.
The CPU has only write access to this register.

	7	6	5	4	3	2	1	0
UARTFCR	FCR[7:0]							
After a reset	0	0	0	0	0	0	0	0

Address: 0xB7B00008
Access: W
Access size: 8 bits

Bit Descriptions

- **FCR[0]** (bit 0)
  This bit switches buffered operation on ("1") and off ("0").

FCR[0]	Description
0	Unbuffered (16450) operation
1	Buffered operation

Note: Changing this bit automatically clears both queues.

- **FCR[1]** (bit 1)
  RCVR queue reset. Setting this bit to "1" clears the receive queue.

FCR[1]	Description
0	Normal operation
1	Clear receive queue

Note: This operation does not clear the receive shift register.

- **FCR[2]** (bit 2)
  XMIT queue reset. Setting this bit to "1" clears the transmit queue.

FCR[2]	Description
0	Normal operation
1	Clear transmit queue

Note: This operation does not clear the transmit shift register.

- **FCR[3]** (bit 3)
  DMA mode select

  Note: This LSI does not support DMA transfers to or from the UART.

- **FCR[5:4]** (bits 5 to 4)
  Reserved

- **FCR[7:6]** (bits 7 to 6)
  RCVR queue interrupt trigger level. These bits specify the trigger level for receive queue interrupts.

FCR[7:6]		Description
7	6	
0	0	1 byte
0	1	4 byte
1	0	8 byte
1	1	14 byte

17.2.6 Line Control Register (UARTLCR)

This register specifies the character format.
The CPU has read/write access to this register.
Allowing reads eliminates the need to save line characters separately in system memory.
The register contents after a reset are 0x00.

	7	6	5	4	3	2	1	0
UARTLCR	LCR[7:0]							
After a reset	0	0	0	0	0	0	0	0

Address: 0xB7B0000C
Access: R/W
Access size: 8 bits

Bit Descriptions

- **LCR[1:0]** (bits 1 to 0)
  These bits specify the character length (5 to 8 bits).

LCR[1:0]		Description
LCR[1]	LCR[0]	
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

- **LCR[2]** (bit 2)
  This bit specifies the number of stop bits for transmitting characters.

LCR[2]	Description
0	1 stop bit
1	2 stop bits (1.5 for character length = 5)

- **LCR[3]** (bit 3)
  This bit controls the use of parity.

LCR[3]	Description
0	No parity
1	Parity used

- **LCR[4]** (bit 4)
  This bit specifies even or odd parity. This setting is only valid, however, when parity is enabled (LCR[3] = "1").

LCR[4]	Description
0	Odd parity
1	Even parity

- **LCR[5]** (bit 5)
  LCR[5] (bit 5) Stick parity. A "1" in this bit sets all parity bits to a fixed value, the inverse of the bit in LCR[4]. This setting is only valid, however, when parity is enabled (LCR[3] = "1"). The receiver can therefore check the parity using a known state.

LCR[5:3]			Description
LCR[5]	LCR[4]	LCR[3]	
0	0	1	Odd parity
0	1	1	Even parity
1	0	1	Fixed parity ("1")
1	1	1	Fixed parity ("0")

- **LCR[6]** (bit 6)
  Break control. Setting this bit to "1" sends a break signal by driving the serial output (SOUT) at Low level, or spacing state. Setting this bit to "0" disables the break.
  This break control function affects only SOUT, masking SOUT output. The transmit operation continues internally.
  The CPU can use this break control function to send a warning to the computer communications system terminal.

LCR[6]	Description
0	Normal operation
1	Transmit break signal

- **LCR[7]** (bit 7)
  Divisor latch access bit (DLAB). This bit switches access between UARTDLL and UARTDLM and UARTRBR, UARTTHR, and UARTIER.

LCR[7]	Description
0	Normal operation (accessing UARTRBR, UARTTHR, UARTIIR)
1	Divisor latch (UARTDLL and UARTDLM) access

17.2.7 Modem Control Register (UARTMCR)

This register controls the modem and data set interface.
The CPU has read/write access to this register.
This register provides direct control over RTS and DTR output. Writing "1" to the corresponding bit drives the pin output at Low level, its active state.
The register contents after a reset are 0x00.

	7	6	5	4	3	2	1	0
UARTMCR	MCR[7:0]							
After a reset	0	0	0	0	0	0	0	0

Address: 0xB7B00010
Access: R/W
Access size: 8 bits

Bit Descriptions

- **MCR[0]** (bit 0)
  Data terminal ready (DTR) output control.

MCR[0]	Description
0	Drive DTR pin output at High level
1	Drive DTR pin output at Low level

- **MCR[1]** (bit 1)
  Request to send (RTS) output control.

MCR[1]	Description
0	Drive RTS pin output at High level
1	Drive RTS pin output at Low level

- **MCR[2]** (bit 2)
  Reserved

  Note: This LSI does not support OUT1.

- **MCR[3]** (bit 3)
  Reserved

  Note: This LSI does not support OUT2.

- **MCR[4]** (bit 4)
  LOOPBACK control. The loop back configuration drives SOUT at High level and connects the transmitter shift register output to the receiver shift register input.

MCR[4]	Description
0	Normal operation
1	Loop back operation

- **MCR[7:5]** (bits 7 to 5)
  Unused. These return "0" for reads.

17.2.8 Line Status Register (UARTLSR)

This register displays the line status.
It is normally the first register that the CPU reads to determine the interrupt source or to poll the serial interface status.
The CPU has read/write access to this register.
The register contents after a reset are 0x60.

	7	6	5	4	3	2	1	0
UARTLSR	LSR[7:0]							
After a reset	0	1	1	0	0	0	0	0

Address: 0xB7B00014
Access: R/W
Access size: 8 bits

Bit Descriptions

- **LSR[0]** (bit 0)
  Data ready bit. This bit goes to "1" when an input character is ready for reading from the UARTRBR register. Reading UARTRBR resets this bit to "0."

LSR[0]	Description
0	UARTRBR does not contain valid data
1	UARTRBR contains valid data

- **LSR[1]** (bit 1)
  A "1" in this bit indicates an overrun error.
  For unbuffered (16450) operation, this indicates that the hardware has overwritten the contents of the UARTRBR register with a new character before the CPU read the former character.
  For buffered operation, this indicates that the queue is full when the next character is completely received. Reading the UARTLSR register clears the error. Note that the contents of the shift register are not stored in the queue, but overwritten by the next character.
  Reading the UARTLSR register resets this bit to "0."

LSR[1]	Description
0	No overrun error pending
1	Overrun error pending

- **LSR[2]** (bit 2)
  A "1" in this bit indicates a parity error. This setting is only valid, however, when parity is enabled (LCR[3] = "1").
  Reading the UARTLSR register resets this bit to "0."
  For buffered operation, this indicates a parity error in the data at the head of the queue. Note that parity errors for other characters in the queue do not affect LSR[2] contents.

LSR[2]	Description
0	No parity error pending
1	Parity error pending

- **LSR[3]** (bit 3)
  A "1" in this bit indicates a framing error.
  A framing error indicates that the corresponding character was not followed by a valid stop bit.
  This bit goes to "1" when the bit following the last data bit (or parity bit) is "0" (spacing level), not "1" (stop bit). Reading the UARTLSR register resets this bit to "0."
  For buffered operation, this bit goes to "1" when the character with the framing error reaches the head of the queue.

LSR[3]	Description
0	No framing error pending
1	Framing error pending

- **LSR[4]** (bit 4)
  LSR[4] (bit 4) A "1" in this bit indicates a break interrupt, "0" (spacing level) input for one frame interval (start bit, data bits, parity bit, and stop bit).
  This bit goes to "1" immediately for unbuffered operation.
  For buffered operation, the interface first adds a zero byte to the queue.
  Later, when that character reaches the head of the queue, the interface sets this bit to "1" and sets the parity, framing, and overrun error bits (LSR[3:1]) to "0" if the CPU has not already done so by reading the UARTLSR register.
  Reading the UARTLSR register resets this bit to "0."

LSR[4]	Description
0	No break interrupt pending
1	Break interrupt pending

LSR[1] to **LSR[4]** transitions to "1" represent sources for receiver line status interrupts, /!/priority/!/ 1 interrupts in the interrupt identification register (IIR). Setting IER[2] in the UARTIER register to "1" enables this interrupt.

- **LSR[5]** (bit 5)
  Transmitter holding register empty (THRE). A "1" in this bit indicates that the ACE is ready to read a new character to transmit.
  This bit goes to "1" when the current character moves from the UARTTHR register to the transmitter shift register. Writing to the UARTTHR register resets this bit to "0." Reading the UARTLSR register does not.
  For buffered operation, this bit goes to "1" when the transmit queue is empty. Writing a byte to the transmit queue resets this bit to "0."
  If IER[1] is "1," enabling THRE interrupts, the transition to "1" produces a THRE interrupt of priority 3 in the UARTIIR register. If THRE is the source for the interrupt indicated by the UARTIIR register, reading the UARTIIR register resets this bit to "0."

LSR[5]	Description
0	UARTTHR contains transmit data
1	UARTTHR ready to accept data

- **LSR[6]** (bit 6)
  Transmitter empty. This bit goes to "1" when the UARTTHR and transmitter shift (TSR) registers are both empty. Writing a character to the UARTTHR register clears the UARTLSR register to "0," driving SOUT at Low level until the hardware transmits that character. Reading the UARTLSR register does not reset this bit to "0."
  For buffered operation, this bit goes to "1" when the transmit queue and the shift register are both empty.

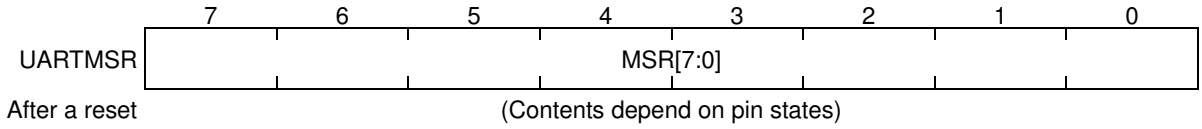
LSR[6]	Description
0	There is data in either UARTTHR or TSR
1	UARTTHR and TSR are both empty

- **LSR[7]** (bit 7)
  This bit is always "0" during unbuffered (16450) operation.
  For buffered operation, this bit goes to "1" when there is a data error (parity error, framing error, or break interrupt) anywhere in the queued data. Reading the data with the error from RBR or discarding it by clearing the entire queue and then reading LSR resets this bit to "0."

LSR[7]	Description
0	No data errors (buffered operation)
1	Parity error, framing error, or break interrupt (buffered operation)

17.2.9 Modem Status Register (UARTMSR)

This register allows the CPU to monitor the status of four control signal inputs from the modem or peripheral equipment: CTS, DSR, RI, and DCD. The CPU uses the UARTMSR register to access the ACE data bus interface and read these inputs.
In addition to the four bits giving the current status, this register also provides four delta bits indicating whether these inputs have changed since the CPU last read this register. A delta bit goes to "1" if the corresponding control signal has changed since the last read and returns to "0" when the CPU reads this register.
Bits MSR[4] to MSR[7] monitor CTS, DSR, RI, and DCD, respectively. A "1" indicates Low level input; "0," High. If IER[3] in the interrupt enable register is "1," enabling modem status interrupts, a "0" to "1" transition in the corresponding delta bits MSR[0] to MSR[3] produces a modem status interrupt of priority 4.
The CPU has read/write access to this register.
The register contents after a reset depend on the input pin states.



Address: 0xB7B00018
Access: R/W
Access size: 8 bits

Bit Descriptions

- **MSR[0]** (bit 0)
  Delta clear to send (DCTS). A "1" in this bit indicates a change in the CTS input state since the last time that the CPU read that state.

MSR[0]	Description
0	No change in CTS input
1	Change in CTS input

- **MSR[1]** (bit 1)
  Delta data set ready (DDSR). A "1" in this bit indicates a change in the DSR input state since the last time that the CPU read that state.

MSR[1]	Description
0	No change in DSR input
1	Change in DSR input

- **MSR[2]** (bit 2)
  Trailing edge of ring indicator (TERI). A "1" in this bit indicates a change from Low level to High in the RI input state since the last time that the CPU read that state. A transition from High level to Low does not affect this bit.

MSR[2]	Description
0	No change in RI input
1	Change in RI input

- **MSR[3]** (bit 3)
  Delta data carrier detect (DDCD). A "1" in this bit indicates a change in the DCD input state since the last time that the CPU read that state.

MSR[3]	Description
0	No change in DCD input
1	Change in DCD input

- **MSR[4]** (bit 4)
  Clear to send (CTS). This bit, the inverse of the CTS input from the modem, indicates whether the modem is ready to receive data from the serial interface's transmit output (SOUT) pin.
  In loop back mode (MCR[4] = "1"), this bit has the same value as MCR[1].

MSR[4]	Description
0	Modem is not ready to receive data from SOUT (CTS = 1)
1	Modem is ready to receive data from SOUT (CTS = 0)

- **MSR[5]** (bit 5)
  Data set ready (DSR). This bit, the inverse of the DSR input from the modem, indicates whether the modem is ready to transmit data to the serial interface's receive circuit.
  In loop back mode (MCR[4] = "1"), this bit has the same value as MCR[0].

MSR[5]	Description
0	Modem is not ready to transmit data (DSR = 1)
1	Modem is ready to transmit data (DSR = 0)

- **MSR[6]** (bit 6)
  Ring indicator (RI). This bit is the inverse of the RI input from the modem.
  In loop back mode (MCR[4] = "1"), this bit has the same value as MCR[2].

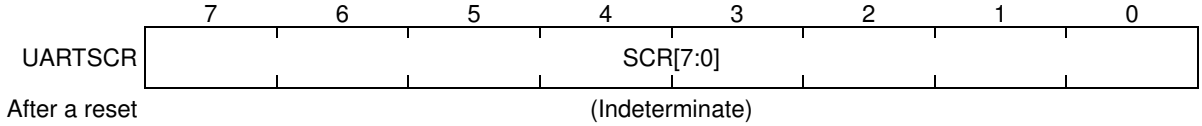
MSR[6]	Description
0	RI = 1
1	RI = 0

- **MSR[7]** (bit 7)
  Data carrier detect (DCD). This bit is the inverse of the DCD input from the modem.
  In loop back mode (MCR[4] = "1"), this bit has the same value as MCR[3].

MSR[7]	Description
0	DCD = 1
1	DCD = 0

17.2.10 Scratch Register (UARTSCR)

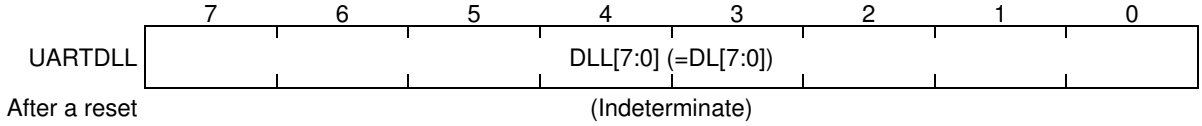
This register is for use as temporary data storage. It plays no role in ACE transfer operations.
The CPU has read/write access to this register.
The register contents after a reset are indeterminate.



Address: 0xB7B0001C
Access: R/W
Access size: 8 bits

17.2.11 Divisor Latch (LSB) (UARTDLL)

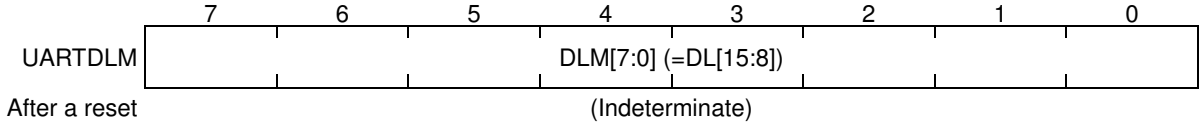
This register contains the lower half of the 16-bit divisor latch for the baud rate generator. For further details, see the section on baud rate clock generation.
The CPU has read/write access to this register when LCR[7] = 1.
The register contents after a reset are indeterminate.



Address: 0xB7B00000
Access: R/W
Access size: 8 bits

17.2.12 Divisor Latch (MSB) (UARTDLM)

This register contains the upper half of the 16-bit divisor latch for the baud rate generator. For further details, see the section on baud rate clock generation.
The CPU has read/write access to this register when LCR[7] = 1.
The register contents after a reset are indeterminate.



Address: 0xB7B00004
Access: R/W
Access size: 8 bits

17.3 Description of Operation

The registers LCR, IER, DLL, DLM, and MCR control serial interface operation. These control registers specify the character length, number of stop bits, parity, baud rate, modem interface, and other parameters. They can be written to in any order except that IER must come last because it enables interrupts. Once the serial interface has been configured for operation, these registers can be updated anytime that the interface is not transferring data.

17.3.1 Transmitting Data

Figure 17.2 gives the timing for transmitting data.

Writing data to the UARTTHR register sends that data to the transmit shift register via the transmit queue. Within 16 baud rate clock cycles after the THRE bit goes to "1," the hardware transmits the start bit and the data bits one bit at a time from the lowest bit. If the character length is 7 bits, the hardware does not transmit the top bit.
If the LCR[3] bit in the UARTLCR register enables parity, the hardware then transmits the parity bit.
Finally, the hardware transmits a stop bit to complete the frame.
When the hardware finishes transmitting the data, the LSR[5] bit in the UARTLSR register goes to "1" to indicate that the hardware is ready to transmit more data. Writing a byte to the transmit queue resets this bit to "0."
If the IER[1] bit enables THRE interrupts, this transition produces an interrupt of priority 3 in the UARTIIR register.
If THRE is the source for the interrupt indicated by the UARTIIR register, reading the UARTIIR register resets this bit to "0."

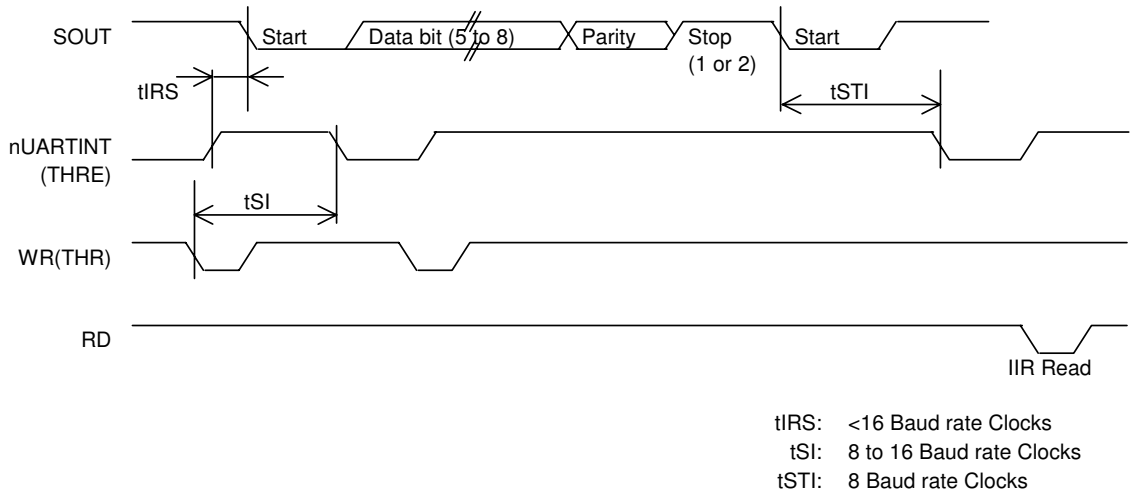


Figure 17.2 Transmit Timing

17.3.2 Receiving Data

Figure 17.4 gives the timing for receiving data; Figure 17.5, that for reading the first byte from the receive queue; Figure 17.6, that for reading the last byte.

The sampling frequency is eight times that of the baud rate clock.
When the hardware detects the start bit from the SIN pin, it first latches the data bits into the receive shift register. It then transfers the received character from the shift register to the UARTRBR register via the receive queue. When the character reaches the UARTRBR register, the LSR[0] bit in the UARTLSR register goes to "1" to indicate that there is valid data in the UARTRBR register. Reading the UARTRBR register resets this bit to "0."

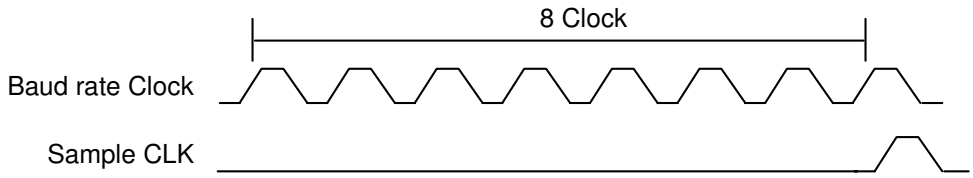


Figure 17.3 Baud Rate and Sampling Clocks

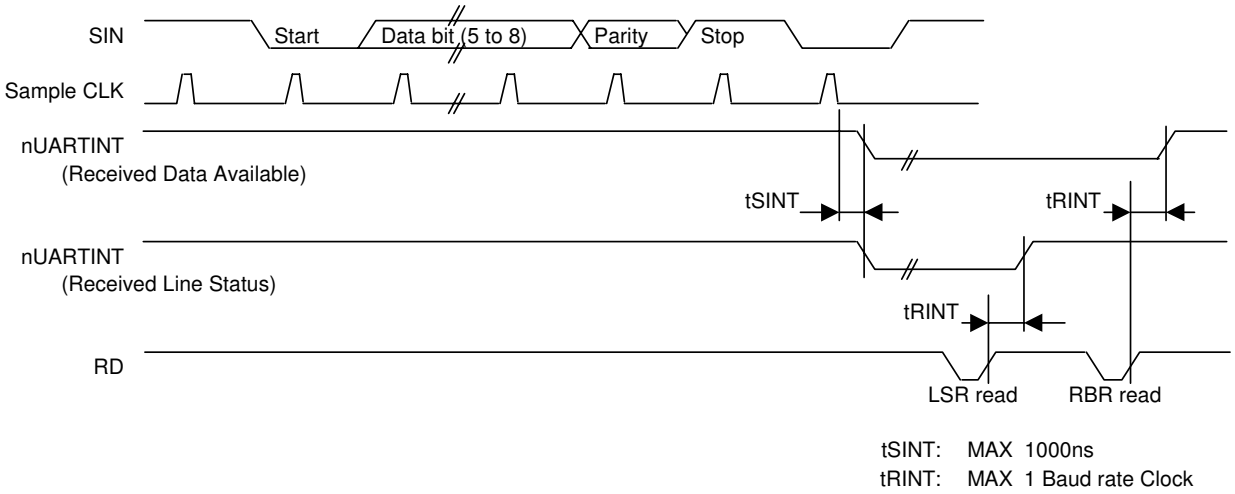


Figure 17.4 Receive Timing

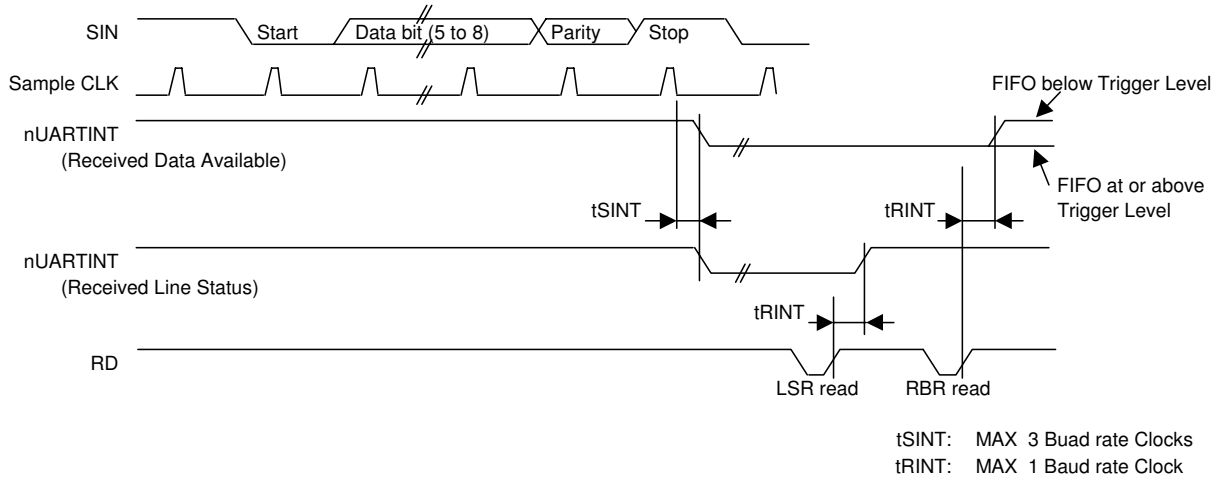


Figure 17.5 Reading First Byte from Receive Queue (Setting RDR)

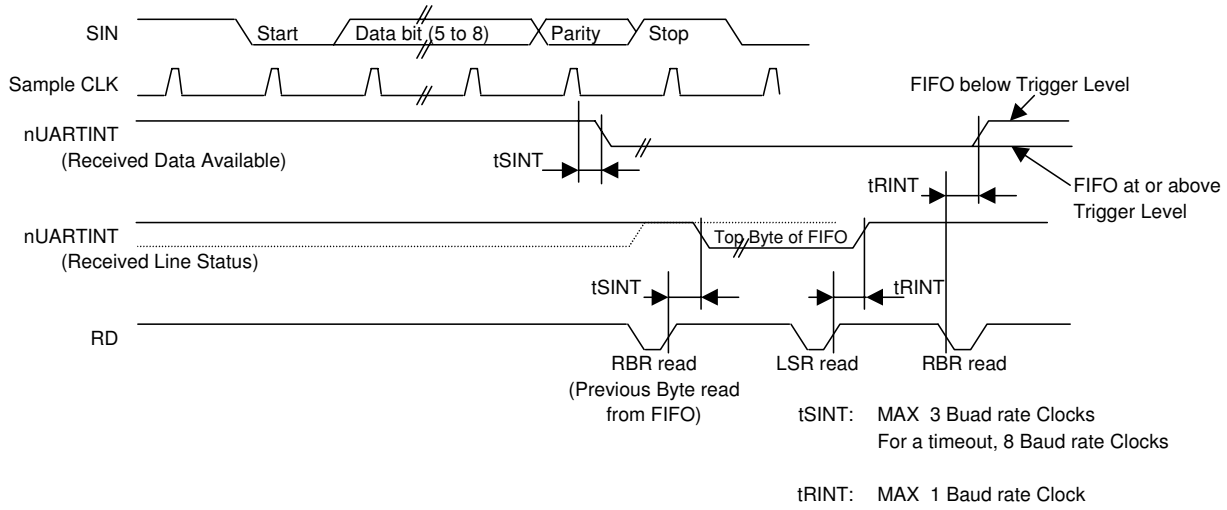


Figure 17.6 Reading Last Byte from Receive Queue

17.3.3 Generating Baud Rate Clock

The following is the formula for calculating the baud rate frequency.

$$\text{Baud rate frequency} = \text{CCLK}/(\text{DL}[15:0] \times 16)$$

Maximum baud rate clock depends on the software, the application, the system load and etc. But there is the capability of the transmit/receive baud rate with DL="2" setting in an ideal condition (ex. UART test) as long as in a small value of the baud rate deviation.

