

Arca210

Hardware Manual

Revision: 1.0

Jan. 2003



方舟科技有限公司

ARCA Technology Corporation

Arca210

Hardware Manual

Copyright © ARCA Technology Corporation. 2003.

Third party brands, logos and names are the property of those respective third parties.

Release history

Date	Revision	Change
Jan 2002	0.9	
Jan 2003	1.0	

Disclaimer

This documentation is provided for use with ARCA Technology Corporation products. No license to ARCA Technology Corporation property rights is granted. ARCA Technology Corporation assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by the ARCA Technology Corporation Terms and Conditions of Sale.

ARCA Technology Corporation products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. ARCA Technology Corporation may make changes to this document without notice. Anyone relying on this documentation should contact ARCA Technology Corporation for the current documentation and errata.

ARCA Technology Corporation

5th Floor, Jade Palace Building,
76 Zhichun Rd, Haidian,
Beijing, P. R. China
Tel: 86-10-62638192
Fax: 86-10-62638348
Http: www.arca.com.cn

Table of Contents

1	OVERVIEW	1
1.1	INTRODUCTION.....	1
1.2	BLOCK DIAGRAM.....	2
1.3	FEATURES.....	3
2	CPU CORE.....	6
3	EXTERNAL MEMORY INTERFACE.....	7
3.1	OVERVIEW.....	7
3.1.1	Features.....	7
3.1.2	Block Diagram.....	8
3.2	PIN CONFIGURATION.....	9
3.3	MEMORY MAP.....	11
3.4	REGISTER CONFIGURATION.....	13
3.4.1	Bus Control Register (BCR).....	14
3.4.2	Static Memory Control Register (SMCRn, n = 0 to 3).....	15
3.4.3	DRAM Control Register (DMCR).....	20
3.4.4	SDRAM Mode Register (SDMR).....	25
3.4.5	Refresh Timer Control/Status Register (RTCSR).....	27
3.4.6	Refresh Timer Counter (RTCNT).....	28
3.4.7	Refresh Time Constant Register (RTCOR).....	28
3.4.8	DRAM Bank Address Configuration Register (DMAR1, 2, 3, 4).....	29
3.5	ENDIAN/ACCESS SIZE AND DATA ALIGNMENT.....	30
3.6	STATIC MEMORY INTERFACE.....	34
3.6.1	Normal Memory Interface.....	34
3.6.2	Byte Control SRAM Interface.....	42
3.6.3	Burst ROM Interface.....	45
3.7	SDRAM INTERFACE.....	46
3.7.1	Example of Connection.....	48
3.7.2	Address Multiplexing.....	53
3.7.3	Non-Page Mode Timing.....	54
3.7.4	Page Mode Timing.....	61
3.7.5	Power-Down Mode.....	74
3.7.6	Refreshing.....	75
3.7.7	Power-On Sequence.....	79
4	DIRECT MEMORY ACCESS.....	82
4.1	OVERVIEW.....	82
4.1.1	Features.....	82
4.1.2	Block Diagram.....	83
4.2	PIN CONFIGURATION.....	84
4.3	REGISTERS CONFIGURATION.....	85
4.3.1	DMA Source Address Register (DSARn, n = 0 to 3).....	86
4.3.2	DMA Destination Address Register (DDARn, n = 0 to 3).....	86
4.3.3	DMA Transfer Counter Register (DCTRn, n = 0 to 3).....	86
4.3.4	DMA Channel Control/Status Register (DCCSRn, n = 0 to 3).....	86
4.3.5	DMA Control Register (DMACR).....	92
4.4	OPERATION.....	94
4.4.1	DMA Transfer Procedure.....	94
4.4.2	DMA Transfer Requests.....	96
4.4.3	Two-Address DMA Transfer.....	97
4.4.4	DMA Transfer Modes.....	99
4.4.5	Channel Priorities.....	101

4.4.6 DMA Transfer Bus Cycle States for Device with DREQ.....	103
4.4.7 Normal DMA Transfer Ending.....	107
4.4.8 Forced Suspension.....	108
5 PCI CONTROLLER.....	110
5.1 OVERVIEW.....	110
5.1.1 Feature.....	110
5.1.2 Signals.....	111
5.2 OPERATION MODE.....	112
5.2.1 Configuration.....	112
5.3 PCI BUS MASTER OPERATION.....	113
5.4 ARCA PCI BUS TARGET OPERATION	114
5.4.1 Base Address Register 0.....	114
5.4.2 Base Address Register 1	115
5.5 PCI CONFIGURATION REGISTER ACCESS	116
5.6 PCI COMMANDS SUPPORT	117
5.7 PCIC REGISTERS.....	118
5.7.1 PCIC Control Registers.....	118
5.7.2 PCI Configuration Registers.....	134
5.8 PCI BUS ARBITER.....	143
5.8.1 Features and Block Diagram.....	143
5.8.2 Arbitration Algorithm.....	144
5.8.3 Request/ Grant Timing.....	146
5.8.4 Examples of Arbiter Operation.....	147
6 ETHERNET CONTROLLER.....	149
6.1 OVERVIEW.....	149
6.1.1 Features.....	149
6.1.2 Block Diagram.....	150
6.2 PIN CONFIGURATION	151
6.3 REGISTER CONFIGURATION.....	154
6.3.1 Bus Mode Register (CSR0).....	155
6.3.2 Transmit Poll Demand Register (CSR1).....	156
6.3.3 Receive Poll Demand Register (CSR2).....	156
6.3.4 Receive/Transmit Descriptor List Base Address Registers (CSR3, CSR4).....	156
6.3.5 DMA Status Register (CSR5).....	157
6.3.6 Control (Operation Mode) Register (CSR6).....	162
6.3.7 Interrupt Enable Register (CSR7).....	164
6.3.8 Missed Frame and Buffer Overflow Counter Register.....	168
6.3.9 Current Host Transmit Buffer Address Register (CSR20)	169
6.3.10 Current Host Receive Buffer Address Register (CSR25)	169
6.3.11 MAC Control Register.....	170
6.3.12 MAC Address Hi Register	173
6.3.13 MAC Address Lo Register	173
6.3.14 Multicast Address Hi/Lo Register.....	173
6.3.15 MII Address Register.....	174
6.3.16 MII Data Register.....	175
6.3.17 Flow Control Register.....	175
6.3.18 VLAN1 Tag Register.....	176
6.3.19 VLAN2 Tag Register.....	177
6.4 DATA DESCRIPTORS AND BUFFERS.....	178
6.4.1 Receive Descriptors.....	179
6.4.2 Transmit Descriptors.....	183
6.5 OPERATION.....	187
6.5.1 Data Flow.....	187
6.5.2 Transmit Operation.....	188
6.5.3 Receive Operation.....	190

6.5.4 Host Bus Burst Access	192
6.5.5 Host Buffer Data Alignment.....	192
6.5.6 Buffer Size Calculations.....	192
6.5.7 DMA Suspended State Behavior.....	193
6.5.8 Interrupts.....	193
6.6 MAC OPERATION	195
6.6.1 Transmit Operation.....	196
6.6.2 Receive Operation.....	197
6.6.3 Address Check	197
6.6.4 Flow Control.....	199
6.6.5 Virtual Local Area Network (VLAN).....	200
6.6.6 MII Management	202
7 USB HOST CONTROLLER	203
7.1 OVERVIEW.....	203
7.1.1 Block Diagram.....	204
7.2 PIN CONFIGURATION.....	205
7.3 REGISTER CONFIGURATION.....	206
8 CLOCK GENERATION UNIT (CGU)	208
8.1 OVERVIEW.....	208
8.1.1 CGU Functional Feature.....	208
8.1.2 CGU Block Diagram.....	209
8.1.3 CGU Pin Configuration.....	210
8.2 CLOCK MODES.....	211
8.3 CGU REGISTERS	214
8.3.1 Clock Frequency Control Register.....	214
8.4 CGU OPERATION.....	217
8.4.1 PLL	217
8.4.2 Change Clock Frequency.....	218
9 POWER/RESET MANAGEMENT CONTROLLER (PMC).....	221
9.1 OVERVIEW.....	221
9.1.1 Low-Power Modes and Function.....	221
9.2 REGISTER DESCRIPTION.....	223
9.2.1 Low-Power Control Register.....	223
9.2.2 Reset Status Register (RSTR).....	226
9.3 DOZE MODE.....	228
9.3.1 Entering Doze Mode.....	228
9.3.2 Exiting Doze Mode.....	228
9.4 SLEEP MODE.....	229
9.4.1 Entering Sleep Mode.....	229
9.4.2 Exiting Sleep Mode	229
9.5 STANDBY MODE.....	231
9.5.1 Entering Standby Mode.....	231
9.5.2 Exiting Standby Mode	231
9.6 MODULE STOP FUNCTION.....	234
9.6.1 Entering Module Stop Function.....	234
9.6.2 Exiting Module Stop Function	234
9.7 RESET CONTROL.....	235
9.7.1 Reset Sources and Pins.....	235
9.7.2 Reset Sequence.....	236
10 INTERRUPT CONTROLLER (INTC).....	239
10.1 OVERVIEW.....	239
10.1.1 Features.....	239
10.1.2 Block Diagram.....	240

10.2	PIN CONFIGURATION.....	241
10.3	REGISTERS CONFIGURATION.....	242
10.3.1	Interrupt Source Register (ISR).....	242
10.3.2	Interrupt Mask Register (IMR).....	242
10.3.3	Interrupt Mask Set Register (IMSR).....	242
10.3.4	Interrupt Mask Clear Register (IMCR).....	243
10.3.5	Interrupt Pending Register (IPR).....	243
10.3.6	Bits Definition for IMR, IMSR, IMCR, ISR, IPR	243
10.4	INTC OPERATION.....	245
11	TIMER UNIT (TMU)	248
11.1	OVERVIEW.....	248
11.1.1	Features.....	248
11.1.2	Block Diagram.....	249
11.2	REGISTER CONFIGURATION.....	250
11.2.1	Timer Enable Register (TER).....	250
11.2.2	Timer Control/Status Register (TCSR).....	251
11.2.3	Timer Reload Data Register (TRDR)	252
11.2.4	Timer Counter (TCNT).....	252
11.3	TMU OPERATION.....	253
11.3.1	Basic Functions.....	253
11.3.2	TMU counter decrease and reload timing.....	254
11.4	USAGE NOTES.....	255
11.4.1	Interrupt handling.....	255
11.4.2	Standby mode.....	255
11.4.3	Interrupt enabling during counting.....	255
12	WATCHDOG TIMER (WDT)	256
12.1	OVERVIEW.....	256
12.1.1	Features.....	256
12.1.2	Block Diagram.....	256
12.2	REGISTER CONFIGURATION.....	257
12.2.1	Watchdog Timer Control/Status Register (WTCSR).....	257
12.2.2	Watchdog Timer Counter (WTCNT).....	258
12.3	USAGE NOTES.....	259
12.3.1	Watchdog Timer Function.....	259
13	GENERAL PURPOSE INPUT/OUTPUT PORT (GPIO)	260
13.1	OVERVIEW.....	260
13.1.1	Features.....	260
13.2	PORT A.....	261
13.2.1	Features.....	261
13.2.2	Block Diagram.....	261
13.2.3	Registers Configuration.....	262
13.2.4	Port A Data Register (PADR).....	262
13.2.5	Port A Control Register (PACR).....	262
13.3	PORT B.....	264
13.3.1	Features.....	264
13.3.2	Block Diagram.....	264
13.3.3	Registers Configuration.....	265
13.3.4	Port B Data Register (PBDR).....	265
13.3.5	Port B Control Register (PBCR).....	265
13.3.6	Port B Edge Detect Register (PBETR).....	266
13.3.7	Port B Edge Flag Register (PBFR).....	267
13.4	PORT C.....	268
13.4.1	Features.....	268
13.4.2	Block Diagram.....	268

13.4.3 Registers Configuration.....	269
13.4.4 Port C Data Register (PCDR).....	269
13.4.5 Port C Control Register (PCCR).....	269
13.5 PORT D.....	271
13.5.1 Features.....	271
13.5.2 Block Diagram.....	271
13.5.3 Registers Configuration.....	272
13.5.4 Port D Data Register (PDDR).....	272
13.5.5 Port D Control Register (PDCR).....	272
13.6 PORT E.....	274
13.6.1 Features.....	274
13.6.2 Block Diagram.....	274
13.6.3 Registers Configuration.....	275
13.6.4 Port E Data Register (PEDR).....	275
13.6.5 Port E Control Register (PECR).....	275
13.7 PORT F.....	277
13.7.1 Features.....	277
13.7.2 Block Diagram.....	277
13.7.3 Registers Configuration.....	278
13.7.4 Port F Data Register (PFDR).....	278
13.7.5 Port F Control Register (PFCR).....	278
13.8 USAGE NOTES.....	280
14 UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER.....	281
14.1 OVERVIEW.....	281
14.1.1 Features.....	281
14.1.2 Block Diagram.....	282
14.2 PIN CONFIGURATION.....	283
14.2.1 RxD Pin.....	283
14.2.2 TxD Pin.....	283
14.2.3 CTS Pin — Clear to Send.....	283
14.2.4 RTS Pin — Request to Send.....	283
14.3 REGISTERS CONFIGURATION.....	284
14.3.1 Receive FIFO Data Register (UARTRDR).....	284
14.3.2 Transmit FIFO Data Register (UARTTDR).....	284
14.3.3 UART Control Register (UARTCR).....	285
14.3.4 UART Status Register (UARTSR).....	290
14.3.5 Bit Rate Divisor Register (UARTBRDR).....	294
14.3.6 Receive FIFO Data Count Register (UARTRFDR).....	295
14.3.7 Transmit FIFO Data Count Register (UARTTFDR).....	295
14.4 RECEIVER/ TRANSMITTER.....	296
14.4.1 Receiver.....	296
14.4.2 Transmitter.....	296
14.5 UART OPERATION.....	297
14.5.1 Overview.....	297
14.5.2 Serial Operation.....	297
14.6 UART INTERRUPT SOURCES AND THE DMAC.....	306
14.7 UART USAGE NOTES.....	307
15 SMART CARD CONTROLLER (SCC).....	308
15.1 OVERVIEW.....	308
15.1.1 Features.....	308
15.1.2 Block Diagram.....	309
15.2 PIN CONFIGURATION.....	310
15.2.1 DATA Pin.....	310
15.2.2 CLK Pin.....	310

15.3	REGISTERS CONFIGURATION.....	311
15.3.1	UART Register (<i>UARTBRDR, UARTCR, UARTCRL, UATCRH, UARTSR, UARTRDR, UARTTDR, UARTRFDR, UARITTFDR</i>).....	311
15.3.2	SCC Control Register (<i>SCCCR</i>).....	311
15.3.3	SCC Status Register (<i>SCCSR</i>).....	313
15.3.4	SCC Transmission Factor Register (<i>SCCTFR</i>).....	314
15.3.5	SCC Extra Guard Time Register (<i>SCCEGTR</i>).....	315
15.3.6	SCC ETU Counter Value Register (<i>SCCECR</i>).....	315
15.3.7	UART Status Register (<i>UARTSR</i>).....	316
15.4	RECEIVER AND TRANSMITTER.....	316
15.4.1	Receiver.....	316
15.4.2	Transmitter.....	316
15.5	PINS CONNECTION.....	317
15.6	DATA FORMAT.....	318
15.7	REGISTER RELATED TO SCC SETTING.....	319
15.8	SCC OPERATION.....	320
15.8.1	Initialization.....	320
15.8.2	Serial Data Transmission.....	322
15.8.3	Serial Data Reception.....	323
15.9	USAGE NOTES.....	324
15.9.1	Interrupt Operation.....	324
15.9.2	Selection of the Operation Class.....	324
15.9.3	Reset operation under asynchronous mode.....	326
16	INFRARED DATA ASSOCIATION (IRDA).....	330
16.1	OVERVIEW.....	330
16.1.1	Features.....	330
16.1.2	Block Diagram.....	331
16.2	PIN CONFIGURATION.....	332
16.2.1	RxD2 Pin.....	332
16.2.2	TxD2 Pin.....	332
16.3	REGISTERS CONFIGURATION.....	333
16.3.1	UART2 Register (<i>UARTBRDR2, UARTCR2, UARTCRL2, UATCRH2, UARTSR2, UARTRDR2, UARITTD2, UARTRFDR2, UARITTFDR2</i>).....	333
16.3.2	Control Register (<i>IRDACR</i>).....	333
16.4	IRDA OPERATION.....	336
16.4.1	Overview.....	336
16.4.2	Transmit.....	336
16.4.3	Receive.....	337
16.4.4	Loopback Mode.....	338
16.5	USAGE NOTES.....	339
17	I²C-BUS INTERFACE (I2C).....	340
17.1	OVERVIEW.....	340
17.1.1	Features.....	340
17.1.2	Block Diagram.....	341
17.2	PIN CONFIGURATION.....	342
17.2.1	SDA Pin.....	342
17.2.2	SCL Pin.....	342
17.3	REGISTERS CONFIGURATION.....	343
17.3.1	Data Register (<i>I2CDR</i>).....	343
17.3.2	Control Register (<i>I2CCR</i>).....	343
17.3.3	Status Register (<i>I2CSR</i>).....	345
17.3.4	Clock Generator Register (<i>I2CGR</i>).....	347
17.4	I ² C-BUS PROTOCOL.....	348
17.4.1	Bit Transfer.....	348
17.4.2	Data Validity.....	348

17.4.3	START and STOP Conditions.....	348
17.4.4	Byte Format.....	348
17.4.5	Data Transfer Format.....	350
17.5	I2CI OPERATION.....	353
17.5.1	I2CI Initialize.....	353
17.5.2	Write Operation.....	353
17.5.3	Read Operation.....	355
17.6	APPENDIX: TERMS REFERRED FROM I2C STANDARD SPECIFICATION.....	356
18	AC97 CONTROLLER	357
18.1	OVERVIEW.....	357
18.1.1	Feature.....	357
18.1.2	Block Diagram.....	357
18.1.3	Pin configuration	358
18.2	REGISTER CONFIGURATION.....	360
18.2.1	Register Descriptions.....	361
18.3	OPERATION.....	376
18.3.1	Initialization and Reset.....	376
18.3.2	Data Transfer	378
18.3.3	DMA Audio Data Transfers	381
18.3.4	External Codec Registers Access Operation.....	381
18.3.5	Power Down Mode.....	384
18.3.6	AC'97 Reset.....	384
18.3.7	ATE Test Mode	386
19	BOOT TIME CONFIGURATION	388
19.1	BANK 0 MEMORY SIZE – MD_MEM.....	388
19.2	CLOCK MODE – MD_CLK.....	388
19.3	PLL FEEDBACK DIVISION RATIO BIT 5-1 – MD_PLL_F.....	388
19.4	PLL OUTPUT DIVISION RATION BIT 0 – MD_PLL_OD.....	389
19.5	PCI HOST MODE – MD_PCI.....	389
19.6	PCI HOST ARBITER – MD_PCIARB.....	389
19.7	TAP SELECT – MD_TAP_SEL.....	389
19.8	OCS-BUS DATA ACCESS TIMEOUT.....	390
19.8.1	Timeout Register.....	390
19.8.2	Address Record register.....	391
20	PIN DESCRIPTION	392
21	SYSTEM MEMORY MAP.....	399
21.1	INTERNAL REGISTER MAP ADDRESS AND DEFINITION	399
21.2	ADDRESS SPACE ALLOCATION.....	399
21.3	REGISTER ADDRESS	400
21.3.1	MMU.....	400
21.3.2	Cache.....	400
21.3.3	OCS-Bus arbiter.....	400
21.3.4	EMI.....	401
21.3.5	PCI.....	402
21.3.6	UART.....	403
21.3.7	INTC.....	403
21.3.8	RTC.....	403
21.3.9	TMU	404
21.3.10	WDT.....	404
21.3.11	CGU	404
21.3.12	PMC.....	404
21.3.13	I2CI.....	405
21.3.14	GPIO	405

Table of Contents

LIST OF FIGURES	406
LIST OF TABLES	409

1 Overview

1.1 Introduction

Arca210 is a member of Arca2 family 32-bit RISC microprocessors. Building around a high performance and low power Arca2 CPU core, Arca210 integrates a varieties of peripherals needed by embedded product such as thin client, Set Top Box and Internext Appliances.

As a highly integrated SOC, the processor core of Arca210 has function units of separate instruction and data cache memory and memory manage unit (MMU). Further, Arca210 integrates external memory interface controller (EMI) supporting direct connection to a variety of memories; ethernet controller supporting 100Mbps ethernet; USB controller supporting USB 1.1; and PCI bus controller, direct memory access controller (DMA) as the high speed data transfer device. The Arca210 microprocessor also integrates an interrupt controller, a timer, a realtime clock, a serail communication interface, and clock generator using phase-locked loop on to a single device.

Advanced system architecture has been introduced into the Arca210, which consists of a high-performance system backbone bus – OCS bus, able to sustain the external memory bandwidth, on which the high-speed CPU, and other direct memory access devices (DMA, PCI-controller) reside. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. Also located on the high-performance bus is a bridge to the lower bandwidth and low-power peripheral bus – OCP bus, where most of the peripheral devices in the system are located.

A powerful built-in power management function keeps power consumption low, even during high-speed operation. This processor core can run at 16 times the frequency of the system bus speed.

With the well designed architecture and implemented with state-of-the-art CMOS technology, the Arca210 provides an ideal low-cost, high-performance, and low-power-consumption solution.

1.2 Block Diagram

The system architecture of Arca210 is shown in the following diagram:

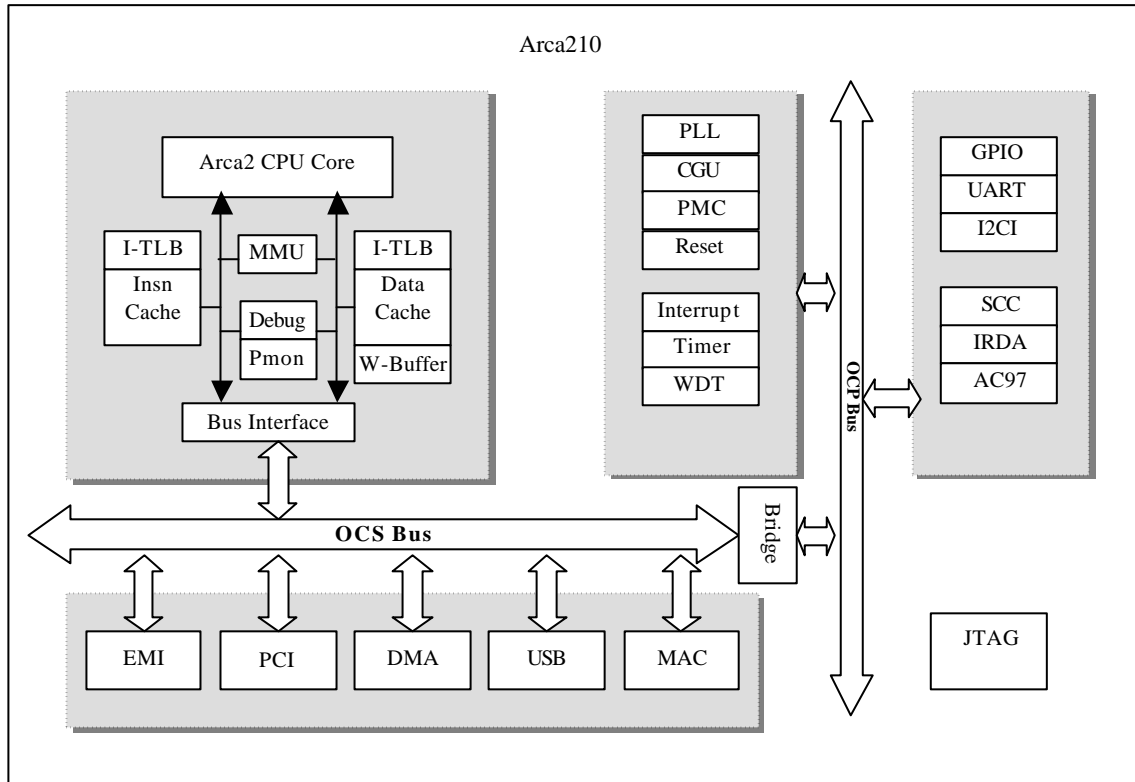


Figure 1-1 Block Diagram

Arca210 consists of the following three functional groups:

■ Arca2 CPU Core

Arca2 CPU core contains CPU, memory management unit, caches and a bus interface unit. The processor core adopts a Harvard memory architecture which contains 16Kbyte instruction cache and 16Kbyte data cache. For data cache, 16 words write back buffer is provided to reduce the performance lost due to bus conflict. Instruction address and data address are translated through separate 32 entry Instruction TLB and 32 entry data TLB. A debug module with JTAG interface is embedded in the core to assist software, especially kernel code debugging. PMON module is used to monitor CPU core's performance such as, clock number, instruction executed number, cache miss number and etc. Please reference to *Arca2 CPU Core Reference Manual* for details description of the CPU core.

■ OCS Bus Devices

Devices requiring high performance memory bandwidth or to be in close proximity to the CPU are connect to OCS bus. Arca210 OCS bus devices include memory controller, PCI bus interface controller, DMA controller, USB controller and Ethernet controller. The high speed of the OCS bus guarantees the high bandwidth requirements between CPU and external high speed devices.

■ OCP Bus Devices

Devices that don't need a high bandwidth connection are connected in OCP (On Chip Peripheral) bus. These include clock generation unit, power management unit, interrupt controller, timer, watchdog timer, general purpose IO, UART, smart card controller, IrDA, I²C interface, AC97 controller.

1.3 Features

The Arca210 is a highly integrated, low-cost implementation of the Arca RISC architecture, that includes many function units, can be used in a single board design. The peripheral and device features of Arca210 are listed in Table 1-1.