Note: A divisor (DL[15:0]) of 1 is not allowed. The setting must be either 0, to stop the clock, or a value of 2 or more.

The following Table lists DL settings for CPU clock and baud rate combinations.

Baud Rate	CCLK = 33MHz		CCLK = 20MHz		CCLK = 8MHz	
	DL [H]	Deviation (%)	DL [H]	Deviation (%)	DL [H]	Deviation (%)
50	A122	0.00000	61A8	0.00000	2710	0.00000
75	6B6C	0.00000	411B	–0.00200	1A0B	–0.00500
110	493E	0.00000	2C64	–0.00320	11C1	0.01000
134.5	3BE7	–0.00279	244E	–0.00344	0E85	0.01270
150	35B6	0.00000	208D	0.00400	0D05	0.01000
300	1ADB	0.00000	1047	–0.00800	0683	–0.02000
600	0D6E	–0.01454	0823	0.01600	0341	0.04002
1200	06B7	–0.01454	0412	–0.03199	01A1	–0.07994
1800	047A	–0.01454	02B6	0.06404	0116	–0.07994
2000	0407	0.02425	0271	0.00000	00FA	0.00000
2400	035B	–0.04366	0209	–0.03199	00D0	0.16026
3600	023D	–0.01454	015B	0.06404	008B	–0.07994
4800	01AE	–0.07267	0104	0.16026	0068	0.16026
7200	011E	0.16026	00AE	–0.22340	0045	0.64412
9600	00D7	–0.07267	0082	0.16026	0034	0.16026
19200	006B	0.39428	0041	0.16026	001A	0.16026
38400	0036	–0.53530	0021	–1.35732	000D	0.16026
56000	0025	–0.45849	0016	1.46104	0009	–0.79365
115200	0012	–0.53530	000B	–1.35732	—	—

17.3.4 Buffered Operation

Received Data Available Interrupts

Enabling both the receive queue and receive interrupts produces a received data available interrupt when the number of characters in the queue exceeds the specified trigger level. The hardware immediately clears this interrupt when the number of characters in the queue falls back to this trigger level.

The received data available bit in IIR is similar in operation. It goes to "1" when the number of characters in the queue exceeds the specified trigger level and returns to "0" when the number of characters in the queue falls back to this trigger level. It goes to "1" immediately after the hardware transfers the triggering data from the receive shift register to the queue and returns to "0" when the queue becomes empty.

Receiver line status interrupts have higher priority than these interrupts.

Character Timeout Interrupts

Enabling both the receive queue and receive interrupts produces a character timeout interrupt when the following conditions are met.

- There is at least one character in the receive queue.
- The time equivalent of at least four characters has elapsed since the last character was received (If the frame format specifies two stop bits, the timer starts after the first stop bit.) or since the last character was read off the queue.

If the frame format specifies one start bit, eight character bits, one parity bit, and two stop bits, for example, the timeout interval for a transfer speed of 300 baud is approximately 160 ms.

The clock used to calculate the character time is CCLK.

Reading a character from the queue clears the character timeout interrupt and resets the timeout detection timer.

If there is no character timeout interrupt pending, the hardware resets the timeout detection timer each time that the CPU reads a character from the queue or the interface receives a new character.

Transmitter Holding Register Empty Interrupts

Enabling both the transmit queue and receive queue interrupts produces a transmitter holding register empty interrupt when the transmit queue is empty. Writing a character to the transmit queue or reading IIR clears this interrupt.

This interrupt is delayed by the time equivalent of the frame less the final stop bit when the following conditions are met.

- There is only a single character in the queue after the transmitter holding register empty (THRE) bit goes to "1."
- The THRE bit goes to "1."

17.3.5 Queue Polled Mode

Enabling buffering, but disabling interrupts by setting the IER[3:0] bits all to "0" produces queue polled mode operation. The receive and transmit blocks have independent controls, so can use separate queue polled modes. Because there are no interrupts, the CPU must check the status of these two blocks by reading LSR.

- A "1" in LSR[0] indicates that there is at least one character in the receive queue.
- LSR[4:1] indicate any errors. Note that setting IER[2] to "0" means that such errors neither produce interrupts nor update the contents of IIR.
- A "1" in LSR[5] indicates that the transmit queue is empty.
- A "1" in LSR[6] indicates that the transmit queue and transmit shift register are both empty.
- A "1" in LSR[7] indicates an error receiving at least one character in the receive queue.

This mode uses the queue, but does not detect the trigger level or timeouts because those functions use interrupts, which are all disabled.

17.3.6 Error Status

Overrun error:

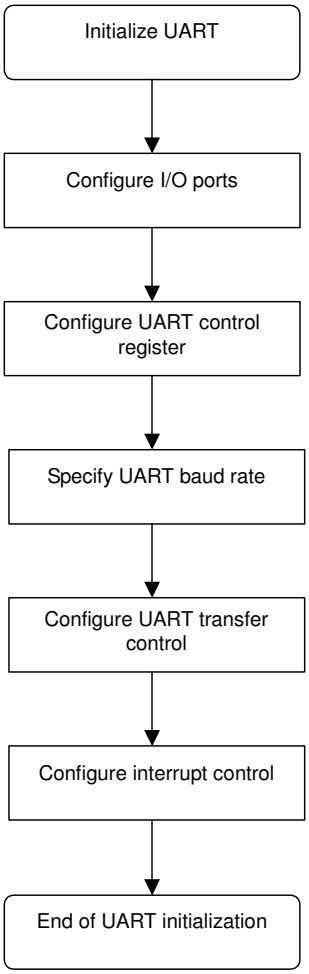
The LSR[1] bit in the UARTLSR register goes to "1" to indicate that the hardware has overwritten the contents of the UARTRBR register with a new character before the CPU read the former character.

(1) Parity error:
The LSR[2] bit in the UARTLSR register goes to "1" to indicate that the parity calculated from the received data does not match that received with the data. This only applies, however, when parity is enabled (LCR[3] = "1").
For buffered operation, LSR[2] indicates an error in the data at the head of the queue. Parity errors for other characters in the queue do not affect its contents.

(2) Framing error:
The LSR[3] bit in the UARTLSR register goes to "1" to indicate that the bit following the last data bit (or parity bit) is "0" (spacing level), not "1" (stop bit).
For buffered operation, LSR[3] goes to "1" when the character with the framing error reaches the head of the queue.

(3) Break interrupt:
The LSR[4] bit in the UARTLSR register goes to "1" to indicate that the input is "0" (spacing level) for one frame interval (start bit, data bits, parity bit, and stop bit).
For buffered operation, LSR[4] goes to "1" when the character with the break interrupt reaches the head of the queue.

17.3.7 Setup Procedure



- Set bit 0 in the port control register (GPCTL) to "1."

- Choose buffered (16550A) or unbuffered (16450) operation.
- Specify buffer trigger levels.
- Specify format (character length, number of stop bits, and parity) in the line control register.

- Set the divisor latch access bit, bit 7 in the line control register, to "1."
- Specify the divisor in DLL and DLM.
- Set the divisor latch access bit, bit 7 in the line control register, to "0."

- Enable or disable interrupts as desired.

- Specify the level and other parameters for the UART interrupt (NIR9) to the interrupt controller.

Chapter 18

Analog-to-Digital Converter

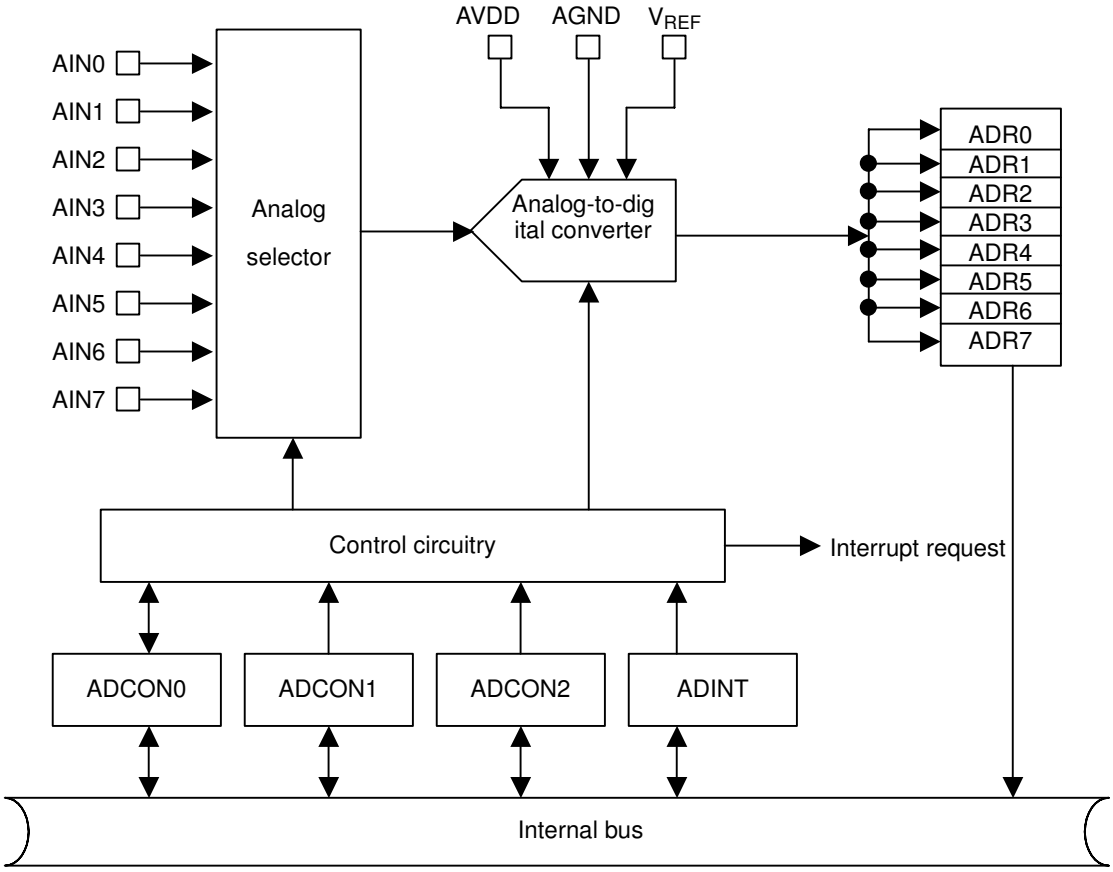
Chapter 18 Analog-to-Digital Converter

18.1 Overview

The built-in 8-channel, 10-bit resolution analog-to-digital converter supports two modes of operation: Scan mode sequentially converts input from the selected range of channels; select mode converts input from a single channel.
The conversion from an analog quantity to a digital one uses consecutive comparison with a sample and hold function.
At the user's option, the analog-to-digital converter issues an interrupt request after one cycle through the specified channels in scan mode and after each conversion in the select mode.

18.1.1 Components

Figure 18.1 shows the analog-to-digital converter components.



AIN0 to AIN7: Analog input pins
ADR0 to ADR7: Converter 10-bit result registers
ADINT: Analog-to-digital converter interrupt control register
ADCON0: Converter control register 0
ADCON1: Converter control register 1
ADCON2: Converter control register 2
AVDD: Analog VDD pin
AGND: Analog ground pin
V_{REF}: Analog reference voltage

Figure 18.1 Analog-to-Digital Converter Components

18.1.2 Pin List

Pin Name	I/O	Description
AVDD	VDD	Power supply for the analog-to-digital converter
VREF	I	Reference voltage for the analog-to-digital converter
AIN[0]	I	Analog-to-digital converter analog input port 0
AIN[1]	I	Analog-to-digital converter analog input port 1
AIN[2]	I	Analog-to-digital converter analog input port 2
AIN[3]	I	Analog-to-digital converter analog input port 3
AIN[4]	I	Analog-to-digital converter analog input port 4
AIN[5]	I	Analog-to-digital converter analog input port 5
AIN[6]	I	Analog-to-digital converter analog input port 6
AIN[7]	I	Analog-to-digital converter analog input port 7
AGND	GND	Ground for the analog-to-digital converter

18.1.3 Control Register List

Address	Name	Abbreviation	R/W	Size	Initial Value
0xB6000000	Analog-to-digital converter control register 0	ADCON0	R/W	16	0x0000
0xB6000004	Analog-to-digital converter control register 1	ADCON1	R/W	16	0x0000
0xB6000008	Analog-to-digital converter control register 2	ADCON2	R/W	16	0x0003
0xB600000C	Analog-to-digital converter interrupt control register	ADINT	R/W	16	0x0000
0xB6000010	Analog-to-digital converter forced interrupt register	ADFINT	R/W	16	0x0000
0xB6000014	Analog-to-digital converter result register 0	ADR0	R/W	16	0x0000
0xB6000018	Analog-to-digital converter result register 1	ADR1	R/W	16	0x0000
0xB600001C	Analog-to-digital converter result register 2	ADR2	R/W	16	0x0000
0xB6000020	Analog-to-digital converter result register 3	ADR3	R/W	16	0x0000
0xB6000024	Analog-to-digital converter result register 4	ADR4	R/W	16	0x0000
0xB6000028	Analog-to-digital converter result register 5	ADR5	R/W	16	0x0000
0xB600002C	Analog-to-digital converter result register 6	ADR6	R/W	16	0x0000
0xB6000030	Analog-to-digital converter result register 7	ADR7	R/W	16	0x0000

Note: Writing to a result register (ADR0 to ADR7) while the analog-to-digital converter is in operation invalidates the contents of the entire set.

18.2 Control Register Descriptions

18.2.1 Analog-to-Digital Converter Control Register 0 (ADCON0)

This register controls scan mode operation.
The program has read/write access to this register.
The contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCON0	– *	– *	– *	– *	– *	– *	– *	– *	– *	SCNC	– *	ADRUN	– *	ADSNM[2:0]		
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB6000000 [H]
Access: R/W
Access size: 16 bits

Note
*: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
It needs 2HCLK + 2CCLK for ADCON0 register write recovery time.

Bit Descriptions

- **ADSNM[2:0]** (bits 2 to 0):
  This field specifies the first channel to scan. The last is always ch7.

ADSNM			Channels	Scan order
2	1	0		
0	0	0	ch0 to ch7	ch0→ch1→ch2→ch3→ch4→ch5→ch6→ch7
0	0	1	ch1 to ch7	ch1→ch2→ch3→ch4→ch5→ch6→ch7
0	1	0	ch2 to ch7	ch2→ch3→ch4→ch5→ch6→ch7
0	1	1	ch3 to ch7	ch3→ch4→ch5→ch6→ch7
1	0	0	ch4 to ch7	ch4→ch5→ch6→ch7
1	0	1	ch5 to ch7	ch5→ch6→ch7
1	1	0	ch6 to ch7	ch6→ch7
1	1	1	ch7	ch7

  Do not change this setting while the analog-to-digital converter is in operation. Writes only take effect when ADRUN (bit 4) is "0."

- **ADRUN** (bit 4):
  This bit turns the analog-to-digital converter operation on and off in scan mode.

ADRUN	Description
0	Stop analog-to-digital converter
1	Start analog-to-digital converter

  Note that the ADRUN bit is a control bit. It in not a flag bit indicating the current analog-to-digital converter status (converting/idle).

- **SCNC** (bit 6):
  This bit specifies the action to take after one cycle through the specified channels.

SCNC	Description
0	Cycle back to first channel after one cycle through the specified channels, analog-to-digital converter
1	Stop after one cycle through the specified channels, analog-to-digital converter

When SCNC is "1", setting the INTSN bit in the ADINT register to "0" will start the next cycle. The INTSN bit indicates that the A-to-D conversion cycle is complete. Note that ADRUN remains "1" all along hence enabling the restart of analog-to-digital operation.

Note: If SCNC is "1," stopping conversion in the middle requires simultaneously setting both SCNC and ADRUN to "0" with the same write. Simultaneously setting both back to "1" then restarts conversion. Note that stopping conversion this way invalidates the contents of the result register for the current channel.

18.2.2 Analog-to-Digital Converter Control Register 1 (ADCON1)

This register controls select mode operation.
The program has read/write access to this register.
The contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCON1	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	STS	– *	ADSTM[2:0]		
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB6000004 [H]
Access: R/W
Access size: 16 bits

Note
*: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
It needs 2HCLK + 2CCLK for ADCON1 register write recovery time.

Bit Descriptions

- **ADSTM[2:0]** (bits 2 to 0):
  This field specifies the channel.

ADSTM			Channel
2	**1**	**0**	
0	0	0	ch0
0	0	1	ch1
0	1	0	ch2
0	1	1	ch3
1	0	0	ch4
1	0	1	ch5
1	1	0	ch6
1	1	1	ch7

  Do not change this setting while the analog-to-digital converter is in operation. Writes only take effect when STS (bit 4) is "0."

- **STS** (bit 4):
  This bit turns the analog-to-digital converter operation on and off in select mode.

STS	Description
0	Stop analog-to-digital converter
1	Start analog-to-digital converter

  The hardware automatically resets this bit to "0" when conversion is complete.

18.2.3 Analog-to-Digital Converter Control Register 2 (ADCON2)

This register specifies the operating clock frequency for the analog-to-digital converter register.
The program has read/write access to this register.
The contents after a reset are 0x0003.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCON1	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	ACKSEL[1:0]	
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Address: 0xB6000008 [H]
Access: R/W
Access size: 16 bits

Note
*: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **ACKSEL[1:0]** (bits 1 and 0):
  This field specifies the divisor for deriving the operating clock from CCLK.

ACKSEL		Channel
1	0	
0	0	Reserved
0	1	CCLK/2
1	0	CCLK/4
1	1	CCLK/8

Notes

Choose the divisor so that the resulting period is between 200 and 1000 ns.

The conversion time, 25 clock cycles, depends on the operating frequency and ACKSEL. The combination of 33 MHz for CCLK and 11B for ACKSEL, for example, yields one of 240 ns × 25 = 6 µs.

Table 18.1 Sample CCLK-ACKSEL Combinations

CCLK (MHz)	ACKSEL		
	01	10	11
33	60 ns	120 ns	240 ns
20	100 ns	200 ns	400 ns
16	120 ns	240 ns	480 ns
8	250 ns	500 ns	1000 ns
2	1000 ns	2000 ns	4000 ns

Note: The shading indicates combinations that produce periods out of range.

18.2.4 Analog-to-Digital Converter Interrupt Control Register (ADINT)

This register contains analog-to-digital converter interrupt settings.
The program has read/write access to this register.
The contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADINT	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	ADSTIE	ADSNIE	INTST	INTSN
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB600000C [H]
Access: R/W
Access size: 16 bits

Note
*: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **INTSN** (bit 0):
  A "1" in this bit indicates completion of one cycle through the specified channels--that is, that channel 7 conversion in scan mode is complete.

INTSN	Description
0	Scan not complete (channel 7 conversion in scan mode not complete)
1	Scan complete (channel 7 conversion in scan mode complete)

  The program must explicitly reset this bit to "0" by writing "1" to it.

- **INTST** (bit 1):
  A "1" in this bit indicates completion of select mode conversion.

INTST	Description
0	Select mode conversion not complete
1	Select mode conversion complete

  The program must explicitly reset this bit to "0" by writing "1" to it.

- **ADSNIE** (bit 2):
  Setting this bit to "1" produces an interrupt request when one cycle through the specified channels is complete--that is, when channel 7 conversion in scan mode is complete.

ADSNIE	Description
0	Disable interrupt after scan
1	Enable interrupt after scan

- **ADSTIE** (bit 3):
  Setting this bit to "1" produces an interrupt request when select mode conversion is complete.

ADSTIE	Description
0	Disable interrupt after conversion
1	Enable interrupt after conversion

18.2.5 Analog-to-Digital Converter Forced Interrupt Register (ADFINT)

This register is for forcing an analog-to-digital converter interrupt.
The program has read/write access to this register.
The contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADFINT	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	– *	ADFAS
After a reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB6000010 [H]
Access: R/W
Access size: 16 bits

Note
*: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.

Bit Descriptions

- **ADFAS** (bit 0):
  Setting this bit to "1" forces assertion of the interrupt signal. This facility is for test purposes. Writing "0" to this bit resets it.

18.2.6 Analog-to-Digital Converter Result Registers (ADR0 to ADR7)

These registers hold the result of converting the corresponding analog input.
The program has read/write access to these registers.
The contents after a reset are 0x0000.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADRn	– *	– *	– *	– *	– *	– *	DTn[9:0]									
After a reset (n = 0 to 7)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: 0xB6000014 to 0xB6000030 [H]
Access: R/W
Access size: 16 bits

Notes

*: These bits are reserved for future expansion. They return "0" for reads. Writes to them are ignored.
Writing to a result register (ADR0 to ADR7) while the analog-to-digital converter is in operation invalidates the contents of the entire set.

Bit Descriptions

- **DTn** (bits 9 to 0):
  This is the result of converting the corresponding analog input.

18.3 Operational Description

The analog-to-digital converter supports two modes of operation: Scan mode sequentially converts input from the selected range of channels; select mode converts input from a single channel.
These two modes cannot be used simultaneously.
Do not change modes while the analog-to-digital converter is in operation.
Changing modes invalidates the contents of the result registers (ADR0 to ADR7).

18.3.1 Scan Mode

Scan mode sequentially converts input from the selected channel (0 to 7) through channel 7. The program has a choice of stopping conversion or cycling back to the first channel when channel 7 conversion is complete.

Figure 18.2 and Figure 18.3 illustrate scan mode operation.

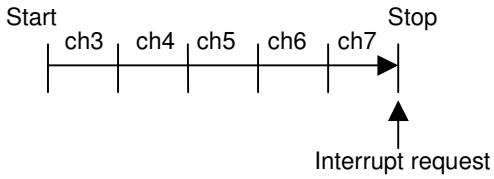


Figure 18.2 Sample Scan Mode Operation over Channels 3 to 7 Stopping after Cycle (SCNC = 1)

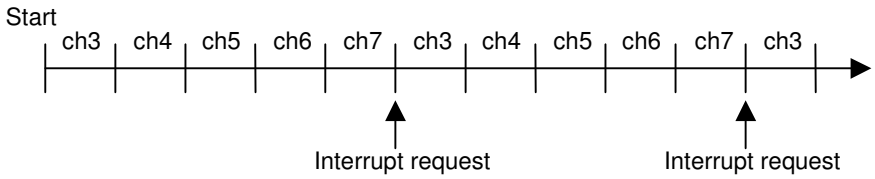


Figure 18.3 Sample Scan Mode Operation over Channels 3 to 7 Recycling (SCNC = 0)

18.3.2 Select Mode

Select mode converts input from a single channel.

Figure 18.4 illustrates select mode operation.