Table 1-1 Arca210 Features

Item	Features
Arca2 CPU core	<ul style="list-style-type: none"> • Arca2 CPU integer core • 5 stage pipeline implementation • Hardward memory architecture with MMU and TLB support • 16K bytes instruction cache • 16K bytes data cache • Hardware debug capability through JTAG port
External Memory Interface (EMI)	<ul style="list-style-type: none"> • Static memory controller <ul style="list-style-type: none"> – Direct interface to ROM, Burst ROM, SRAM, Flash and memory like devices – 8, 16 or 32-bit bus width – 4 banks up to 256M (64M X 4) – Programmable wait and external wait signal • Synchronous DRAM controller <ul style="list-style-type: none"> – Both DIMM and SODIMM are supported – 32-bit bus width – 1 bank up to 2G, total 4 banks up to 2.75G – Supports burst operation – Page mode supported – Has both auto-refresh and self-refresh functions
PCI Controller	<ul style="list-style-type: none"> • PCI revision 2.1 • 32-bit, 33MHz • Supports (in host mode) up to 4 external bus masters or slaves • Supports (in host mode) bus arbitration between host and 4 external masters • Support host and satellite mode • Support transaction between Arca210 CPU and PCI slaves • Support direct transaction between external master and system device • Four 8-deep FIFO for each transaction
Direct Memory Access controller (DMA)	<ul style="list-style-type: none"> • Four independent DMA channels with fix or round robin priority • Up to 16M transfer count • Transfer data unit: byte, half-word, word, 4-word or 8-word • Support external DMA request
Ethernet Controller (ETHC)	<ul style="list-style-type: none"> • Compliant with IEEE802.3, 802.3u Specification • 10/100 Mbps data transfer rates • IEEE802.3 compliant MII interface to talk to an external PHY • VLAN support • Full and half duplex modes

	<ul style="list-style-type: none"> • Support CSMA/CD protocol in half duplex mode • Supports flow-control for full-duplex operation • Provides External and internal loop back capability on the MII Interface
USB Host Controller (UHC)	<ul style="list-style-type: none"> • Open Host Controller Interface (OHCI)-compatible and USB Revision 1.1-compatible • Support low-speed (1.5 Mbps) and full-speed (12 Mbps) USB devices • Two downstream ports are provided
Clock Generation Unit (CGU)	<ul style="list-style-type: none"> • On-chip phase-locked loop (PLL) with programmable multiple-ratio • On-chip oscillator circuit • 5 clock modes can be selected by MD0 ~ MD2 • The PLL on/off is programmable by software • Input clock source can be selected from oscillator or an external clock input • Generates three clocks: internal clock (iclk) for CPU core, peripheral clock (pclk) for peripheral devices, system clock (sclk) for system bus devices • iclk, pclk and sclk frequency can be changed separately for software by setting division ratio
Interrupt controller (INT)	<ul style="list-style-type: none"> • Eight external interrupt pins (IRQ0 – IRQ7) • On-chip peripheral interrupts
Universal Asynchronous Receiver/Transmitter (UART)	<ul style="list-style-type: none"> • Full-duplex communication • 16-byte FIFO for transmission/reception • DMA can be transferred • Modem control functions (RTS and CTS) are provided
Watchdog Timer (WDT)	<ul style="list-style-type: none"> • Generates power-on reset or manual reset • 16-bit counter
Timer Unit (TMU)	<ul style="list-style-type: none"> • 3-channel auto-reload-type 32-bit timer • 5 types of counter input clocks can be selected • Auto-reload function is provided for each channel
Power/reset Management Controller (PMC)	<ul style="list-style-type: none"> • Supports three low-power modes and function: sleep mode, standby mode, module stop function • reset sequence control
I ² C-Bus Interface (I2CI)	<ul style="list-style-type: none"> • Supports only single master mode • Support of I²C standard-mode and F/S-mode up to 400 kHz • Independent, programmable serial clock generator • The number of devices that you can connect to the same I²C-bus is limited only by the maximum bus capacitance of 400pF
Smart Card Controller (SCC)	<ul style="list-style-type: none"> • Conforms to the ISO/IEC standard 7816-3 • Support asynchronous character (T=0) and block (T=1) communication modes • Receiver and Transmit mode error signal detection and automatic re-transmission of data (T = 0) • Supports both direct convention and inverse convention • A straightforward extension of UART: When SCC is disabled, UART can work as a normal UART
Infrared Serial Interface (IrDA)	<ul style="list-style-type: none"> • Based on the IrDA 1.0 specification • Polarity of transmitted and received signals selectable • When transmitting, support normal 3/16 and IrDA low-power mode bit duration • When receiving, normal 3/16 and low-power mode bit duration signal will both be received as low-power mode bit duration • A straightforward extension of UART2: When IrDA is disabled, UART2 can work as a normal UART

AC97 Controller (AC97)	<ul style="list-style-type: none">• Compliant with AC'97 Component Specification 2.2• Support 16, 18, 20bit sample size• Programmable Output channels and Input channels Support• Power Down Mode• Two Wake-Up mode Support• Support DMA transfer mode
General Purpose Input/Output Port (GPIO)	<ul style="list-style-type: none">• 8 bits dedicated ports• 39 shared ports

Table 1-2 Characteristics

Item	Characteristics
Power supply voltage	I/O: $3.3 \pm 0.3V$; Internal: $1.8 \pm 0.2V$
Operating frequency	300MHz – 400MHz
Process	0.18- μm CMOS
Package	304PBGA

2 CPU Core

Arca210 integartes a high performance Arca2 cpu core. Refer document “*Arca2 CPU Core reference manual*” for detail description of this processor core.

3 External Memory Interface

3.1 Overview

EMI (External Memory Interface) module divides physical address space and outputs control signals for various types of memory and bus interface. EMI functions enable it to link directly to Synchronous DRAM, SRAM, ROM, Burst ROM and Flash without an external circuit, simplifying system design and allowing high-speed data transfers in a compact system.

3.1.1 Features

The EMI has the following features:

- Static memory interface
 - Direct interface to ROM, Burst ROM, SRAM and Flash
 - Four-bank static memory is supported from bank 0 to 3. Each bank can be configured separately
 - Bank 0 memory bus width is set by external pin. Bank 1 to 3 memory bus width may be 8, 16 or 32 bits
 - Output of control signals allowing direct connection of memory to each bank Write strobe setup time and hold time periods can be inserted in an access cycle to enable connection to low-speed memory
 - Wait state insertion can be controlled by program
 - Wait insertion by WAIT pin
 - Automatic wait cycle insertion to prevent data bus collisions in case of consecutive memory accesses to different banks, or a read access followed by a write access to the same bank
- Synchronous DRAM Interface
 - Four banks with changeable size are supported
 - Both DIMM and SODIMM are supported
 - Only 32-bit bus width is supported
 - Multiplexes row/column addresses according to SDRAM capacity
 - One or two chip-select signals for each bank
 - Two-bank or four-bank SDRAM is supported
 - Supports burst operation
 - Has both auto-refresh and self-refresh functions
 - Controls timing of SDRAM direct-connection control signals according to register setting
 - Supports power-down mode to minimize the power consumption of SDRAM
 - Supports page mode

3.1.2 Block Diagram

The following figure shows the functional block diagram of the EMI.

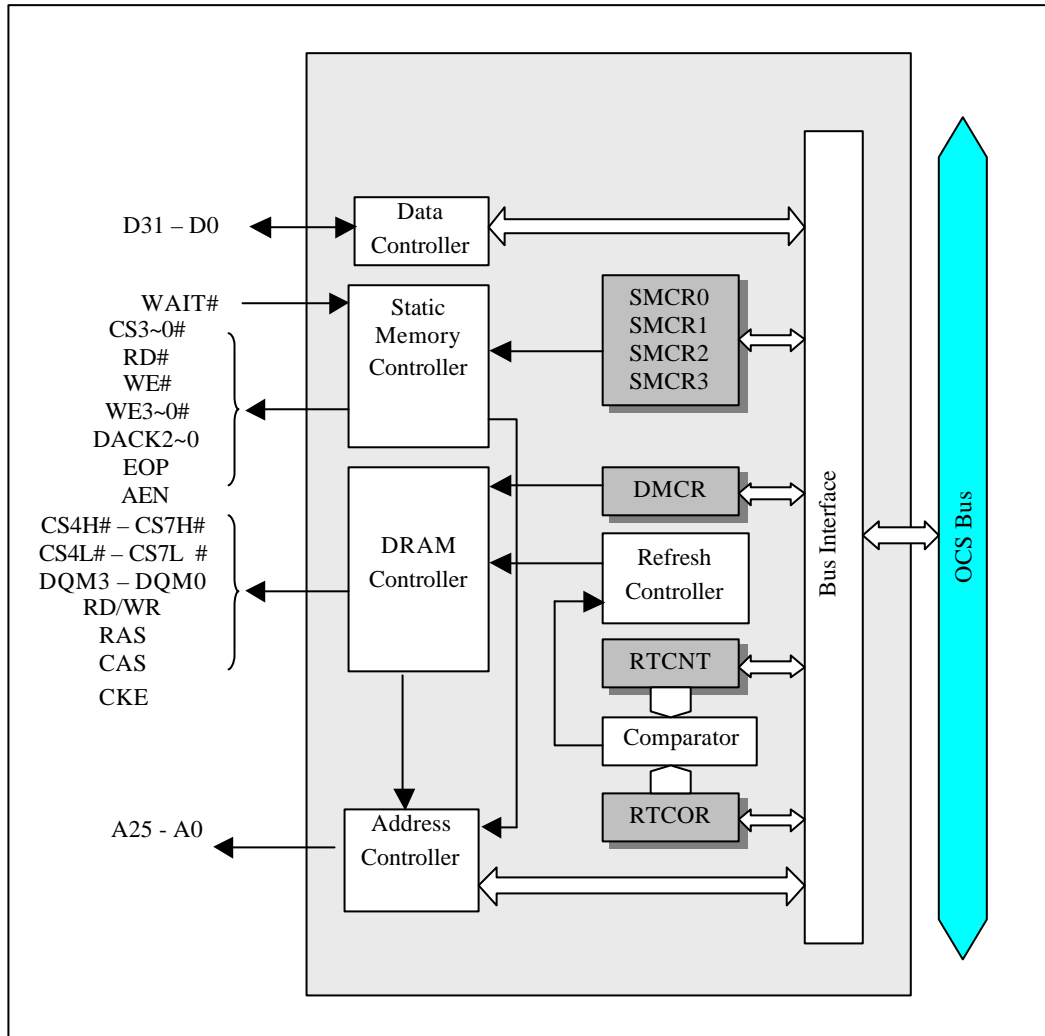


Figure 3-1 EMI Block Diagram

3.2 Pin Configuration

The following table lists the EMI pins of Arca210.

Table 3-1 EMI Pin Configuration

Pin Name	I/O	Signal	Description
Data bus	I/O	D31 – D0	Data I/O
Address bus	O	A25–A0	Address output
Chip select 3 - 0	O	CS3# - CS0#	Chip select signal that indicates the bank being accessed
chip select 4L	O	CS4L#	When SDRAM is used, chip select signal to indicate that bank 4 is being accessed
Chip select 4H/data enable 4	O	CS4H#/DQM4#	When SDRAM SODIMM is used, selection signal for D32-D39. When SDRAM DIMM is used, chip select signal to indicate that upper half of bank 4 is being accessed
chip select 5L	O	CS5L#	When SDRAM is used, chip select signal to indicate that bank 5 is being accessed
Chip select 5H/data enable 5	O	CS5H#/DQM5#	When SDRAM SODIMM is used, selection signal for D40-D47. When SDRAM DIMM is used, chip select signal to indicate that upper half of bank 5 is being accessed
chip select 6L	O	CS6L#	When SDRAM is used, chip select signal to indicate that bank 6 is being accessed
Chip select 6H/data enable 6	O	CS6H#/DQM6#	When SDRAM SODIMM is used, selection signal for D48-D55. When SDRAM DIMM is used, chip select signal to indicate that upper half of bank 6 is being accessed
Chip select 7L	O	CS7L#	When SDRAM is used, chip select signal to indicate that bank 7 is being accessed
Chip select 7H/data enable 7	O	CS7H#/DQM7#	When SDRAM SODIMM is used, selection signal for D56-D63. When SDRAM DIMM is used, chip select signal to indicate that upper half of bank 7 is being accessed
Read	O	RD#/OE#	When using static memory: RD# signal
Write	O	WE#	When using static mameory: WE# signal, indicates a write cycle
Column address strobe	O	CAS#	SDRAM CAS# signal
Row address strobe	O	RAS#	SDRAM RAS# signal
Read/write	O	RD/WR	Data bus direction designation signal Also used as SDRAM write designation signal
Data enable 0	O	WE0#/DQM0	When static memory is used, selects D7-0 write strobe signal. When SDRAM is used, selection signal for D7–D0
Data enable 1	O	WE1#/DQM1	When static memory is used, selects D15-8 write strobe signal When SDRAM is used, selection signal for D15–D8

Pin Name	I/O	Signal	Description
Data enable 2	O	WE2#/DQM2	When static memory is used, selects D23-16 write strobe signal When SDRAM is used, selection signal for D23–D16
Data enable 3	O	WE3#/DQM3	When static memory is used, selects D31-24 write strobe signal When SDRAM is used, selection signal for D31–D24.
Clock enable	O	CKE	Enable the SDRAM clock
Wait	I	Wait#	External Wait state request signal for static and PC Card
DMAC0 acknowledge signal	O	DACK0	DMAC channel 0 data acknowledge signal
DMAC1 acknowledge signal	O	DACK1	DMAC channel 1 data acknowledge signal
DMAC2 acknowledge signal	O	DACK2	DMAC channel 2 data acknowledge signal
End of process	O	EOP	Indicates the end of DMA transfer
Address enable	O	AEN	Active high to indicate that the system is in DMA transfer mode

3.3 Memory Map

Space Allocation: In the Arca210 architecture, both logical space and physical space have 32-bit address space. The 4Gbyte physical space is divided into several partitions for static memory, SDRAM, PCI and internal I/O. The following table shows the physical space map.

Table 3-2 Physical Address Space Map

Physical Address	Connectable Memory	Capacity	Chip Select Signals	Access Size
H'0000 0000 to H'03FF FFFF	Static memory bank 0	64 Mbytes	CS0#	8, 16, 32*
H'0400 0000 to H'07FF FFFF	Static memory bank 1	64 Mbytes	CS1#	8, 16, 32
H'0800 0000 to H'0BFF FFFF	Static memory bank 2	64 Mbytes	CS2#	8, 16, 32
H'0C00 0000 to H'0FFF FFFF	Static memory bank 3	64 Mbytes	CS3#	8, 16, 32
H'1000 0000 to H'BFFF FFFF	SDRAM bank 4 ~ 7 with changeable base address and changeable size	2816 Mbytes	CS4H#, CS4L#, CS5H#, CS5L#, CS6H#, CS6L#, CS7H#, CS7L#	32
H'C000 0000 to H'DFFF FFFF	PCI space	512 Mbytes		
H'E000 0000 to H'FFFF FFFF	Internal I/O	512 Mbytes		

Notes:

1. Use external pin to specify static memory bank 0 bus width.
 - The memory data bus width can be set for bank 0 at power-on reset. The correspondence between the external pin (MD3, MD4) and memory size is listed in the table below.

MD4	MD3	Bank 0 Memory Size
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	Reserved

- Bank 1 to 3 data bus width is programmable. Bank 4 to 7 data bus width is fixed on 32 bits.
2. Do not access the reserved area. If the reserved area is accessed, the correct operation can not be guaranteed.

Following figure shows the physical space map.

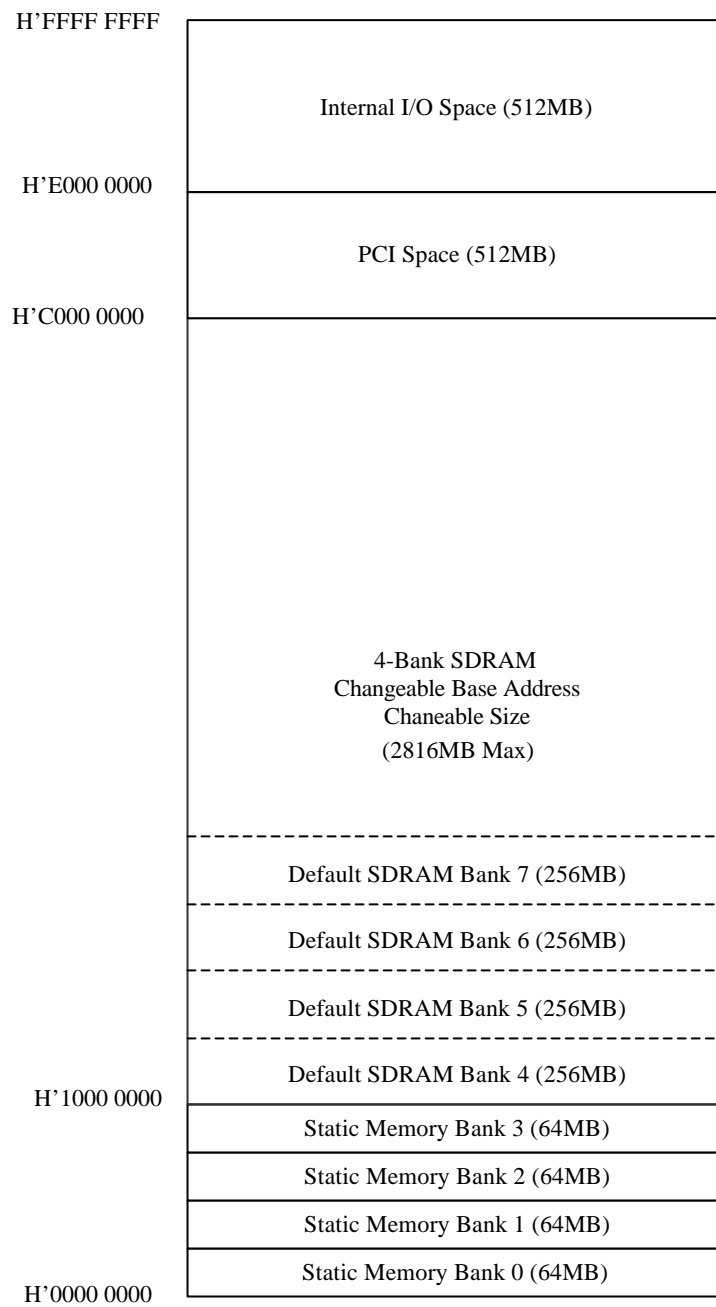


Figure 3-2 Physical Address Space Map

3.4 Register Configuration

The EMI has seventeen registers, including mode registers of SDRAM. These registers control the operation of memory. They are used to configure the memory type, wait states insertion and refreshing.

Table 3-3 EMI Register Configuration

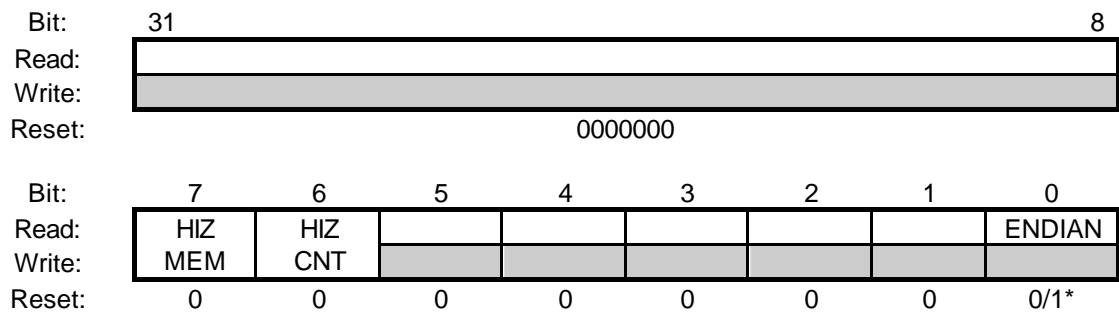
Name	Full Name	R/W	Initial Value	Address	Access Width
BCR	Bus control register	R/W	H'0000 000x	H'E1020000	32
SMCR0	Static memory control register 0	R/W	H'0FFF 7700	H'E1020004	32
SMCR1	Static memory control register 1	R/W	H'0FFF 7700	H'E1020008	32
SMCR2	Static memory control register 2	R/W	H'0FFF 7700	H'E102000C	32
SMCR3	Static memory control register 3	R/W	H'0FFF 7700	H'E1020010	32
DMCR	DRAM control register	R/W	H'0000 0000	H'E1020014	32
RTCSR	Refresh time control/status register	R/W	H'0000	H'E1020018	16
RTCNT	Refresh timer counter	R/W	H'0000	H'E102001C	16
RTCOR	Refresh time constant register	R/W	H'0000	H'E1020020	16
DMAR1	DRAM bank 4 address configuration register	R/W	H'0000 10F0	H'E1020040	32
DMAR2	DRAM bank 5 address configuration register	R/W	H'0000 20F0	H'E1020044	32
DMAR3	DRAM bank 6 address configuration register	R/W	H'0000 30F0	H'E1020048	32
DMAR4	DRAM bank 7 address configuration register	R/W	H'0000 40F0	H'E102004C	32
SDMR	SDRAM mode register, bank 4	W	—	H'FEFFA000-H'FEFFAFFF	8/16/32
	SDRAM mode register, bank 5		—	H'FEFFB000-H'FEFFBFFF	
	SDRAM mode register, bank 6		—	H'FEFFC000-H'FEFFCFFF	
	SDRAM mode register, bank 7		—	H'FEFFD000-H'FEFFDFFF	

3.4.1 Bus Control Register (BCR)

The bus control register (BCR) is a 32-bit read/write register that specifies the status of EMI pins. It is initialized to H'0000000x by a power-on reset, but is not initialized by a manual reset or in standby mode.

BCR— Bus control Register

H'E1020000



* Samples the value of pin MD_ENDIAN upon a power-on reset

- **Bits 31~8, 5~1:** Reserved: Writes to these bits have no effect and always read as 0.
- **Bit 7 HIZMEM:** Hi-Z memory control. Specifies the status of A[25:0], CS4H#, CS5H#, CS6H#, CS7H#, RD/WR, WE#, RD#, RAS# and CAS# standby mode or during clock frequency change.

Bit 7: HIZMEM	Description	
0	A[25:0], CS4H#, CS5H#, CS6H#, CS7H#, RD/WR, WE#, RD#, RAS# and CAS# are Hi-Z in standby mode or during clock frequency change	(Initial value)
1	A[25:0], CS4H#, CS5H#, CS6H#, CS7H#, RD/WR, WE#, RD#, RAS# and CAS# are High in standby mode or during clock frequency change	

- **Bit 6 HIZCNT:** High-Z Control. Specifies the status of the CS4L#, CS5L#, CS6L#, CS7L#, and DQMn/WEn# in standby mode.

Bit 6: HIZCNT	Description	
0	CS4L#, CS5L#, CS6L#, CS7L#, and DQMn/WEn# are high-impedance status (High-Z) in standby mode or during clock frequency change	(Initial value)
1	CS4L#, CS5L#, CS6L#, CS7L#, and DQMn/WEn# are driven in standby mode or during clock frequency change	

- **Bit 0 ENDIAN:** Endian flag. Samples the value of pin MD_ENDIAN designating endian upon a power-on reset. The endian for all physical space is decided by this bit. This bit is read only.

Bit 0: ENDIAN	Description	
---------------	-------------	--

0	(On reset) Endian setting pin (MD_ENDIAN) is low. Indicates big endian.	(Initial value)
1	(On reset) Endian setting pin (MD_ENDIAN) is high. Indicates little endian.	

3.4.2 Static Memory Control Register (SMCRn, n = 0 to 3)

SMCRn are 32-bit read/write registers that contain control bits for configuring static memory. On a power-on reset, SMCR0 is initialized to B'0000 1111 1111 1111 0111 0111 xy00 0000 (binary) where x represents the value of pin MD4 and y represents the value of pin MD3, SMCR1~3 are initialized to H'0fff7700. They are not initialized by a manual reset or in standby mode.

Static memory other than bank 0 should not be accessed until register initialization is completed.

SMCR0— Static Memory Control Register

H'E1020004

Bit:	31	30	29	28	27	26	25	24
Read:					STRV3	STRV2	STRV1	STRV0
Write:								
Reset:	0	0	0	0	1	1	1	1

Bit:	23	22	21	20	19	18	17	16
Read:	TAW3	TAW2	TAW1	TAW0	TBP3	TBP2	TBP1	TBP0
Write:								
Reset:	1	1	1	1	1	1	1	1

Bit:	15	14	13	12	11	10	9	8
Read:		TAH2	TAH1	TAH0		TAS2	TAS1	TAS0
Write:								
Reset:	0	1	1	1	0	1	1	1

Bit:	7	6	5	4	3	2	1	0
Read:	BW1	BW0			BCM	BL1	BL0	SMT
Write:								
Reset:	x	y	0	0	0	0	0	0

SMCR1— Static Memory Control Register
SMCR2— Static Memory Control Register
SMCR3— Static Memory Control Register

H'E1020008
H'E102000C
H'E1020010

Bit:	31	30	29	28	27	26	25	24
Read:					STRV3	STRV2	STRV1	STRV0
Write:								
Reset:	0	0	0	0	1	1	1	1

Bit:	23	22	21	20	19	18	17	16
Read:	TAW3	TAW2	TAW1	TAW0	TBP3	TBP2	TBP1	TBP0
Write:								
Reset:	1	1	1	1	1	1	1	1

Bit:	15	14	13	12	11	10	9	8
Read:		TAH2	TAH1	TAH0		TAS2	TAS1	TAS0
Write:								
Reset:	0	1	1	1	0	1	1	1

Bit:	7	6	5	4	3	2	1	0
Read:	BW1	BW0			BCM	BL1	BL0	SMT
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 31~28, 15, 11, 5~4: Reserved bits. Writes to these bits have no effect and always read as 0.

- **Static Memory Recovery Time (STRV3, STRV2, STRV1, STRV0):** Specify the number of idle cycles inserted between bus cycles when switching from one bank to another bank or between a read access to a write access in the same bank.

Bit 27~24: STRV3~0	Description	
0000	0 idle cycle	
0001	1 idle cycle	
0010	2 idle cycles	
0011	3 idle cycles	
0100	4 idle cycles	
0101	5 idle cycles	
0110	6 idle cycles	
0111	7 idle cycles	
1000	8 idle cycles	
1001	9 idle cycles	
1010	10 idle cycles	
1011	11 idle cycles	
1100	12 idle cycles	
1101	13 idle cycles	
1110	14 idle cycles	
1111	15 idle cycles	(Initial value)

- **Access Wait Time (TAW3, TAW2, TAW1, TAW0):** For normal memory, these bits specify the number of wait cycles to be inserted in read strobe time. For burst ROM, these bits specify the number of wait cycles to be inserted in first data read strobe time.

Bit 23~20: TAW3~0	Waits Inserted	WAIT# Pin	
0000	0 cycle	Ignored	
0001	1 cycle	Enabled	
0010	2 cycles	Enabled	
0011	3 cycles	Enabled	
0100	4 cycles	Enabled	
0101	5 cycles	Enabled	
0110	6 cycles	Enabled	
0111	7 cycles	Enabled	
1000	8 cycles	Enabled	
1001	9 cycles	Enabled	
1010	10 cycles	Enabled	
1011	12 cycles	Enabled	
1100	15 cycles	Enabled	
1101	20 cycles	Enabled	
1110	25 cycles	Enabled	
1111	31 cycles	Enabled	(Initial value)

- **Burst Pitch Time (TBP3, TBP2, TBP1, TBP0):** For burst ROM, these bits specify the number of wait cycles to be inserted in subsequent access. For normal memory, these bits specify the number of wait cycles to be inserted in write strobe time.

Bit 19~16: TBP3~0	Waits Inserted in Burst Pitch	WAIT# Pin	
0000	0 cycle	Ignored	
0001	1 cycle	Enabled	
0010	2 cycles	Enabled	
0011	3 cycles	Enabled	
0100	4 cycles	Enabled	
0101	5 cycles	Enabled	
0110	6 cycles	Enabled	
0111	7 cycles	Enabled	
1000	8 cycles	Enabled	
1001	9 cycles	Enabled	
1010	10 cycles	Enabled	
1011	12 cycles	Enabled	
1100	15 cycles	Enabled	
1101	20 cycles	Enabled	
1110	25 cycles	Enabled	
1111	31 cycles	Enabled	(Initial value)

- **Address Hold Time (TAH2, TAH1, TAH0):** These bits specify the number of wait cycles to be inserted from negation of read/write strobe to address.

Bit 14~12: TAH2~0	Waits Inserted in Hold	
000	0 cycle	
001	1 cycle	
010	2 cycles	
011	3 cycles	
100	4 cycles	
101	5 cycles	
110	6 cycles	
111	7 cycles	(Initial value)

- **Address Setup Time (TAS2, TAS1, TAS0):** These bits specify the number of wait cycles to be inserted from address to assertion of read/write strobe.

Bit 10~8: TAS2~0	Waits Inserted in Setup	
000	0 cycle	
001	1 cycle	
010	2 cycles	
011	3 cycles	
100	4 cycles	
101	5 cycles	
110	6 cycles	
111	7 cycles	(Initial value)

- **Bus Width (BW1, BW0):** These bits specify the bus width. For bank 0, these bits are read only and represent the value of MD4 and MD3 in a power-on reset. For bank 1 to 3, these bits are writable and are initialized to 0 by a power-on reset.