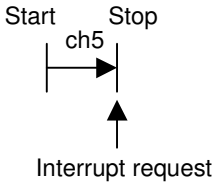


Figure 18.4 Sample Select Mode Operation Using Channel 5

Flowcharts

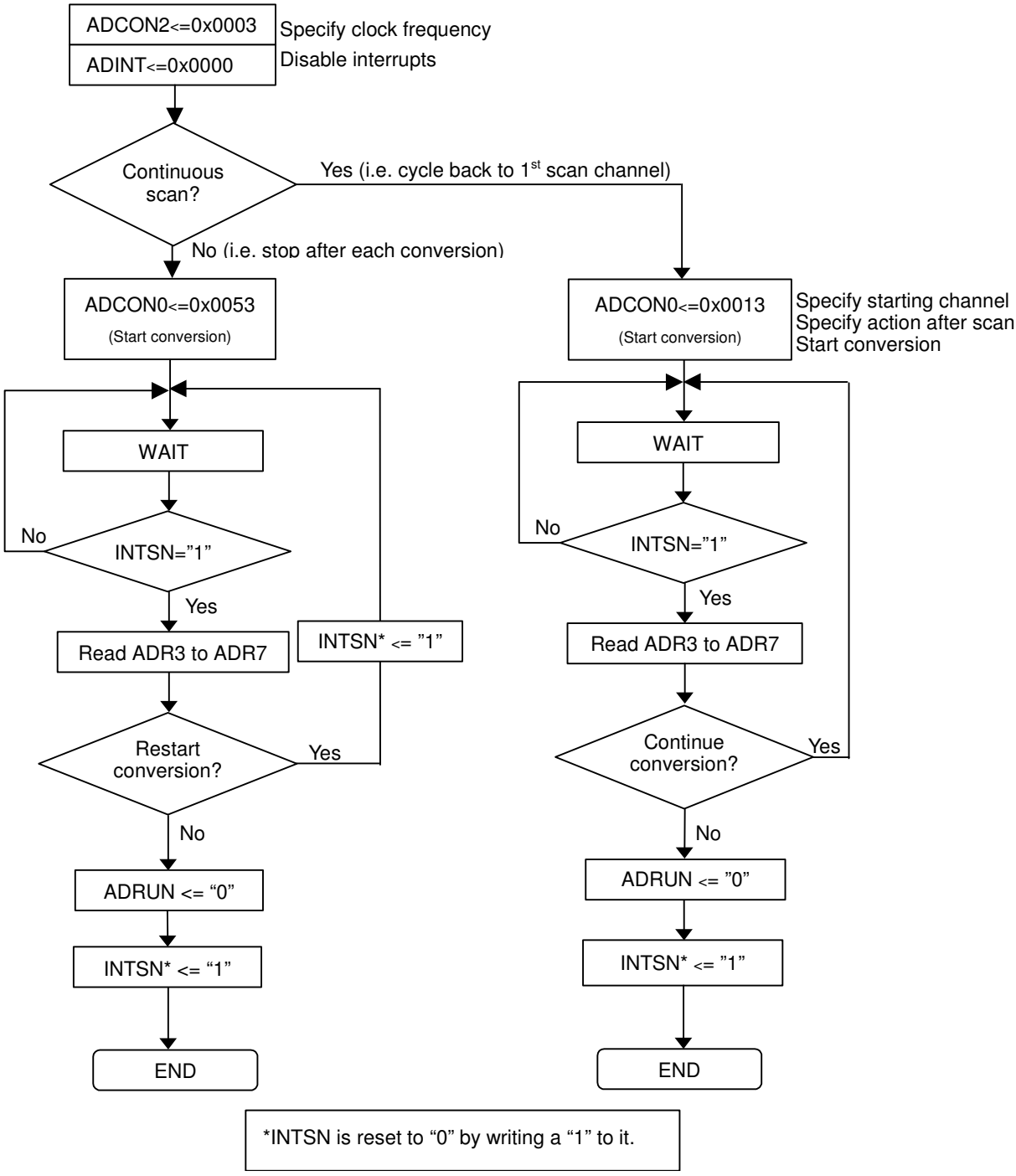


Figure 18.5 Scan Mode Example

Operating conditions
- Operating clock: 1/8 CCLK
- Interrupt requests: No
- Scan channels: 3 to 7

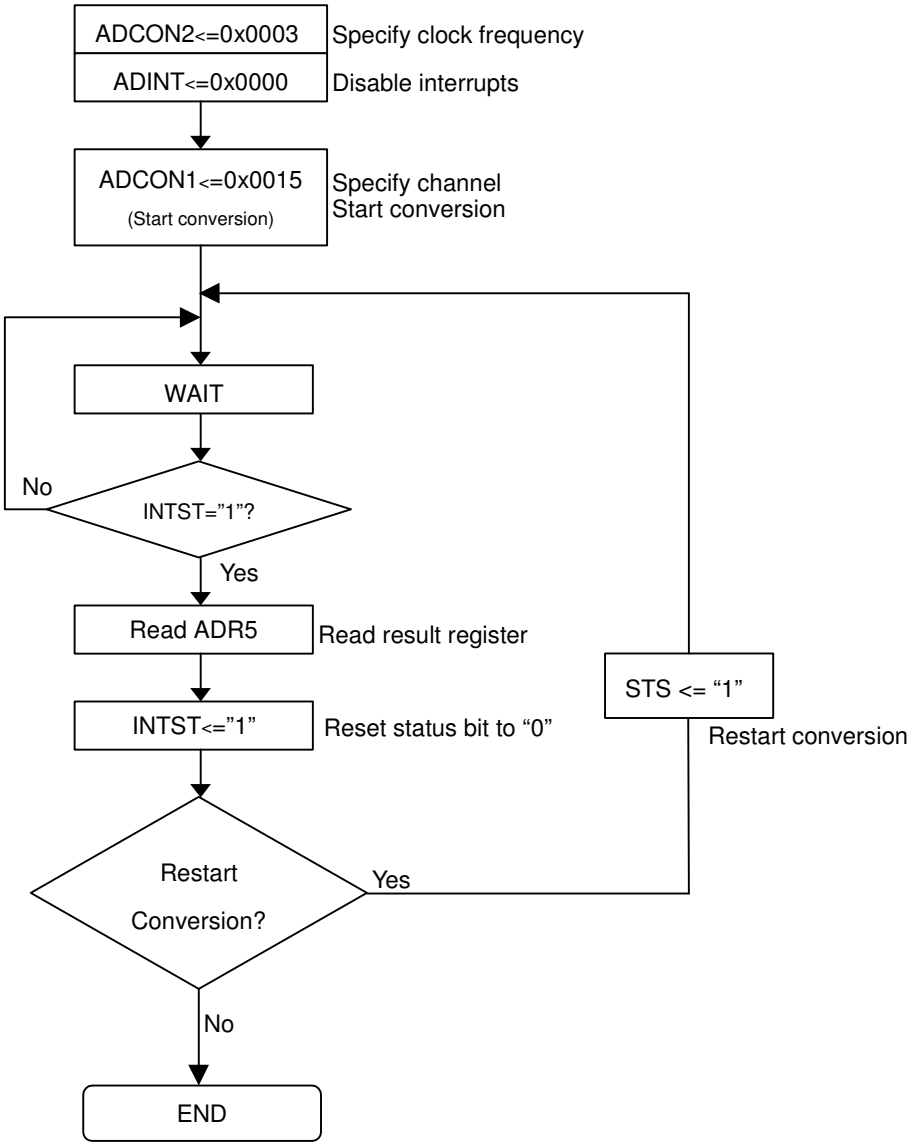


Figure 18.6 Select Mode Example

Operating conditions
- Operating clock: 1/8 CCLK
- Interrupt requests: No
- Channel: 5

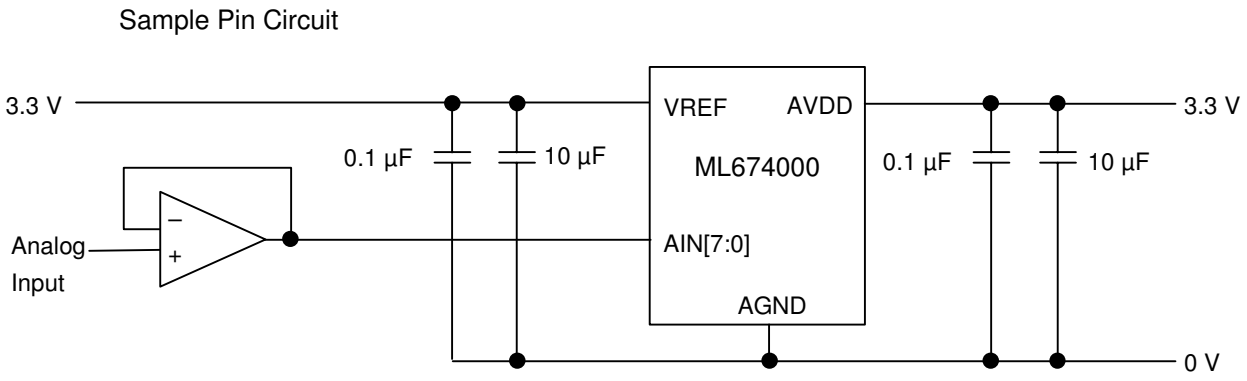


Figure 18.7 Sample Analog-to-Digital Converter Connections

Chapter 19

Electrical Characteristics

Chapter 19 Electrical Characteristics

19.1 Absolute Maximum Ratings

Item	Symbol	Conditions	Rating	Unit
Digital power supply voltage (core)	V_{DD_CORE}	GND = AGND = 0 V Ta = 25°C	–0.3 to +3.6	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	
Analog power supply voltage	AV_{DD}		–0.3 to V_{DD_IO}+0.3	
Analog reference voltage	V_{REF}		–0.3 to V_{DD_IO}+0.3 and –0.3 to AV_{DD} +0.3	
Analog input voltage	V_{AI}		–0.3 to V_{REF}	
Input current	I_I		–10 to +10	mA
High level output current	I_{OH}		+10	
Low level output current *1	I_{OL}		–20	
Low level output current *2			–30	
Power dissipation	P_D	Ta = 85°C per package	530	mW
Storage temperature	T_{STG}	—	–50 to +150	°C

Note

1. All output pins except XA[15:0]
2. XA[15:0]

19.2 Recommended Operating Conditions

(GND = 0 V)

Item	Symbol	Conditions	Minimum	Typical	Maximum	Unit
Digital power supply voltage (core)	V_{DD_CORE}	$V_{DD\_IO} \geq V_{DD\_CORE}$	2.25	2.5	2.75	V
Digital power supply voltage (I/O)	V_{DD_IO}		3.0	3.3	3.6	
Analog power supply voltage	AV_{DD}	$A_{VDD} = V_{DD\_IO}$	3.0	3.3	3.6	
Analog reference voltage	V_{REF}	$V_{REF} = A_{VDD} = V_{DD\_IO}$	3.0	3.3	3.6	
SRAM Storage hold voltage	V_{DDH}	f_{OSC} = 0 Hz	2.25	—	3.6	
Operating frequency	f_{OSC}	V_{DD_CORE} = 2.25 to 2.75 V_{DD_IO} = 3.0 to 3.6 *1	1	—	33.333	MHz
Ambient temperature	Ta	—	–40	25	85	°C

Note

1. Crystal frequencies between 16 MHz and 33 MHz. External clock input frequencies between 1MHz to 33MHz. Minimum of 2.56 MHz for external SDRAM. Minimum of 6.4 MHz for external EDO DRAM. Minimum of 2 MHz for analog-to-digital converter.

19.3 Electrical Characteristics

19.3.1 DC Characteristics

(V_{DD_CORE} = 2.25 to 2.75V, V_{DD_IO} = 3.0 to 3.6V, Ta = –40 to +85°C)

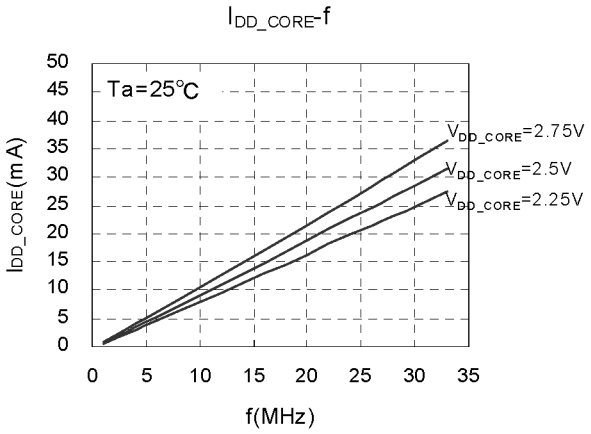
Item	Symbol	Conditions	Minimum	Typical	Maximum	Unit
High level input voltage	V_{IH}	—	2.0	—	V_{DD_IO}+0.3	V
Low level input voltage	V_{IL}		–0.3	—	0.8	
Schmitt Input Buffer Threshold Voltage	V_{T+}		—	1.6	2.1	
	V_{T-}		0.7	1.1	—	
	V_{HYS}		0.4	0.5	—	
High level output voltage	V_{OH}	I_{OH} = –100 µA	V_{DD}–0.2	—	—	
		I_{OH} = –4 mA	2.4	—	—	
Low level output voltage	V_{OL}	I_{OL} = 100 µA	—	—	0.2	
Low level output voltage *1		I_{OL} = 4 mA	—	—	0.4	
Low level output voltage *2		I_{OL} = 6 mA	—	—	0.4	
Input leak current *3	I_{IH} /I_{IL}	V_I = 0V / V_{DD_IO}	–10	—	10	µA
Input leak current *4		V_I = 0V Pull-up resistance of 50 kΩ	10	66	200	
Output leak current	I_{LO}	V_O = 0V / V_{DD_IO}	–10	—	10	
Input pin capacitance	C_I	—	—	6	—	pF
Output pin capacitance	C_O	—	—	9	—	
I/O pin capacitance	C_{IO}	—	—	10	—	
Analog reference power supply current	I_{REF}	Analog-to-digital converter operative*5	—	320	650	µA
		Analog-to-digital converter stopped	—	1	2	
Current consumption (STANDBY)	I_{DDS_CORE}	Ta = 25°C *6	—	3	45	
	I_{DDS_IO}			1	5	
Current consumption (HALT) *7	I_{DDH_CORE}	f_{OSC} = 16 MHz CL=50pF	—	8	15	mA
	I_{DDH_IO}		—	2	5	
Current consumption (RUN)	I_{DD_CORE}		—	15	25	
	I_{DD_IO}		—	18	30	

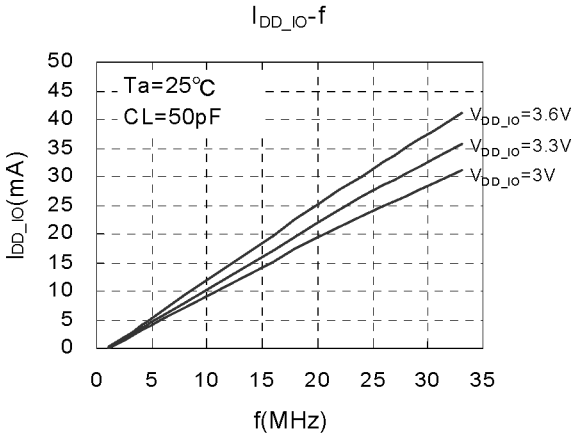
Notes

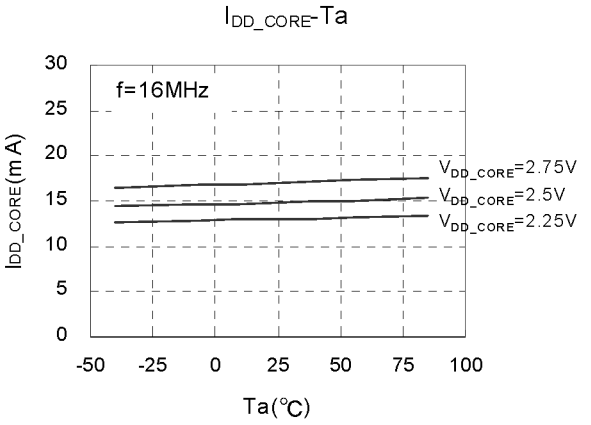
1. All output pins except XA[15:0]
2. XA[15:0]
3. All input pins except RESET_N
4. RESET_N pin, with 50 kΩ pull-up resistance
5. Analog-to-Digital Converter operation ratio is 20%
6. V_{DD_IO} or 0 V for input ports; no load for other pins
7. DRAM blocks stopped by MODE pins settings

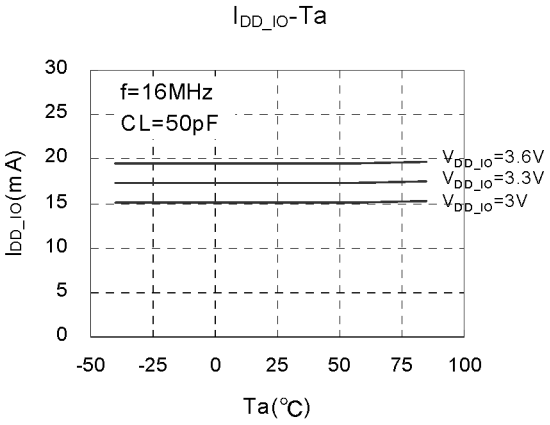
Power Consumption

The values in the following charts are measured values in the operating conditions indicated.
The samples were taken during normal operation in ARM mode with all peripheral clocks activated. Instructions were being executed from external memory.









19.3.2 AC Characteristics

■ Power Supply On/OFF Timing

(V_{DD_CORE} = 2.25 to 2.75V, V_{DD_IO} = 3.0 to 3.6V, Ta = –40 to +85°C)

Item	Symbol	Condi-tions	Minimum	Typical	Maximum	Unit	Notes
AVDD Supply on Delay	t_{AVDD_ON}		0	—	—	ns	
VDDcore Supply on Delay	$t_{VDDCORE\_ON}$		0	—	—	ns	
AVDD Supply off Delay	t_{AVDD_OFF}		0	—	—	ns	
VDDio Supply off Delay	t_{VDDIO_OFF}		0	—	—	ns	

■ Clock Timing

(V_{DD_CORE} = 2.25 to 2.75V, V_{DD_IO} = 3.0 to 3.6V, Ta = –40 to +85°C)

Item	Symbol	Condi-tions	Minimum	Typical	Maximum	Unit	Notes
Clock frequency	f_C		1	—	33.333	MHz	
Clock cycle time	t_C		30	—	1000		
Clock High level pulse width	t_{CH}		14	—	—		
Clock Low level pulse width	t_{CL}		14	—	—	ns	
Clock Rise time	t_{CR}		—	—	4		
Clock Fall time	t_{CF}		—	—	4		
XSDCLK frequency	f_{SDC}		1	—	33.333	MHz	
XSDCLK cycle time	t_{SDC}		30	—	1000		
XSDCLK High level pulse width	t_{SDCH}	—	12	—	—		
XSDCLK Low level pulse width	t_{SDCL}		12	—	—	ns	
XSDCLK Rise time	t_{SDCR}		—	—	2		
XSDCLK Fall time	t_{SDCF}		—	—	2		
HCLK frequency	f_{HC}		0.125 *1	—	33.333	MHz	
HCLK cycle time	t_{HC}		30	—	8000 *1	ns	
CCLK frequency	f_{CC}		0.125 *2	—	33.333	MHz	
CCLK cycle time	f_{CC}		30	—	8000 *2	ns	

Notes

1. Minimum of 2.56 MHz / Maximum of 390.6ns for external SDRAM. Minimum of 6.4 MHz / Maximum of 156ns for external EDO DRAM.
2. Minimum of 2 MHz / Maximum of 500ns for analog-to-digital converter.

Item	Symbol	Condi-tions	Minimum	Typical	Maximum	Unit	Notes
RESET_N pulse width 1	t_{RSTW1}	—	20 t_C	—	—	ns	Except for when power is first applied or returning from HALT or STANDBY mode
RESET_N pulse width 2	t_{RSTW2}		Oscillation stabilization interval	—	—	—	When power is first applied or returning from HALT or STANDBY mode
EFIQ_N pulse width	t_{EFIQW}		2 t_{HC}	—	—	ns	
EXINT pulse width 1	$t_{EXINTW1}$		2 t_{HC}	—	—		Except for STANDBY mode
EXINT pulse width 2	$t_{EXINTW2}$		t_{HC}	—	—		Release from STANDBY mode
DREQCLR0/DREQCLR1 delay 1	t_{DCLRD1}	CL = 30 pF	8 t_{HC} + 10.5	—	—		
TCOUT0/TCOUT1 delay 1	$t_{TCOUTD1}$		8 t_{HC} + 10.5	—	—		
DREQCLR0/DREQCLR1 delay 2	t_{DCLRD2}		2 t_{HC} + 11	—	—		
TCOUT0/TCOUT1 delay 2	$t_{TCOUTD2}$		2 t_{HC} + 11	—	—		
DREQ0/DREQ1 hold time	t_{DREQH}	—	t_{HC}	—	—		

■ Control signal timing

SRAM/ROM							
Item	**Symbol**	**Condi-tions**	**Minimum**	**Typical**	**Maximum**	**Unit**	**Notes**
XROMCS_N/XRAMCS_N access time1 (SRAM/ROM) READ ACCESS	t_{CSR1}	CL = 30 pF	$(n_{R1} + n_{R3})\ t_{HC} - 1$	—	$(n_{R1} + n_{R3})\ t_{HC} + 1$	ns	The ROMAC and RAMAC registers specify the OE/WE pulse width and read off time for ROM and SRAM access, respectively. For further details, see the following Tables. n_{R1} = 0 (CPU Access) n_{R1} = 1 (DMA Access)) n_{R2} = 0.5 (CPU Access) n_{R2} = 1.5 (DMA Access) n_{R3} = (OE/WE pulse width) n_{R4} = (OE/WE pulse width)+ 0.5 $n_{R5} = n_{R2}$+ (OE/WE pulse width) + 0.5
XROMCS_N/XRAMCS_N access time2 (SRAM/ROM) READ ACCESS	t_{CSR2}		$2*(n_{R1} + n_{R3})\ t_{HC} - 1$	—	$2*(n_{R1} + n_{R3})\ t_{HC} + 1$		
XA[23:0] access time (SRAM/ROM)	t_{ACC}		$(n_{R1} + n_{R3})\ t_{HC}$ - 2.5	—	$(n_{R1} + n_{R3})\ t_{HC}$ + 0.5		
XBS_N[1:0] access time (SRAM/ROM)	t_{BS}		$(n_{R1} + n_{R3})\ t_{HC} - 2.5$	—	$(n_{R1} + n_{R3})\ t_{HC} - 0.5$		
XOE_N delay (SRAM/ROM)	t_{OED}		$n_{R1}\ t_{HC} - 1$	—	$n_{R1}\ t_{HC} + 0.5$		
XWE_N delay (SRAM/ROM)	t_{WED}		$n_{R2}\ t_{HC} - 3$	—	$n_{R2}\ t_{HC} - 0.5$		
XROMCS_N/XRAMCS_N access time1 (SRAM/ROM) WRITE ACCESS	t_{CSW1}		$(n_{R2} + n_{R3})\ t_{HC} - 3.5$	—	$(n_{R2} + n_{R3})\ t_{HC}$ - 1		
XROMCS_N/XRAMCS_N access time2 (SRAM/ROM) WRITE ACCESS	t_{CSW2}		$(n_{R5} + n_{R2} + n_{R3})\ t_{HC} - 3.5$	—	$(n_{R5} + n_{R2} + n_{R3})\ t_{HC}$ - 1		
XBWE_N[1:0] delay (SRAM/ROM)	t_{WELHD}		$n_{R2}\ t_{HC} - 3.5$	—	$n_{R2}\ t_{HC}$ - 1		
XBWE_N[1:0] hold time (SRAM/ROM)	t_{WELHH}		-0.5	—	0.5		
XOE_N, XWE_N pulse width (SRAM/ROM)	$t_{OE/WEW}$		$n_{R3}\ t_{HC} - 1$	—	$n_{R3}\ t_{HC} + 0.5$		
XOE_N pulse width (SRAM/ROM) ARM ACCESS	t_{OEW}		$2*\ n_{R3}t_{HC} - 1$	—	$2*\ n_{R3}t_{HC} + 0.5$		
XBS_N[1:0] delay (SRAM/ROM)	t_{BSBD}		0	—	2		
XBS_N[1:0] output hold time 1 (SRAM/ROM)	t_{BSBH1}		$n_{R1}\ t_{HC} - 0.5$	—	—		
XBS_N[1:0] output hold time 2 (SRAM/ROM)	t_{BSBH2}		$0.5\ t_{HC} + 1$	—	—		
XA[23:0] delay (SRAM/ROM)	t_{XAD}		-0.5	—	2.5		
XA[23:0] output hold time 1 (SRAM/ROM)	t_{XAH1}		$n_{R1}\ t_{HC} - 0.5$	—	—		
XA[23:0] output hold time 2 (SRAM/ROM)	t_{XAH2}		$0.5\ t_{HC} + 1$	—	—		

SRAM/ROM (continued)							
Item	**Symbol**	**Condi-tions**	**Minimum**	**Typical**	**Maximum**	**Unit**	**Notes**
XD[15:0] input setup time (SRAM/ROM) ARM ACCESS	t_{XDIS}	—	19	—	—	ns	
XD[15:0] input setup time (SRAM/ROM) DMAC ACCESS	t_{XDIS}		17	—	—		
XD[15:0] input hold time (SRAM/ROM)	t_{XDIH}		0	—	—		
XD[15:0] output delay (SRAM/ROM) ARM ACCESS	t_{XDOD}	CL = 30 pF	$n_{R4}t_{HC}$ – 14.5	—	—		
XD[15:0] output delay (SRAM/ROM) DMAC ACCESS	t_{XDOD}		n_{R4} t_{HC} – 7.5	—	—		
XD[15:0] output Enable time (SRAM/ROM)	t_{XDOE}		n_{R1} t_{HC} + 1.5	—	—		
XD[15:0] output Disable time (SRAM/ROM)	t_{XDODE}		-6	—	—		
XD[15:0] output hold time (SRAM/ROM) ARM ACCESS	t_{XDOH}		0.5 t_{HC} + 3	—	—		
XD[15:0] output hold time (SRAM/ROM) DMAC ACCESS	t_{XDOH}		0.5 t_{HC} + 5.5	—	—		
XROMCS_N, XRAMCS_N output hold time 1 (SRAM/ROM)	t_{CSH1}		n_{R1} t_{HC} – 1.5	—	—		
XROMCS_N, XRAMCS_N Output hold time 2 (SRAM/ROM)	t_{CSH2}		0.5 t_{HC} + 0.5	—	—		

■ ROMAC Register Settings for Timing Parameters OE/WE Pulse Width and Read Off Time

ROMTYPE[2:0]: Timing parameter combination

ROMTYPE[2:0]	OE/WE pulse width	Read off time	Notes
000	1	0	
001	2	0	
010	3	2	
011	4	2	
100, 101, 110	—	—	Operation is not guaranteed
111	8	4	

■ RAMAC register Settings for Timing Parameters OE/WE Pulse Width and Read Off Time

RAMTYPE[2:0]: Timing parameter combination

RAMTYPE[2:0]	OE/WE pulse width	Read off time	Notes
000	1	0	
001	2	0	
010	3	2	
011	4	2	
100, 101, 110	—	—	Operation is not guaranteed
111	8	4 (3)*1	

*1: Read off timing is 3 in case of Accessing by DMA

IO0/IO1							
Item	**Symbol**	**Condi-tions**	**Minimum**	**Typical**	**Maximum**	**Unit**	**Notes**
XIOCS_N[0]/XIOCS_N[1] access time 1(external I/O banks 0 and 1) READ ACCESS	$t_{XIOCSR1}$	—	$n_{IO1}\ t_{HC} - 0.5$	—	$n_{IO1}\ t_{HC} + 1$	ns	The IO0AC and IO1AC registers specify the OE/WE pulse width and read off time for accessing external I/O banks 0 and 1, respectively. For further details, see the following Table.
XIOCS_N[0]/XIOCS_N[1] access time 2(external I/O banks 0 and 1) READ ACCESS	$t_{XIOCSR2}$		$2*n_{IO1}\ t_{HC} - 0.5$	—	$2*n_{IO1}\ t_{HC} + 1$		
XA[23:0] access time (external I/O banks 0 and 1)	$t_{XIOXACC}$		$n_{IO1}\ t_{HC} - 1.5$	—	$n_{IO1}\ t_{HC} + 0.5$		
XBS_N[1:0] access time (external I/O banks 0 and 1)	t_{XIOBS}		$n_{IO1}\ t_{HC} - 1.5$	—	$n_{IO1}\ t_{HC} + 0.5$		
XWR delay (external I/O banks 0 and 1)	$t_{XIOXWRD}$	CL = 30 pF	1	—	3.5		
XWR hold time (external I/O banks 0 and 1)	$t_{XIOXWRH}$		t_{HC}	—	$t_{HC} + 1.5$		n_{IO1} = (Address Setup) + (OE/WE Pulse width)
XWAIT sampling timing delay 1 (external I/O banks 0 and 1)	$t_{XIOXWAITD1}$	—	$n_{IO2}\ t_{HC}$	—	$n_{IO2}\ t_{HC}$		n_{IO2} = (Address Setup) - 1 + (OE/WE Pulse width)
XWAIT sampling timing delay 2 (external I/O banks 0 and 1)	$t_{XIOXWAITD2}$		$n_{IO1}\ t_{HC}$	—	$n_{IO1}\ t_{HC}$		n_{IO3} = (Address Setup)
XWAIT setup time (external I/O banks 0 and 1)	$t_{XIOXWAITS}$		15.5	—	—		n_{IO4} = (Address Setup) + 1
XWAIT hold time (external I/O banks 0 and 1)	$t_{XIOXWAITH}$		0	—	—		n_{IO5} = (OE/WE pulse width)
XOE_N delay (external I/O banks 0 and 1)	t_{XIOOED}	CL = 30 pF	$n_{IO3}\ t_{HC} - 0.5$	—	$n_{IO3}\ t_{HC} + 1$		n_{IO6} = (Address Setup) + 1 + (OE/WE Pulse width)
XWE_N delay 1 (external I/O banks 0 and 1)	t_{XIOWED}		$n_{IO4}\ t_{HC}$	—	$n_{IO4}\ t_{HC} + 1.5$		n_{IO7} = (Address Setup) + 2 + (OE/WE Pulse width)
XIOCS_N[0]/XIOCS_N[1] access time 1(external I/O banks 0 and 1) WRITE ACCESS	$t_{XIOCSW1}$		$n_{IO6}\ t_{HC} - 0.5$	—	$n_{IO6}\ t_{HC} + 0.5$		
XIOCS_N[0]/XIOCS_N[1] access time 2(external I/O banks 0 and 1) WRITE ACCESS	$t_{XIOCSW2}$		$(n_{IO6}+n_{IO7})t_{HC} - 0.5$	—	$(n_{IO6}+n_{IO7})t_{HC} + 0.5$		
XBWE_N[1:0] delay (external I/O banks 0 and 1)	$t_{XIOWELHD}$		$n_{IO4}\ t_{HC} - 0.5$	—	$n_{IO4}\ t_{HC} + 1$		
XOE_N, XWE_N pulse width (external I/O banks 0 and 1)	$t_{XIOoE/WEW}$		$n_{IO5}\ t_{HC} - 1.5$	—	$n_{IO5}\ t_{HC}$		
XBWE_N[1:0] hold time (external I/O banks 0 and 1)	$t_{XIOWELHH}$		-0.5		0.5		
XBS_N[1:0] delay (external I/O banks 0 and 1)	t_{XIOBSD}		0	—	2		
XBS_N[1:0] output hold time (external I/O banks 0 and 1)	t_{XIOBSH}		$t_{HC} - 1$	—	—		