Bit 7~6: BW1~0	Bus Width	
00	8 bits	(Initial value)*
01	16 bits	
10	32 bits	
11	Reserved	

- **SRAM Byte Control Mode (BCM):** When SRAM is connected, this bit specifies the type of SRAM. This bit is valid only when SMT is set to 0.

Bit 3: BCM	Description	
0	SRAM is set to normal mode	(Initial value)
1	SRAM is set to byte control mode	

- **Burst Length (BL1, BL0):** When Burst ROM is connected, these bits specify the number of burst in an access. These bits are valid only when SMT is set to 1.

Bit 2~1: BL1~0	Burst Length	
00	4 consecutive accesses Can be used with 8-, 16-, or 32-bit bus width	(Initial value)
01	8 consecutive accesses Can be used with 8-, 16-, or 32-bit bus width	
10	16 consecutive accesses Can only be used with 8- or 16-bit bus width. Do not specify for 32-bit bus width	
11	32 consecutive accesses Can only be used with 8-bit bus width	

- **Static Memory Type (SMT):** This bit specifies the type of static memory.

Bit 0: SMT	Description	
0	Normal memory	(Initial value)
1	Burst ROM	

3.4.3 DRAM Control Register (DMCR)

The DRAM control register (DMCR) is a 32-bit read/write register that specifies the timing, address multiplexing and refresh control of synchronous DRAM (bank 4 to 7). This enables direct connection of synchronous DRAM without external circuits.

The DMCR is initialized to H'00000000 by a power-on reset, but is not initialized by manual reset or in standby mode. When using synchronous DRAM, bank 0 to 3 should not be accessed until initialization is completed.

DMCR— DRAM Control Register

H'E1020014

Bit:	31	30	29	28	27	26	25	24
Read:		SODIM	PGM	CA2	CA1	CA0	RMODE	RFSH
Write:		M						
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:	MRSET	CS	RA1	RA0	BA	PDM		
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	TRAS2	TRAS1	TRAS0	RCD1	RCD0	TPC2	TPC1	TPC0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:		TRWL1	TRWL0	TRC2	TRC1	TRC0	TCL1	TCL0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit 32, 17~16, 7: Reserved bits. Writes to these bits have no effect and always read as 0.

- **SODIMM Supporting (SODIMM):** This bit specifies the type of synchronous DRAM.

Bit 30: SODIMM	Description	
0	SDRAM chip or DIMM is used	(Initial value)
1	SDRAM SODIMM is used	

- **Page Mode (PGM):** Set SDRAM page mode. Refer to 3.7.4 for detail.

Bit 29: PGM	Description	
0	Non-page mode	(Initial value)
1	Page mode	

- **Column Address Width (CA2, CA1, CA0):** These bits specify the column address width of connected synchronous DRAM.

Bit 28~26: CA2~0	Description	
000	8-bit column address is used.	(Initial value)
001	9-bit column address is used.	
010	10-bit column address is used.	
011	11-bit column address is used.	
100	12-bit column address is used.	
101	Reserved	
110	Reserved	
111	Reserved	

- **Refresh Mode (RMODE):** Specifies whether auto-refresh or self-refresh is performed when the RFSH bit is set to 1. When the RFSH bit is 1 and this bit is cleared to 0, CAS-before-RAS refresh (also named auto-refresh) is performed for synchronous DRAM, using the cycle set by refresh-related registers. If a refresh request is issued during an external bus cycle, the refresh cycle is executed when the bus cycle ends. When the RFSH bit is 1 and this bit is set to 1, the self-refresh state is set for synchronous DRAM after waiting for the end of any currently executing external bus cycle. All access to memory is ignored in the self-refresh state.

Bit 25: RMODE	Description	
0	Auto-refresh is performed	(Initial value)
1	Self-refresh is performed	

- **Refresh Control (RFSH):** The RFSH bit determines whether refresh is performed for synchronous DRAM.

Bit 24: RFSH	Description	
0	No refresh is performed	(Initial value)
1	Refresh is performed	

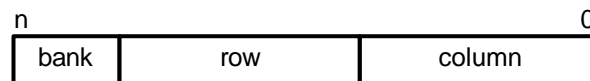
- **Mode Register Set (MRSET):** Set when a SDRAM mode register setting is used. When this bit is 0 and SDRAM mode register is written, a Pre-charge all banks command (PALL) is performed. When this bit is 1 and SDRAM mode register is written, a Mode Register Set command (MRS) is performed.

Bit 23: MRSET	Description	
0	All-bank pre-charge	(Initial value)
1	Mode register setting	

- **Chip Select Signal Number (CS):** This bit specifies the number of chip select signals for each bank. When this bit is 0, only CSnL# is valid. When this bit is 1, both CSnH# and CSnL# are valid. (n = 4 to 7)

Bit 22: CS	Description	
0	1 chip select signal is used	(Initial value)
1	2 chip select signals are used (inhibit when SODIMM is used)	

The address of synchronous DRAM is composed of following sections:



The CA2~0 bits set the column address width, RA1~0 bits set the row address width and BA bit sets the bank address width. When CS bit is 0, CSnL# is always asserted if bank n is accessed, while CSnH# keeps negative. When CS bit is 1, both CSnL# and CSnH# are valid. Which is asserted according to bit n+1 of physical address. When the bit n+1 is 0, CSnL# is asserted, otherwise, CSnH# is asserted.

- **Row Address Width (RA1, RA0):** These bits specify the column address width of connected synchronous DRAM.

Bit 21~20: RA1~0	Description	
00	11-bit row address is used	(Initial value)
01	12-bit row address is used	
10	13-bit row address is used	
11	Reserved	

- **Bank Address Width (BA):** This bit specifies the number of bank select signals for one chip select.

Bit 19: BA	Description	
0	1-bit bank address is used (2 banks each chip select)	(Initial value)
1	2-bit bank address is used (4 banks each chip select)	

- **Power Down Mode (PDM):** Sets the power-down mode. When power-down mode is set, SDRAM will be driven to power-down mode when it is not accessing and refreshing. Clock supply to SDRAM will be stopped also. Please refer to 3.7.5 for detail.

Bit 18: PDM	Description	
0	Non-power-down mode	(Initial value)
1	Power-down mode	

- **RAS Assertion Time (TRAS2, TRAS1, TRAS0):** When synchronous DRAM is connected, these bits set the minimum CKE negation time after self-refresh command is issued.

Bit 15~13: TRAS2~0	Synchronous DRAM minimum CKE negation Time	
000	4	(Initial value)
001	5	
010	6	
011	7	
100	8	
101	9	
110	10	
111	11	

- **RAS–CAS Delay (RCD1, RCD0):** These bits set the synchronous DRAM bank active-read/write command delay time.

Bit 12~11: RCD1~0	Description	
00	1	(Initial value)
01	2	
10	3	
11	4	

- **RAS Precharge Time (TPC2, TPC1, TPC0):** When the synchronous DRAM interface is selected, these bits specify the minimum number of cycles until the next bank active command is output after precharging.

Bit 10~8: TPC2~0	Description	
000	1 cycle	(Initial value)
001	2 cycles	
010	3 cycles	
011	4 cycles	
100	5 cycles	
101	6 cycles	
110	7 cycles	
111	8 cycles	

- **Write Precharge Time (TRWL1, TRWL0):** These bits set the synchronous DRAM write precharge delay time. In auto-precharge mode, they specify the time until the next bank active command is issued after a write cycle. After a write cycle, the next active command is not issued for a period of TRWL + TPC.

Bit 6~5: TRWL1~0	Description	
00	1 cycle	(Initial value)
01	2 cycles	
10	3 cycles	
11	4 cycles	

- **RAS Cycle Time (TRC2, TRC1, TRC0):** For synchronous DRAM, no bank active command is issued during the period TRC after an auto-refresh command. In self-refresh, these bits also specify the delay cycles to be inserted after CKE assertion.

Bit 4~2: TRC2~0	Description	
000	1 cycle	(Initial value)
001	3 cycles	
010	5 cycles	
011	7 cycles	
100	9 cycles	
101	11 cycles	
110	13 cycles	
111	15 cycles	

- **CAS Latency (TCL1, TCL0):** For synchronous DRAM, these bits specify the delay from read command to data becomes available at the outputs.

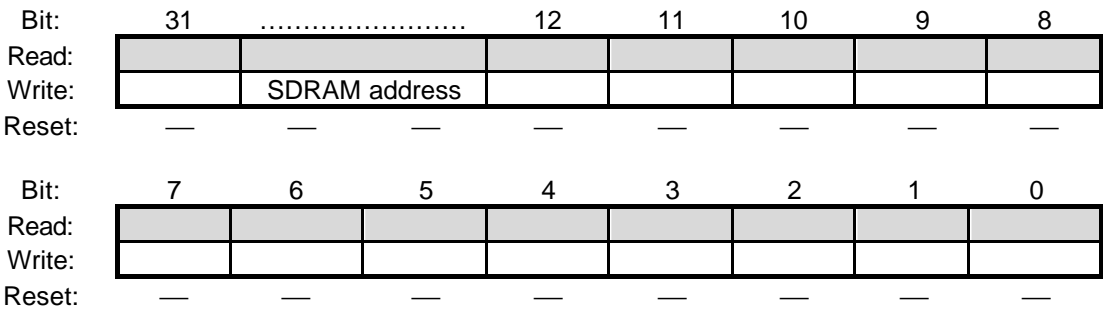
Bit 1~0: TCL1~0	Description	
00	Inhibit	(Initial value)
01	2 cycles	
10	3 cycles	
11	Inhibit	

3.4.4 SDRAM Mode Register (SDMR)

The SDRAM mode register (SDMR) is written to via the synchronous DRAM address bus and is a 10-bit write-only register. It sets SDRAM mode for bank 4 to 7. SDMR is undefined after a power-on reset.

Writing to the SDRAM mode register uses the address bus rather than the data bus. If the value to be set is X and the SDMR address is Y, the value X is written in the SDRAM mode register by writing in address X + Y. Since A0 of the synchronous DRAM is connected to A2 of the Arca210 and A1 of the Synchronous DRAM is connected to A3 of the Arca210, the value actually written to the synchronous DRAM is the X value shifted two bits right. For example, when H'230 is written to the SDMR register of bank 4, random data is written to the address H'E102B000 (address Y) + H'8C0 (value X), or H'E102B8C0. As a result, H'230 is written to the SDMR register. When H'230 is written to the SDMR register of bank 5, random data is written to the address H'E102C000 (address Y) + H'8C0 (value X), or H'E102C8C0. As a result, H'230 is written to the SDMR register. The range for value X is H'000 to H'FFC.

SDMR— SDRAM Mode Register



The Mode Register is used to define the specific mode of operation of the SDRAM. This definition includes the section of a burst length, a burst type, a CAS latency, an operating mode and a write burst mode, as shown in following figure.

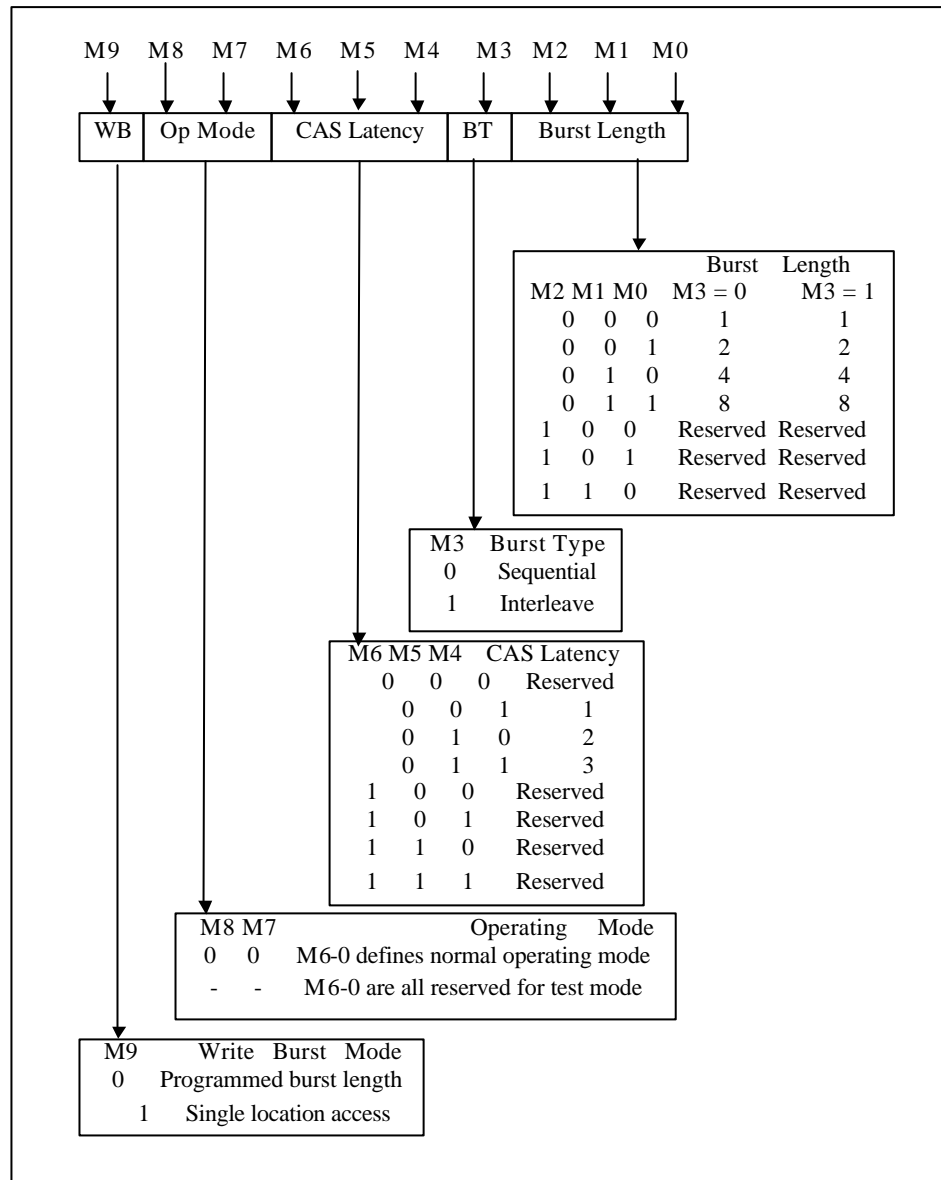


Figure 3-3 Synchronous DRAM Mode Register Configuration

3.4.5 Refresh Timer Control/Status Register (RTCSR)

The refresh timer control/status register (RTCSR) is a 16-bit readable/writable register that specifies the refresh cycle and the status of RTCNT.

RTCSR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or in standby mode.

RTCSR— Refresh Timer Control/Status Register

H'E1020018

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	CMF					CKS2	CKS1	CKS0
Write:								
Reset:	0	0	0	0	0	0	0	0

- **Bit 15~8, 6~3:** Reserved bits. Writes to these bits have no effect and read always as 0.
- **Compare-Match Flag (CMF):** Status flag that indicates a match between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR) values. Writes to 1 of this bit have no effect.

Bit 7: CMF	Description	
0	RTCNT and RTCOR values do not match Clear condition: When 0 is written	(Initial value)
1	RTCNT and RTCOR values match Set condition: When RTCNT = RTCOR	

- **Refresh Clock Select Bits (CKS2, CKS1, CKS0):** These bits select the clock input to RTCNT. The source clock is the external bus clock (CKO). The RTCNT count clock is CKO divided by the specified ratio.

Bit 2~0: CKS2~0	Description	
000	Disables clock input	(Initial value)
001	Bus clock (CKO)/4	
010	CKO/16	
011	CKO/64	
100	CKO/256	
101	CKO/1024	
110	CKO/2048	
111	CKO/4096	

3.4.6 Refresh Timer Counter (RTCNT)

RTCNT is a 16-bit read/write register. RTCNT is a 16-bit counter that counts up with input clocks. The clock select bits (CKS2–CKS0) of RTCSR select the input clock. When the refresh bit (RFSH) of the memory control register (DMCR) is set to 1 and the refresh mode is set to auto-refresh, a memory refresh cycle starts when RTCNT matches RTCOR. RTCNT is initialized to H'0000 by a power-on reset.

RTCNT— Refresh Timer Counter

H'E102001C

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

3.4.7 Refresh Time Constant Register (RTCOR)

The **refresh time constant register (RTCOR)** is a 16-bit read/write register. The values of RTCOR and RTCNT (bottom 8 bits) are constantly compared. When the refresh bit (RFSH) of the memory control register (DMCR) is set to 1 and the refresh mode bit (RMODE) is set to auto-refresh, a memory refresh cycle starts when RTCNT matches RTCOR. RTCOR is initialized to H'0000 by a power-on reset.

RTCOR— Refresh Timer Constant Register

H'E1020020

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

3.4.8 DRAM Bank Address Configuration Register (DMAR1, 2, 3, 4)

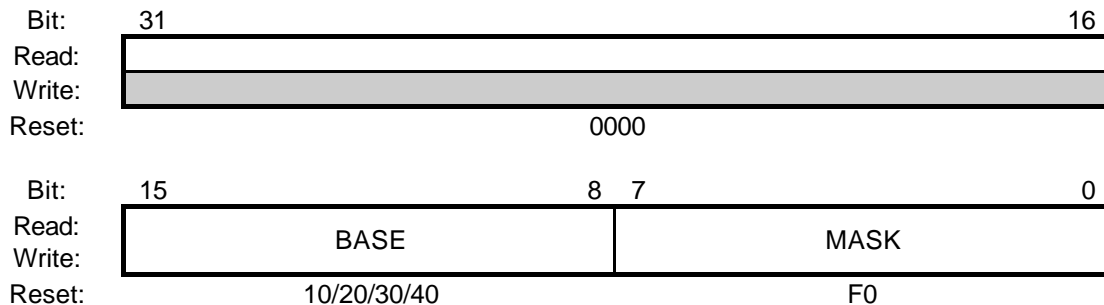
The DRAM Bank Address Configuration Registers (DMAR1, 2, 3 and 4) define the physical address for DRAM bank 4, 5, 6 and 7, respectively. Each register contains a base address and a mask. When the following equation is met:

$$(physical_address [31:24] \& MASK_n) == BASE_n$$

The bank n is active. The *physical_address* is address output on internal system bus. DRAM bank regions must be programmed so that each bank occupies a unique area of the physical address space. Programming overlapping bank regions will result in unpredictable error.

These registers are initialized by a power-on reset.

DMAR1 —	DRAM Bank 4 Address Configuration Register	H'E1020040
DMAR2 —	DRAM Bank 5 Address Configuration Register	H'E1020044
DMAR3 —	DRAM Bank 6 Address Configuration Register	H'E1020048
DMAR4 —	DRAM Bank 7 Address Configuration Register	H'E102004C



- **Bit 31~16:** Reserved bits. Writes to these bits have no effect and read always as 0.
- **Bit 15~8 Address Base (BASE):** Defines the base address of DRAM Bank n ($n = 4, 5, 6$ and 7). The initial value is:

DMAR1.BASE	0x10
DMAR2.BASE	0x20
DMAR3.BASE	0x30
DMAR4.BASE	0x40
- **Bit 7~0 Address Mask (MASK):** Defines the address of DRAM Bank n ($n = 4, 5, 6$ and 7).

3.5 Endian/Access Size and Data Alignment

The Arca210 supports both big endian, in which the 0 address is the most significant byte, and little endian, in which the 0 address is the least significant. It is designated by external pin MD_ENDIAN at the time of a power-on reset. After a power-on reset, big endian is engaged when MD_ENDIAN is low, little endian is engaged when MD_ENDIAN is high.

A data bus width of 8, 16 or 32 bits can be used for static memory, bank 0 to 3. Only 32-bit memory bus width is available for SDRAM, bank 4 to 7. This means data alignment is done by matching the device's data width. The access unit must also be matched to the device's bus width. This also means that when word data is read from a byte-width device, the read operation must happen 4 times. In the Arca210, data alignment and conversion of data length is performed automatically between the respective interfaces.

Following tables show the relationship between endian, device data width, and access unit.

Table 3-4 32-Bit External Device/Big Endian Access and Data Alignment

WE3, CAS3, DQM3	WE2, CAS2, DQM2	WE1, CAS1, DQM1	WE0, CAS0, DQM0	D31- D24	D23- D16	D15-D8	D7-D0	Operation
ASSERT	———	———	———	Data 7-0	———	———	———	Byte access at 0
———	ASSERT	———	———	———	Data 7-0	———	———	Byte access at 1
———	———	ASSERT	———	———	———	Data 7-0	———	Byte access at 2
———	———	———	ASSERT	———	———	———	Data 7-0	Byte access at 3
ASSERT	ASSERT	———	———	Data 15-8	Data 7-0	———	———	Half-word access at 0
———	———	ASSERT	ASSERT	———	———	Data 15-8	Data 7-0	Half-word access at 2
ASSERT	ASSERT	ASSERT	ASSERT	Data 31-24	Data 23-16	Data 15-8	Data 7-0	Word access at 0

Table 3-5 16-Bit External Device/Big Endian Access and Data Alignment

WE3, CAS3, DQM3	WE2, CAS2, DQM2	WE1, CAS1, DUM1	WE0, CAS0, DUM0	D31- D24	D23- D16	D15-D8	D7-D0	Operation
—	—	ASSERT	—	—	—	Data 7-0	—	Byte access at 0
—	—	—	ASSERT	—	—	—	Data 7-0	Byte access at 1
—	—	ASSERT	ASSERT	—	—	Data 15-8	Data 7-0	Half-word access
—	—	ASSERT	ASSERT	—	—	Data 31-24	Data 23-16	Word access at 0, 1st time at 0
—	—	ASSERT	ASSERT	—	—	Data 15-8	Data 7-0	Word access at 0, 2nd time at 1

Table 3-6 8-Bit External Device/Big Endian Access and Data Alignment

WE3, DQMUU	WE2, DQMUL	WE1, DUMLU	WE0, DUMLL	D31- D24	D23- D16	D15-D8	D7-D0	Operation
—	—	—	ASSERT	—	—	—	Data 7-0	Byte access
—	—	—	ASSERT	—	—	—	Data 15-8	Half-word access 1st time at 0
—	—	—	ASSERT	—	—	—	Data 7-0	Half-word access 2nd time at 1
—	—	—	ASSERT	—	—	—	Data 31-24	Word access at 0, 1st time at 0
—	—	—	ASSERT	—	—	—	Data 23-16	Word access at 0, 2nd time at 1
—	—	—	ASSERT	—	—	—	Data 15-8	Word access at 0, 3rd time at 2
—	—	—	ASSERT	—	—	—	Data 7-0	Word access at 0, 4th time at 3

Table 3-7 32-Bit External Device/Little Endian Access and Data Alignment

WE3, CAS3, DQM3	WE2, CAS2, DQM2	WE1, CAS1, DQM1	WE0, CAS0, DQM0	D31- D24	D23- D16	D15-D8	D7-D0	Operation
—	—	—	ASSERT	—	—	—	Data 7-0	Byte access at 0
—	—	ASSERT	—	—	—	Data 7-0	—	Byte access at 1
—	ASSERT	—	—	—	Data 7-0	—	—	Byte access at 2
ASSERT	—	—	—	Data 7-0	—	—	—	Byte access at 3
—	—	ASSERT	ASSERT	—	—	Data 15-8	Data 7-0	Half-word access at 0
ASSERT	ASSERT	—	—	Data 15-8	Data 7-0	—	—	Half-word access at 2
ASSERT	ASSERT	ASSERT	ASSERT	Data 31-24	Data 23-16	Data 15-8	Data 7-0	Word access at 0

Table 3-8 16-Bit External Device/Little Endian Access and Data Alignment

WE3, CAS3, DQM3	WE2, CAS2, DQM2	WE1, CAS1, DUM1	WE0, CAS0, DUM0	D31- D24	D23- D16	D15-D8	D7-D0	Operation
—	—	—	ASSERT	—	—	—	Data 7-0	Byte access at 0
—	—	ASSERT	—	—	—	Data 7-0	—	Byte access at 1
—	—	ASSERT	ASSERT	—	—	Data 15-8	Data 7-0	Half-word access
—	—	ASSERT	ASSERT	—	—	Data 15-8	Data 7-0	Word access at 0, 1st time at 0
—	—	ASSERT	ASSERT	—	—	Data 31-24	Data 23-16	Word access at 0, 2nd time at 1

Table 3-9 8-Bit External Device/Little Endian Access and Data Alignment

WE3, DQMUU	WE2, DQMUL	WE1, DUMLU	WE0, DUMLL	D31- D24	D23- D16	D15-D8	D7-D0	Operation
_____	_____	_____	ASSERT	_____	_____	_____	Data 7-0	Byte access
_____	_____	_____	ASSERT	_____	_____	_____	Data 7-0	Half-word access 1st time at 0
_____	_____	_____	ASSERT	_____	_____	_____	Data 15-8	Half-word access 2nd time at 1
_____	_____	_____	ASSERT	_____	_____	_____	Data 7-0	Word access at 0, 1st time at 0
_____	_____	_____	ASSERT	_____	_____	_____	Data 15-8	Word access at 0, 2nd time at 1
_____	_____	_____	ASSERT	_____	_____	_____	Data 23-16	Word access at 0, 3rd time at 2
_____	_____	_____	ASSERT	_____	_____	_____	Data 31-24	Word access at 0, 4th time at 3

3.6 Static Memory Interface

The static memory interface is comprised of four chip selects, CSn# (n = 0 to 3), and each configurable for ROM, burst ROM, SRAM, Flash or memory-like devices. The data bus width for each chip select region may be programmed to be 8, 16 or 32 bits. RD# is asserted for all reads. WE#, WEn# (n = 0 to 3) are asserted for SRAM and Flash writes. Arca210 supplies 26 bits address for access up to 64M space per chip select. A0 is not used in 16-bit wide bus and A1~0 are not used in 32-bit wide bus.

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in the case of a 32-bit device, and 16 bits in the case of a 16-bit device. When writing, only the WEn# signal for the byte to be written is asserted. For details, see section 3.6.2.

The SMT field in SMCRn registers specifies the type of memory and BW1~0 field specifies the bus width. For bank 0, the bus width is determined by the value of external pins, MD3 and MD4, in a power-on reset.

3.6.1 Normal Memory Interface

When SMT field in SMCRn (n = 0 to 3) is 0 and BCM field is 0, normal memory (nonburst ROM, Flash, normal SRAM or memory-like device) is connected to bank n. When bank n (n = 0 to 3) is accessed, CSn# is asserted as soon as address is output. In addition, the RD# signal, which can be used as OE#, and write control signals, WE0# to WE3#, are asserted.

The TAS1~0 field in SMCRn is the latency from CSn# to read/write strobe. TAW3~0 field is the delay time of RD# in read access. TBP3~0 field is the delay time of WE# and WEn# in write access. In addition, any number of waits can be inserted by means of the external pin (WAIT#). The TAH2~0 field is the latency from RD# and WEn# negation to CSn# negation, also the hold time to address and write data.

All kinds of normal memories (nonburst ROM, normal SRAM and Flash) have the same read and write timing. There are some requirements for writes to Flash memory. Flash memory space must be uncacheable and unbuffered. Writes must be exactly the width of the populated Flash devices on the data bus (no byte writes to a 32-bit bus or word writes to a 16-bit bus, and so on). Software is responsible for partitioning commands and data, and writing them out to Flash in the appropriate sequence.

The following figures show examples of connection to 32-, 16- and 8-bit data width normal memory.