IO0/IO1 (continued)							
Item	**Symbol**	**Condi-tions**	**Minimum**	**Typical**	**Maximum**	**Unit**	**Notes**
XA[23:0] delay (external I/O banks 0 and 1)	t_{XIOXAD}	—	-0.5	—	2	ns	
XA[23:0] output hold time 1 (external I/O banks 0 and 1)	$t_{XIOXAH1}$		-0.5	—	—		
XA[23:0] output hold time 2 (external I/O banks 0 and 1)	$t_{XIOXAH2}$		t_{HC} + 1	—	—		
XD[15:0] input setup time (external I/O banks 0 and 1)	t_{XIODIS}		17	—	—		
XD[15:0] input hold time (external I/O banks 0 and 1)	t_{XIODIH}		0	—	—		
XD[15:0] output delay (external I/O banks 0 and 1)	t_{XIODOD}	CL = 30 pF	n_{IO1} t_{HC} –5	—	n_{IO1} t_{HC} –2		
XD[15:0] output Enable time (external I/O banks 0 and 1)	t_{XIODOE}		t_{HC} + 2	—	—		
XD[15:0] output Disable time (external I/O banks 0 and 1)	$t_{XIODODE}$		-6	—	—		
XD[15:0] output hold time (external I/O banks 0 and 1)	t_{XIODOH}		t_{HC} + 3	—	—		
XIOCS_N[1:0] output hold time (external I/O banks 0 and 1)	t_{XIOCSH}		t_{HC} – 2	—	—		

■ IO0AC/IO1AC Register Settings and Parameters Address Setup, OE/WE Pulse Width, and Read Off Time

IOTYPE[2:0]: Timing parameter combination

IOTYPE[2:0]	AS	R/W	OF	Notes
000	1	1	1	
001	1	4	2	
010	—	—	—	Operation is not guaranteed
011	2	8	4	
100	2	12	6	
101	2	16	7	
110	—	—	—	Operation is not guaranteed
111	4	24	10	

AS = address setup; R/W = OE/WE pulse width; OF = read off time

(V_{DD_CORE} = 2.25 to 2.75V, V_{DD_IO} = 3.0 to 3.6V, Ta = –40 to +85°C)

SDRAM							
Item	**Symbol**	**Condi-tions**	**Minimum**	**Typical**	**Maximum**	**Unit**	**Notes**
XSDCS_N delay (SDRAM)	t_{SDCSD}	CL = 30 pF	$0.5t_{SDC}$ + 1.5	—	$0.5t_{SDC}$ + 5	ns	The DRPC register specifies the SDRAM access parameters tRAS, tRCD, tRP, tDPL. For further details, refer to the table on page 19-8.
XSDCKE delay (SDRAM)	t_{SDCKED}		$0.5t_{SDC}$ + 1	—	$0.5t_{SDC}$ + 5		
XDQM[0]/XDQM[1] delay (SDRAM)	t_{SDDQMD}		$0.5t_{SDC}$ + 1	—	$0.5t_{SDC}$ + 5.5		n_{SD1} = tRCD
XRAS_N delay (SDRAM)	t_{SDRASD}		$0.5t_{SDC}$ + 1.5	—	$0.5t_{SDC}$ + 5.5		n_{SD2} = tRAS
XCAS_N delay (SDRAM)	t_{SDCASD}		$0.5t_{SDC}$ + 1.5	—	$0.5t_{SDC}$ + 4.5		n_{SD3} = tRP
RASCAS minimum delay (SDRAM)	t_{SDRCD}	—	$n_{SD1}t_{SDC}$	—	—		
RAS active time (SDRAM)	t_{SDRAS}		$n_{SD2}t_{SDC}$	—	—		
RAS precharge time (SDRAM)	t_{SDRP}		$n_{SD3}t_{SDC}$	—	—		
XWE_N delay (SDRAM)	t_{SDWED}	CL = 30 pF	$0.5t_{SDC}$ – 0.5	—	$0.5t_{SDC}$ + 4		
XD[15:0] input setup time (SDRAM)	t_{SDXDIS}	—	6.5	—	—		
XD[15:0] input hold time (SDRAM)	t_{SDXDIH}		0	—	—		
XA[23:0] output delay (SDRAM)	t_{SDXAD}	CL = 30 pF	$0.5t_{SDC}$ + 1.5	—	$0.5t_{SDC}$ + 5		
XD[15:0] output delay (SDRAM)	t_{SDXDOD}		$0.5t_{SDC}$ + 3	—	$0.5t_{SDC}$ + 6		
XD[15:0] output hold time (SDRAM)	t_{SDXDOH}		$0.5t_{SDC}$ + 3	—	—		
XD[15:0] output enable time (SDRAM)	t_{SDXDOE}		$0.5t_{SDC}$ + 3	—	—		
XD[15:0] output disable time (SDRAM)	$t_{SDXDODE}$		$0.5t_{SDC}$ + 5	—	—		

The DRAM Characteristics Control Register (DRPC) specifies the DRAM timing parameters in clock cycles. The values for tRCD, tRAS, and tRP provided in this table are used in determining the timing characteristics noted in the preceding SDRAM timing table. For detailed information about this register, refer to Chapter 10 of this manual.

DRAMSPEC[3:0]				SDRAM operation				
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	Low speed (high frequency)
1	0	0	1	3	6	3	2	
1	0	1	0	(reserved) *1				
1	0	1	1	(reserved) *1				
1	1	0	0	(reserved) *1				
1	1	0	1	(reserved) *1				
1	1	1	0	(reserved) *1				
1	1	1	1	(reserved) *1				

Notes

1. Settings marked as (reserved) should not be used

EDODRAM							
Item	**Symbol**	**Condi-tions**	**Minimum**	**Typical**	**Maximum**	**Unit**	**Notes**
RASCAS delay (EDO DRAM)	t_{EDRCD}	—	$n_{ED1}\ t_{HC} - 0.5$	—	$n_{ED1}\ t_{HC} + 1$	ns	The DRPC register specifies the EDO DRAM access parameters tRAH, tCAC, tCAS, tRCD, tRP. For further details, refer to the table on page 19-10.
CAS pulse width (EDO DRAM)	t_{EDCAS}	CL = 30 pF	$n_{ED2}\ t_{HC} - 2.5$	—	$n_{ED2}\ t_{HC} - 0.5$		
RAS pulse width (EDO DRAM)	t_{EDRAS}		$n_{ED7}\ t_{HC} - 1.5$	—	$n_{ED7}\ t_{HC}$		n_{ED1} = tRCD
RAS precharge time (EDO DRAM)	t_{EDRP}	—	$n_{ED3}\ t_{HC}$	—	—		n_{ED2} = tCAS
CAS precharge time (EDO DRAM)	t_{EDCP}		$n_{ED4}\ t_{HC} + 1$	—	—		n_{ED3} = tRP
XRAS_N delay (EDO DRAM)	t_{EDRASD}	CL = 30 pF	t_{HC}	—	$t_{HC} + 2$		n_{ED4} = tCAC + 1 - tCAS
XOE_N delay 1 (EDO DRAM)	t_{EDOED1}		$n_{ED5}\ t_{HC} - 4$	—	$n_{ED5}\ t_{HC} - 1$		n_{ED5} = tRAH
XOE_N delay 2 (EDO DRAM)	t_{EDOED2}		$n_{ED4}\ t_{HC} - 2$	—	$n_{ED4}\ t_{HC}$		n_{ED6} = tCAC + 1
XWE_N delay 1 (EDO DRAM)	t_{EDWED1}		$n_{ED5}\ t_{HC} - 1.5$	—	$n_{ED5}\ t_{HC}$		n_{ED7} = tRCD + N(tCAC + 1)
XWE_N delay 2 (EDO DRAM)	t_{EDWED2}		$n_{ED4}\ t_{HC} - 0.5$	—	$n_{ED4}\ t_{HC} + 1$		N = Data Size / Bus Width
Row address hold time (EDO DRAM)	t_{EDRAH}		$n_{ED5}\ t_{HC} - 2$	—	$n_{ED5}\ t_{HC} + 1$		
Column address delay (EDO DRAM)	t_{EDCAD}		t_{HC}	—	$t_{HC} + 2.5$		
Column address hold time (EDO DRAM)	t_{EDCAH}		$n_{ED2}\ t_{HC} - 2.5$	—	$n_{ED2}\ t_{HC}$		
XD[15:0] sampling timing delay (EDO DRAM)	$t_{EDXDSMPLD}$	—	$n_{ED6}\ t_{HC}$	—	$n_{ED6}\ t_{HC}$		
XD[15:0] input setup time (EDO DRAM)	t_{EDXDIS}		16	—	—		
XD[15:0] input hold time (EDO DRAM)	t_{EDXDIH}		0	—	—		
XD[15:0] output delay 1 (EDO DRAM)	$t_{EDXDOD1}$	CL = 30 pF	0	—	2		
XD[15:0] output delay 2 (EDO DRAM)	$t_{EDXDOD2}$		0.5	—	3		
XD[15:0] output hold time (EDO DRAM)	t_{EDXDOH}		$n_{ED2}\ t_{HC} - 2$	—	—		
XD[15:0] output enable time (EDO DRAM)	t_{EDXDOE}		$-(n_{ED5} + 1)\ t_{HC} + 1$	—	—		
XD[15:0] output disable time (EDO DRAM)	$t_{EDXDODE}$		$(n_{ED2}+n_{ED4})\ t_{HC} + 0.5$	—	—		

The DRAM Characteristics Control Register (DRPC) specifies the DRAM timing parameters in clock cycles. The values for tRAH, tCAS, tRCD, tCAC, tOEZ, and tRP provided in this table are used in determining the timing characteristics noted in the preceding EDO-DRAM timing table. For detailed information about this register, refer to Chapter 10 of this manual.

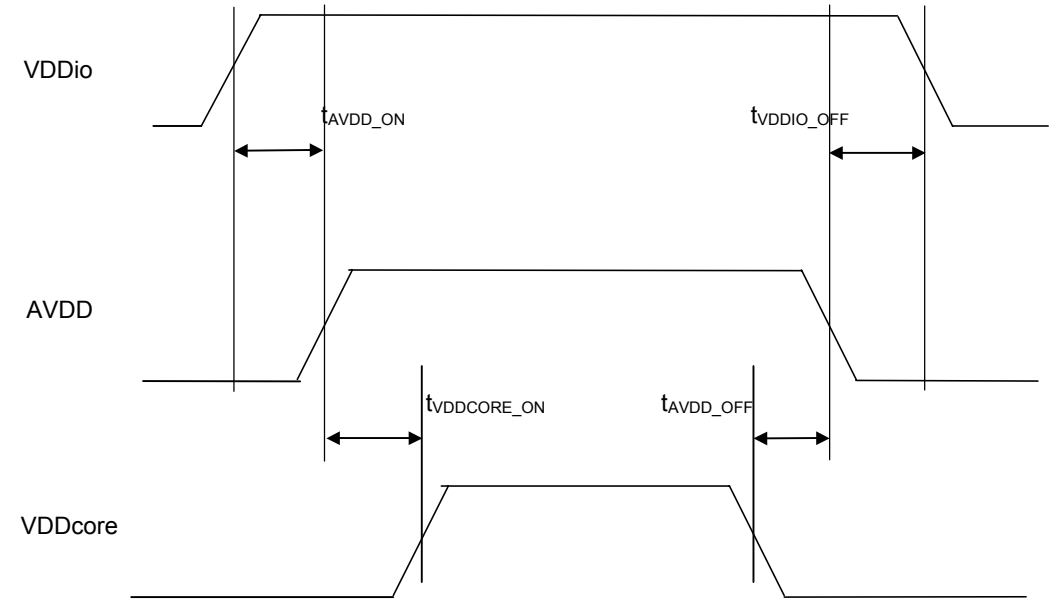
DRAMSPEC[3:0]				EDO DRAM operation				
3	**2**	**1**	**0**	**tRAH tCAS**	**tRCD**	**tCAC tOEZ**	**tRP**	
0	0	0	0	1	2	1	1	High speed (low frequency)
0	0	0	1	1	2	1	2	
0	0	1	0	1	3	1	2	
0	0	1	1	1	3	1	3	
0	1	0	0	1	3	2	3	
0	1	0	1	1	4	2	4	
0	1	1	0	1	5	2	5	
0	1	1	1	2	4	2	4	
1	0	0	0	2	5	2	5	
1	0	0	1	2	6	2	6	Low speed (high frequency)
1	0	1	0	3	8	3	7	
1	0	1	1	(reserved) *1				
1	1	0	0	(reserved) *1				
1	1	0	1	(reserved) *1				
1	1	1	0	(reserved) *1				
1	1	1	1	(reserved) *1				

Notes

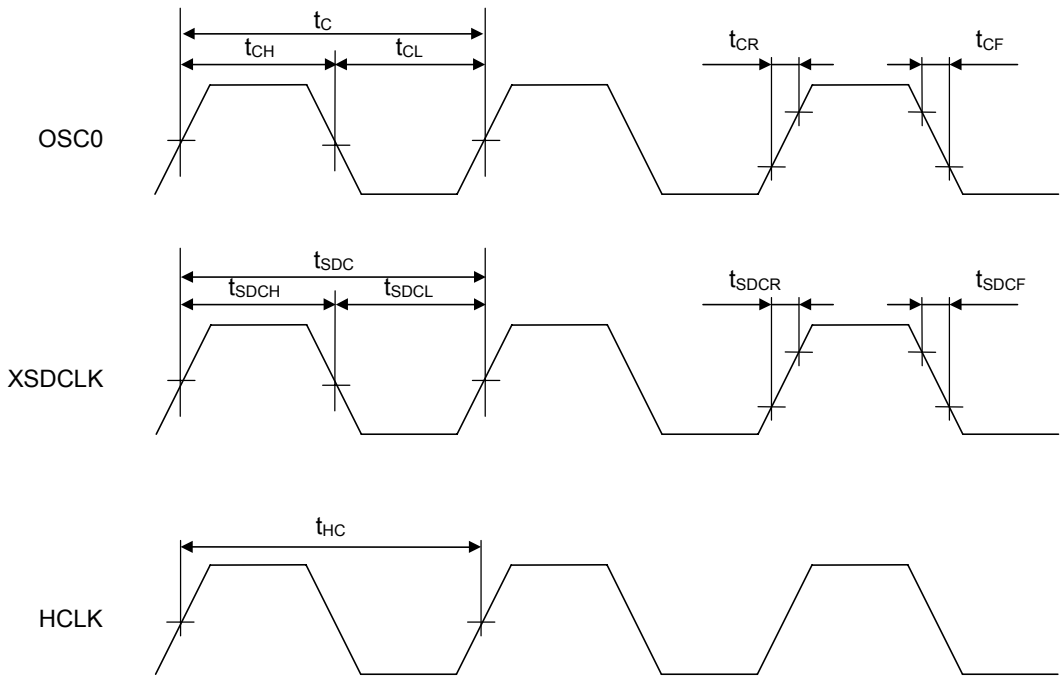
1. Settings marked as (reserved) should not be used

19.3.3 Timing Charts

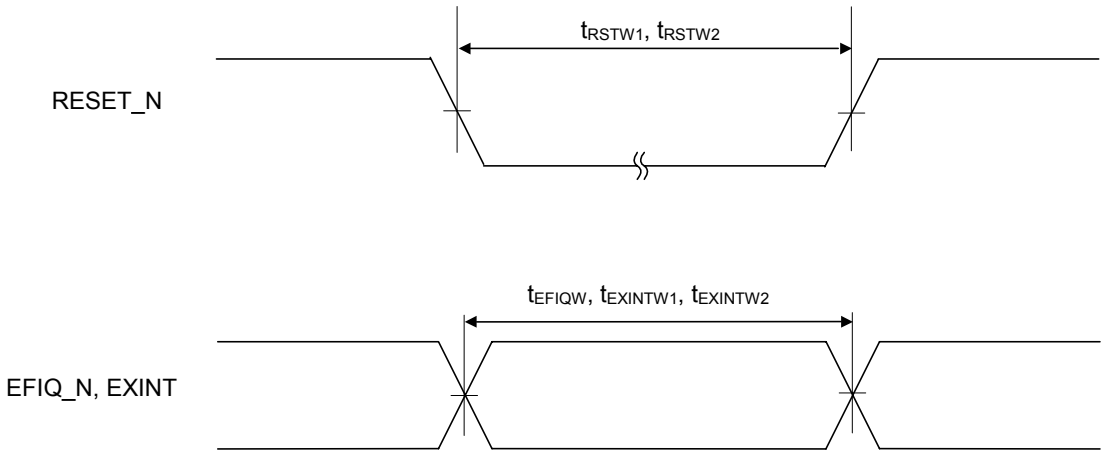
■ Power Supply On/OFF Timing



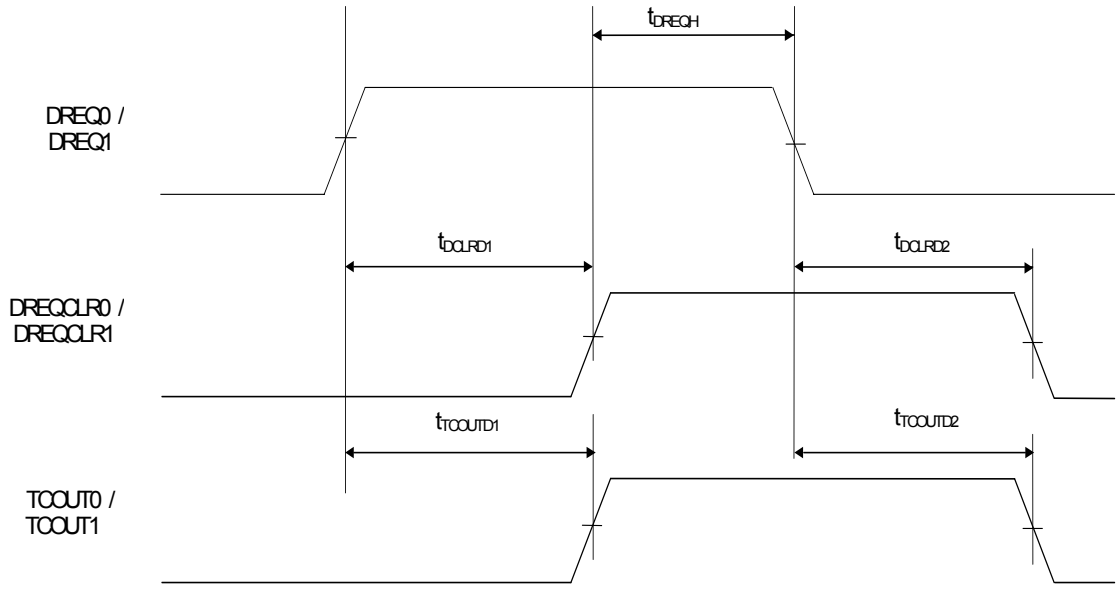
■ Clock Timing



■ Control Signal Timing

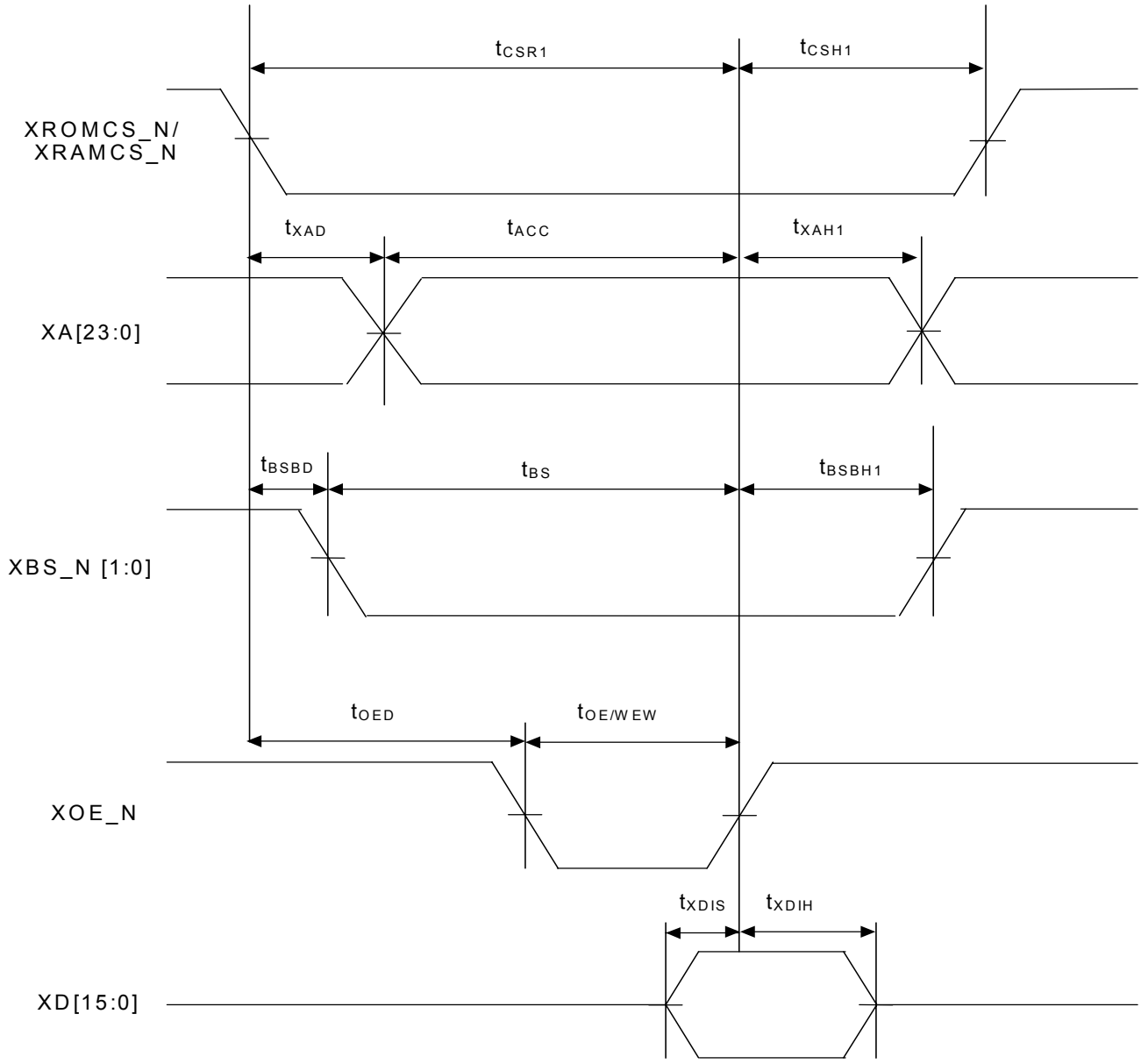


■ DMA Timing

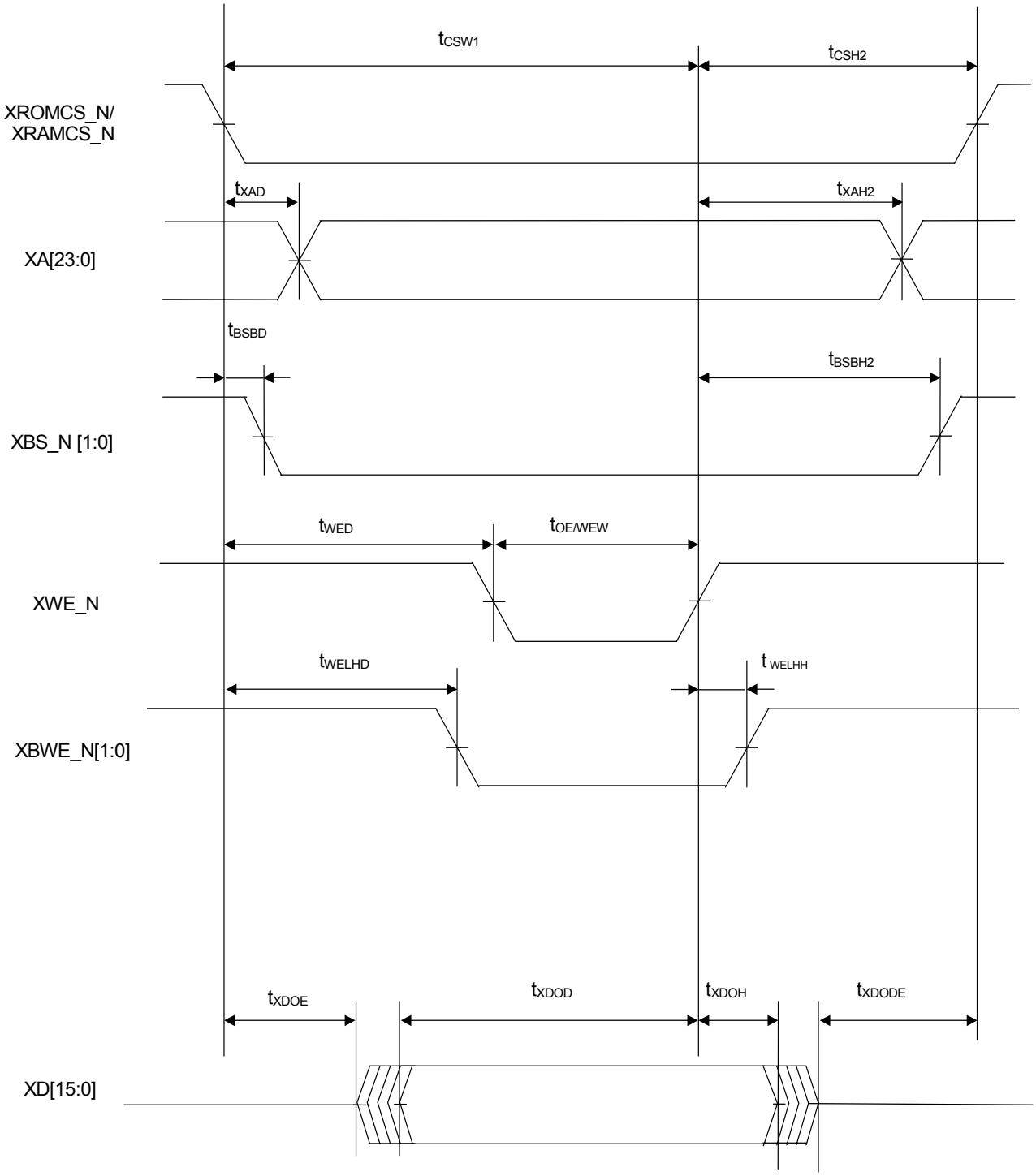

tDREQH
DREQ0 /
DREQ1
tDCLRD1
tDCLRD2
DREQCLR0 /
DREQCLR1
tTCOUTD1
tTCOUTD2
TCOUT0 /
TCOUT1

■ External Bus Timing

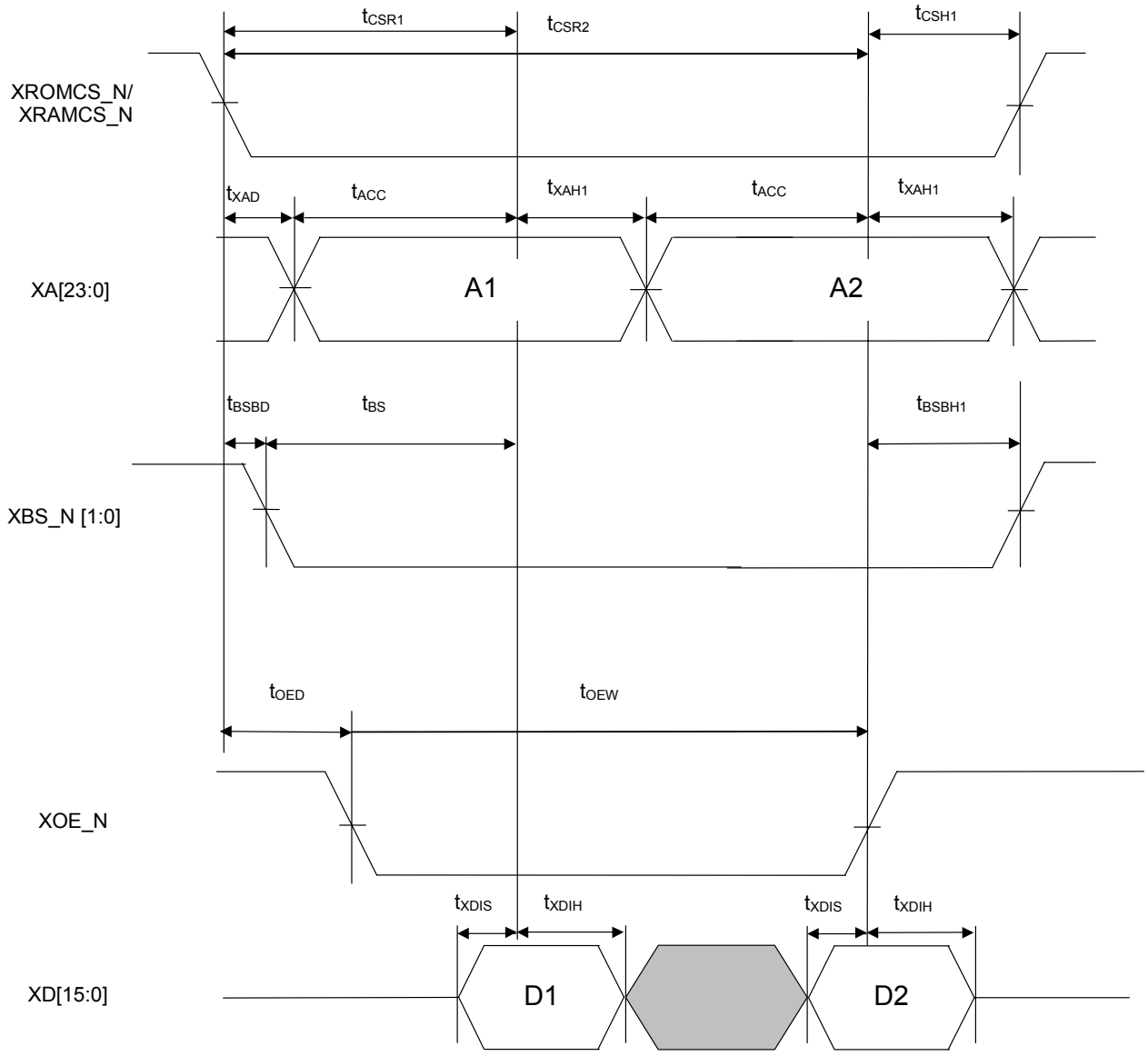
- External ROM/RAM Read Cycle
  (Bus Width 16 bit External ROM/RAM Byte/Half Word Access)



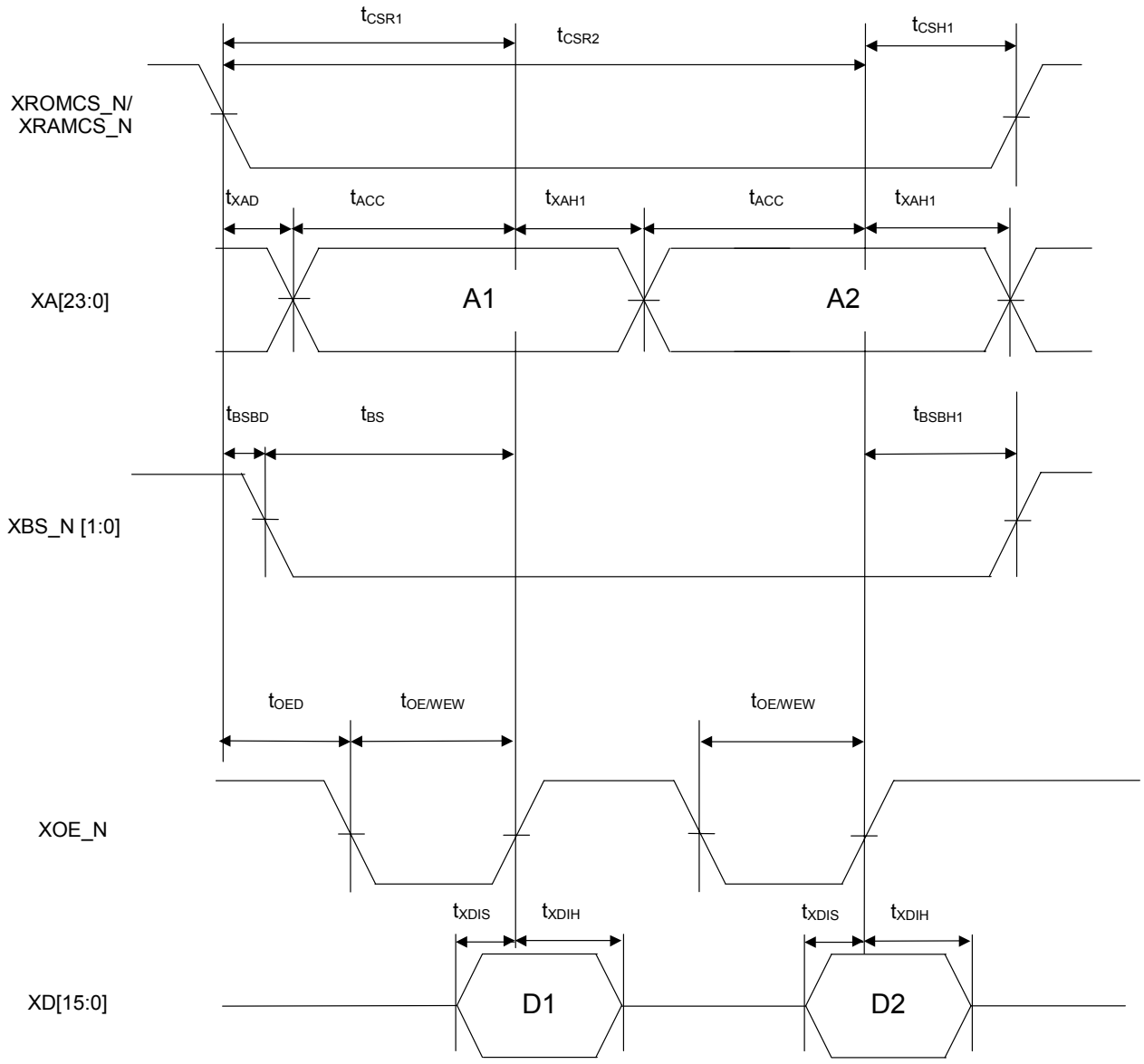
- External ROM/RAM Write Cycle
  (Bus Width16 bit External ROM/RAM Byte/Half Word Access)



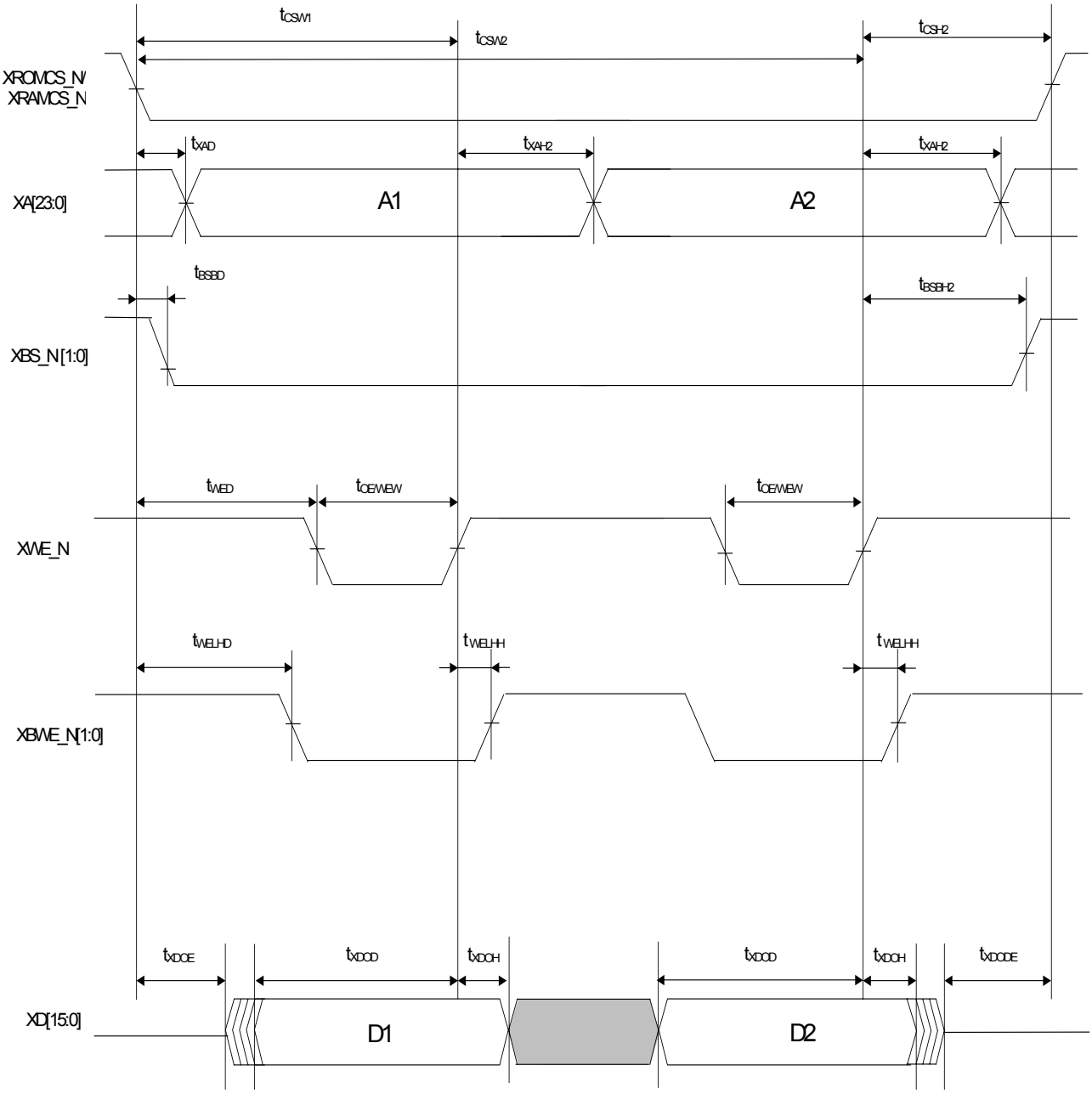
- External ROM/RAM Read Cycle
  (Bus Width16 bit External ROM/RAM Word Access from CPU)



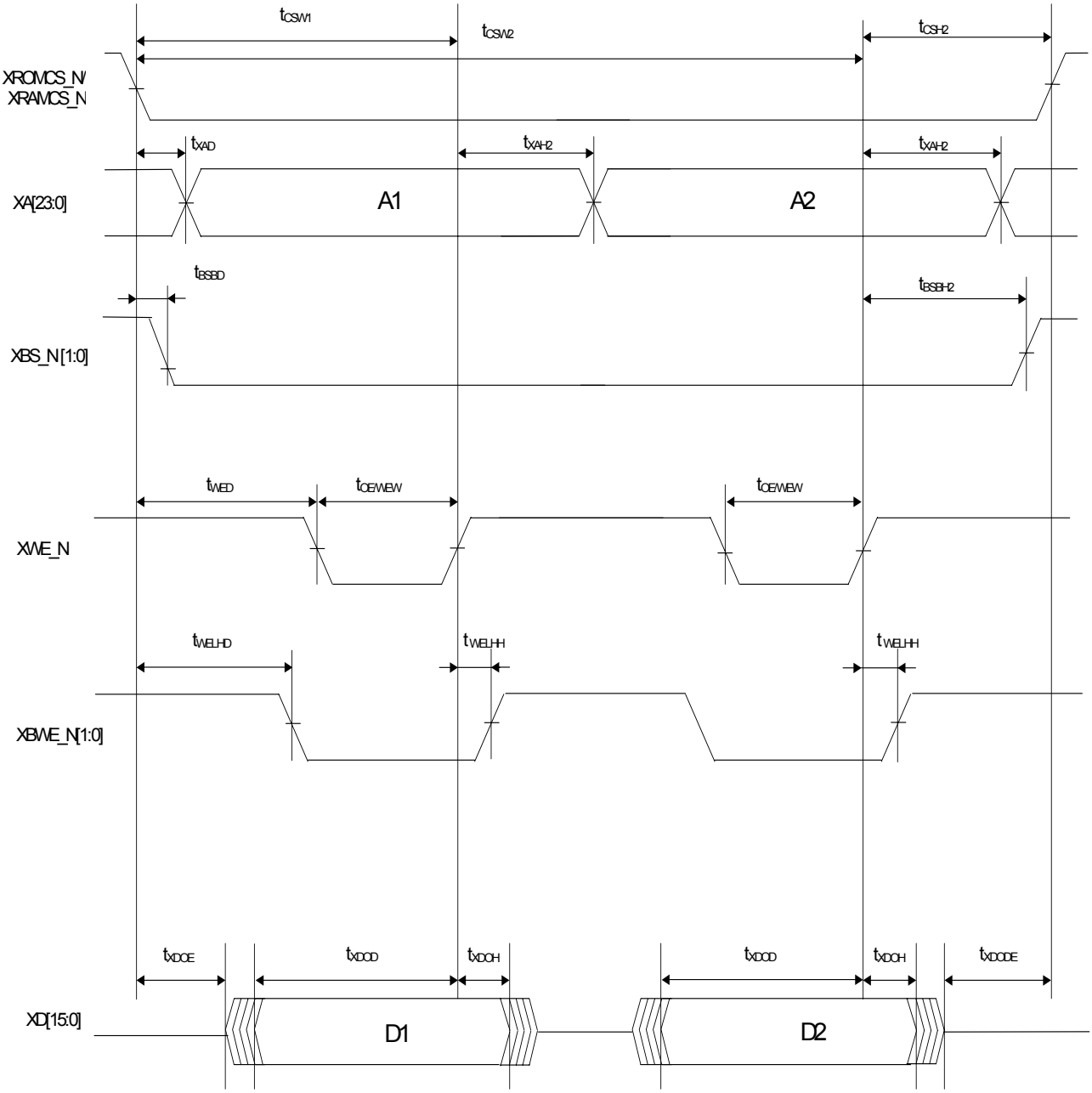
- External ROM/RAM Read Cycle
  (Bus Width16 bit External ROM/RAM Word Access from DMAC)



- External ROM/RAM Write Cycle
  (Bus Width 16 bit External ROM/RAM Word Access from CPU)

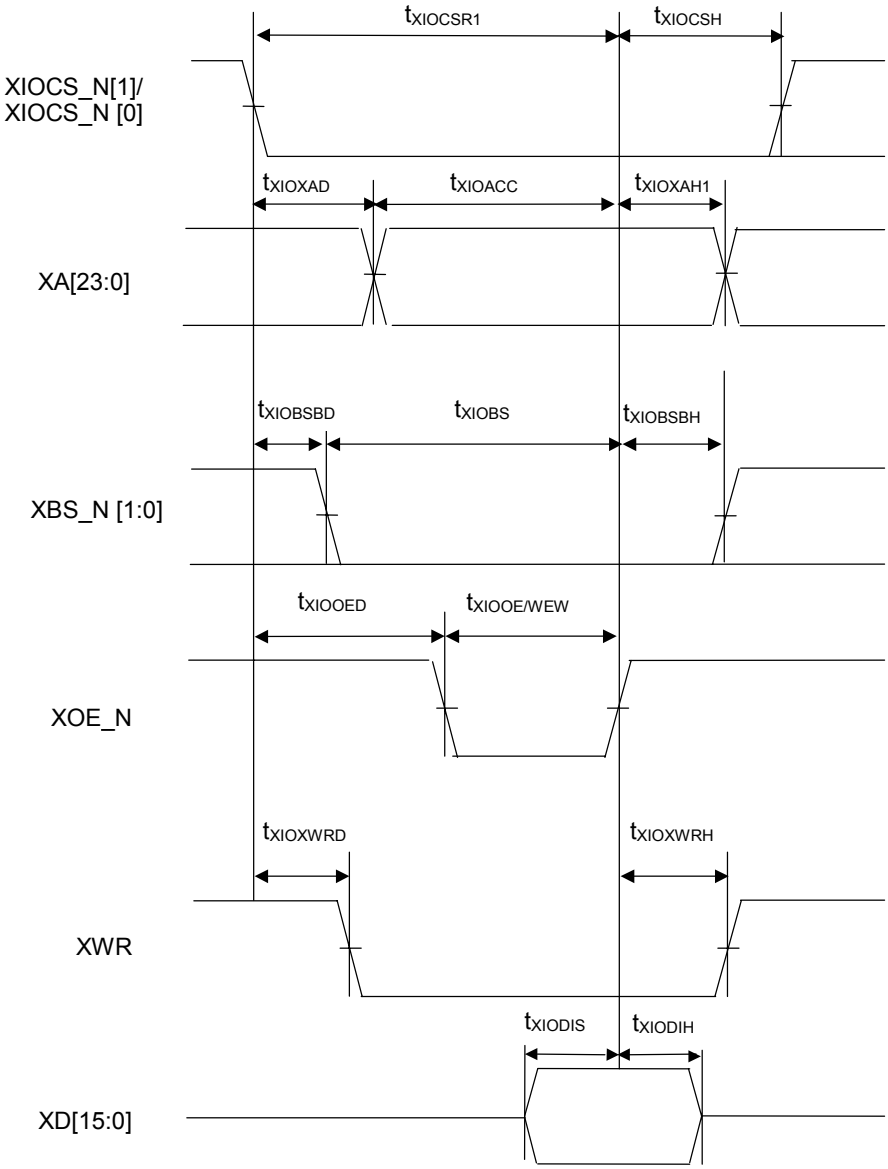


- External ROM/RAM Write Cycle
  (Bus Width 16 bit External ROM/RAM Word Access from DMAC)

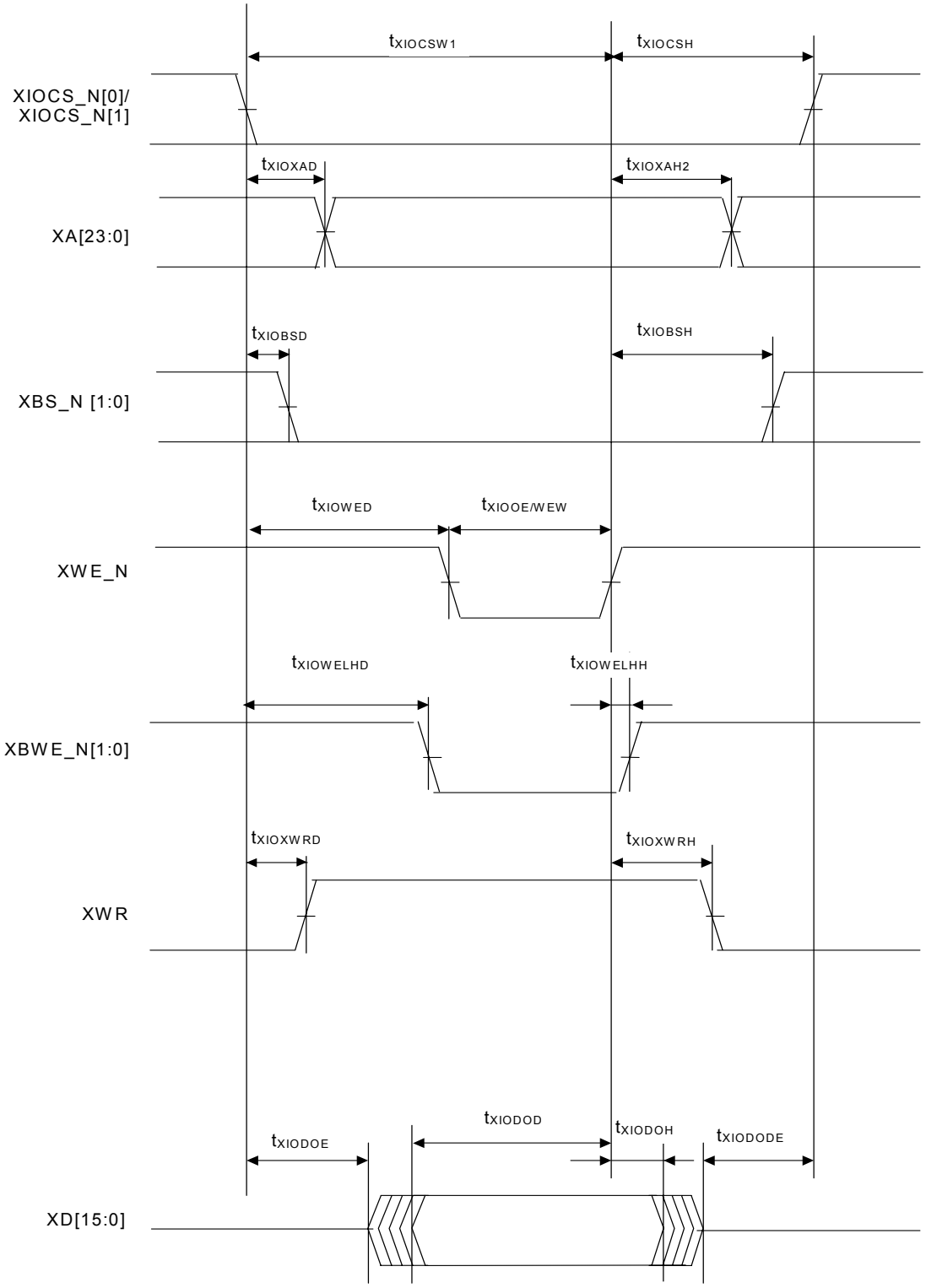


- External I/O Banks 0 and 1 Read Cycle

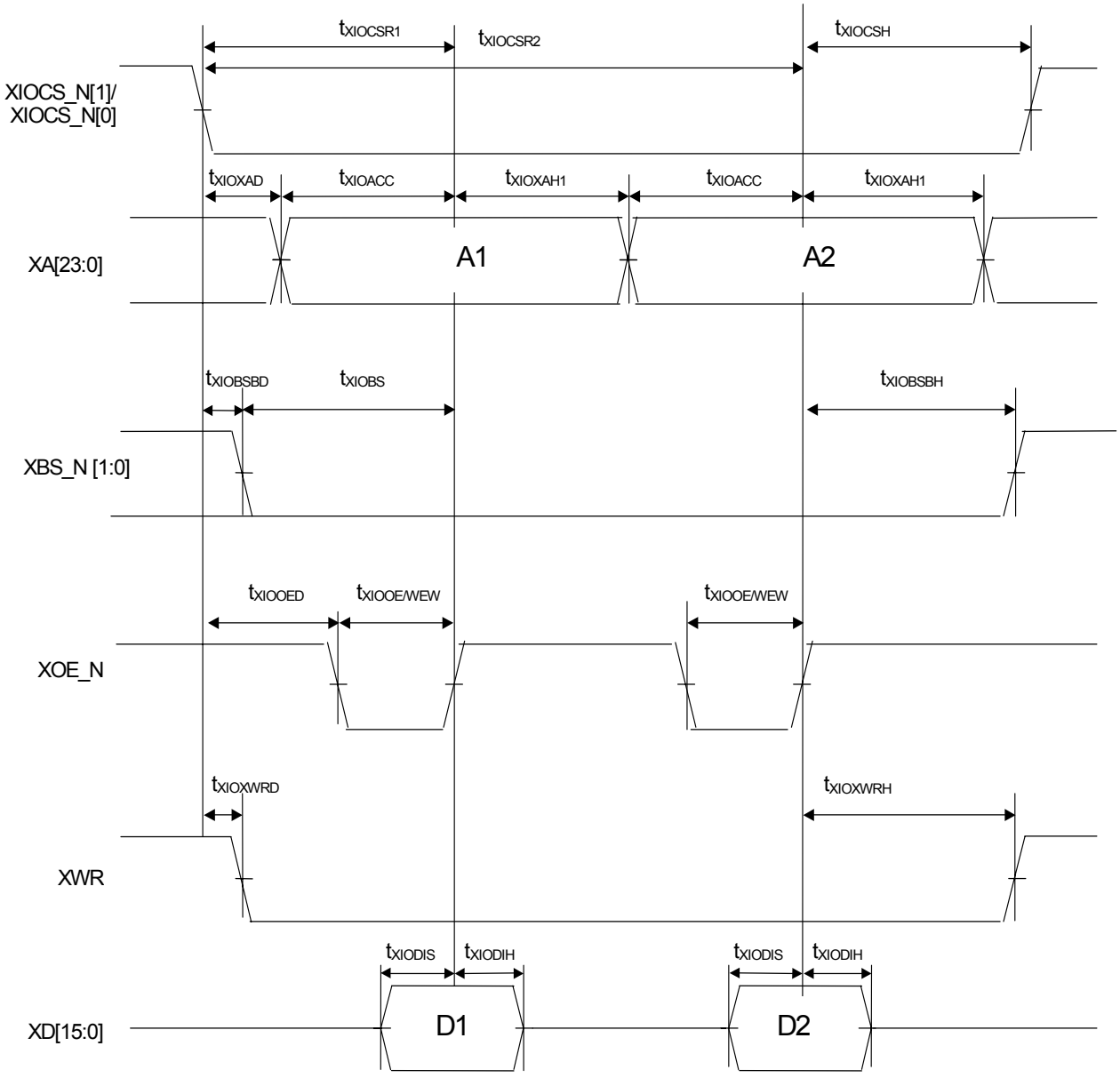
(Bus Width 16 bit External I/O Banks 0 and 1 Byte/Half Word Access)