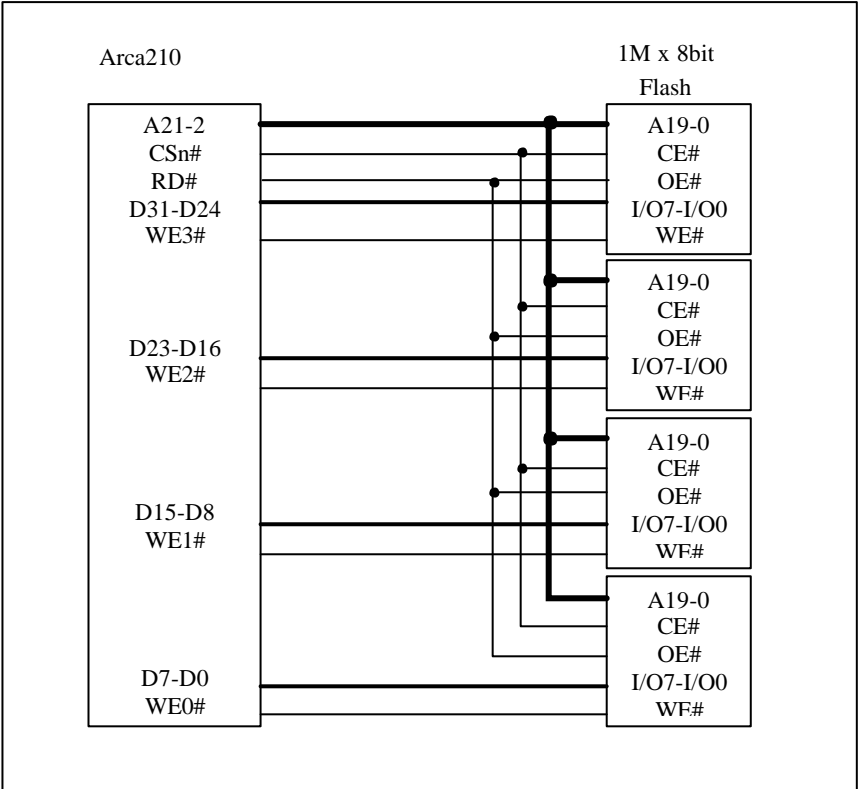


Figure 3-4 Example of 32-Bit Data Width Flash Connection

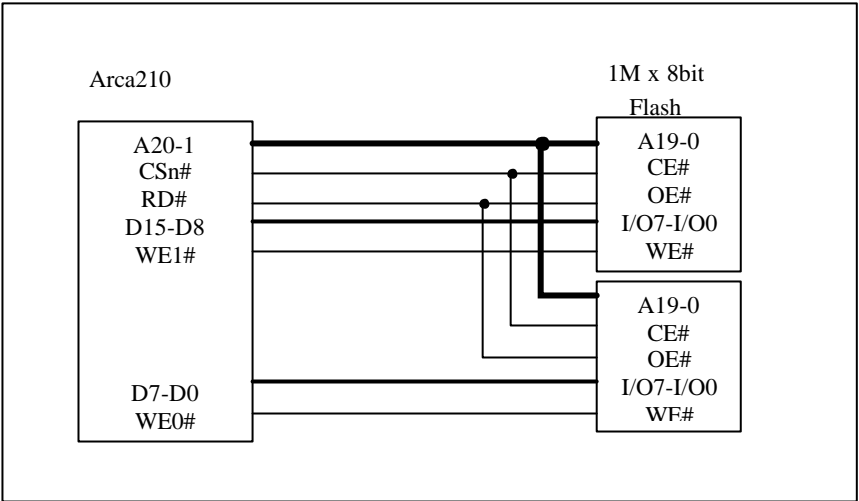


Figure 3-5 Example of 16-Bit Data Width Flash Connection

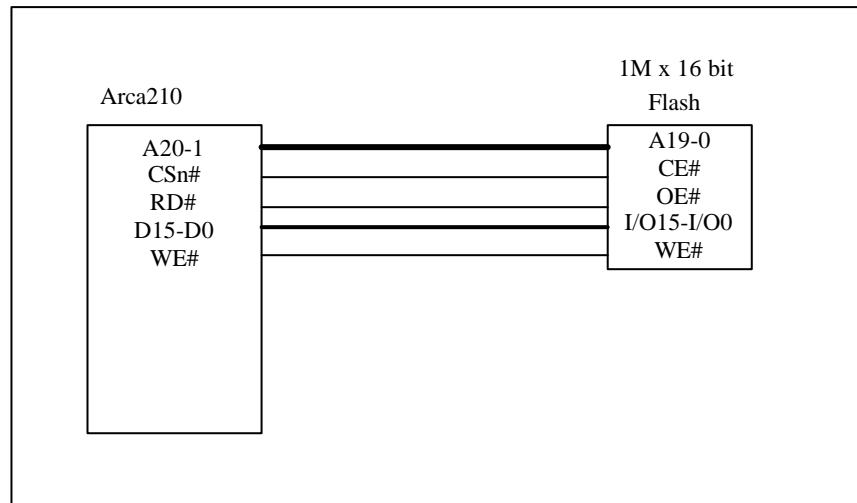


Figure 3-6 Example of 16-Bit Data Width Flash Connection

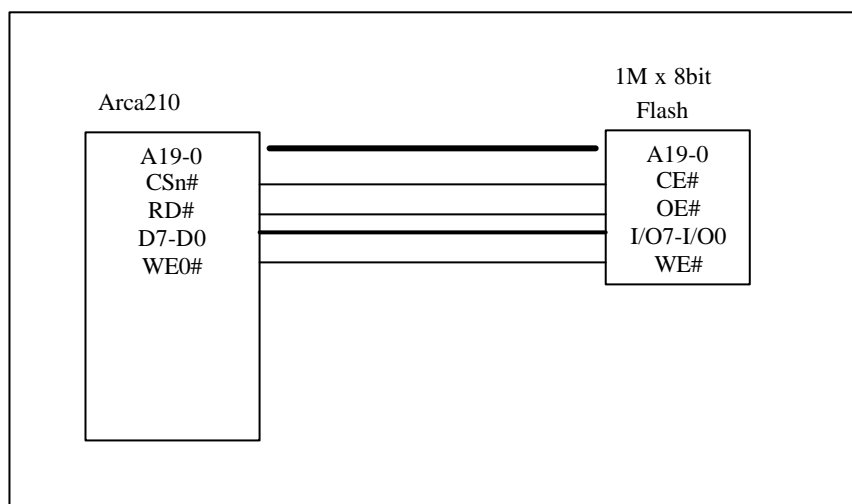


Figure 3-7 Example of 8-Bit Data Width Flash Connection

Basic Timing of Normal Memory

Following figures show the timing of normal memory. A no-wait read access is completed in one cycle and a no-wait write access is completed in two cycles. Therefore, there is no negation period in case of access at minimum pitch.

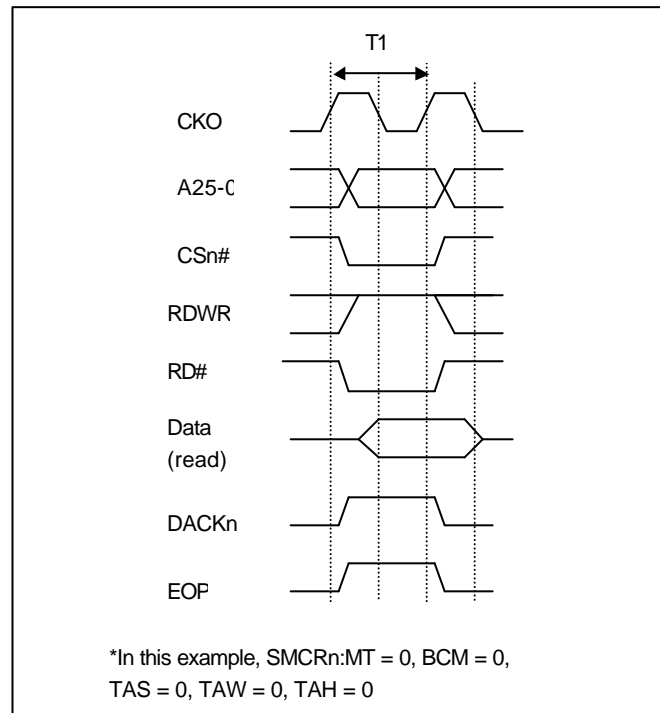


Figure 3-8 Basic Timing of Normal Memory Read

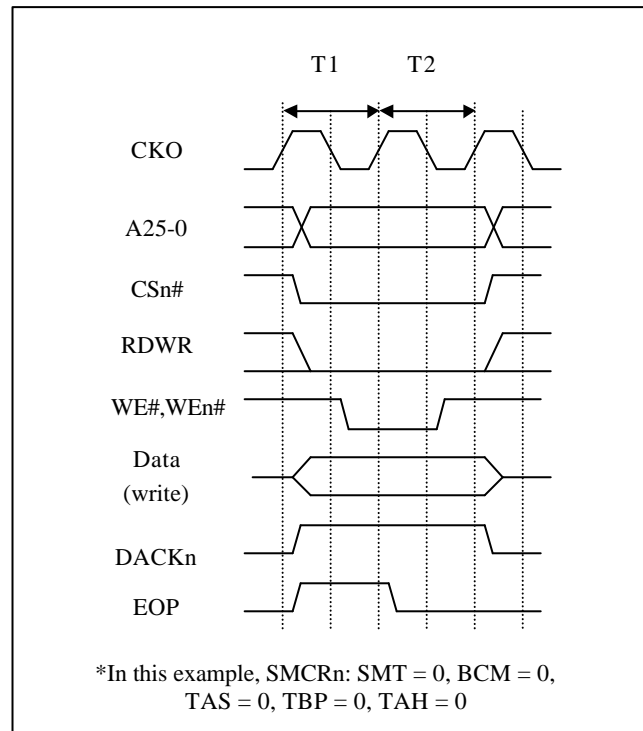


Figure 3-9 Basic Timing of Normal Memory Write

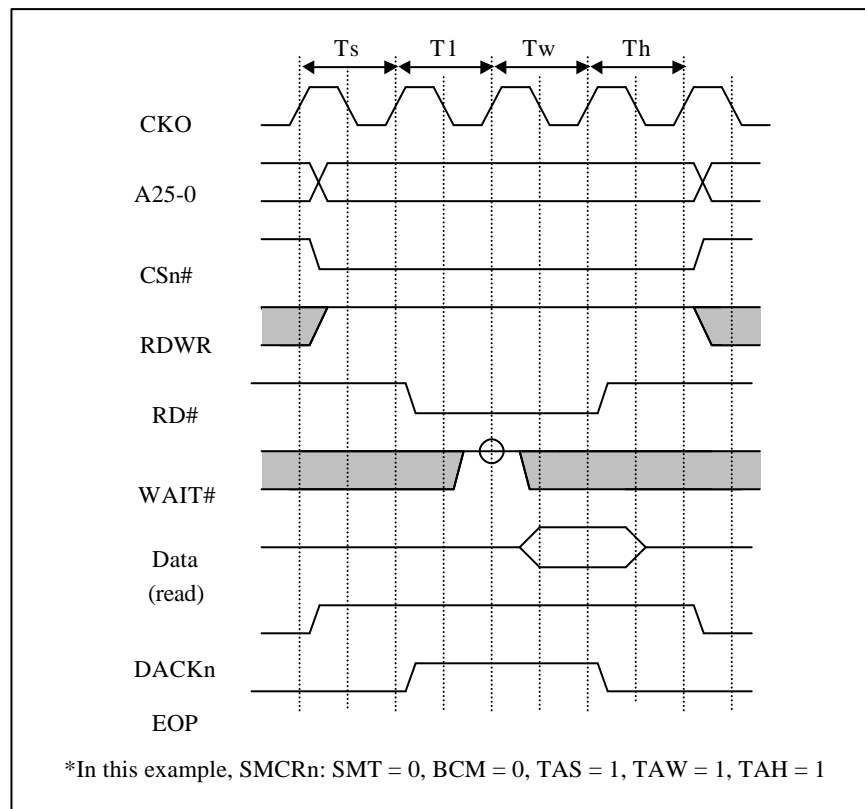


Figure 3-10 Normal Memory Read Timing With Wait (Software Wait Only)

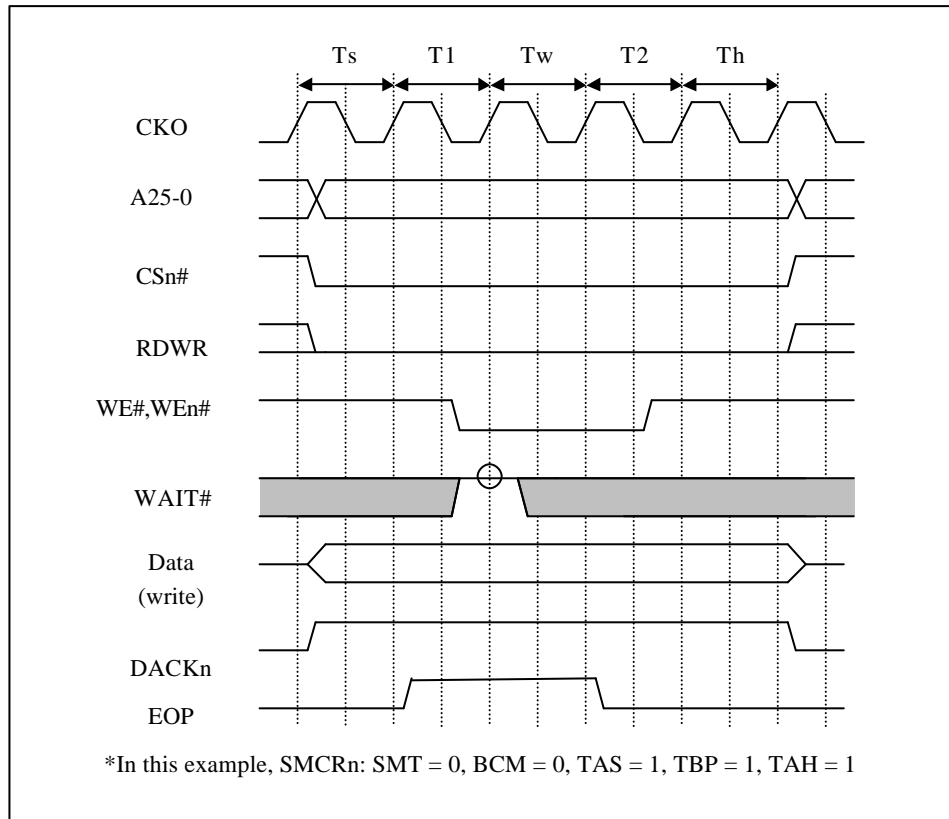


Figure 3-11 Normal Memory Write Timing With Wait (Software Wait Only)

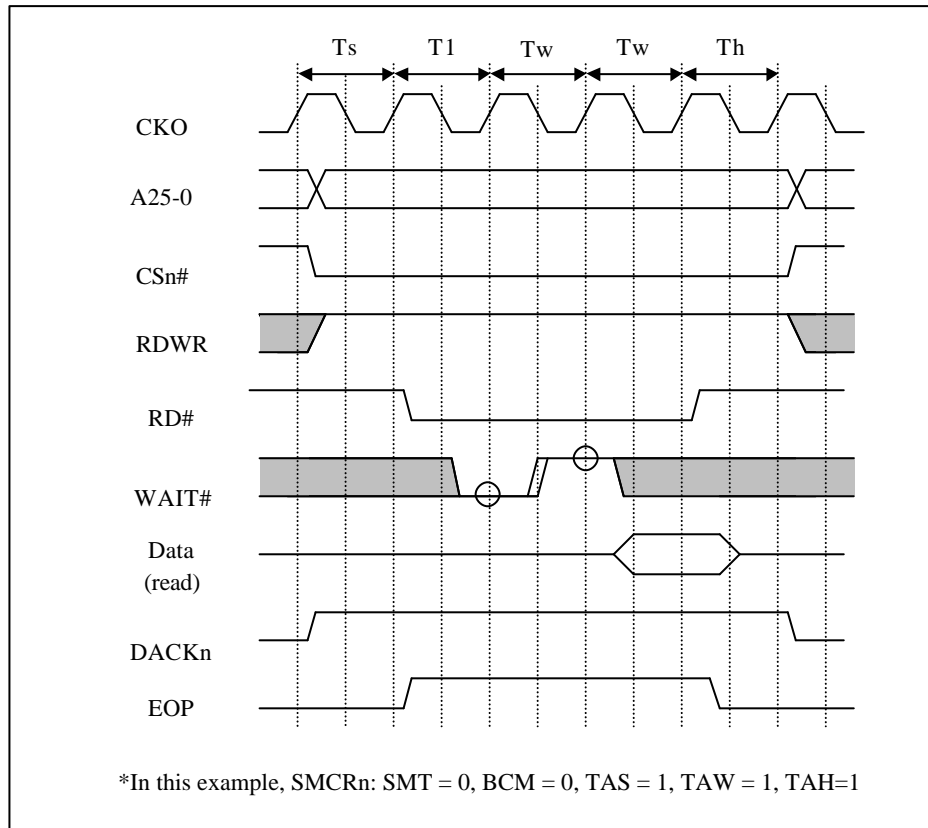


Figure 3-12 Normal Memory Read Timing With Wait (Wait Cycle Insertion by WAIT# pin)

3.6.2 Byte Control SRAM Interface

The byte control SRAM interface is a memory interface that outputs a byte select strobe WEn# in both read and write bus cycles. It has 16 bit data pins, and can be directly connected to SRAM which has an upper byte select strobe and lower byte select strobe function such as UB# and LB#.

In read/write access, RD#/WE# is used as read/write strobe signal and WEn# are used as byte select signals.

Following figure shows an example of byte control SRAM connection to Arca210.

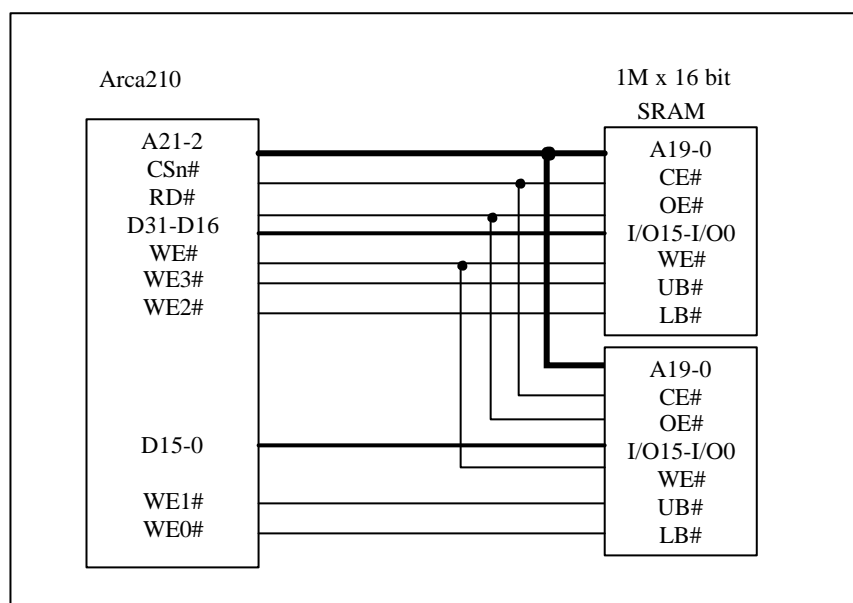


Figure 3-13 Example of 32-Bit Data Width Byte Control SRAM Connection

Following figures show examples of Byte Control SRAM timing.

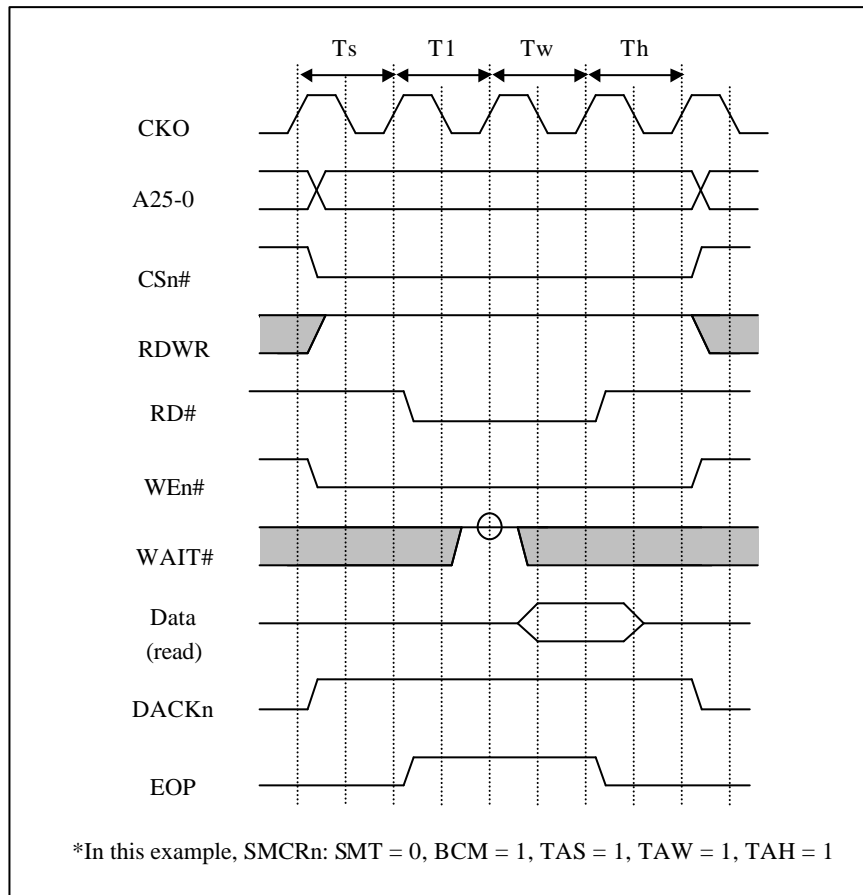


Figure 3-14 Byte Control SRAM Read Timing

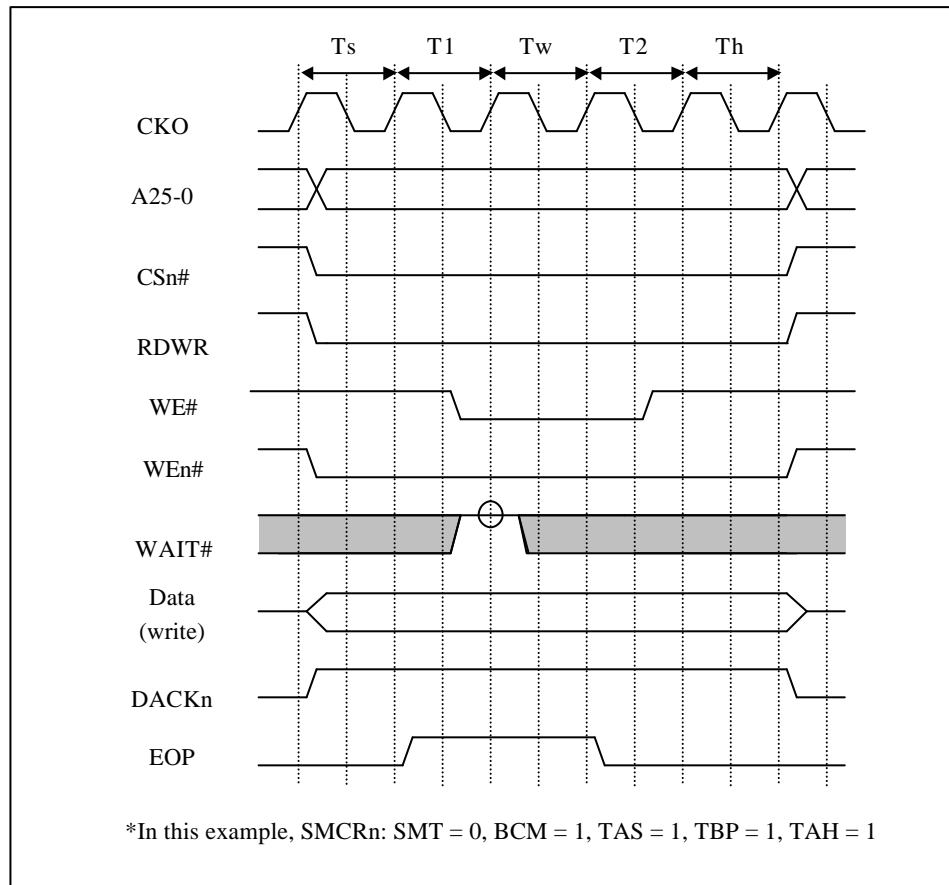


Figure 3-15 Byte Control SRAM Write Timing

3.6.3 Burst ROM Interface

Setting SMT to 1 in SMCRn allows burst ROM to be connected to bank n ($n = 0$ to 3). The burst ROM interface provides high-speed access to ROM that has a nibble access function. Basically, access is performed in the same way as for normal memory, but when the first cycle ends, only the address is changed before the next access is executed. When 8-bit burst ROM is connected, the number of consecutive accesses can be set as 4, 8, 16, or 32 with bits BL1~0. When 16-bit ROM is connected, 4, 8, or 16 can be set in the same way. When 32-bit ROM is connected, 4 or 8 can be set.

For burst ROM read, TAW sets the delay time from read strobe to the first data, TBP sets the delay time from consecutive address to data. Burst ROM writes have the same timing as normal memory except TAW instead of TBP is used to set the delay time of write strobe.

WAIT# pin sampling is always performed when one or more wait states are set.

Following figures show the timing of burst ROM.

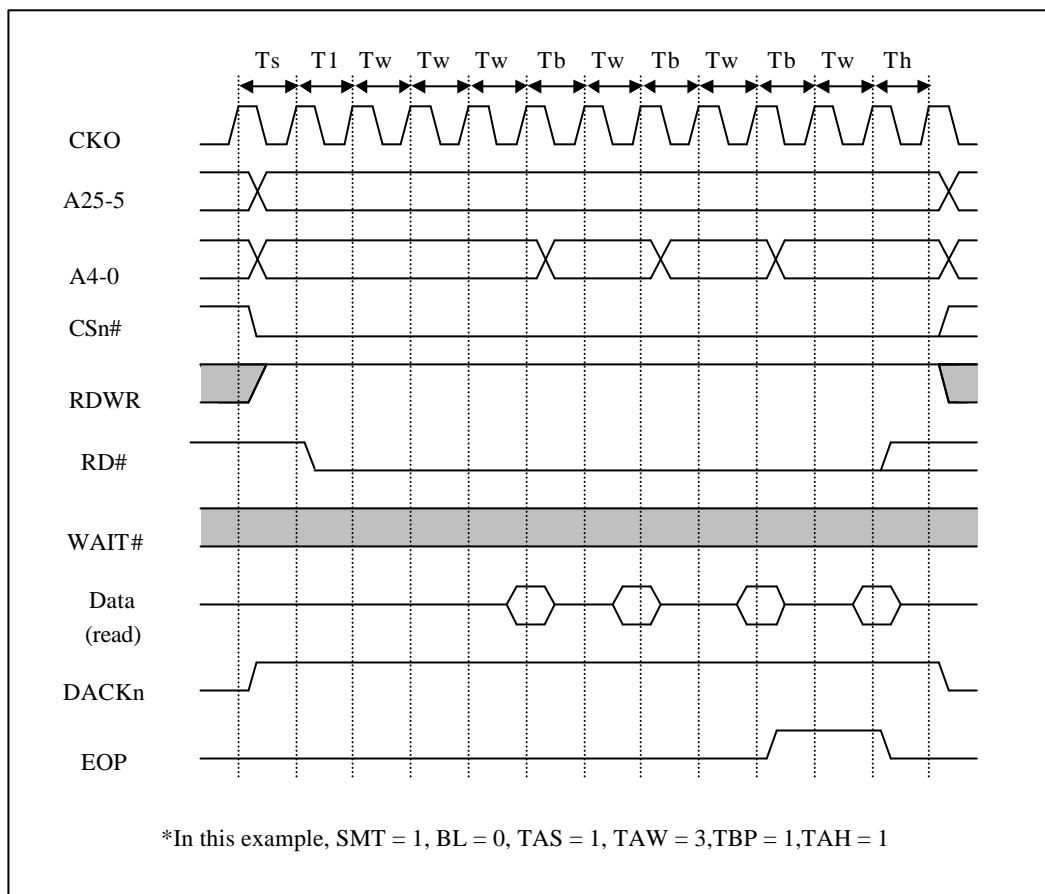


Figure 3-16 Burst ROM Read Timing (Software Wait Only)

3.7 SDRAM Interface

Synchronous DRAM Direct Connection: Since synchronous DRAM can be selected by the CS# signal, bank 4 to 7 can be connected using RAS# and other control signals in common.