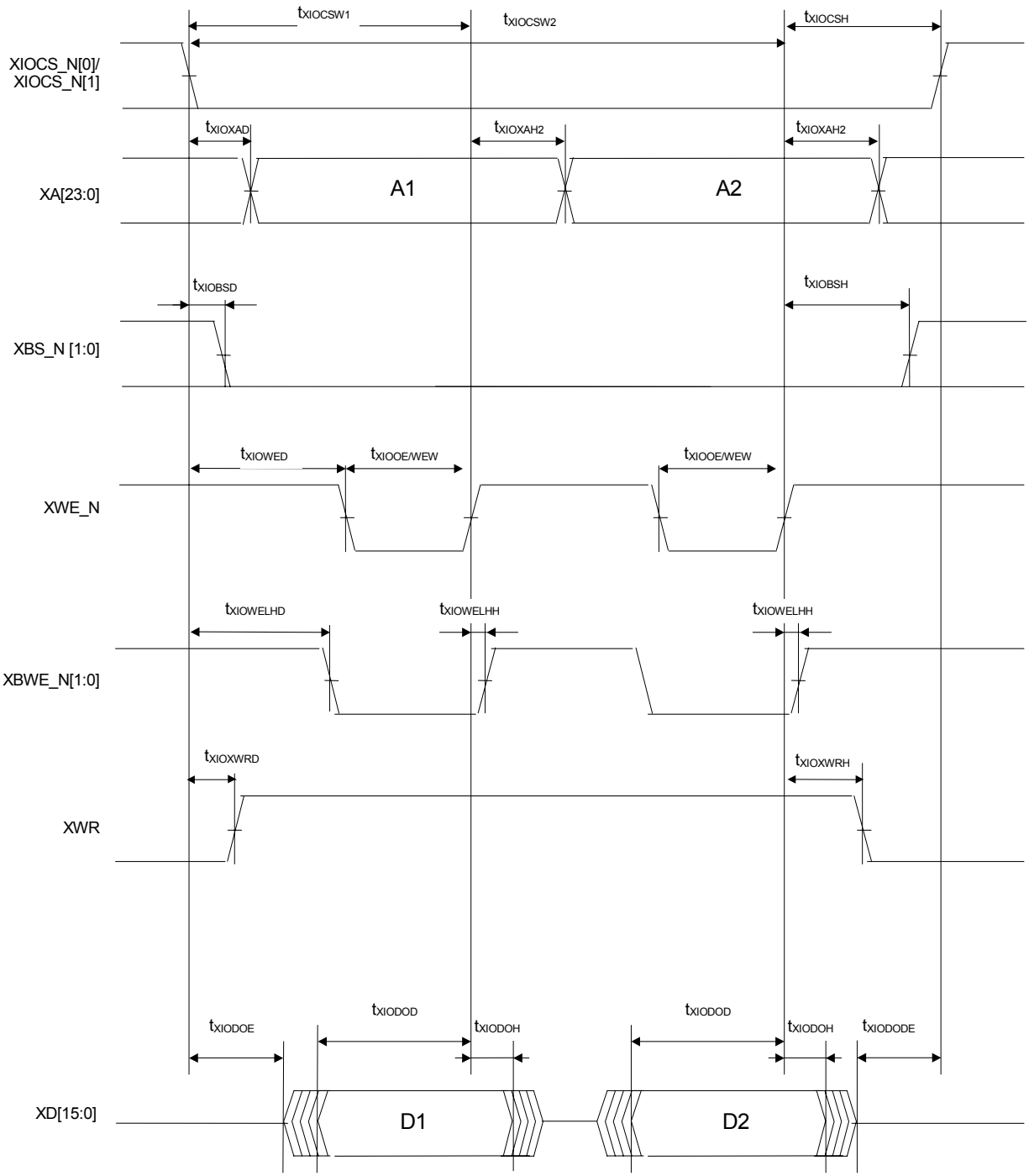
- External I/O Banks 0 and 1 Write Cycle
  (Bus Width 16 bit External I/O Banks 0 and 1 Byte/Half Word Access)



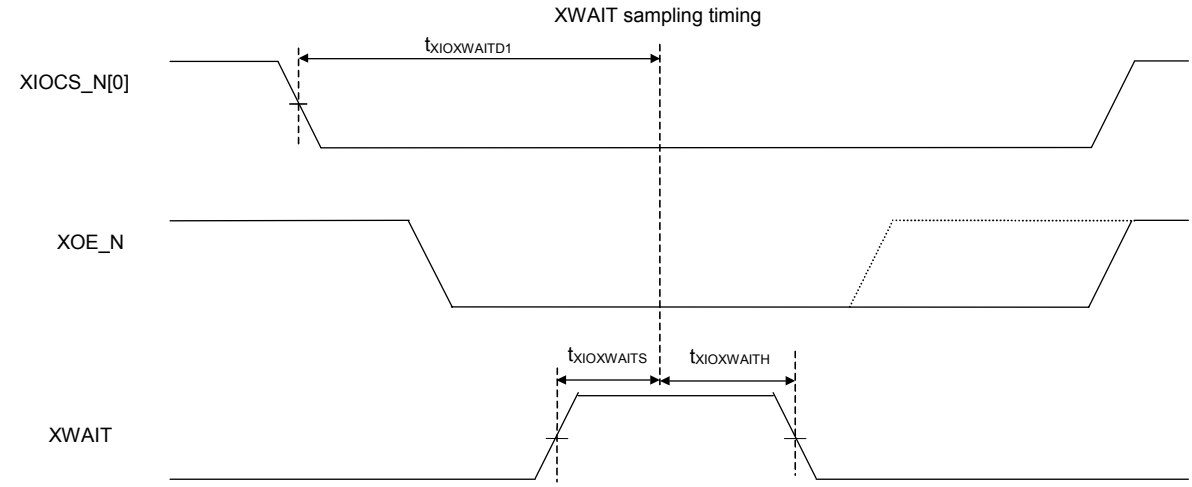
- External I/O Banks 0 and 1 Read Cycle
  (Bus Width 16 bit External I/O Banks 0 and 1 Word Access/
  Bus Width 8 bit External I/O Banks 0 and 1 Half Word Access)



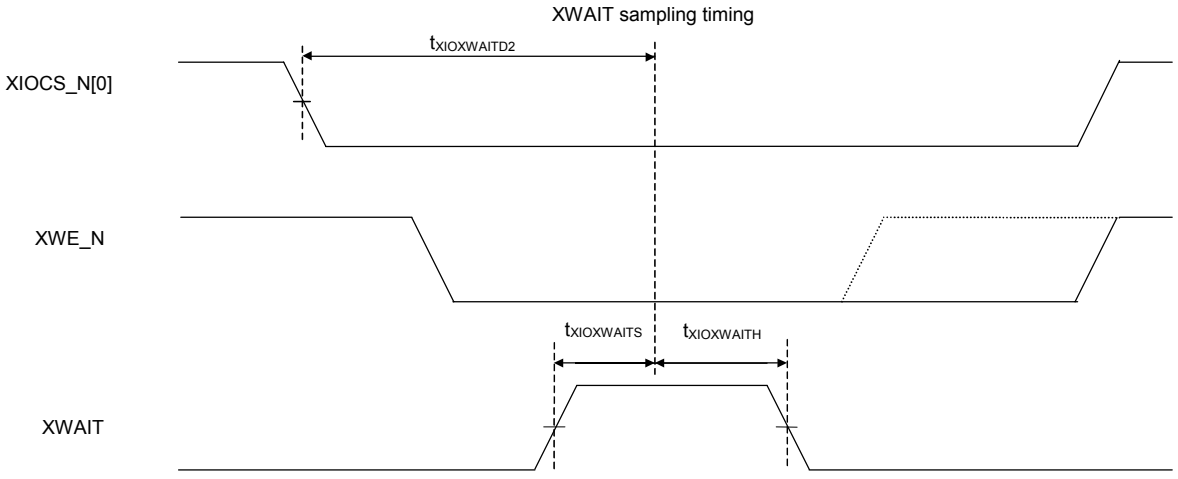
- External I/O Banks 0 and 1 Write Cycle
  (Bus Width 16 bit External I/O Banks 0 and 1 Word Access/
  Bus Width 8 bit External I/O Banks 0 and 1 Half Word Access)



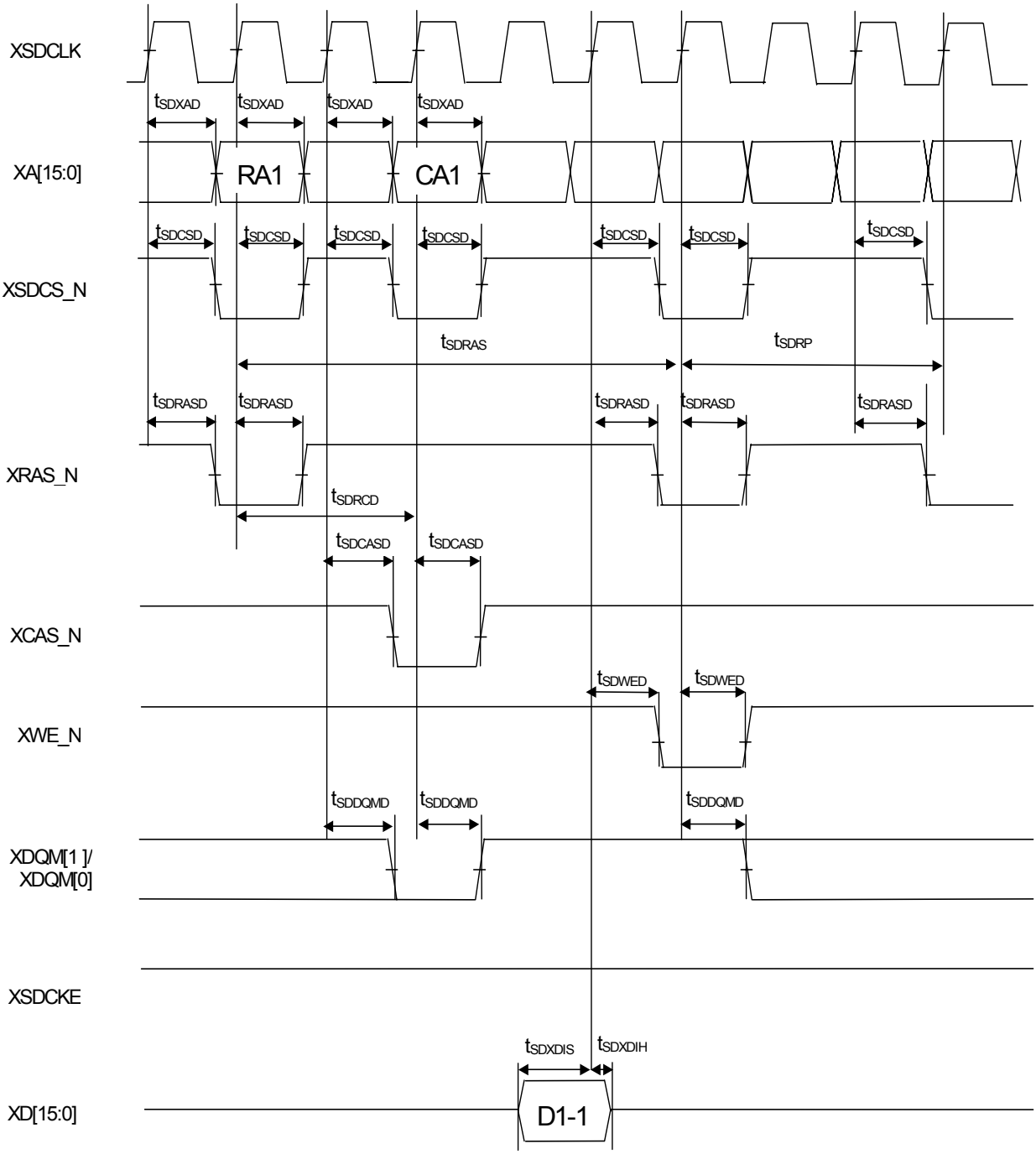
■XWAIT Signal Input Timing



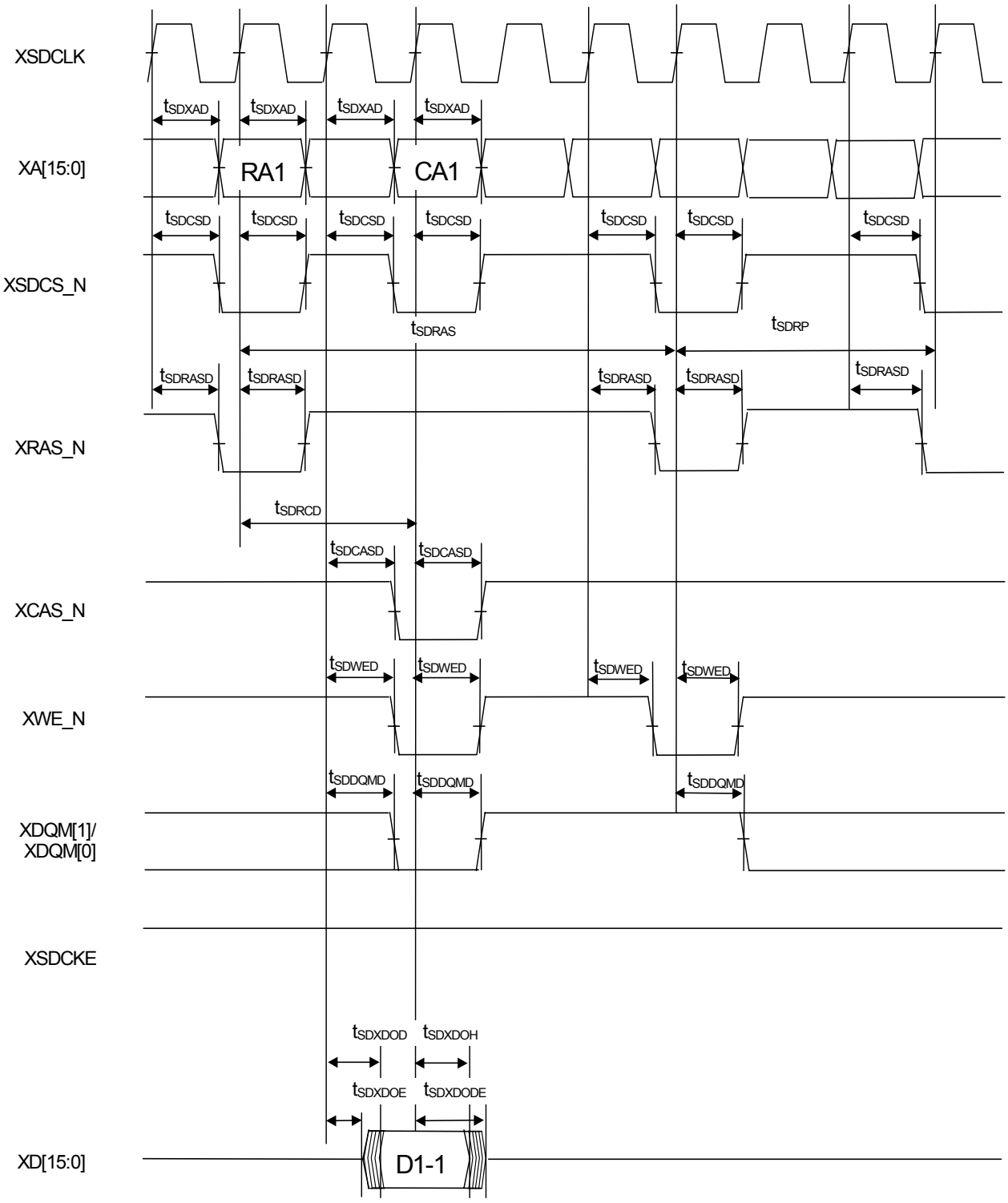
■ XWAIT Signal Input Timing



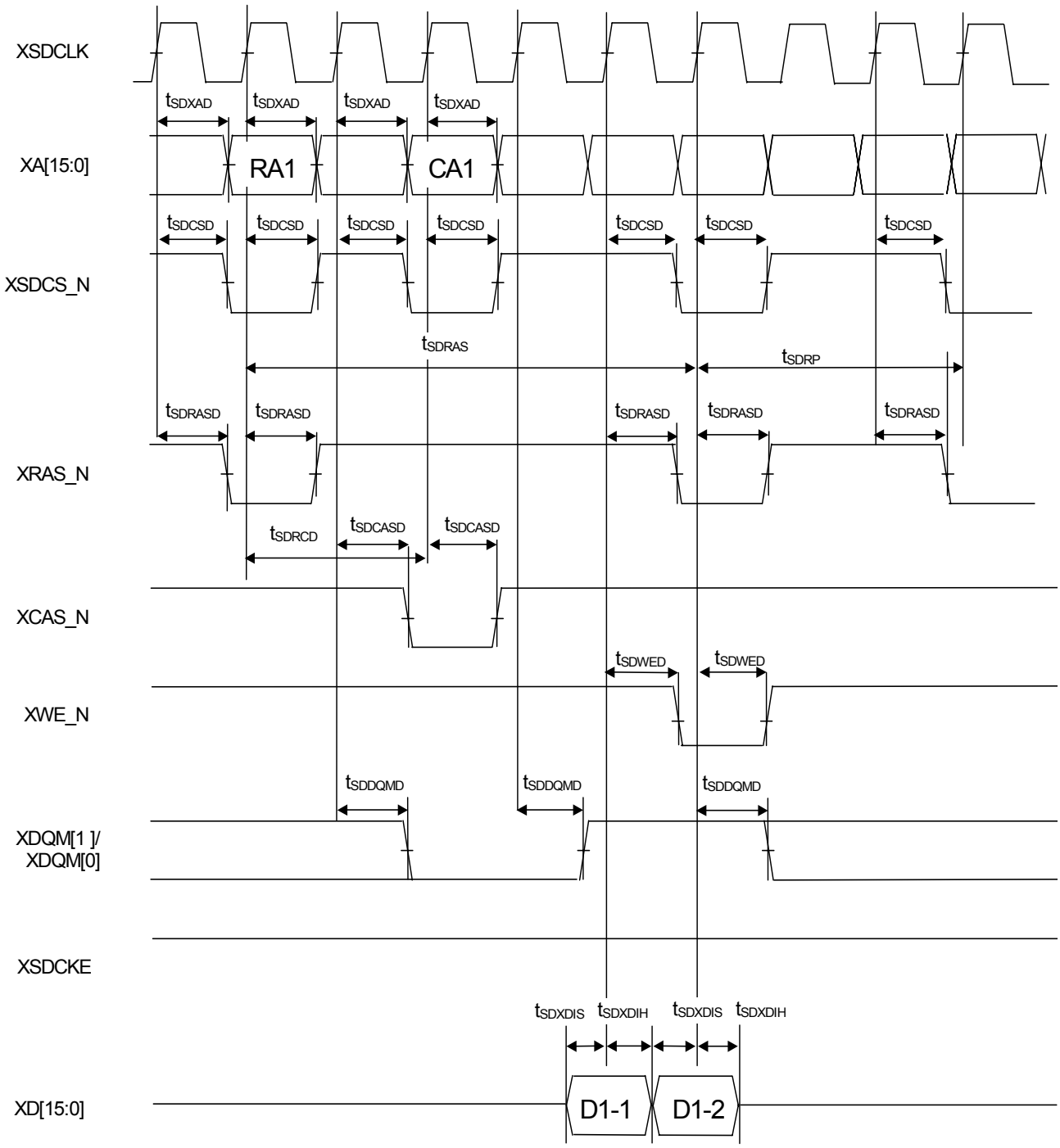
■ SDRAM Read Cycle
(Bus Width 16 bit SDRAM Byte/Half Word Access)



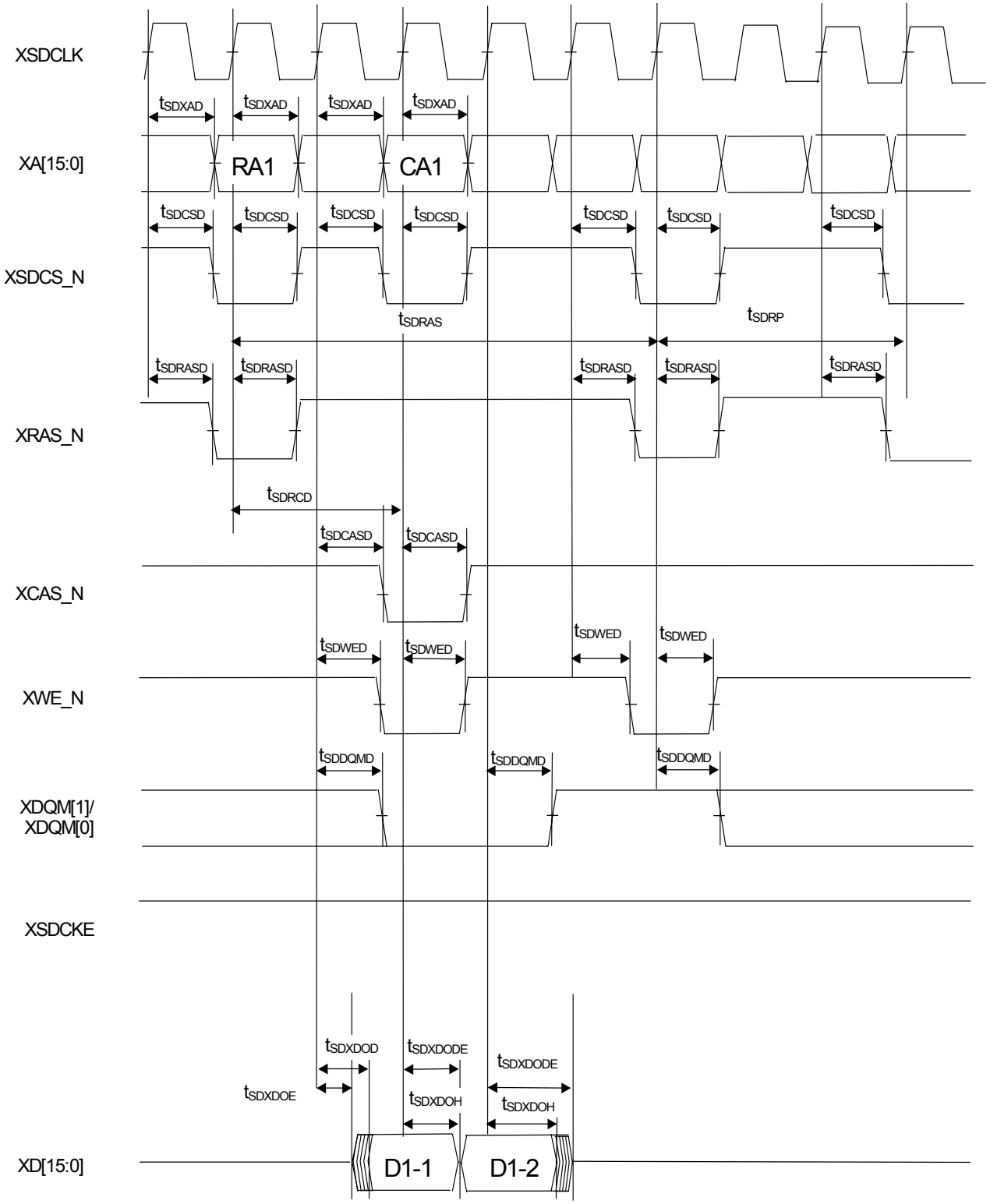
■ SDRAM Write Cycle
(Bus Width 16 bit SDRAM Byte/Half Word Access)



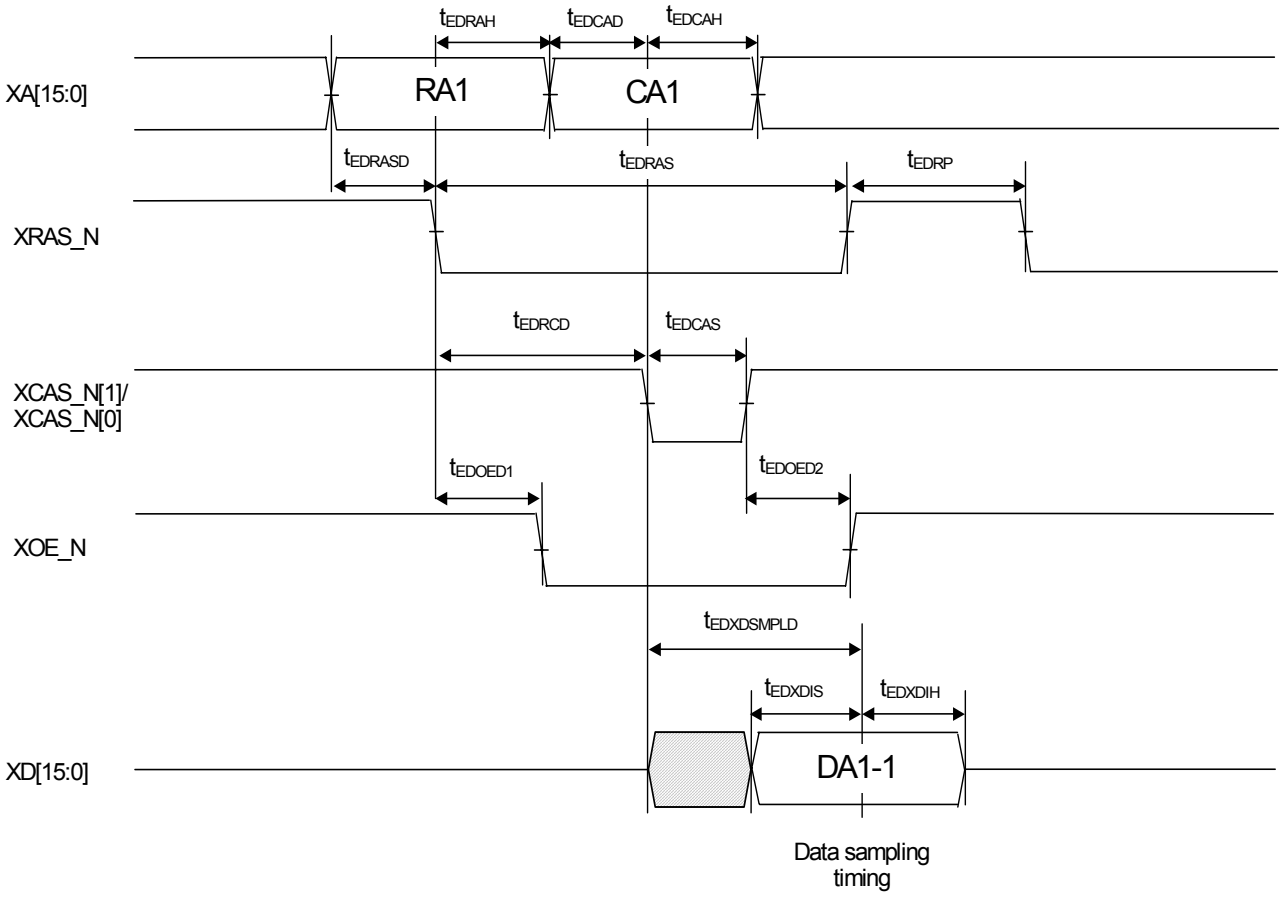
■ SDRAM Read Cycle
(Bus Width 16 bit SDRAM Word Access)