With the Arca210, burst length 4, burst read/burst write mode is supported as the synchronous DRAM operating mode. A 16-byte or 32-byte burst transfer is performed in a cache fill/copy-back cycle and DMA operation. 16-byte data is read even in a single read in order to access synchronous DRAM with a burst read/write access. 16-byte transfer is also performed in a single write, but DQMn# is not asserted when unnecessary data is transferred.

The control signals for direction connection of synchronous DRAM are CS4H# - CS7H#, CS4L# - CS7L#, RAS#, CAS#, RD/WR, DQM0#, DQM1#, DQM2#, DQM3# and CKE. All signals other than CS# signal are common to all areas. Synchronous DRAM can therefore be connected in parallel to a number of areas.

Both 2-bank and 4-bank SDRAM are supported. The bank select signals are always output from the A15 pin and A16 pin of Arca210.

Byte specification is performed by DQM0#, DQM1#, DQM2# and DQM3#. When SODIMM is used, byte specification is performed by DQM4#, DQM5#, DQM7# and DQM8# also. A read/write is performed for the byte for which the corresponding DQMn# is low. DQM3# and DQM7# specify an access to address 4n, and DQM0# and DQM4# specify an access to address 4n+3.

Both synchronous DRAM chip, 100/168-pin DIMM and 144-pin SODIMM are supported.

- In system using 168-pin DIMMs, 4 or 8 chip selects correspond to 2 DIMM cards. One or two CS# signals can be selected for each bank according to CS bit of MCR. When one CS# signal is used, CSnL# are asserted during access and CSnH# are ignored (n = 4 to 7). When two CS# signals are used, the CSnH# (n = 4 to 7) is asserted when access upper half of bank n, the CSnL# (n = 4 to 7) is asserted when access lower half of bank n. Figure 3-19 shows an example of 168-pin DIMMs connection.
- In system using 100-pin DIMMs, the CS bit of DMCR should be set to 0. 4 chip selects correspond to 2 DIMM cards. CSnL# are used as chip-select signals and CSnH# is ignored (n = 4 to 7). Figure 3-20 shows an example of 100-pin DIMMs connection.
- In system using 144-pin SODIMMs, SODIMM bit of DMCR should be set to 1 and CS bit should be set to 0. 4 chip selects correspond to 2 SODIMM cards. CSnL# are used as chip-select signals and CSnH# are used as DQMn# (n = 4 to 7). When lower half of bank n is accessed, DQM0# - DQM3# are used as byte-mask-enable signals. When upper half of bank n is accessed, DQM4# - DQM7# become byte-mask-enable signals (n = 4 to 7). Figure 3-21 shows an example of 144-pin DIMMs connection.

Commands for SDRAM are specified by RAS#, CAS#, RD/WR and special address signals. The Arca210 accesses SDRAM by using the following subset of standard interface commands.

- Mode Register Set (MRS)
- Bank Activate (ACTV)
- Read (READ)
- Read with Auto-Precharge (READA)
- Write (WRIT)
- Write with Auto-Precharge (WRITA)
- Precharge All Banks (PALL)

- Auto-Refresh (CBR)
- Enter Self-Refresh (SLFRSH)
- No Operation (NOP)

Table 3-10 SDRAM Command Encoding (Notes: 1)

Command	CS#	RAS#	CAS#	RD/WR	DQM	A14-11, A9-0	A10	Note
NOP	H	X	X	X	X	X	X	
NOP	L	H	H	H	X	X	X	
MRS	L	L	L	L	X	Op-Code		
ACTV	L	L	H	H	X	Bank, Row	X	2
READ	L	H	L	H	L/H	Bank, Col	L	3
READA	L	H	L	H	L/H	Bank, Col	H	3
WRIT	L	H	L	L	L/H	Bank, Col	L	3
WRITA	L	H	L	L	L/H	Bank, Col	H	3
PRE	L	L	H	L	X	Bank	L	
PALL	L	L	H	L	X	X	H	
CBR/SLFRSH	L	L	L	H	X	X	X	4

Note:

1. CKE is HIGH for all commands shown except SLFRSH
2. A0-12 provide row address, and A13, A14 determine which bank is active.
3. A0-9 provide column address, and A13, A14 determine which bank is being read from or written to.
4. This command is CBR if CKE is HIGH, SLFRSH if CKE is LOW.

3.7.1 Example of Connection

Following figure shows an example of connection of 512K x 16-bit x 2-bank synchronous DRAM.

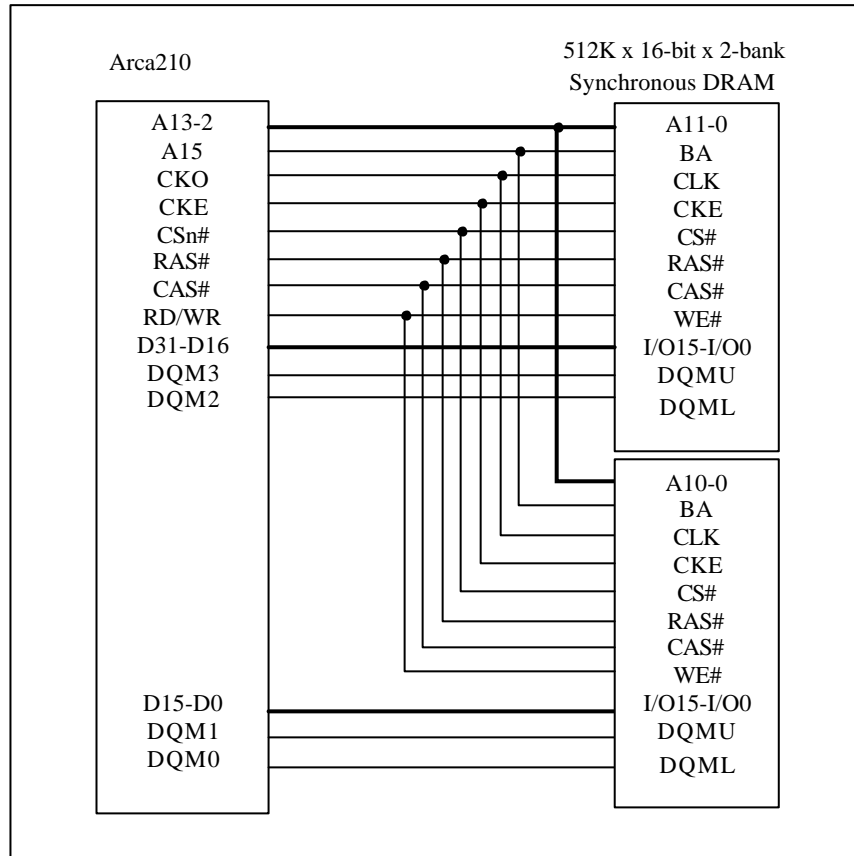


Figure 3-17 Example of Synchronous DRAM Chip Connection (1)

Following figure shows an example of connection of 1M x 16-bit x 4-bank synchronous DRAM.

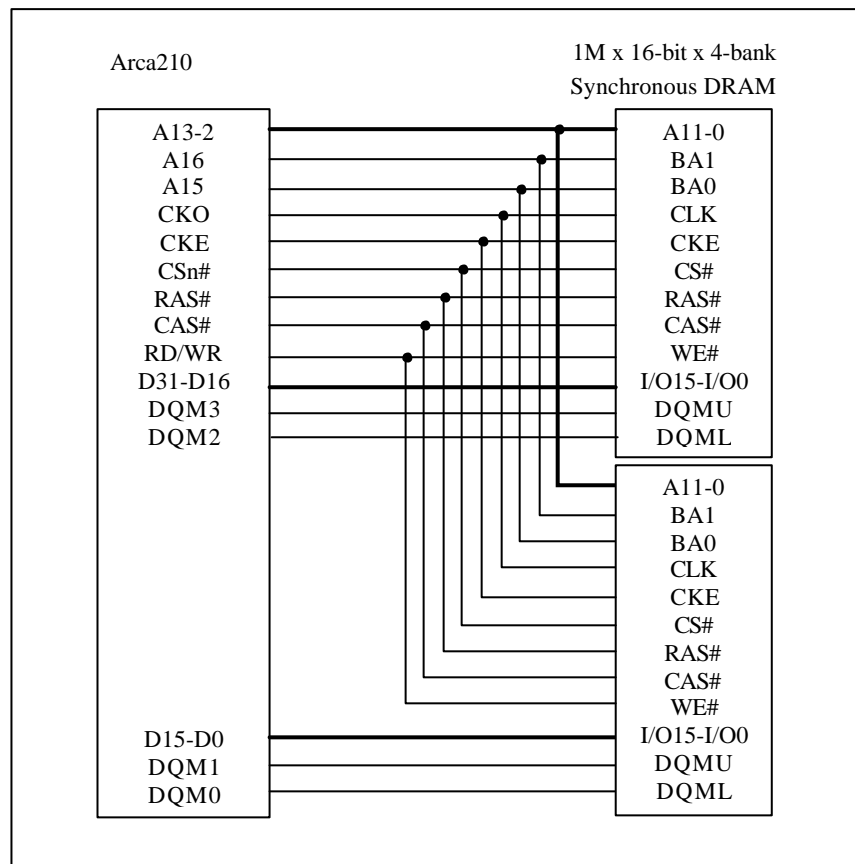


Figure 3-18 Example of Synchronous DRAM Chip Connection (2)

Following figure shows an example of connection of 168-pin synchronous DRAM DIMMs.

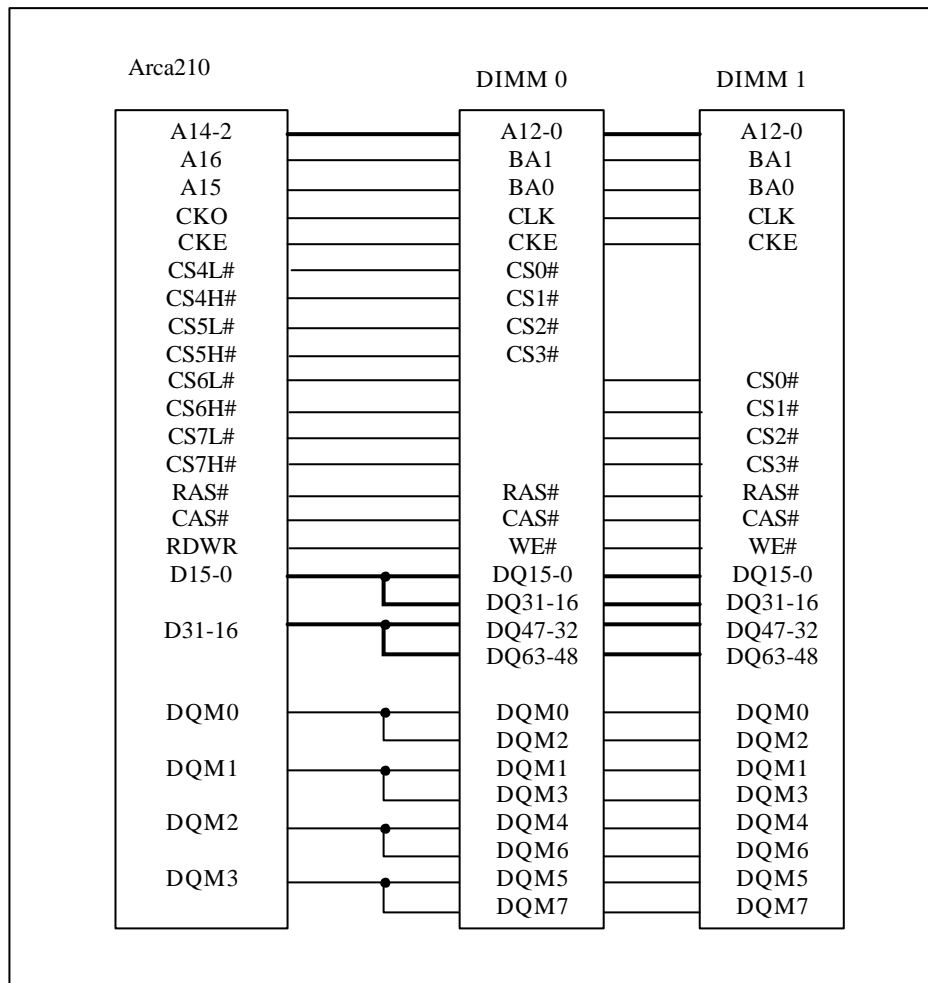


Figure 3-19 Example of Synchronous DRAM 168-pin DIMMs Connection

Following figure shows an example of connection of 100-pin synchronous DRAM DIMMs.

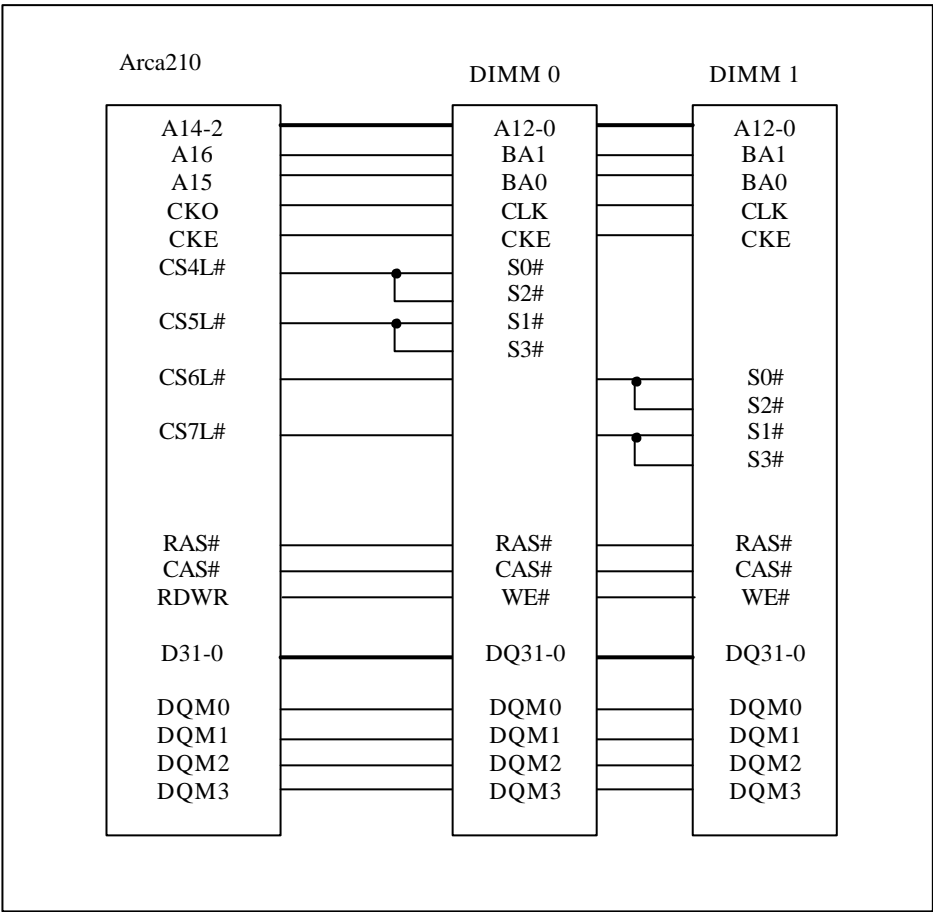


Figure 3-20 Example of Synchronous DRAM 100-pin DIMMs Connection

Following figure shows an example of connection of 144-pin synchronous DRAM DIMMs.

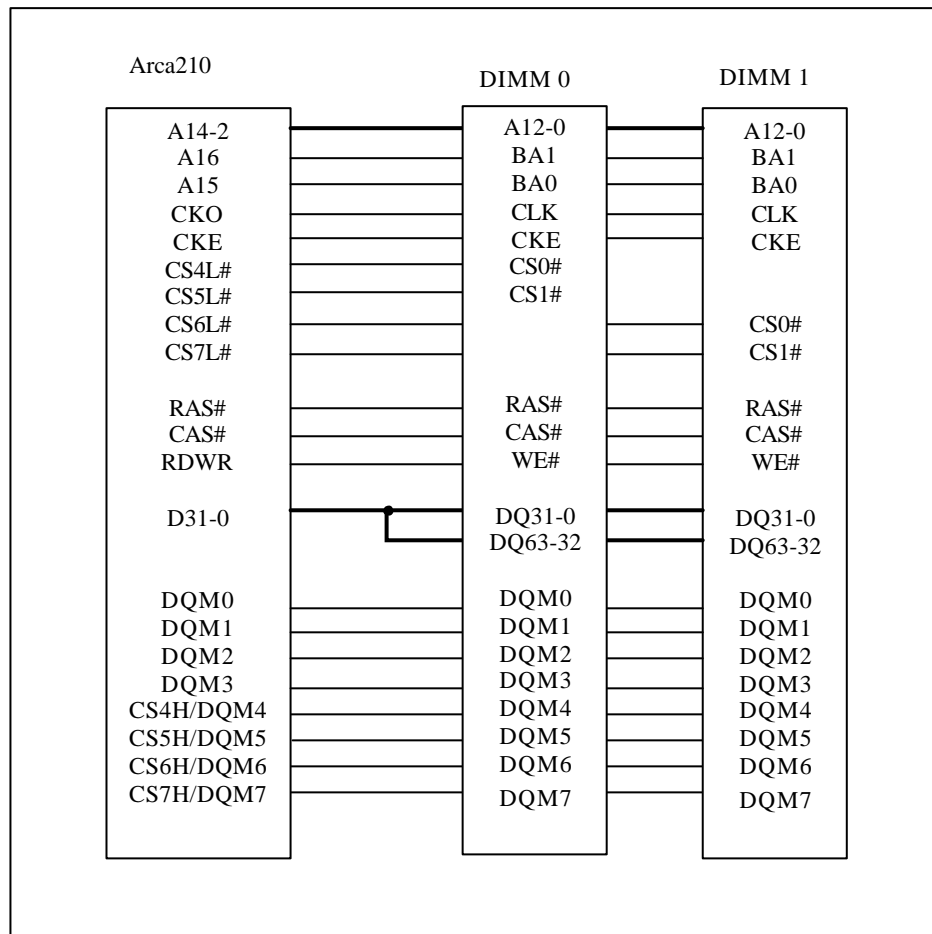


Figure 3-21 Example of Synchronous DRAM 144-pin DIMMs Connection

3.7.2 Address Multiplexing

Synchronous DRAM can be connected without external multiplexing circuitry in accordance the address multiplex specification bits CA2~0, RA1~0 and BA in DMCR. Table 3-11 shows the relationship between the address multiplex specification bits and the bits output at the address pins.

A25~A17 and A1-A0 are not multiplexed; the original values are always output at these pins.

When A0, the LSB of the synchronous DRAM address, is connected to the Arca210, it performs word address specification. Therefore the connection should be made in this order: connect pin A0 of the synchronous DRAM to pin A2 of the Arca210, then connect pin A1 to pin A3.

Table 3-11 Relationship between CA, BA and Address Multiplex Output*⁶

CA2~0	RA1~0	Output Timing	A2-A11, A12, A13-A14	A15* ²	A16* ²	Note
8 bits	11 bits	Column	A2-A11, L/H* ¹ , A12, A13	A21	A22	3, 4
		Row	A10-A22			
	12 bits	Column	A2-A11, L/H* ¹ , A12, A13	A22	A23	3, 4
		Row	A10-A22			
	13 bits	Column	A2-A11, L/H* ¹ , A12, A13	A23	A24	3, 4
		Row	A10-A22			
9 bits	11 bits	Column	A2-A11, L/H* ¹ , A12, A13	A22	A23	3, 4
		Row	A11-A23			
	12 bits	Column	A2-A11, L/H* ¹ , A12, A13	A23	A24	3, 4
		Row	A11-A23			
	13 bits	Column	A2-A11, L/H* ¹ , A12, A13	A24	A25	3, 4
		Row	A11-A23			
10 bits	11 bits	Column	A2-A11, L/H* ¹ , A12, A13	A23	A24	3, 4
		Row	A12-A24			
	12 bits	Column	A2-A11, L/H* ¹ , A12, A13	A24	A25	3, 4
		Row	A12-A24			
	13 bits	Column	A2-A11, L/H* ¹ , A12, A13	A25	A26	3, 4
		Row	A12-A24			
11 bits	11 bits	Column	A2-A11, L/H* ¹ , A12, A13	A24	A25	3, 4
		Row	A13-A25,			
	12 bits	Column	A2-A11, L/H* ¹ , A12, A13	A25	A26	3, 4
		Row	A13-A25,			
	13 bits	Column	A2-A11, L/H* ¹ , A12-A17	A26	A27	3, 4
		Row	A13-A25,			
12 bits	11 bits	Column	A2-A11, L/H* ¹ , A12, A13	A25	A26	3, 4
		Row	A14-A26			
	12 bits	Column	A2-A11, L/H* ¹ , A12, A13	A26	A27	3, 4
		Row	A14-A26			
	13 bits	Column	A2-A11, L/H* ¹ , A12, A13	A27	A28	3, 4
		Row	A14-A26			

Notes:

1. L/H is a bit used in the command specification; it is fixed at L or H according to the Access mode.
2. Bank address specification
3. If one bank select signal is used (BA = 0), take A15 as bank select signal. If two bank select signals are used (BA = 1), take A15 and A16 as bank select signals

4. If one chip-select signal is used ($CS = 0$), only $CSnL$ ($n = 4$ to 7) are asserted during access. If two chip-select signals are used ($CS = 1$), $CSnH$ or $CSnL$ is asserted according to the high address bit.
- 5.
6. The A2 to A16 in table head are output pins. The A2 to A28 in table body are physical address.

3.7.3 Non-Page Mode Timing

16-Byte Burst Read: 16-byte burst read occurs in the event of cache fill or a 16-byte transfer of DMA. In the following example it is assumed that four 2M 8-bit synchronous DRAM chips are connected and a 32-bit data width is used, and the burst length is 4. Following the T_r cycle, ACTV command is performed, a READA command is issued in the T_{c1} , and the read data is accepted on the rising edge of the external command clock (CKO) from cycle T_{d1} to cycle T_{d4} . The T_{pc} cycle is used to wait for completion of auto-pre-charge based on the READA command inside the synchronous DRAM; no new access command can be issued to the same bank during this cycle, but access to synchronous DRAM for another area is possible. In the Arca210, the number of T_{pc} cycles is determined by the TPC bit specification in DMCR, and commands cannot be issued for the same synchronous DRAM during this interval.

The example in Figure 3-22 shows the basic timing. To connect slower synchronous DRAM, the cycle can be extended by setting DMCR bits. The number of cycles from the ACTV command output cycle (T_r) to the READ command output cycle (T_{c1}) can be specified by the RCD bit in DMCR, with a value of 0 to 3 specifying 1 to 4 cycles, respectively. In case of 2 or more cycles, T_{rw} cycles, in which an NOP command is issued for the synchronous DRAM, are inserted between the T_r cycle and the T_{c1} cycle. The number of cycles from READA command output cycles T_{c1} to the first read data valid cycle (T_{d1}) can be specified as 1 to 3 cycles by means of $TCL1$ - $TCL0$ in DMCR. This number of cycles corresponds to the number of synchronous DRAM CAS latency cycles.

Following figure shows the 16-byte burst read timing.

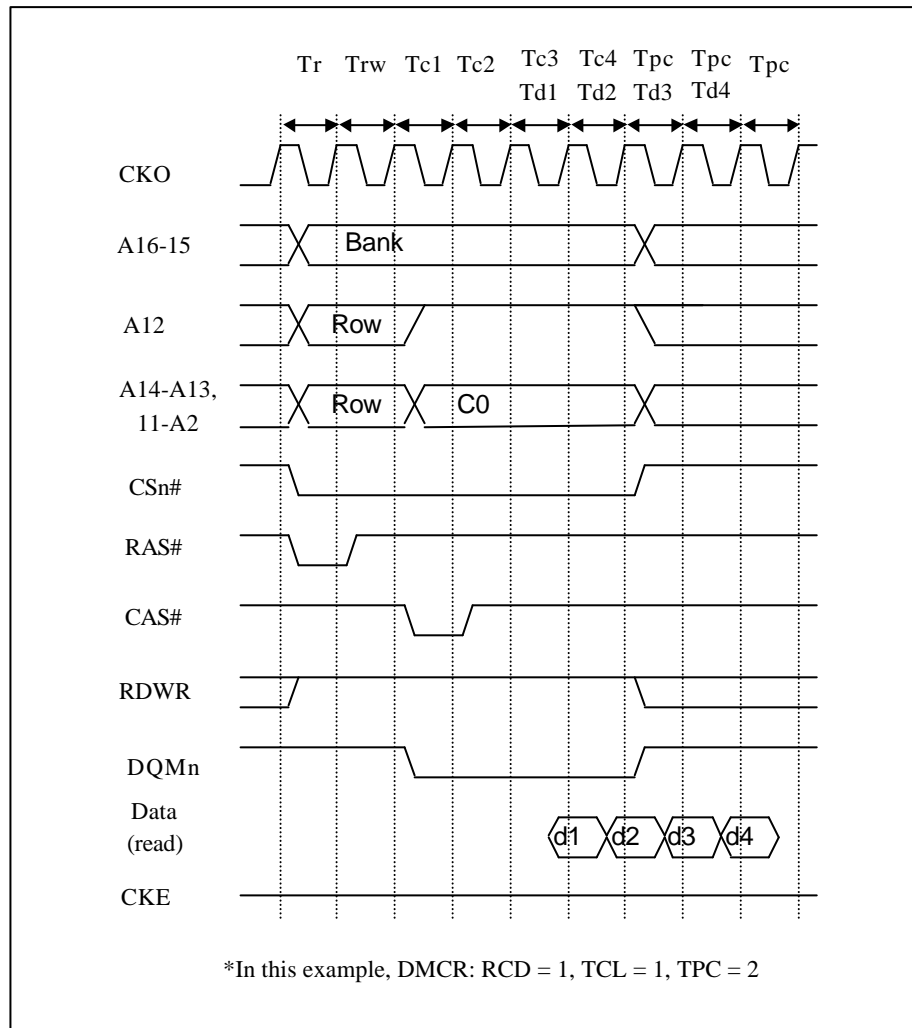


Figure 3-22 Synchronous DRAM 16-Byte Burst Read Timing (Non-Page Mode)

32-Byte Burst Read: 32-byte burst read occurs in the event of cache fill or a 32-byte transfer of DMAC. Following figure shows an example of the 32-byte burst read timing. Synchronous DRAM burst length is 4. Following the ACTV command, a READ command is issued for the first 16-byte data. After Tc4, a READA command is issued for the second 16-byte data.

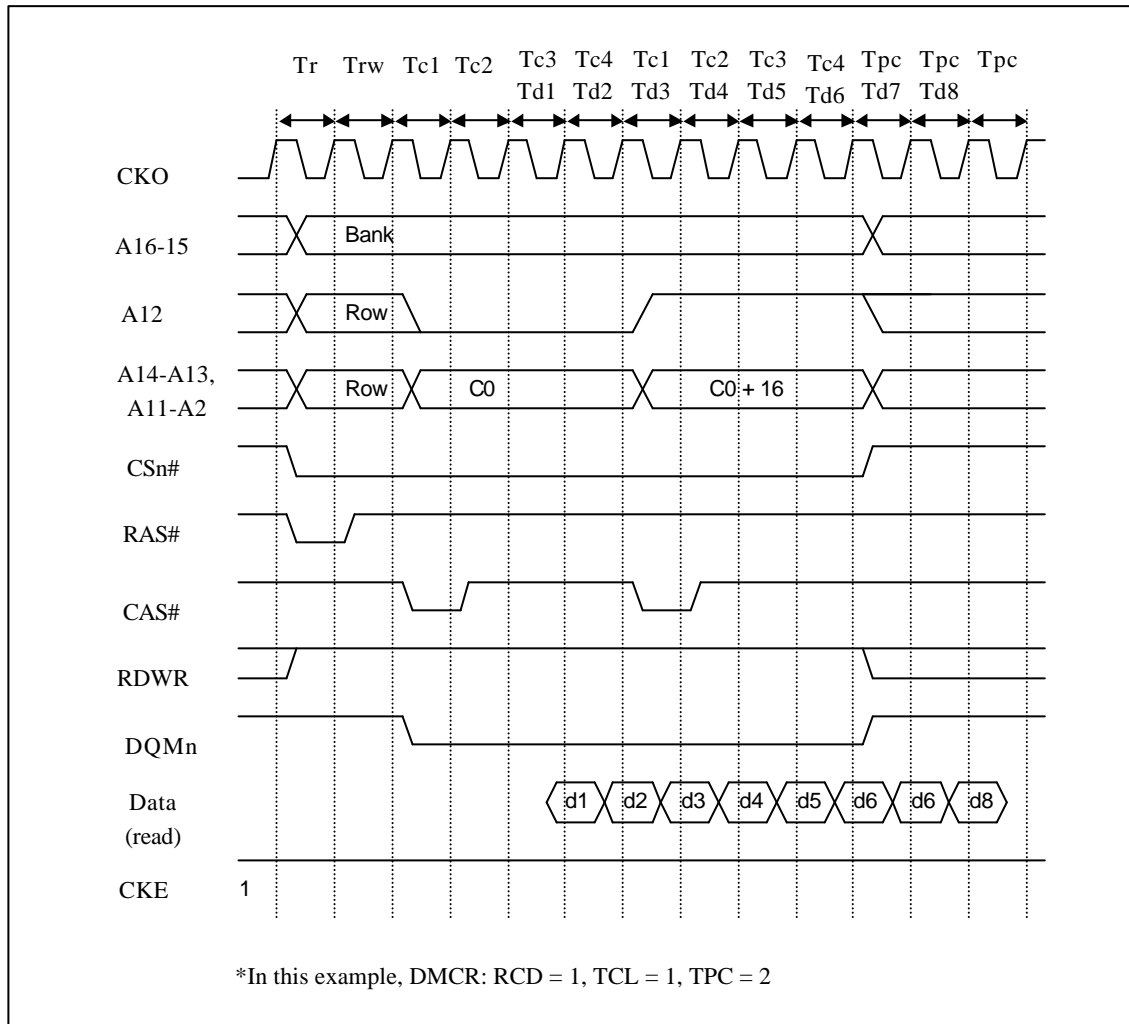


Figure 3-23 Synchronous DRAM 32-Byte Burst Read Timing (Non-Page Mode)

Single Read: Following figure shows the timing when a single data read is performed. With the Arca210, as synchronous DRAM is set to burst read/burst write mode, read data output continues after the required data has been read. To prevent data collisions, after the required data is read in Td1, empty read cycles Td2 to Td4 are performed.

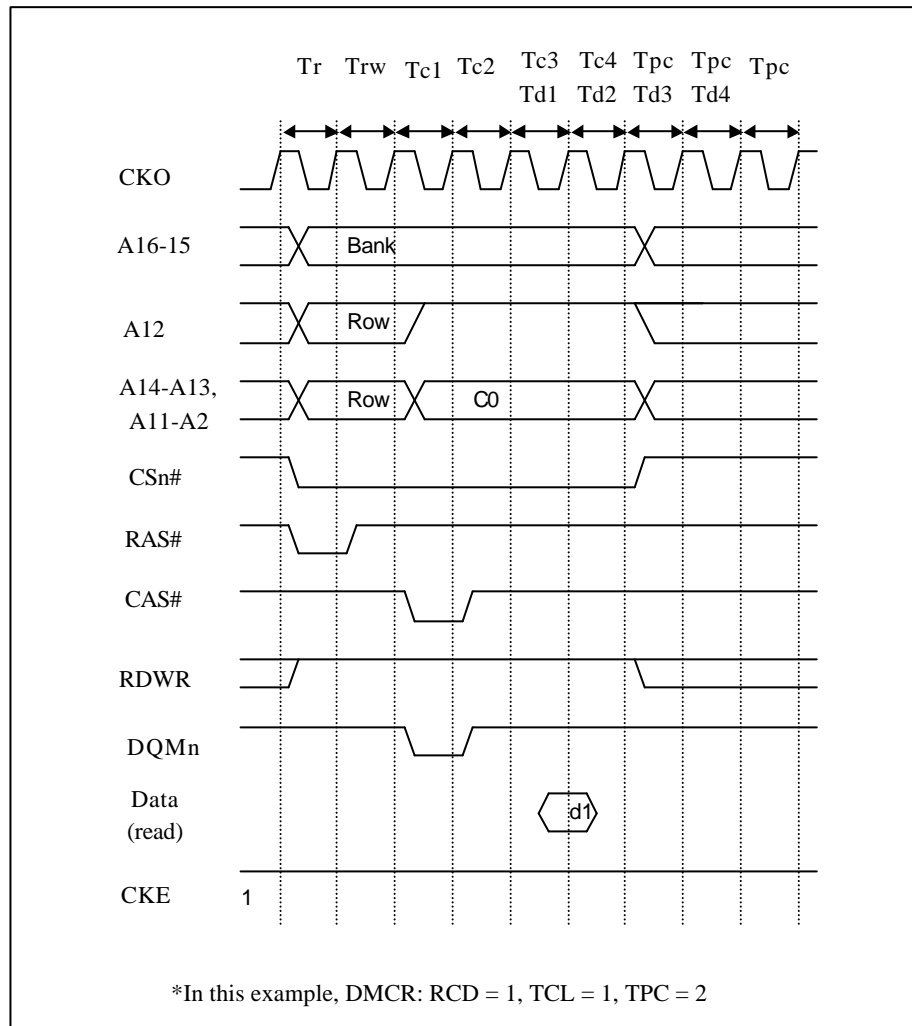


Figure 3-24 Synchronous DRAM Single Read Timing (Non-Page Mode)

16-Byte Burst Write: The timing chart for a burst write is shown in following figure. In the Arca210, a 16-byte burst write occurs in the event of cache copy-back or a 16-byte transfer of DMAC. In a 16-byte burst write operation, following the T_r cycle in which ACTV command output is performed, a WRITEEA command that performs auto-pre-charge is issued in the T_{c1} cycle. In the write cycle, the write data is output at the same time as the write command. In the case of the write with auto-pre-charge command, pre-charging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until pre-charging is completed. Consequently, in addition to the pre-charge wait cycle, T_{pc} , used in a read access, cycle $Trwl$ is also added as a wait interval until pre-charging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of $Trwl$ cycles can be specified by the $TRWL$ bit in $DMCR$. 32-byte boundary data is written in wraparound mode.

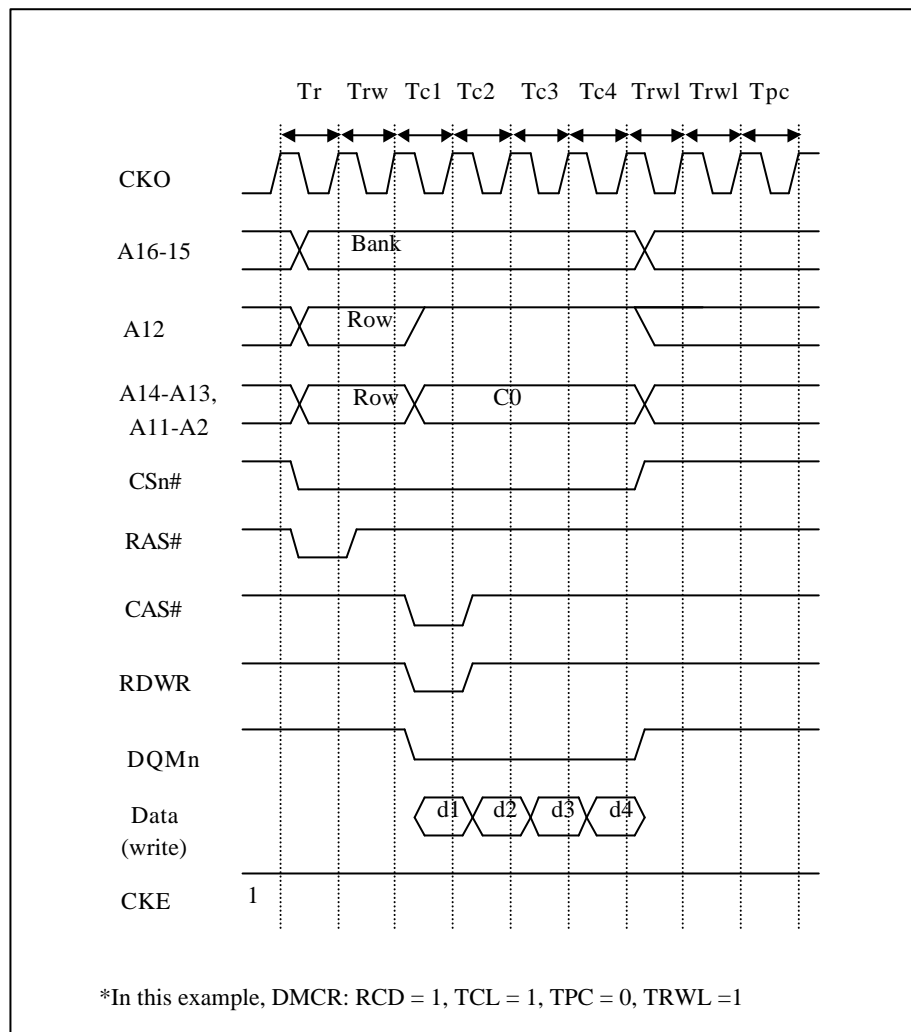


Figure 3-25 Synchronous DRAM 16-Byte Burst Write Timing (Non-Page Mode)

32-Byte Burst Write: 32-byte burst write occurs in the event of cache copy-back or a 32-byte transfer of DMAC. Following figure shows the timing of 32-byte burst write. The burst length of synchronous DRAM is 4. A WRITE command is issued for the first 16-byte data following ACTV command. After T_{c4} , a WRITEEA command is issued for next 16-byte data.

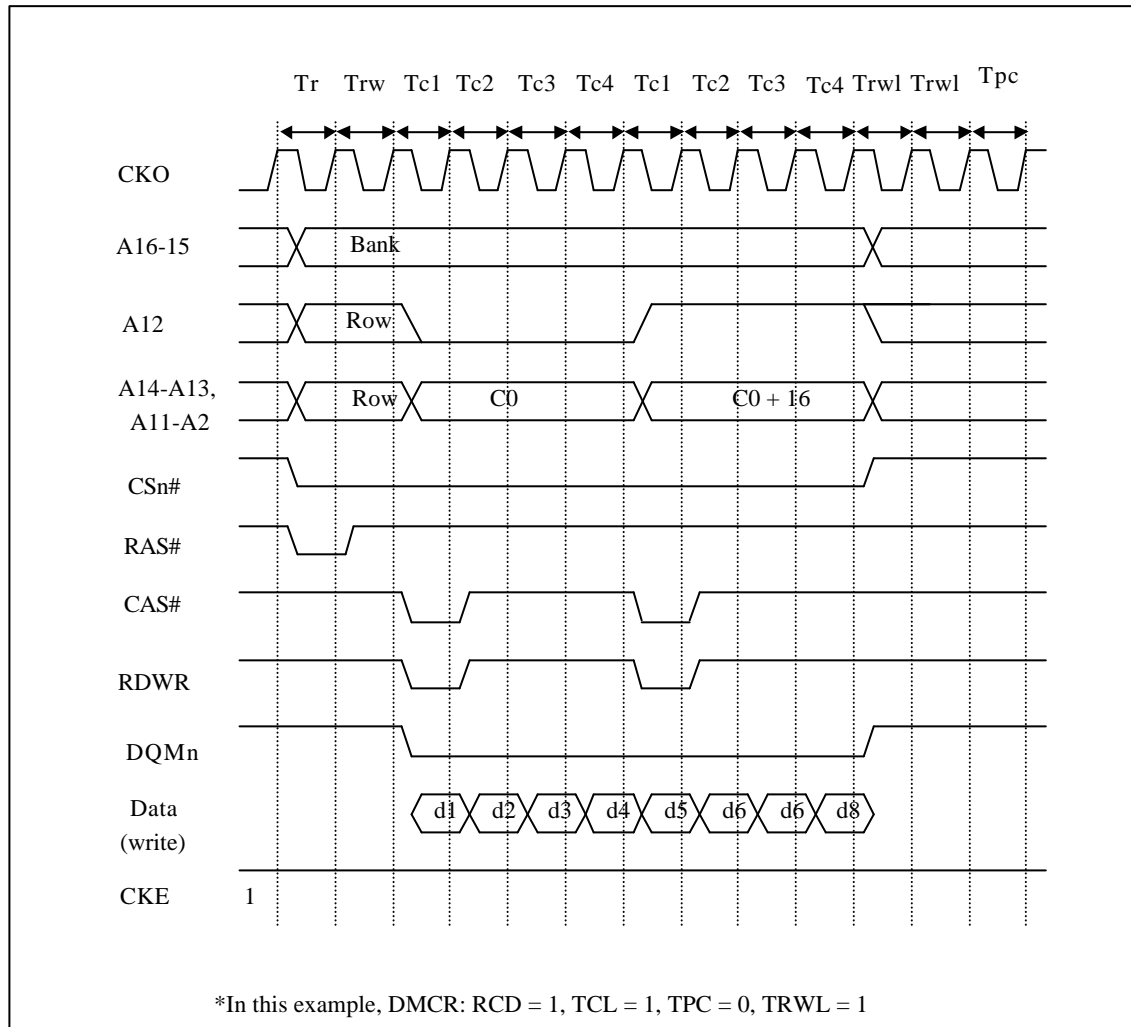


Figure 3-26 Synchronous DRAM 32-Byte Burst Write Timing (Non-Page Mode)

Single Write: The basic timing chart for write access is shown in following figure. In a single write operation, following the T_r cycle in which ACTV command output is performed, a WRITEA command that performs auto-pre-charge is issued in the T_{c1} cycle. In the write cycle, the write data is output at the same time as the write command. In the case of the write with auto-pre-charge command, pre-charging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until pre-charging is completed. Consequently, in addition to the pre-charge wait cycle, T_{pc} , used in a read access, cycle $Trwl$ is also added as a wait interval until pre-charging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of $Trwl$ cycles can be specified by the $TRWL$ bit in $DMCR$.

As the Arca210 supports burst read/burst write operations for synchronous DRAM, a single write requires the same number of cycles as a burst write.

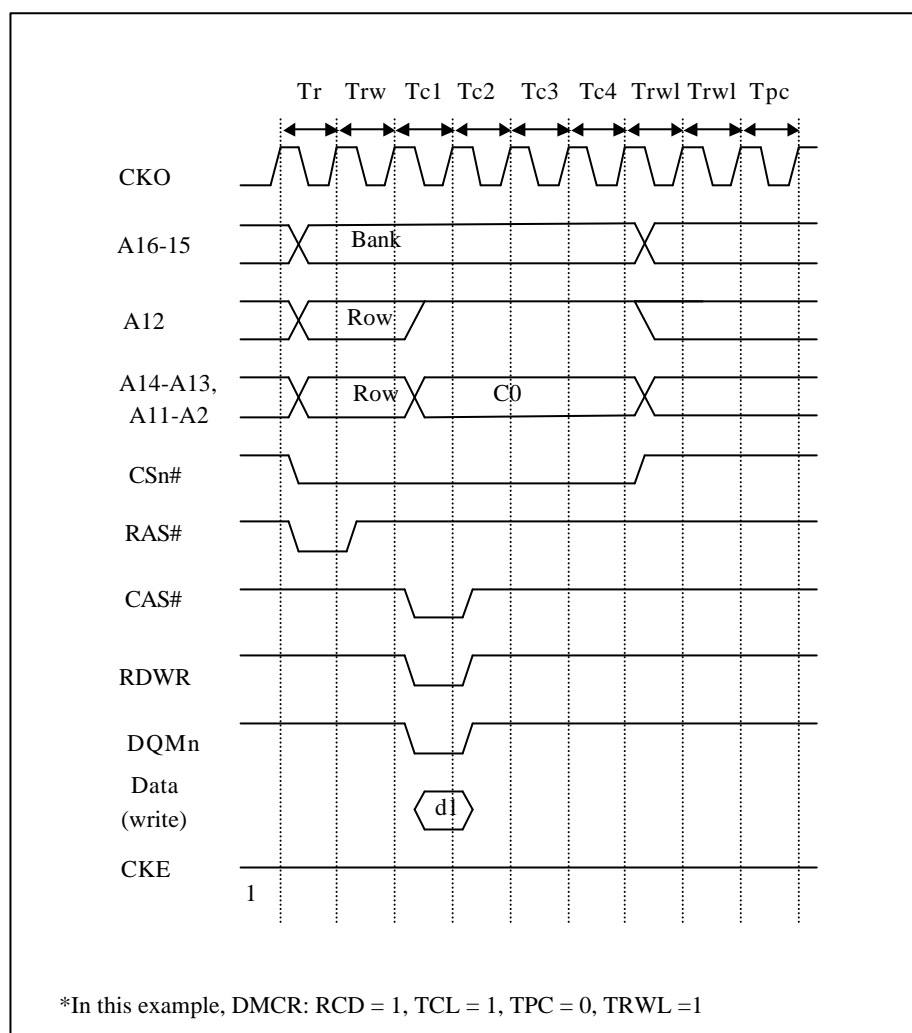


Figure 3-27 Synchronous DRAM Single Write Timing (Non-Page Mode)

3.7.4 Page Mode Timing

The synchronous DRAM bank function is used to support high-speed accesses to the same row address. When the PGM bit in DMCR is 1, read/write command accesses are performed using commands without auto-precharge (READ, WRIT). In this case, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As synchronous DRAM is internally divided into two or four banks, it is possible to activate one row address in each bank. If the next access is to a different bank or a different row address in the same bank, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued.

In a write, when auto-precharge is performed, a command cannot be issued for a period of $Trwl + Tpc$ cycles after write cycle. When page mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by $Trwl + Tpc$ cycles for each write.

There is a limit on $Tras$, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of $Tras$. In this way, it is possible to observe the restrictions on the maximum active state time for each bank. If auto-refresh is not used, measures must be taken in the program to ensure that the banks do not remain active for longer than the prescribed time.

When change from page mode to non-page mode, a PALL command must be issued. To issue PALL command, clear MRSET bit of DMCR to 0, then write mode register of all chip select.

Following figures show the timing of 16-byte burst access, 32-byte burst access and single access in page mode.

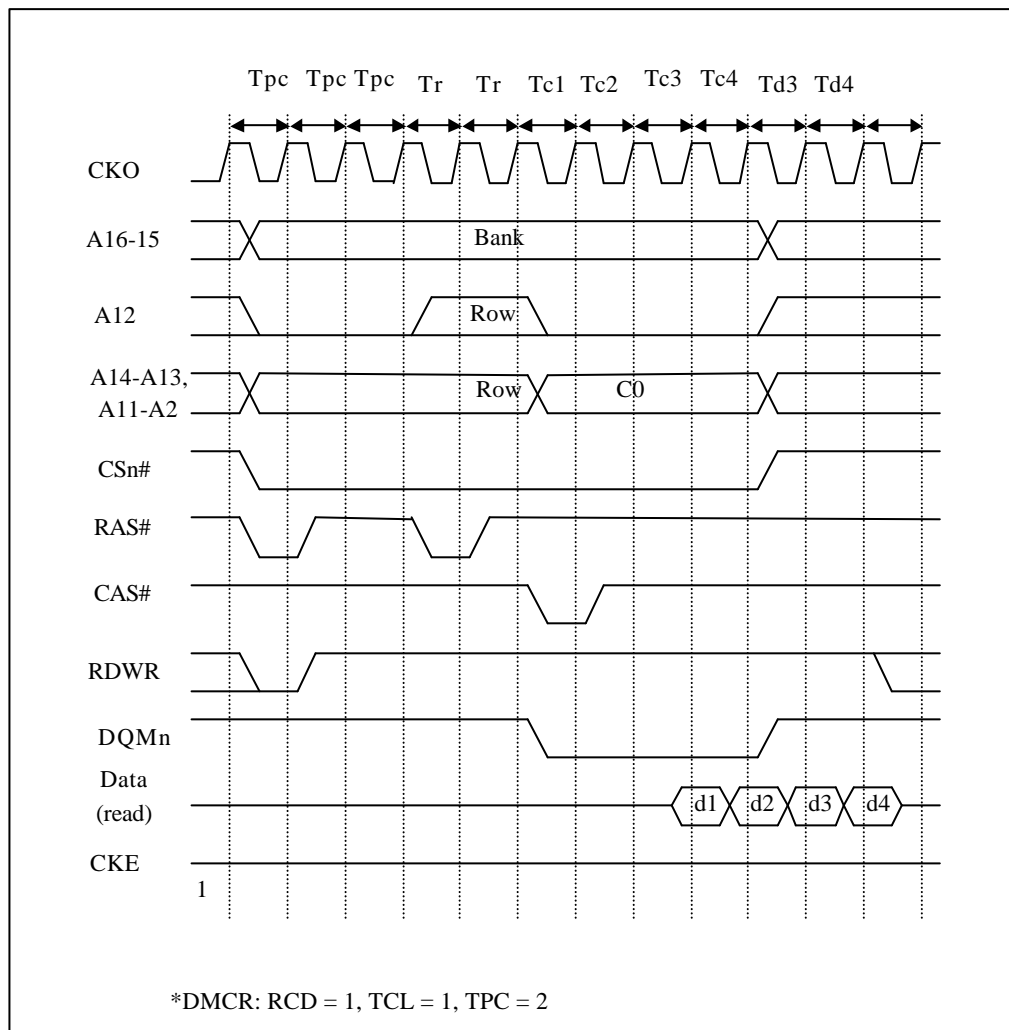


Figure 3-28 Synchronous DRAM 16-byte Burst Read Timing (Page Mode, Different Row)

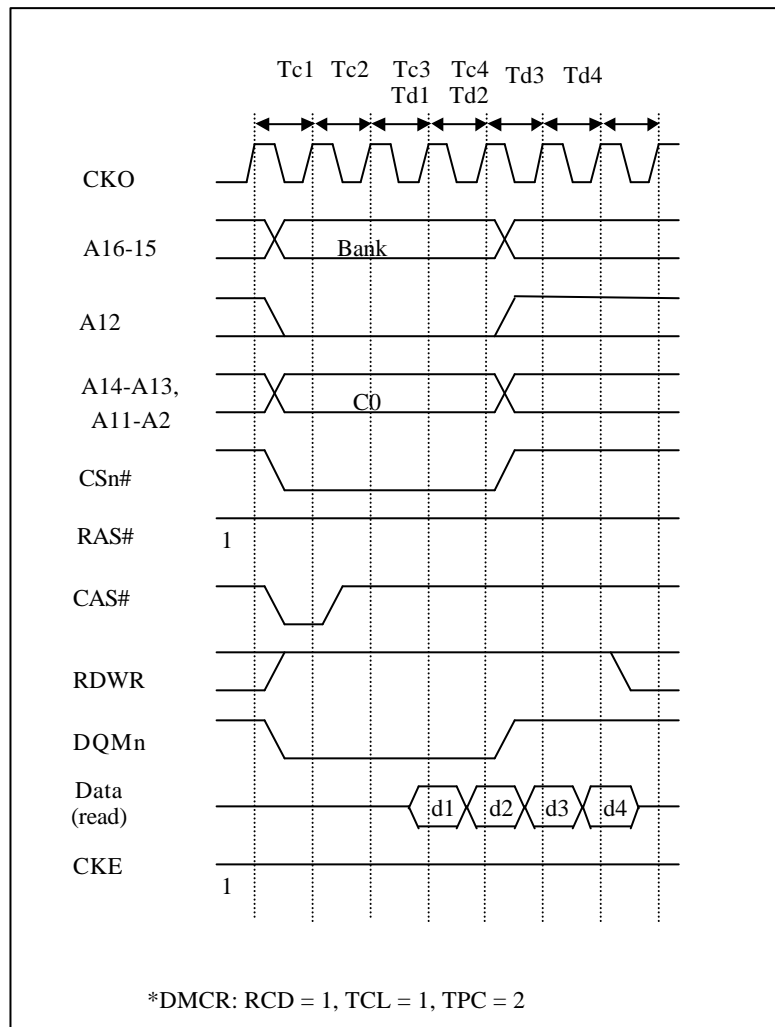


Figure 3-29 Synchronous DRAM 16-byte Burst Read Timing (Page mode, Same Row)

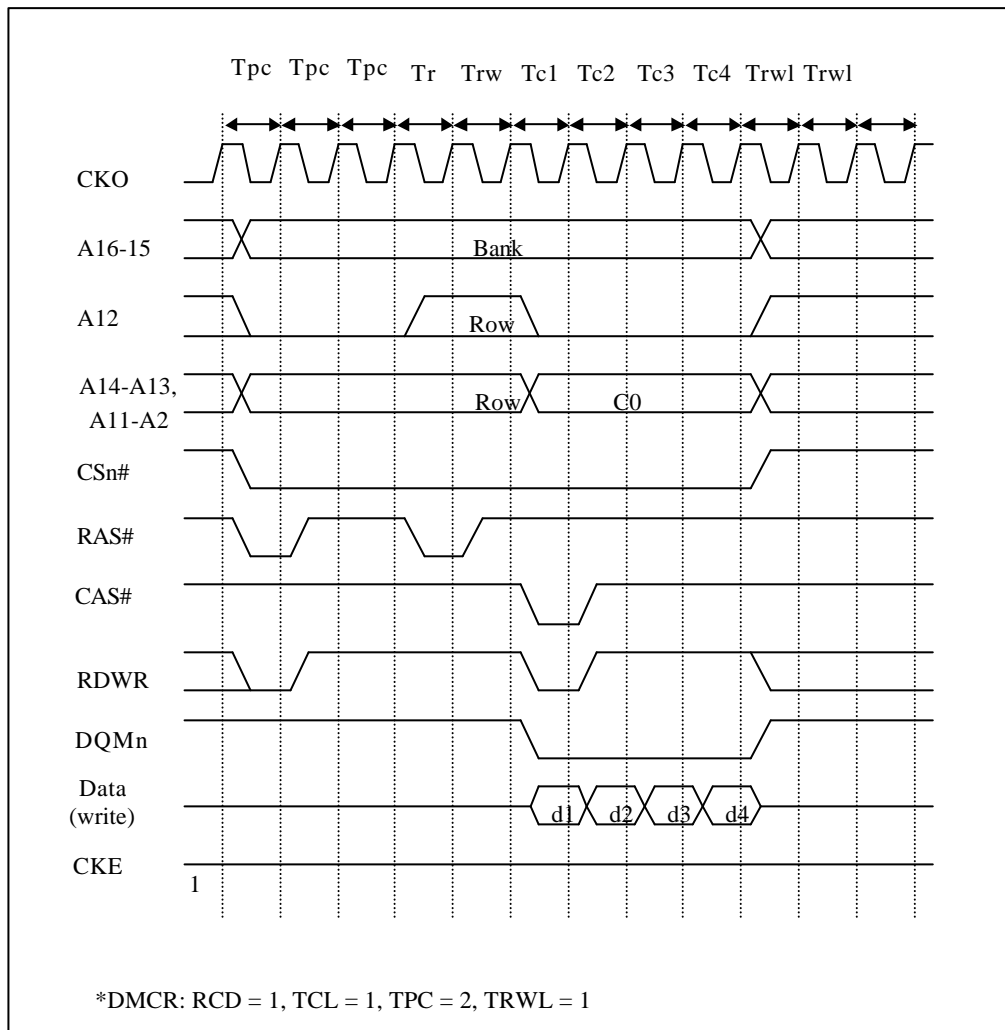


Figure 3-30 Synchronous DRAM 16-byte Burst Write Timing (Page Mode, Different Row)