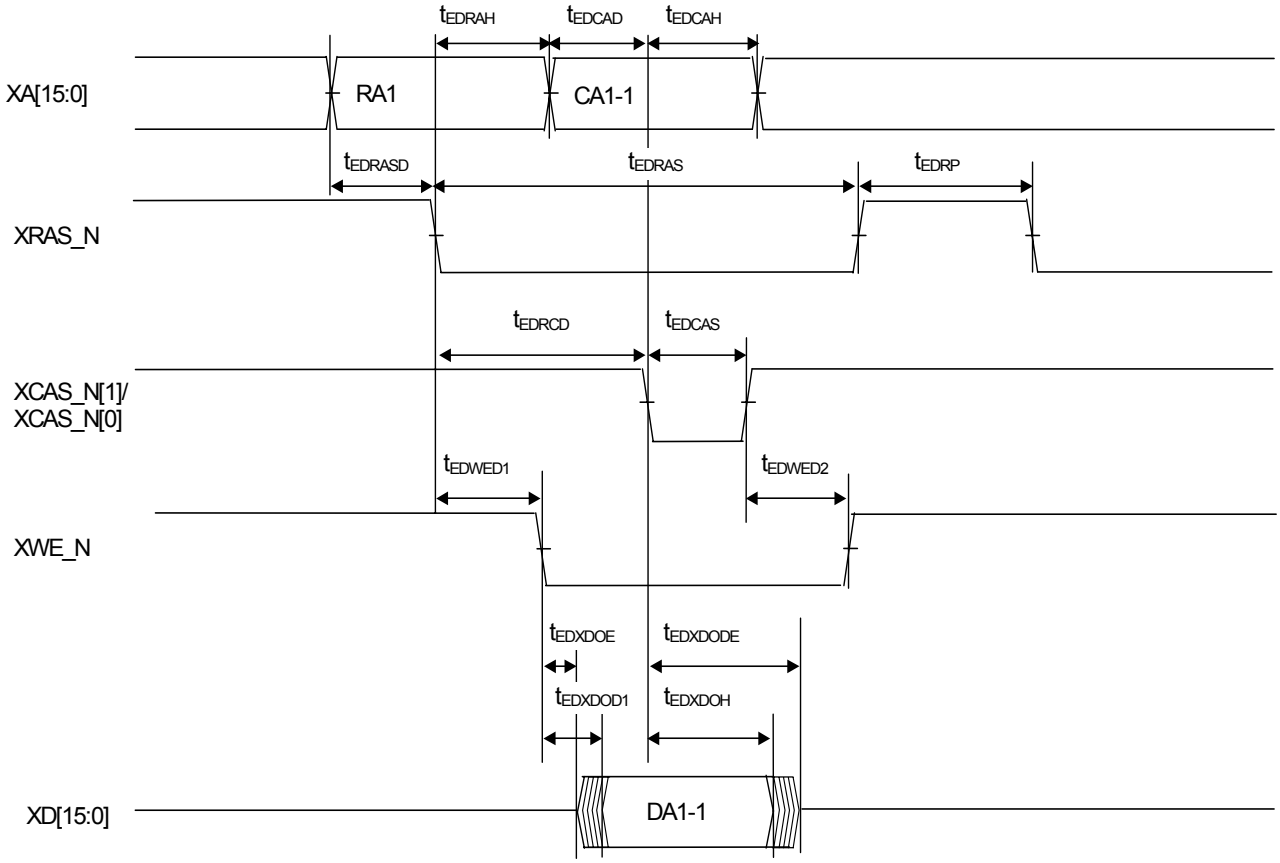
■ SDRAM Write Cycle
(Bus Width 16 bit SDRAM Word Access)



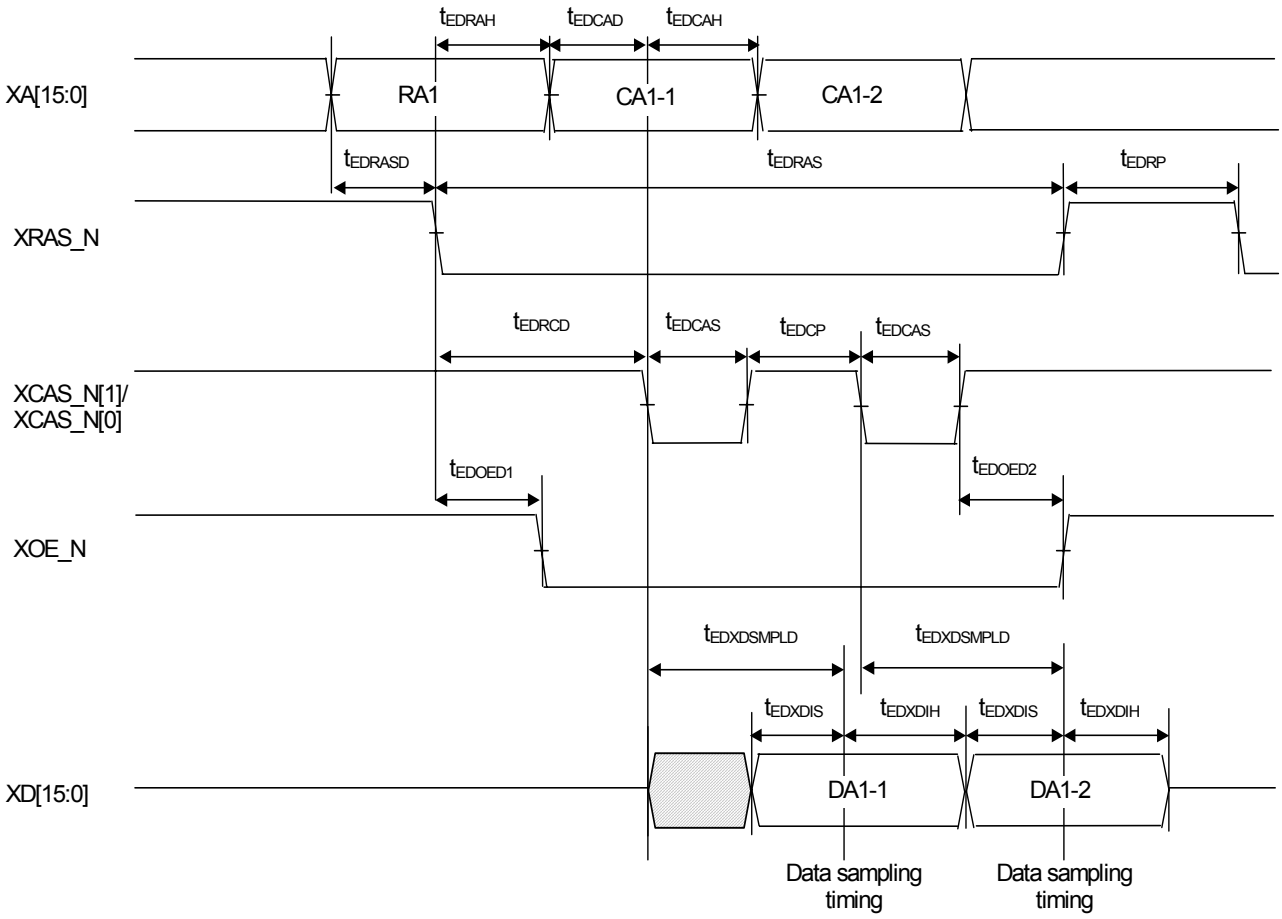
■ EDO DRAM Read Cycle
(Bus Width 16 bit EDO DRAM Byte/Half Word Access)



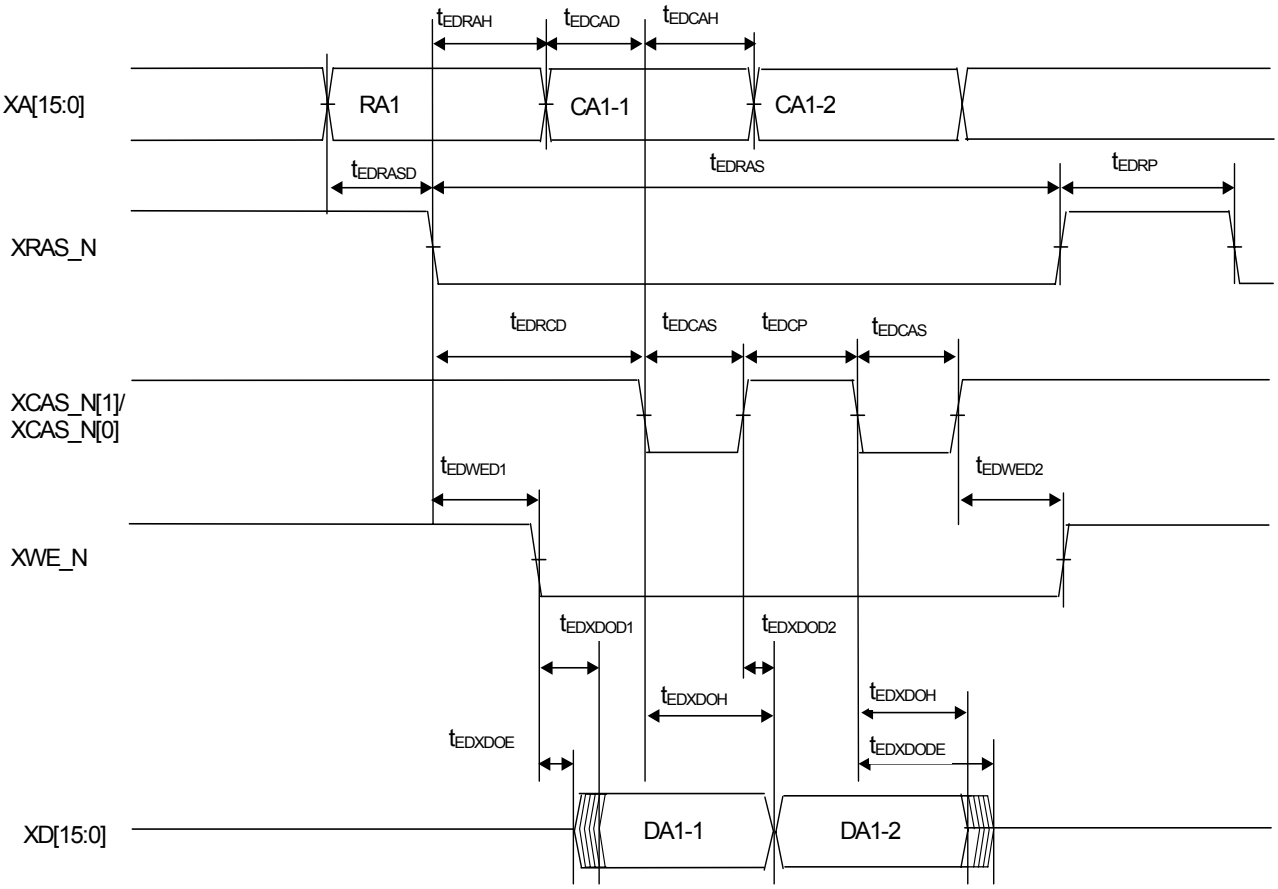
■ EDO DRAM Write Cycle
(Bus Width 16 bit EDO DRAM Byte/Half Word Access)



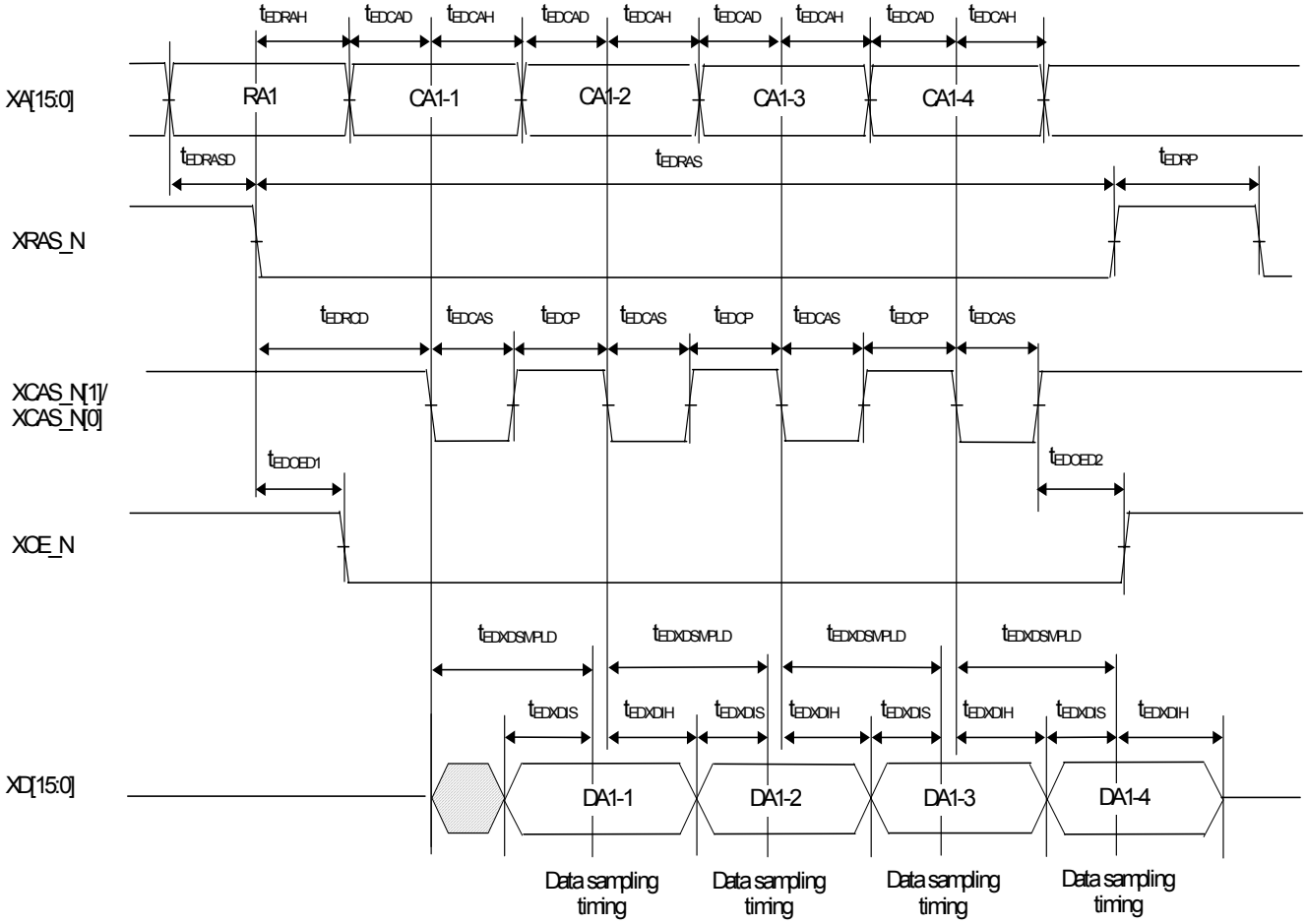
■ EDO DRAM Read Cycle
(Bus Width 8 bit EDO DRAM Half Word Access/
Bus Width 16 bit EDO DRAM Word Access)



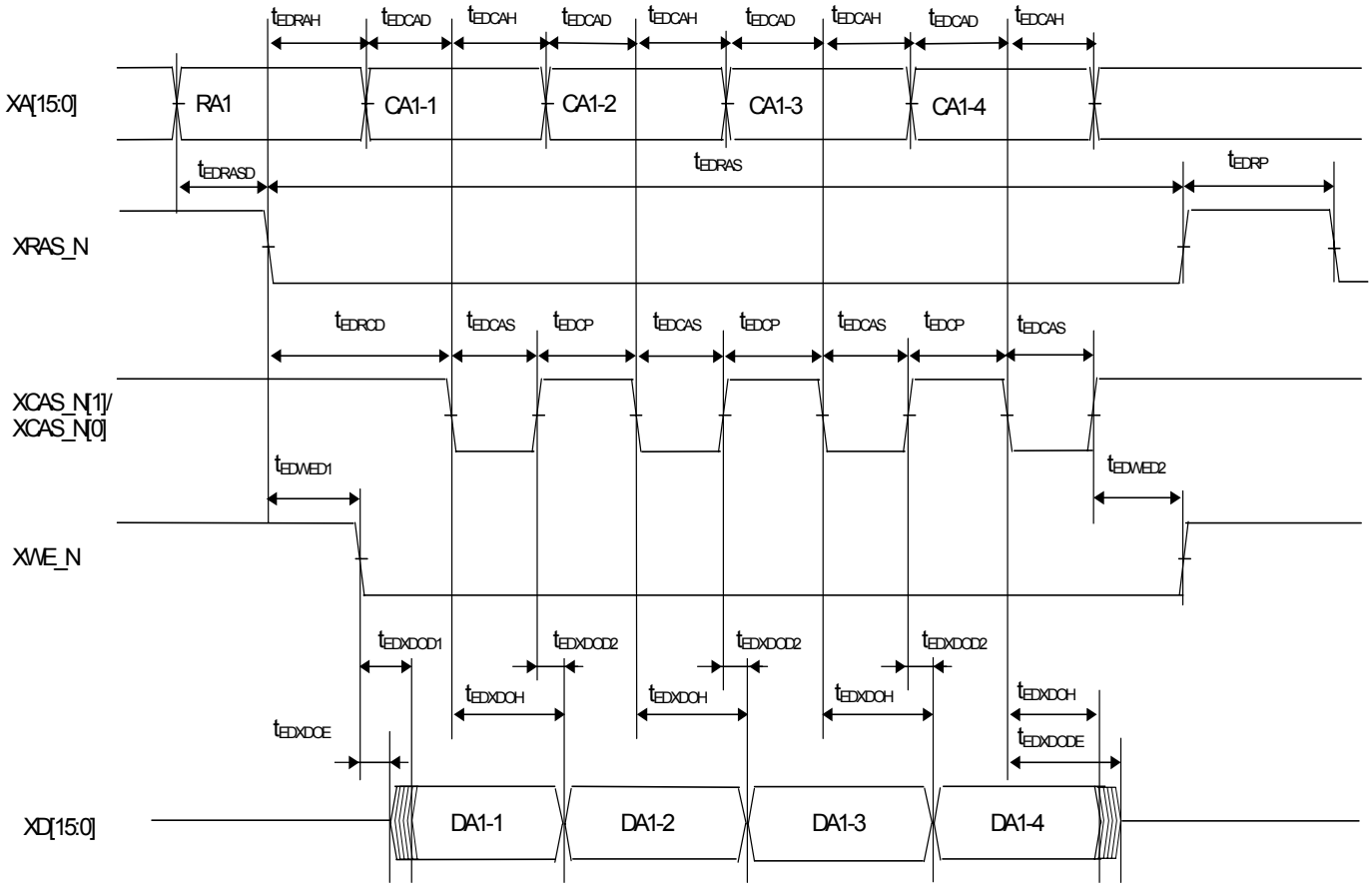
■ EDO DRAM Write Cycle
(Bus Width 8 bit EDO DRAM Half Word Access/
Bus Width 16 bit EDO DRAM Word Access)



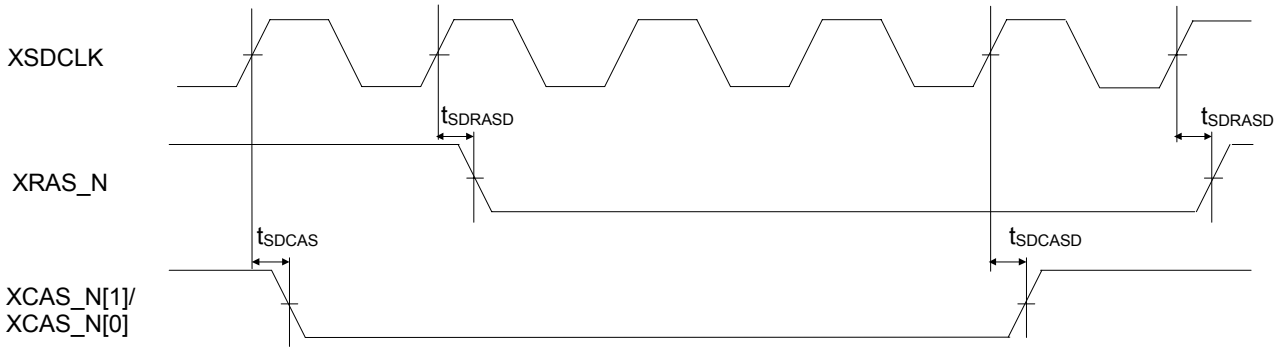
■ EDO DRAM Read Cycle
(Bus Width 8 bit EDO DRAM Word Access)



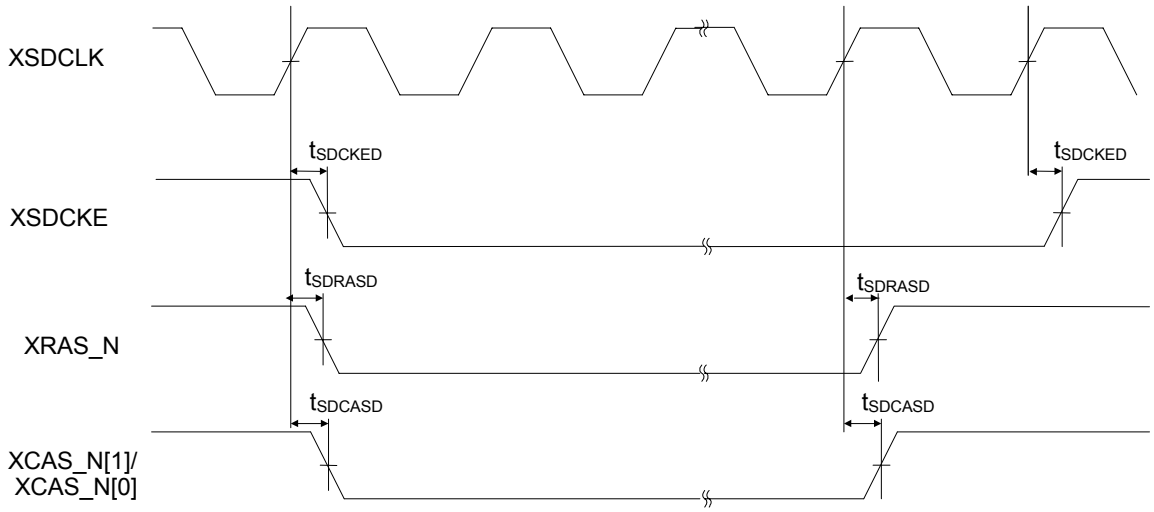
■ EDO DRAM Write Cycle
(Bus Width 8 bit EDO DRAM Word Access)



■ CAS before RAS (CBR) Refresh Operation



■ Self Refresh Operation



19.3.4 Analog-to-Digital Converter Characteristics

■ Analog-to-Digital Converter Characteristics

(V_{DD_CORE} = 2.50V, V_{DD_IO} = 3.3V, Ta = 25C)

Item	Symbol	Conditions	Minimum	Typical	Maximum	Unit
Resolution	n	—	—	—	10	bit
Linearity error	E_L	Analog input source impedance Ri ≤ 1kΩ	—	±3	—	LSB
Differential linearity error	E_D		—	±3	—	
Zero scale error	E_{ZS}		—	±3	—	
Full scale error	E_{FS}		—	±3	—	
Conversion time	t_{CONV}	—	5	—	—	µs
Throughput		—	10	—	200	kHz

Notes: VDD_io and AVDD should be supplied separately

- Definition of Terms

(1) Resolution: Minimum input analog value recognized. For 10-bit resolution, this is (VREF – Aground) ÷ 1024.

(2) Linearity error: Difference between the theoretical and actual conversion characteristics. (Note that it does not include quantization error.) The theoretical conversion characteristic divides the voltage range between VREF and AGND into 1024 equal steps.

(3) Differential linearity error: Difference between the theoretical and actual input voltage change producing a 1-bit change in the digital output anywhere within the conversion range. This is an indicator of conversion characteristic smoothness. The theoretical value is (VREF – Aground) ÷ 1024.

(4) Zero scale error: Difference between the theoretical and actual conversion characteristics at the point where the digital output switches from "0x000" to "0x001."

(5) Full scale error: Difference between the theoretical and actual conversion characteristics at the point where the digital output switches from "0x3FE" to "0x3FF."