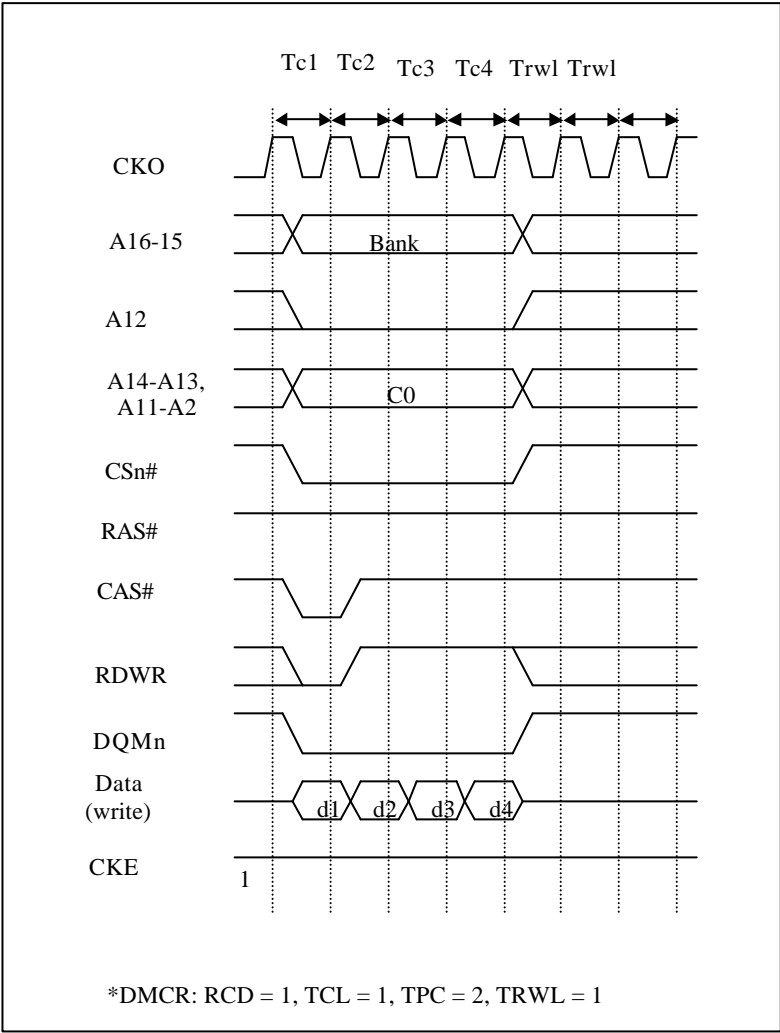


Figure 3-31 Synchronous DRAM 16-byte Burst Write Timing (Page Mode, Same Row)

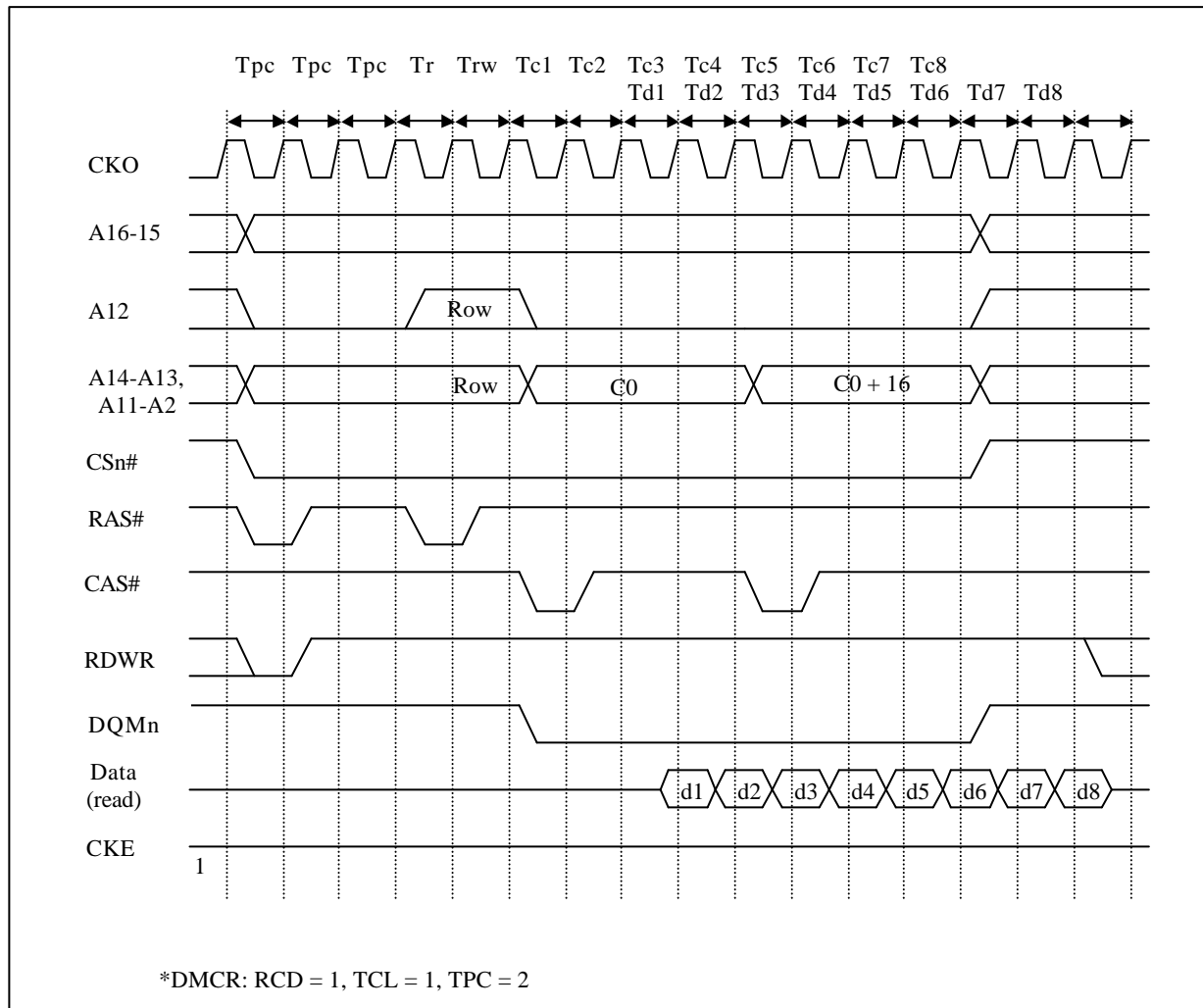


Figure 3-32 Synchronous DRAM 32-byte Burst Read Timing (Page Mode, Different Row)

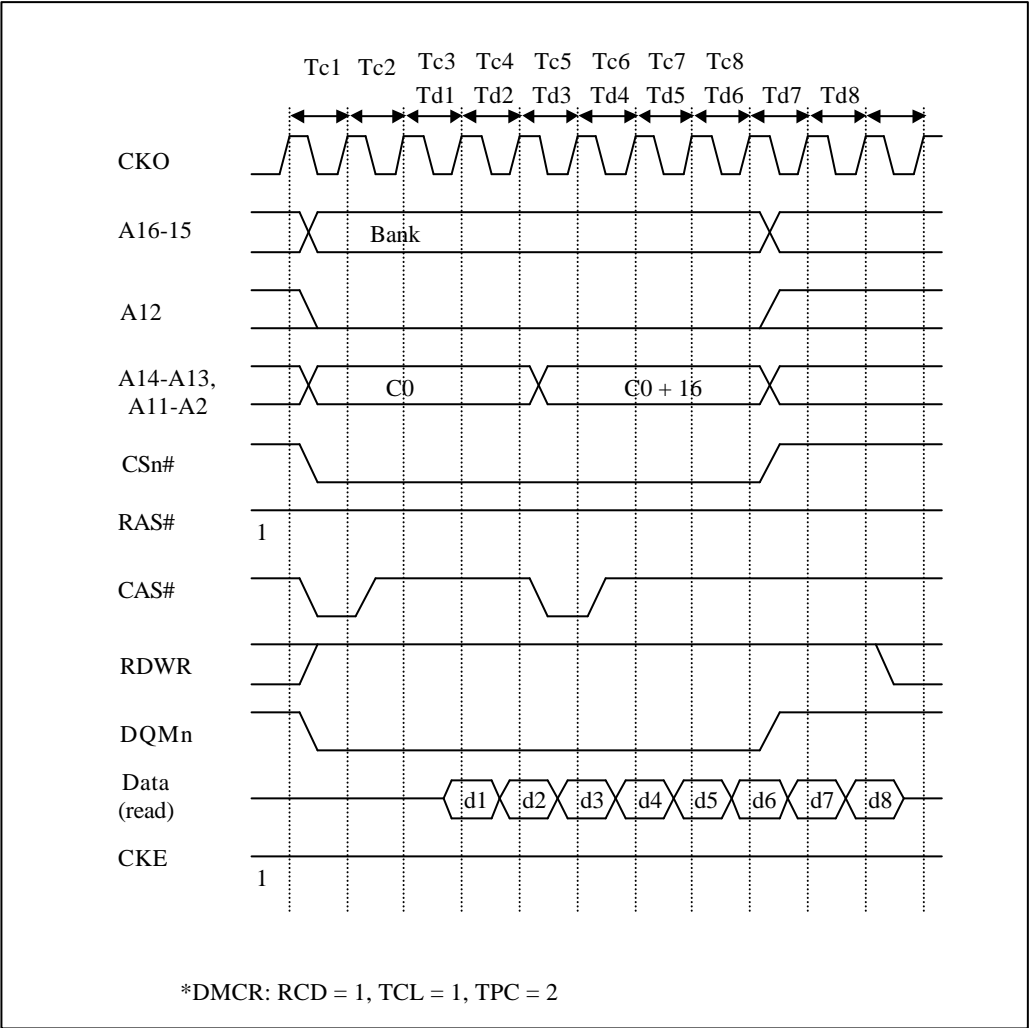


Figure 3-33 Synchronous DRAM 32-byte Burst Read Timing (Page Mode, Same Row)

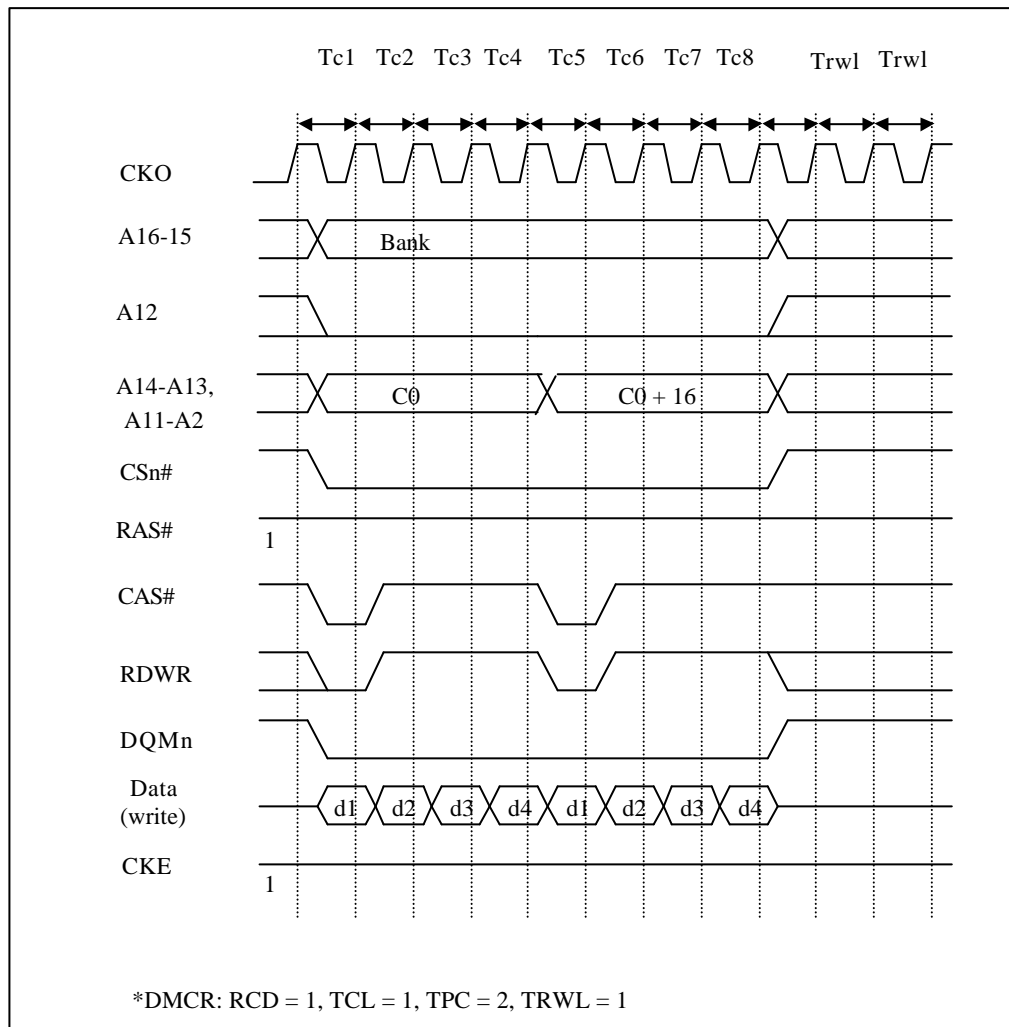


Figure 3-34 Synchronous DRAM 32-byte Burst Write Timing (Page Mode, Same Row)

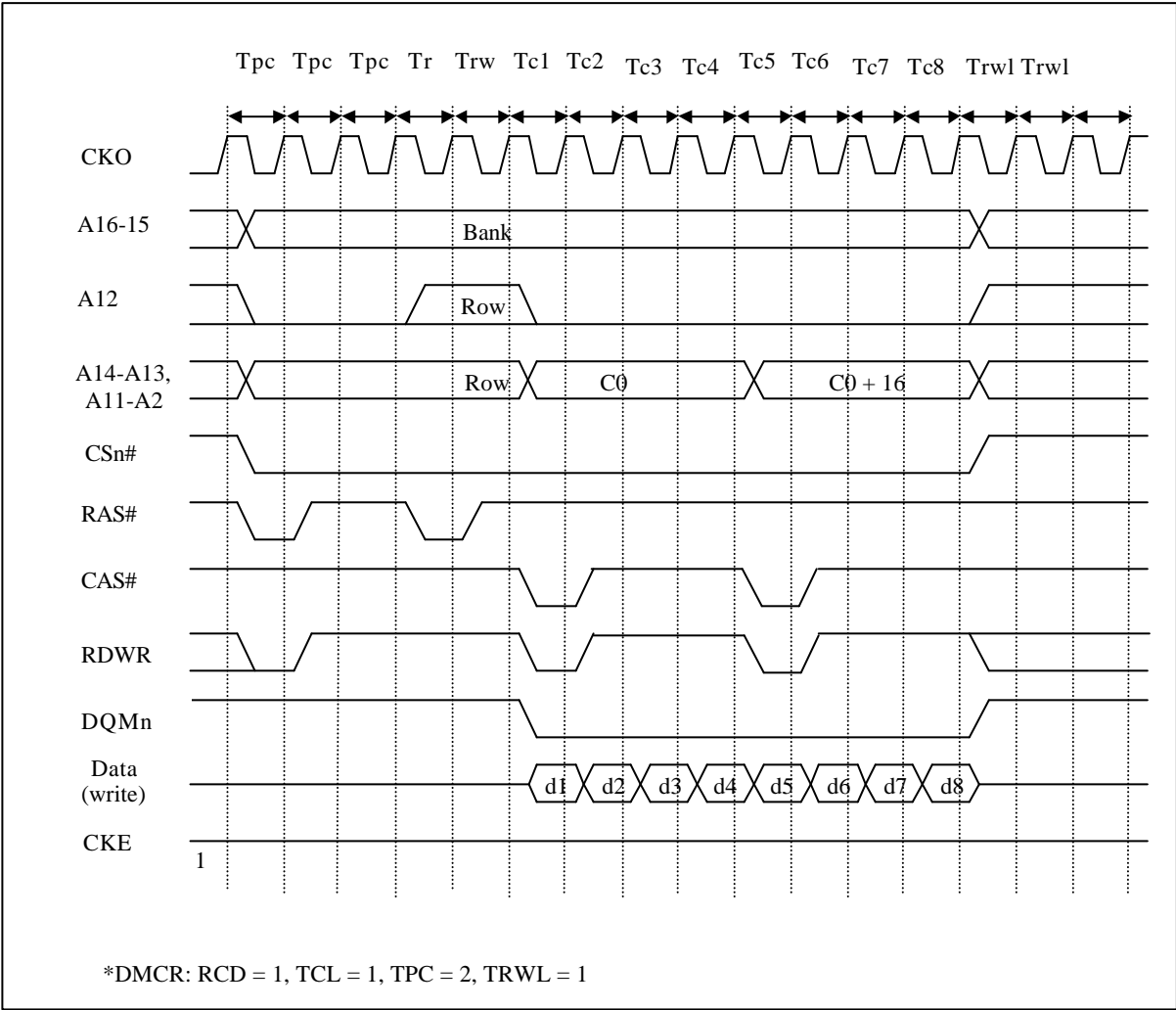


Figure 3-35 Synchronous DRAM 32-byte Burst Write Timing (Page Mode, Different Row)

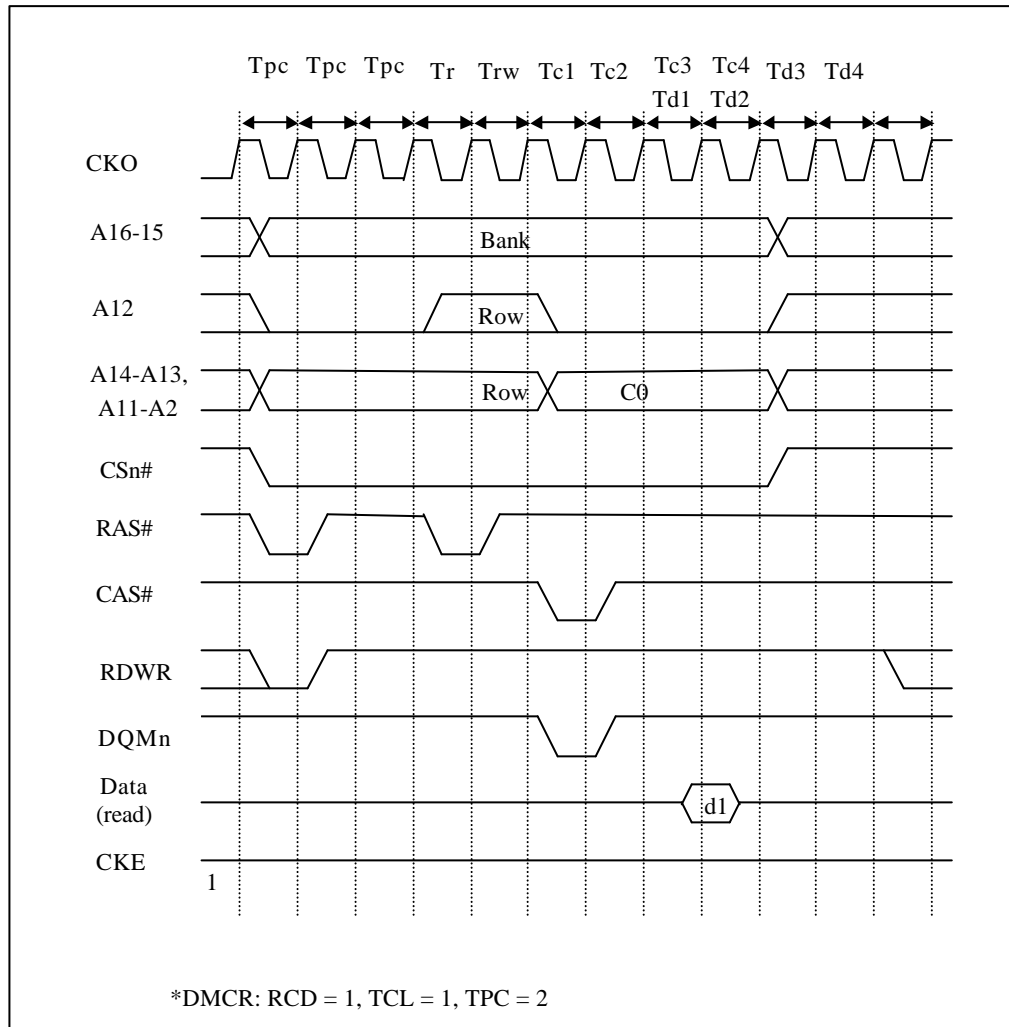


Figure 3-36 Synchronous DRAM Single Read Timing (Page Mode, Different Row)

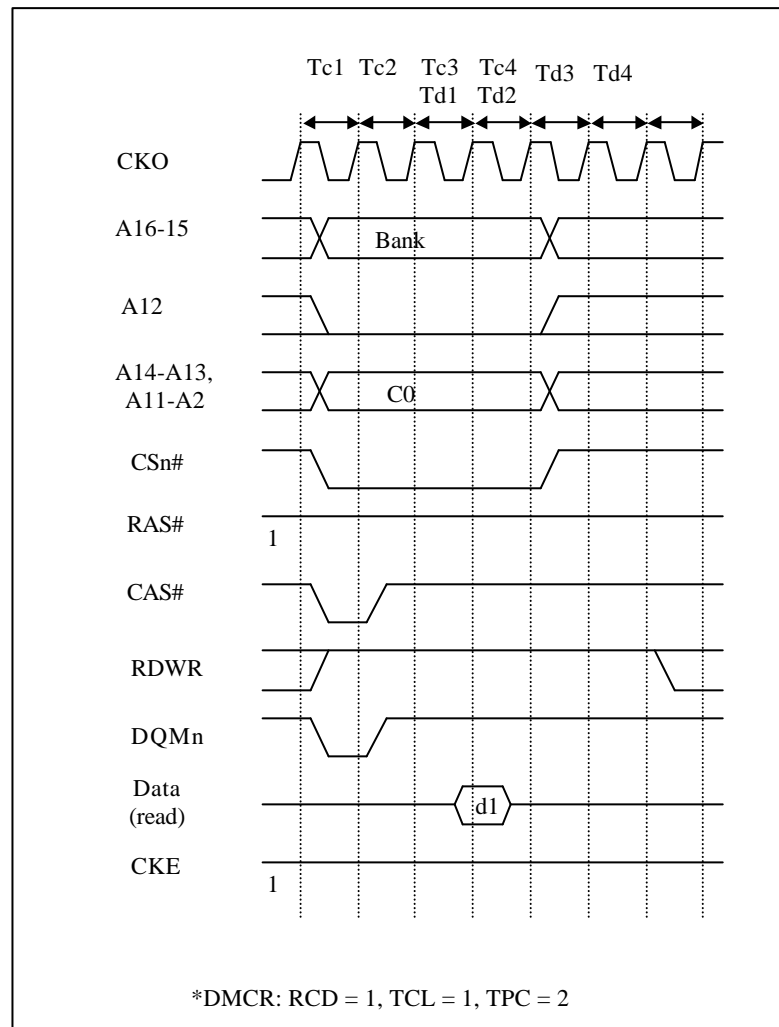


Figure 3-37 Synchronous DRAM Single Read Timing (Page mode, Same Row)

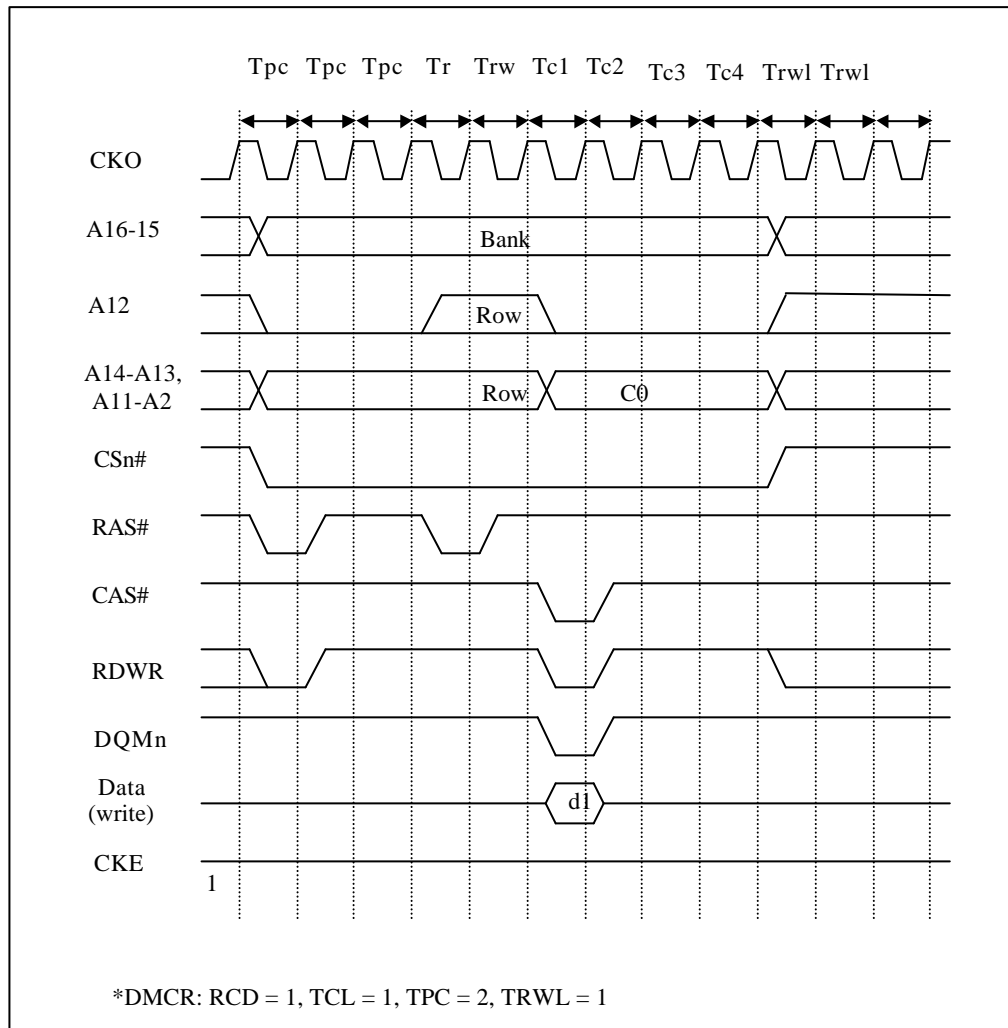


Figure 3-38 Synchronous DRAM Single Write Timing (Page Mode, Different Row)

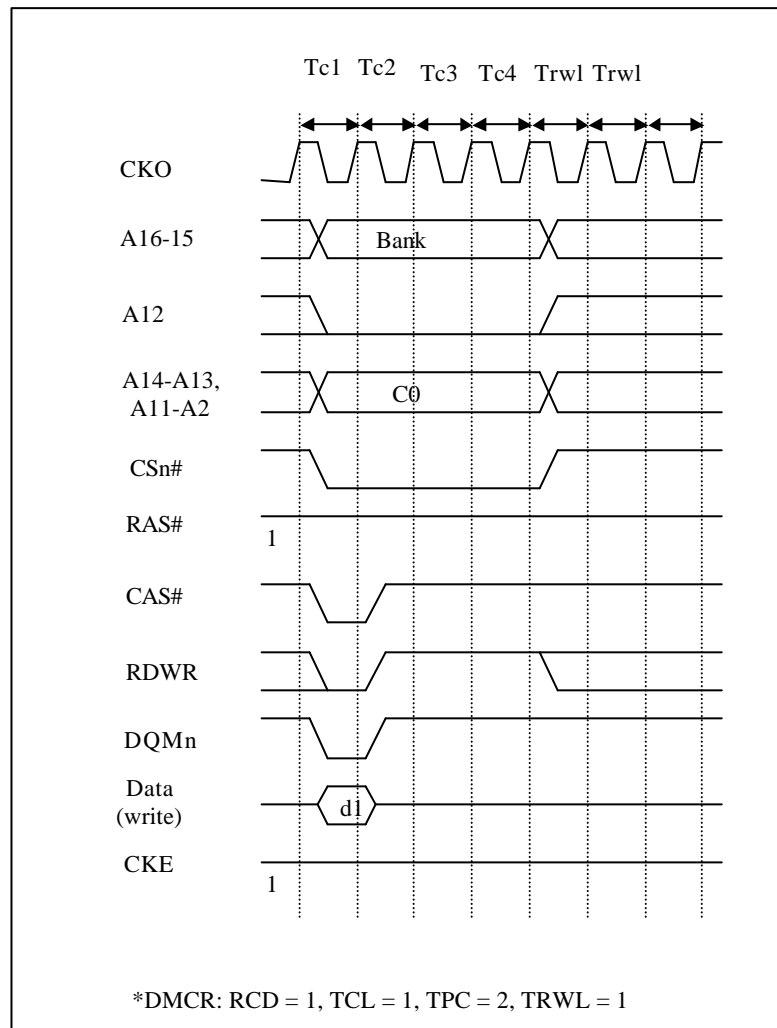


Figure 3-39 Synchronous DRAM Single Write Timing (Page Mode, Same Row)

3.7.5 Power-Down Mode

The SDRAM power-down mode is supported to minimize the power consumption. CKE going to low level when SDRAM is idle/active state will drive SDRAM to precharge/active power-down mode. The clock supplies to SDRAM may be stopped also when CKE keep in low level more than two cycles. When a new access starts or a refresh requests, CKE is driven to high level and clock supplies are re-enabled.