Appendixes

Appendixes

Appendix A. Onboard Debugging

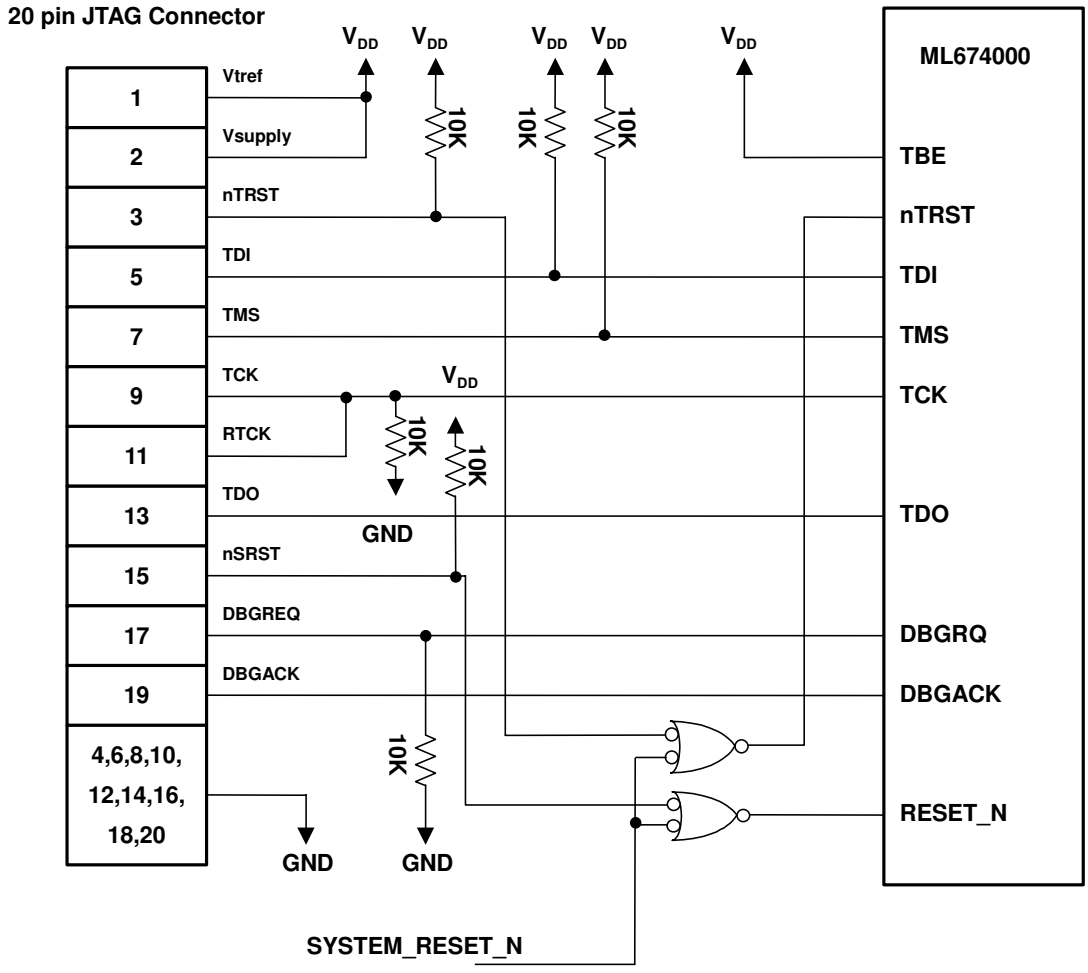
A.1 Necessary Conditions

This LSI has the built-in Embedded ICE debug extensions, made available externally through JTAG interface pins. The JTAG pins allow access to the processor core's extensive debugging facilities.
Onboard debugging requires the following.

- A standard JTAG to JTAG-ICE hardware such as ARM Multi-ICE.
- ARM Debugger for Windows (ADW) included in ARM's software development tools or other debugger software compatible with the JTAG-ICE hardware being utilized.
- Development host with the above debugger installed
- Appropriate connector cables

A.2 Connections

The following Figure gives the circuits for connecting this LSI to the JTAG-ICE hardware.



A.3 Usage Notes

The following are only the most important usage notes. For full details, refer to the JTAG-ICE manual and other references.
The following notes apply to the Oki ADI-Board.

- Set Adaptive=off.
- Set the communications speed to 10 MHz.

Appendix B. Register List

Address	Name	Abbreviation	R/W	Size	Initial Value	Ref. Pages
0x78000000	IRQ register	IRQ	R	32	0x00000000	8-7
0x78000004	Software interrupt register	IRQS	R/W	32	0x00000000	8-8
0x78000008	FIQ register	FIQ	R	32	0x00000000	8-9
0x7800000C	FIQRAW register	FIQRAW	R	32	Reflects EFIQ_N interrupt input	8-10
0x78000010	FIQ enable register	FIQEN	R/W	32	0x00000000	8-11
0x78000014	IRQ number register	IRN	R	32	0x00000000	8-12
0x78000018	Current interrupt level register	CIL	R/W	32	0x00000000	8-13
0x7800001C	(Reserved)					
0x78000020	Interrupt level control register 0	ILC0	R/W	32	0x00000000	8-14
0x78000024	Interrupt level control register 1	ILC1	R/W	32	0x00000000	8-16
0x78000028	Current interrupt level clear register	CILCL	W	32	—	8-18
0x7800002C	Current interrupt level encode register	CILE	R	32	0x00000000	8-19
0x78100000	Bus width control register	BWC	R/W	32	0x00000008	10-3
0x78100004	External ROM access control register	ROMAC	R/W	32	0x00000007	10-5
0x78100008	External RAM access control register	RAMAC	R/W	32	0x00000007	10-6
0x7810000C	External I/O bank 0 access control register	IO0AC	R/W	32	0x00000007	10-7
0x78100010	External I/O bank 1 access control register	IO1AC	R/W	32	0x00000007	10-8
0x78180000	DRAM bus width control register	DBWC	R/W	32	0x00000000	10-9
0x78180004	DRAM control register	DRMC	R/W	32	0x00000000	10-10
0x78180008	DRAM characteristics control register	DRPC	R/W	32	0x00000000	10-12
0x7818000C	SDRAM mode register	SDMD	R/W	32	0x00000001	10-13
0x78180010	DRAM command register	DCMD	R/W	32	0x00000000	10-15
0x78180014	DRAM refresh cycle control register 0	RFSH0	R/W	32	0x00000000	10-16
0x78180018	DRAM power down control register	RDWC	W	32	0x00000003	10-19
0x7818001C	DRAM refresh cycle control register 1	RFSH1	R/W	32	0x00000000	10-17
0x7BE00000	DMA mode register	DMAMOD	R/W	32	0x00000000	11-5
0x7BE00004	DMA status register	DMASTA	R	32	0x00000000	11-6
0x7BE00008	DMA transfer complete status register	DMAINT	R	32	0x00000000	11-7
0x7BE00100	DMA channel mask register	DMACMSK0	R/W	32	0x00000001	11-9
0x7BE00104	DMA transfer mode register	DMACTMOD0	R/W	32	0x00000040	11-10
0x7BE00108	DMA transfer source address register	DMACSAD0	R/W	32	0x00000000	11-12
0x7BE0010C	DMA transfer destination address register	DMACDAD0	R/W	32	0x00000000	11-13
0x7BE00110	DMA transfer count register	DMACSIZ0	R/W	32	0x00000000	11-14
0x7BE00114	DMA transfer complete status clear register	DMACCINT0	W	32	—	11-15
0x7BE00200	DMA channel mask register	DMACMSK1	R/W	32	0x00000001	11-9
0x7BE00204	DMA transfer mode register	DMACTMOD1	R/W	32	0x00000040	11-10
0x7BE00208	DMA transfer source address register	DMACSAD1	R/W	32	0x00000000	11-12

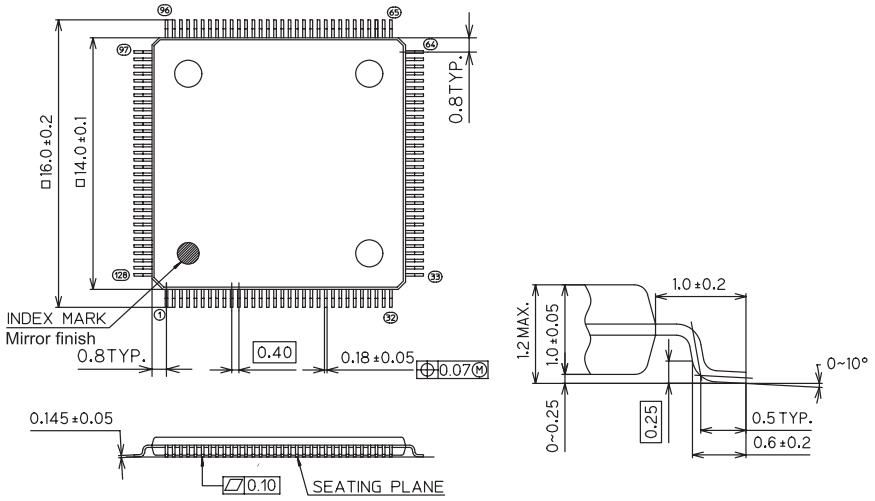
Address	Name	Abbreviation	R/W	Size	Initial Value	Ref. Pages
0x7BE0020C	DMA transfer destination address register	DMACDAD1	R/W	32	0x00000000	11-13
0x7BE00210	DMA transfer count register	DMACSIZ1	R/W	32	0x00000000	11-14
0x7BE00214	DMA transfer complete status clear register	DMACCINT1	W	32	—	11-15
0x7BF00000	(Reserved)					
0x7BF00004	IRQ clear register	IRCL	W	32	—	8-20
0x7BF00010	IRQA register	IRQA	R/W	32	0x00000000	8-21
0x7BF00014	IRQ detection mode setting register	IDM	R/W	32	0x00000000	8-23
0x7BF00018	Interrupt level control register	ILC	R/W	32	0x00000000	8-24
0xB6000000	Analog-to-digital converter control register 0	ADCON0	R/W	16	0x0000	18-4
0xB6000004	Analog-to-digital converter control register 1	ADCON1	R/W	16	0x0000	18-6
0xB6000008	Analog-to-digital converter control register 2	ADCON2	R/W	16	0x0003	18-7
0xB600000C	Analog-to-digital converter interrupt control register	ADINT	R/W	16	0x0000	18-8
0xB6000010	Analog-to-digital converter forced interrupt register	ADFINT	R/W	16	0x0000	18-10
0xB6000014	Analog-to-digital converter result register 0	ADR0	R/W	16	0x0000	18-11
0xB6000018	Analog-to-digital converter result register 1	ADR1	R/W	16	0x0000	18-11
0xB600001C	Analog-to-digital converter result register 2	ADR2	R/W	16	0x0000	18-11
0xB6000020	Analog-to-digital converter result register 3	ADR3	R/W	16	0x0000	18-11
0xB6000024	Analog-to-digital converter result register 4	ADR4	R/W	16	0x0000	18-11
0xB6000028	Analog-to-digital converter result register 5	ADR5	R/W	16	0x0000	18-11
0xB600002C	Analog-to-digital converter result register 6	ADR6	R/W	16	0x0000	18-11
0xB6000030	Analog-to-digital converter result register 7	ADR7	R/W	16	0x0000	18-11
0xB7000000	Port function select register	GPCTL	R/W	16	0x0000	12-11
0xB7000004	Block clock control register	BCKCTL	R/W	16	0x0000	7-4
0xB7A00000	Port A output register	GPPOA	R/W	16	Indeterminate	12-5
0xB7A00004	Port A input register	GPPIA	R	16	Reflects pin states	12-6
0xB7A00008	Port A mode register	GPPMA	R/W	16	0x0000	12-7
0xB7A0000C	Port A interrupt enable register	GPIEA	R/W	16	0x0000	12-8
0xB7A00010	Port A interrupt polarity register	GPIPA	R/W	16	0x0000	12-9
0xB7A00014	Port A interrupt status register	GPISA	R/W	16	0x0000	12-10
0xB7A00020	Port B output register	GPPOB	R/W	16	Indeterminate	12-5

Address	Name	Abbreviation	R/W	Size	Initial Value	Ref. Pages
0xB7A00024	Port B input register	GPPIB	R	16	Reflects pin states	12-6
0xB7A00028	Port B mode register	GPPMB	R/W	16	0x0000	12-7
0xB7A0002C	Port B interrupt enable register	GPIEB	R/W	16	0x0000	12-8
0xB7A00030	Port B interrupt polarity register	GPIPB	R/W	16	0x0000	12-9
0xB7A00034	Port B interrupt status register	GPISB	R/W	16	0x0000	12-10
0xB7B00000	Divisor Latch (LSB)	UARTDLL	R/W	8	0x00	17-22
0xB7B00000	Receiver Buffer Register	UARTRBR	R	8	Indeterminate	17-4
0xB7B00000	Transmitter Holding Register	UARTTHR	W	8	Indeterminate	17-5
0xB7B00004	Divisor Latch (MSB)	UARTDLM	R/W	8	0x00	17-23
0xB7B00004	Interrupt Enable Register	UARTIER	R/W	8	0x00	17-6
0xB7B00008	FIFO Control Register	UARTFCR	W	8	0x00	17-10
0xB7B00008	Interrupt Identification Register	UARTIIR	R	8	0x01	17-8
0xB7B0000C	Line Control Register	UARTLCR	R/W	8	0x00	17-12
0xB7B00010	Modem Control Register	UARTMCR	R/W	8	0x00	17-14
0xB7B00014	Line Status Register	UARTLSR	R/W	8	0x60	17-16
0xB7B00018	Modem Status Register	UARTMSR	R/W	8	Contents depend on pin states	17-19
0xB7B0001C	Scratch Register	UARTSCR	R/W	8	Indeterminate	17-21
0xB7D00000	PWM register 0	PWR0	R/W	16	0x0000	15-3
0xB7D00004	PWM period register 0	PWCY0	R/W	16	0x0000	15-4
0xB7D00008	PWM counter 0	PWC0	R/W	16	0x0000	15-5
0xB7D0000C	PWM control register 0	PWCON0	R/W	16	0x0000	15-6
0xB7D00020	PWM register 1	PWR1	R/W	16	0x0000	15-3
0xB7D00024	PWM period register 1	PWCY1	R/W	16	0x0000	15-4
0xB7D00028	PWM counter 1	PWC1	R/W	16	0x0000	15-5
0xB7D0002C	PWM control register 1	PWCON1	R/W	16	0x0000	15-6
0xB7D0003C	PWM Interrupt status register	PWINTSTS	R/W	16	0x0000	15-7
0xB7E00000	Watchdog timer control register	WDTCON	W	8	—	13-2
0xB7E00004	Time base counter control register	WDTBCON	R/W	8	0x00	13-3
0xB7E00014	Status register	WDSTAT	R/W	8	0x00	13-5
0xB7F00000	Timer 0 control register	TIMECNTL0	R/W	16	0x0000	14-7
0xB7F00004	Timer 0 base register	TIMEBASE0	R/W	16	0x0000	14-9
0xB7F00008	Timer 0 counter register	TIMECNT0	R	16	0x0000	14-10
0xB7F0000C	Timer 0 compare register	TIMECMP0	R/W	16	0xFFFF	14-11
0xB7F00010	Timer 0 status register	TIMESTAT0	R/W	16	0x0000	14-12
0xB7F00020	Timer 1 control register	TIMECNTL1	R/W	16	0x0000	14-7
0xB7F00024	Timer 1 base register	TIMEBASE1	R/W	16	0x0000	14-9
0xB7F00028	Timer 1 counter register	TIMECNT1	R	16	0x0000	14-10
0xB7F0002C	Timer 1 compare register	TIMECMP1	R/W	16	0xFFFF	14-11
0xB7F00030	Timer 1 status register	TIMESTAT1	R/W	16	0x0000	14-12
0xB7F00040	Timer 2 control register	TIMECNTL2	R/W	16	0x0000	14-7
0xB7F00044	Timer 2 base register	TIMEBASE2	R/W	16	0x0000	14-9
0xB7F00048	Timer 2 counter register	TIMECNT2	R	16	0x0000	14-10

Address	Name	Abbreviation	R/W	Size	Initial Value	Ref. Pages
0xB7F0004C	Timer 2 compare register	TIMECMP2	R/W	16	0xFFFF	14-11
0xB7F00050	Timer 2 status register	TIMESTAT2	R/W	16	0x0000	14-12
0xB7F00060	Timer 3 control register	TIMECNTL3	R/W	16	0x0000	14-7
0xB7F00064	Timer 3 base register	TIMEBASE3	R/W	16	0x0000	14-9
0xB7F00068	Timer 3 counter register	TIMECNT3	R	16	0x0000	14-10
0xB7F0006C	Timer 3 compare register	TIMECMP3	R/W	16	0xFFFF	14-11
0xB7F00070	Timer 3 status register	TIMESTAT3	R/W	16	0x0000	14-12
0xB7F00080	Timer 4 control register	TIMECNTL4	R/W	16	0x0000	14-7
0xB7F00084	Timer 4 base register	TIMEBASE4	R/W	16	0x0000	14-9
0xB7F00088	Timer 4 counter register	TIMECNT4	R	16	0x0000	14-10
0xB7F0008C	Timer 4 compare register	TIMECMP4	R/W	16	0xFFFF	14-11
0xB7F00090	Timer 4 status register	TIMESTAT4	R/W	16	0x0000	14-12
0xB7F000A0	Timer 5 control register	TIMECNTL5	R/W	16	0x0000	14-7
0xB7F000A4	Timer 5 base register	TIMEBASE5	R/W	16	0x0000	14-9
0xB7F000A8	Timer 5 counter register	TIMECNT5	R	16	0x0000	14-10
0xB7F000AC	Timer 5 compare register	TIMECMP5	R/W	16	0xFFFF	14-11
0xB7F000B0	Timer 5 status register	TIMESTAT5	R/W	16	0x0000	14-12
0xB8000004	Clock stop register	CLKSTP	R/W	32	0x00000000	7-7
0xB8000008	Clock select register	CGBCNT0	R/W	32	0x00000000	7-9
0xB800000C	Clock wait register	CKWT	R/W	32	0x000000FF	7-10
0xB8000010	Remap control register	RMPCON	R/W	32	0x00000000	3-3
0xB8001004	System timer enable register	TMEN	R/W	32	0x00000000	14-4
0xB8001008	System timer reload register	TMRLR	R/W	32	0x00000000	14-5
0xB8001010	System counter overflow register	TMOVFR	R/W	32	0x00000000	14-6
0xB8002000	SIO transfer buffer register	SIOBUF	R/W	32	0x00000000	16-3
0xB8002004	SIO status register	SIOSTA	R/W	32	0x00000000	16-4
0xB8002008	SIO control register	SIOCON	R/W	32	0x00000000	16-6
0xB800200C	Baud rate control register	SIOBCN	R/W	32	0x00000000	16-7
0xB8002010	(Reserved)					
0xB8002014	Baud rate timer register	SIOBT	R/W	32	0x00000000	16-8
0xB8002018	SIO test control register	SIOTCN	R/W	32	0x00000000	16-9

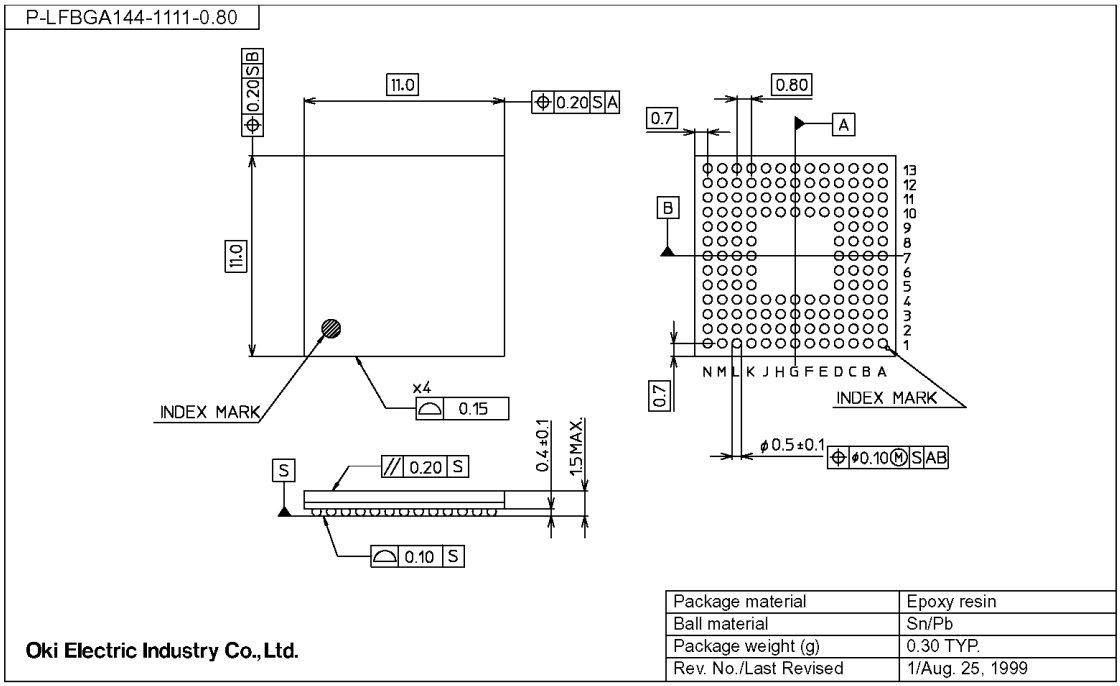
Appendix C. Package Dimensions

TQFP128-P-1414-0.40-K



Package material	Epoxy resin
Lead frame material	42 alloy
Pin treatment	Solder plating (≥ 5 µm)
Package weight (g)	0.55 TYP.
Rev.No./Last Revised	5 1/Nov.18,1999

Oki Electric Industry Co., Ltd.



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 Oki'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	
PEUL674000-01	June 10, 2002	–	–	Preliminary edition 1
FEUL674000-01	July 4, 2002	10-6	10-6	Parameter Table Changed
		10-7	10-7	Parameter Table Changed
		10-8	10-8	Parameter Table Changed
		16-2	16-2	Register Address Table Changed
		19-1 to 19-38	19-1 to 19-40	Most part replaced Added items Changed numerical values Changed figures
		A-3	A-6	SIO Register Addresses changed
FEUL674000-02	December 10, 2002	1-1 to 1-2	1-1 to 1-2	The list of features changed (correct some misdescription)
		1-3	1-3	Changed signal name in Figure 1.1 (correct some misdescription)
		1-4	1-4	Changed signal names in pin layout figure (correct some misdescription)
		-	1-5	Add Pin layout for BGA Package
		1-5 to 1-7	1-6 to 1-8	Changed table of pin list (correct some misdescription)
		1-8 to 1-11	1-9 to 1-12	Changed table of pin description (correct some misdescription)
		1-14	1-14	Change specification. MODE[1] pin does not provide enable/ disable control for the analog-to-digital converter.
		1-17	1-17	Change description for MODE[1] pin
		3-2	3-2	Changed Address Map (correct a misdescription)
		4-1	4-1	Change description for MODE[1]
		5-1	5-1	Add note (reference of clock gear)
		6	6	Change chapter title
		6-1	6-1	Changed minimum reset pulse width (correct a misdescription)
		7-1	7-1	Change description of power down by MODE[2:0]. Add description about clock stop to individual Function Block feature
		7-7	7-7	Add description about write protection
		7-9	7-9	Changed description (correct a misdescription)

FEUL674000-02	December 10, 2002	7-10	7-10	Add notes. Changed Description about CKWT Parameter table changed
		7-11	7-11	Table of the functional blocks with clock signal control is changed. Add notes
		7-12	7-12	Correct a misdescription in HALT/STANDBY description Add note
		8-1	8-1	Correct a erratum in features
		8-2	8-2	Correct some erratum in figure 8.1
		8-3	8-3	Correct some errata in Pin name
		8-23	8-23	Correct a misdescription in Bit Description Add Note
		8-32	8-32	Correct some errata
		10-2	10-2	Correct R/W description of RDWC
		10-6	10-6	Change parameter table
		10-16	10-16	Correct the bit neme
		-	10-21	Add IO banks control description
		-	10-22	Add description about DRAM controller disable
		10-27	10-28	Changed description
		10-30	10-31	Changed Figure 10.5/10.6
		10-40	10-41	Changed Figure 10.21
		10-44	10-45	Changed Figure 10.26
		10-45	10-46	Changed Figure title
		11-9	11-9	Changed Table
		11-16	11-16	Add description
		11-21	11-21	Add note 5. DMA restriction to external ROM area.
		11-23	11-23	Changed Figure 11.2
		12	12	Changed chapter title
		12-2	12-2	Changed Figure 12.1
		12-12	12-12	Changed tables in description about GPCTL3,4,5
		12-13	12-13	Changed Figure 12.2
		-	12-13	Add explanation about primary/secondary function configureation
		13-5	13-5	Correct some errata
		14-3 to 14-6	14-3 to 14-6	Changed the description about size of system timer registers

FEUL674000-02	December 10, 2002	14-7	14-7	Add note item
		14-14	14-14	Correct some errata
		16	16	Changed chapter title
		16-2	16-2	Correct one register name in Control register list
		16-5	16-5	Correct the description how to reset TRIRQ and RVIRQ.
		16-3 to 16-10	16-3 to 16-10	Correct the register size
		17	17	Changed chapter title
		17-27	17-27	Add the description about maximum baud rate frequency
		18-3	18-3	Correct Pin name in Pin list
		18-4	18-4	Add note
		18-6	18-6	Add note
		19-2	19-2	Change description of note 7.
		19-2	19-2 to 19-3	Change Power consumption specification and description
		19-11	19-12	Change parameter table
		19-24	19-25	Change Figure
		19-28	19-29	Changed Figure
		A-3/A-6	A-3/A-6	Changed Register List
FEUL674000-03	February 4, 2003	5-2	5-2 to 5-4	Add sample crystals and oscillation circuit information.
		8-23	8-23	Correct a misdescription about polarity in the table of Bit Description
		13-4	13-4	Correct description of OFINTMODE bit position in WDTBCON register. Correct description of WDHLT bit settings in WDTBCON register.

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Before using the product, please make sure that the information being referred to is up-to-date.

2. The outline of action and examples for application circuits described herein have been chosen as an explanation for the standard action and performance of the product. When planning to use the product, please ensure that the external conditions are reflected in the actual circuit, assembly, and program designs.

3. When designing your product, please use our product below the specified maximum ratings and within the specified operating ranges including, but not limited to, operating voltage, power dissipation, and operating temperature.

4. Oki assumes no responsibility or liability whatsoever for any failure or unusual or unexpected operation resulting from misuse, neglect, improper installation, repair, alteration or accident, improper handling, or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified operating range.

5. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of the product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.

6. The products listed in this document are intended for use in general electronics equipment for commercial applications (e.g., office automation, communication equipment, measurement equipment, consumer electronics, etc.). These products are not, unless specifically authorized by Oki, authorized for use in any system or application that requires special or enhanced quality and reliability characteristics nor in any system or application where the failure of such system or application may result in the loss or damage of property, or death or injury to humans.
   Such applications include, but are not limited to, traffic and automotive equipment, safety devices, aerospace equipment, nuclear power control, medical equipment, and life-support systems.

7. Certain products in this document may need government approval before they can be exported to particular countries. The purchaser assumes the responsibility of determining the legality of export of these products and will take appropriate and necessary steps at their own expense for these.

8. No part of the contents contained herein may be reprinted or reproduced without our prior permission.