The following figure shows the timing of power-down mode and clock stopping.

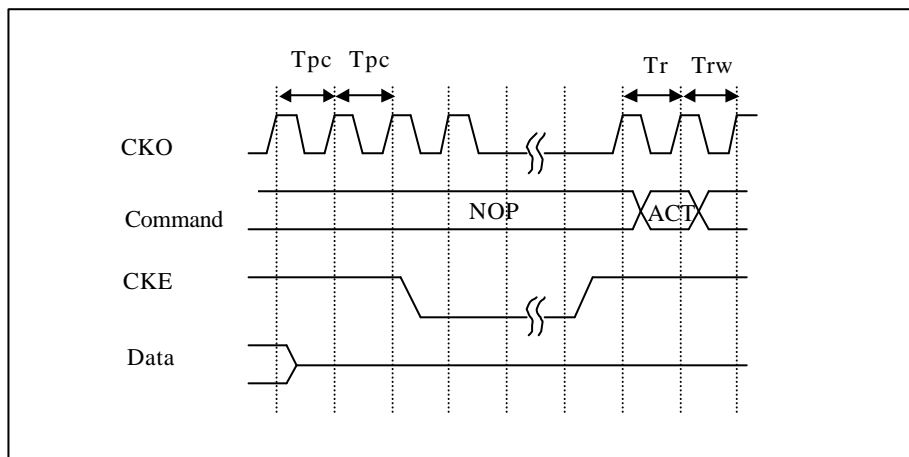


Figure 3-40 SDRAM Power-Down Mode Timing (CKO Stopped)

The following figure shows the power-down mode timing that CKE low level less than two cycles and clock is not stopped.

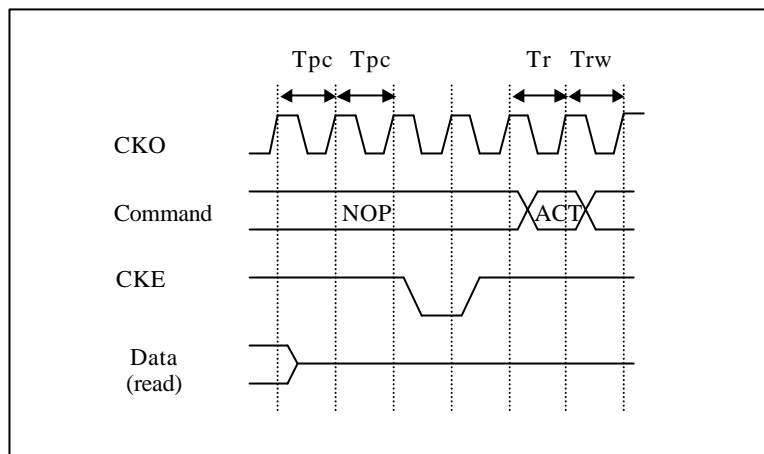


Figure 3-41 SDRAM Power-Down Mode Timing (Clock Supplied)

3.7.6 Refreshing

EMI provides a function for controlling the refresh of synchronous DRAM. Auto-refresh can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in DMCR. If synchronous DRAM is not accessed for a long period, self-refresh mode can be activated by set both the RMODE bit and the RFSH bit to 1.

- AUTO-REFRESH

Refreshing is performed at intervals determined by the input clock selected by bits CKS2-0 in RTCSR, and the value set in RTCOR. The value of bits CKS2-0 in RTCSR should be set so as to satisfy the refresh interval stipulation for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, then make the CKS2-CKS0 setting. When the clock is selected by CKS2-CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed. At the same time, RTCNT is cleared to zero and the count-up is restarted. Figure 3-42 shows the auto-refresh cycle operation.

In non-page mode, first, a REF command is issued in the TRr cycle. After the TRr cycle, new command output cannot be performed for the duration of the number of cycles specified by the TRC bits in DMCR. The TRC bits must be set so as to satisfy the synchronous DRAM refresh cycle time stipulation (active/active command delay time). Figure 3-43 shows the auto-refresh timing when TRC is set to 2.

Auto-refresh is performed in normal operation, in sleep mode, and in the case of a manual reset.

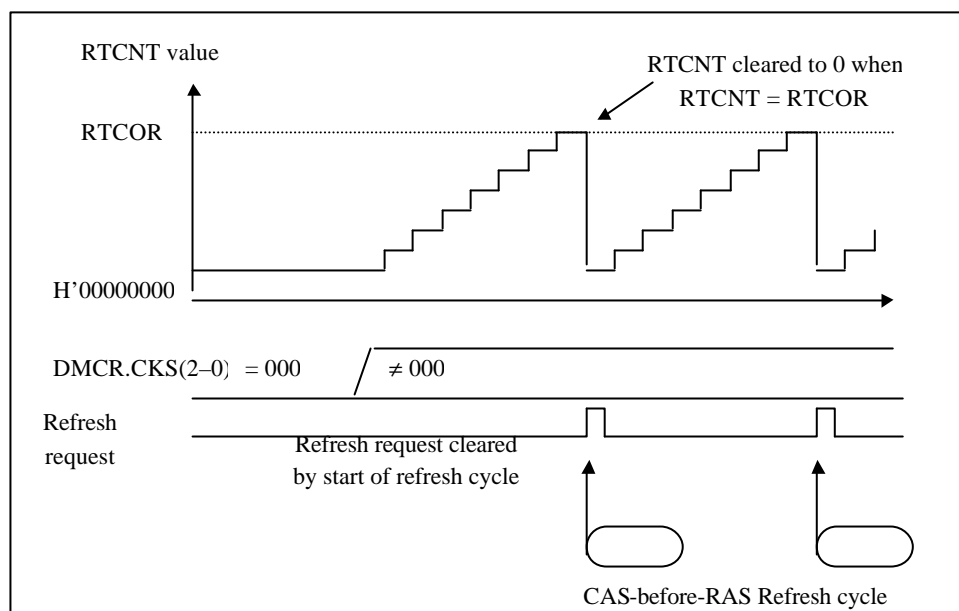


Figure 3-42 Synchronous DRAM Auto-Refresh Operation

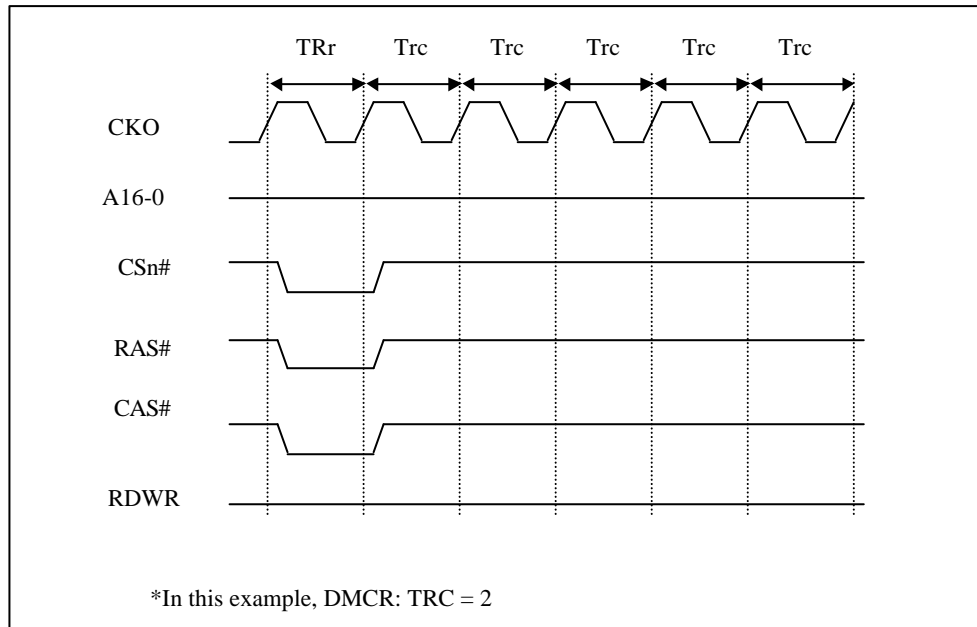


Figure 3-43 Synchronous DRAM Auto-Refresh Timing (Non-page mode)

In page mode, a PALL command is issued firstly to precharge all banks. Then a REF command is issued in the TRr cycle.

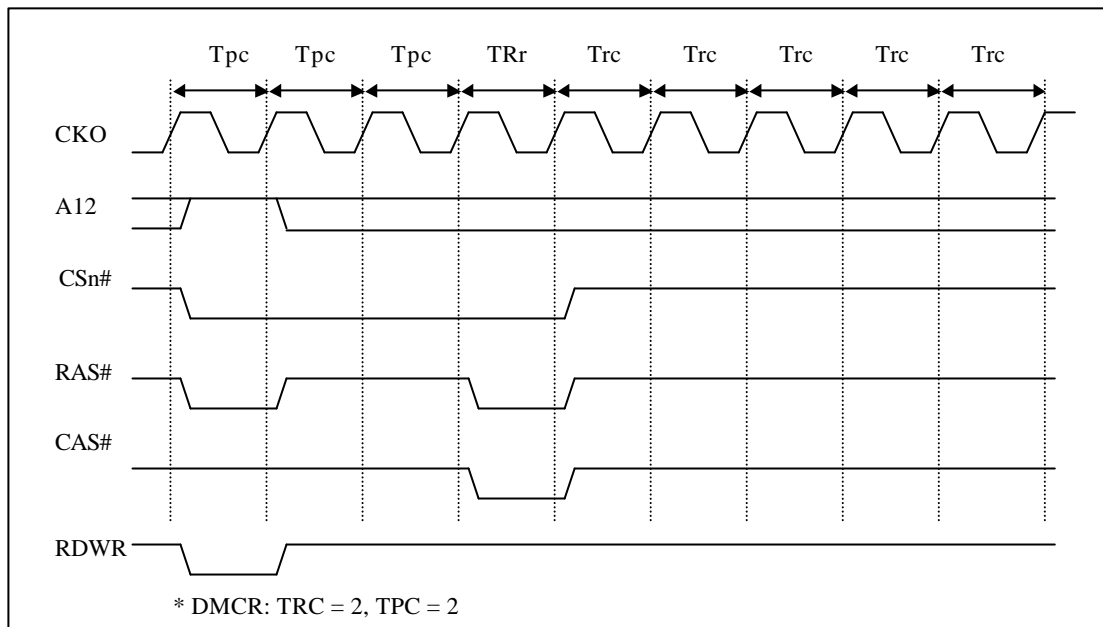


Figure 3-44 Synchronous DRAM Auto-Refresh Timing (Page Mode)

- SELF-Refresh

Self-refresh mode is a kind of standby mode in which the refresh timing and refresh addresses are generated within the synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit to 1. The self-refresh state maintaining while the CKE signal is low. Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh mode is clear by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the TRC bits in DMCR. Trsw cycles are inserted to meet the minimum CKE negation time specified by the TRAS bits in DMCR. Self-refresh timing is shown in Figure 3-45. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refresh is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if RFSH is set to 1 and RMODE is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refresh takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately. After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the Arca210's standby function, and is maintained even after recovery from standby mode other than through a power-on reset. In the case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in the case of a manual reset. In standby mode, if RFSH bit in DMCR is 1, self-refresh is always performed in spite of RMODE field in DMCR until standby mode is canceled.

Relationship between Refresh Requests and Bus Cycle Requests:

If a refresh request is generated during execution of a bus cycle, execution of the refresh is deferred until the bus cycle is completed. If a match between RTCNT and RTCOR occurs while a refresh is waiting to be executed, so that a new Refresh request is generated, the previous refresh request is eliminated. In order for refreshing to be performed normally, care must be taken to ensure that no bus cycle is longer than the refresh interval.

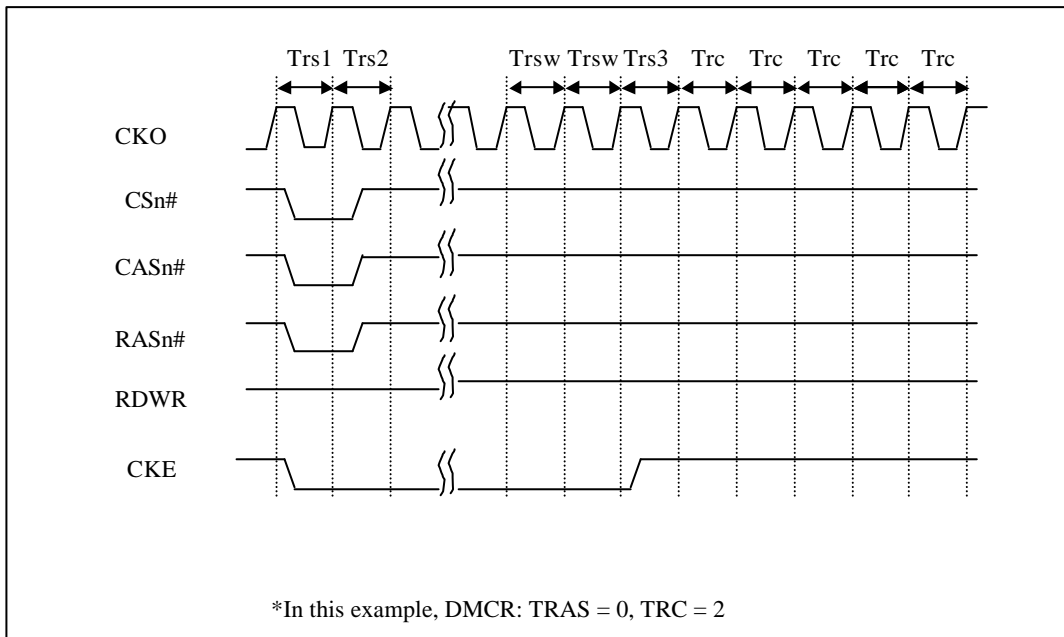


Figure 3-45 Synchronous DRAM Self-Refresh Timing

In page mode, a PALL command is issued firstly to precharge all banks.

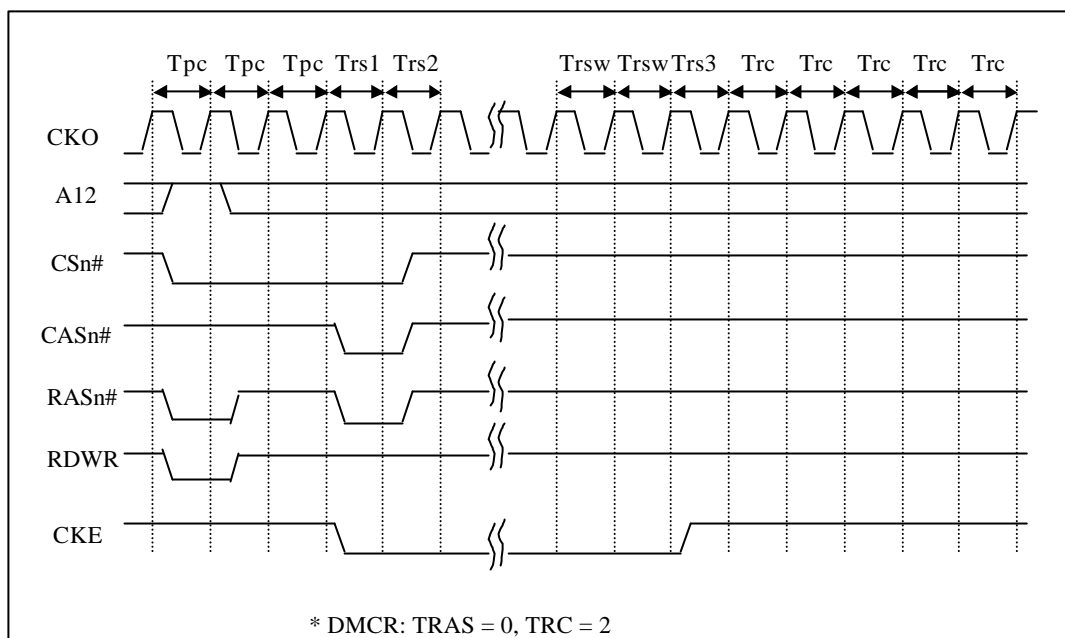


Figure 3-46 Synchronous DRAM Self-Refresh Timing (Page Mode)

3.7.7 Power-On Sequence

In order to use synchronous DRAM, mode setting must first be performed after powering on. To perform synchronous DRAM initialization correctly, the external memory interface registers must first be set, followed by a write to the synchronous DRAM mode register.

In synchronous DRAM mode register setting, the address signal value at that time is latched by MRS command. If the value to be set is X , the bus state controller provides for value X to be written to the synchronous DRAM mode register by performing a write to address H'E102A000 + X for bank 4, address H'E102B000 + X for bank 5, address H'E102C000 + X for bank 6, and address H'E102D000 + X for bank 7. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/write, CAS latency 2 to 3, wrap type sequential, and burst length 4 supported by the Arca210, arbitrary data is written in a byte-size access to the following addresses.

	Bank 4	Bank 5	Bank 6	Bank 7
CAS latency 2	E102A088	E102B088	E102C088	E102D088
CAS latency 3	E102A0C8	E102B0C8	E102C0C8	E102D0C8

The value set in DMCR.MRSET is used to select whether a Pre-charge All Banks command (PALL) or a Mode Register Set command (MRS) is issued. The timing for the Pre-charge All Banks command is shown in Figure 3-47, and the timing for the Mode Register Set command in Figure 3-48.

Before mode register setting, a 200 μ s idle time (depending on the memory manufacturer) must be guaranteed after powering on requested by the synchronous DRAM. If the reset signal pulse width is greater than this idle time, there is no problem in performing power-on sequence immediately.

- First, a pre-charge all bank (PALL) command must be issued by performing a write to address H'E102A000 + X for bank 4, H'E102B000 + X for bank 5, H'E102C000 + X for bank 6 and H'E102D000 + X for bank 7 while DMCR.MRSET = 0.
- Next the number of dummy auto-refresh cycles specified by the manufacturer (usually 8) or more must be executed. This is usually achieved automatically while various kinds of initialization are being performed after auto-refresh setting, but a way of carrying this out more dependably is to set a short refresh request generation interval just while these dummy cycles are being executed. With simple read or write access, the address counter in the synchronous DRAM used for auto-refreshing is not initialized, and so the cycle must always be an auto-refresh cycle.
- After auto-refresh has been executed at least the prescribed number of times, a Mode Register Set command (MRS) is issued in the TMw1 cycle by setting DMCR.MRSET to 1 and performing a write to address H'E102A000 + X , H'E102B000 + X , H'E102C000 + X and H'E102D000 + X .

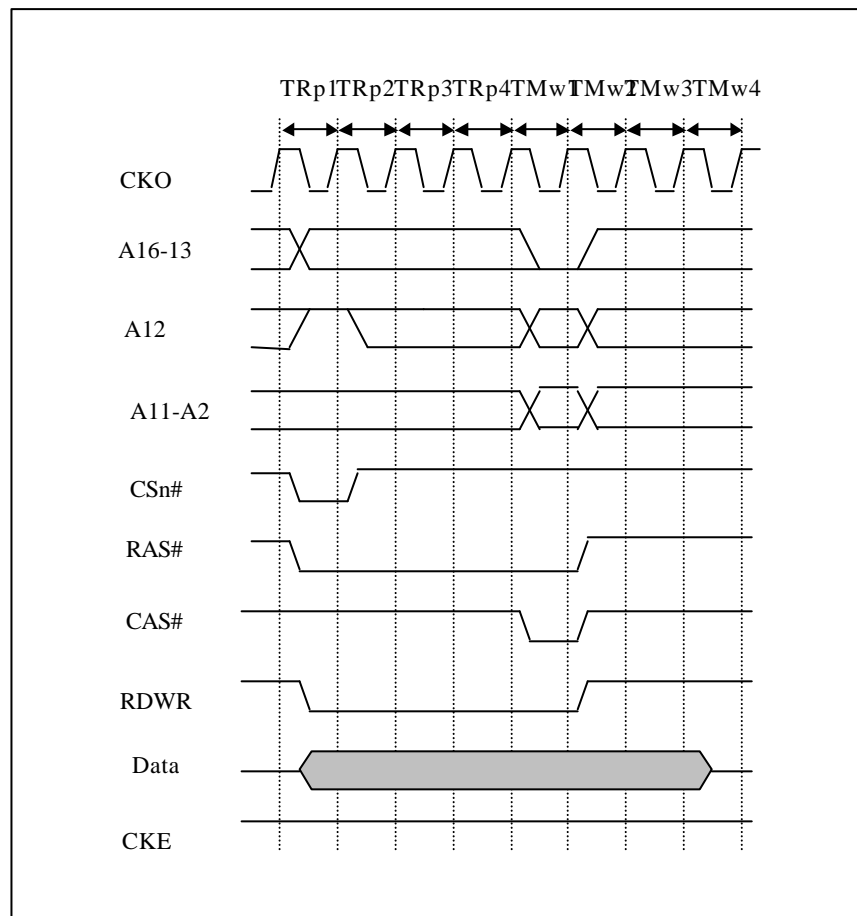


Figure 3-47 SDRAM Mode Register Write Timing 1 (Pre-charge All Banks)

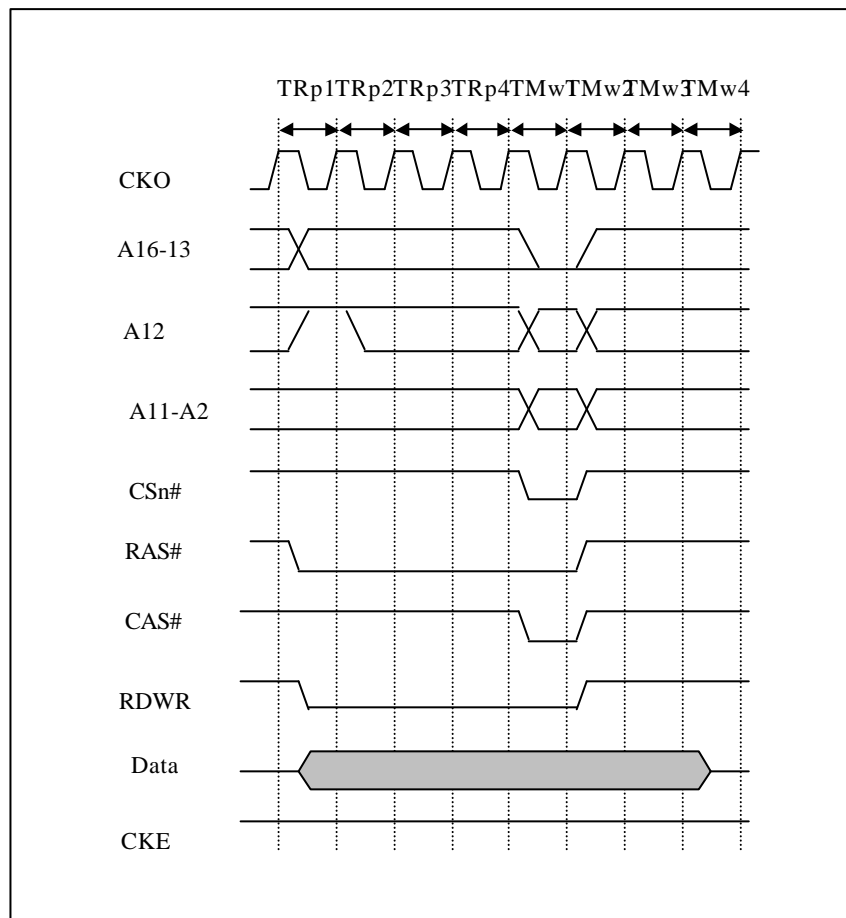


Figure 3-48 SDRAM Mode Register Write Timing 2 (Mode Register Set)

4 Direct Memory Access

4.1 Overview

The DMA controller (DMA) controls data transfers between external memory, external devices with DACK, memory-mapped external devices and on-chip modules (TMU, UART, etc) in place of CPU to relieve it of the interrupt overhead in serving those data transfers.

4.1.1 Features

The DMA has the following features:

- Four independent DMA channels
- Maximum transfer count: $16M - 1$ (16,777,215) transfers
- Transfer data units: 8-bit, 16-bit, 32-bit, 16-byte or 32-byte
- One transfer address type: two-address transfer
- Two transfer modes: single mode or block mode
- Two external DMA request detection modes: low level or falling edge detection
- Three types of transfer requests:
 - External request from DREQ0 to DREQ2 which can be accepted only on channel 0, 1 and 2.
 - On-chip peripheral modules DMA request from TMU, UART, IrDA/UART2 and AC97 that can be accepted on all channels.
 - Auto-request within DMA.
- Transfer objects
 - Between external memory and external device with DACK
 - Between external memory and external memory or memory-mapped devices
 - Between external memory and on-chip modules
- Channel functions:
 - Channel 0: External requests are accepted.
 - Channel 1: External requests are accepted.
 - Channel 2: External requests are accepted.
 - Channel 3: External requests are not accepted.
- Two types of channel priority ranking: fixed priority or round robin mode.
- Interrupt request can be sent to the CPU on completion of the specified number of transfers or on address error during transfer.

4.1.2 Block Diagram

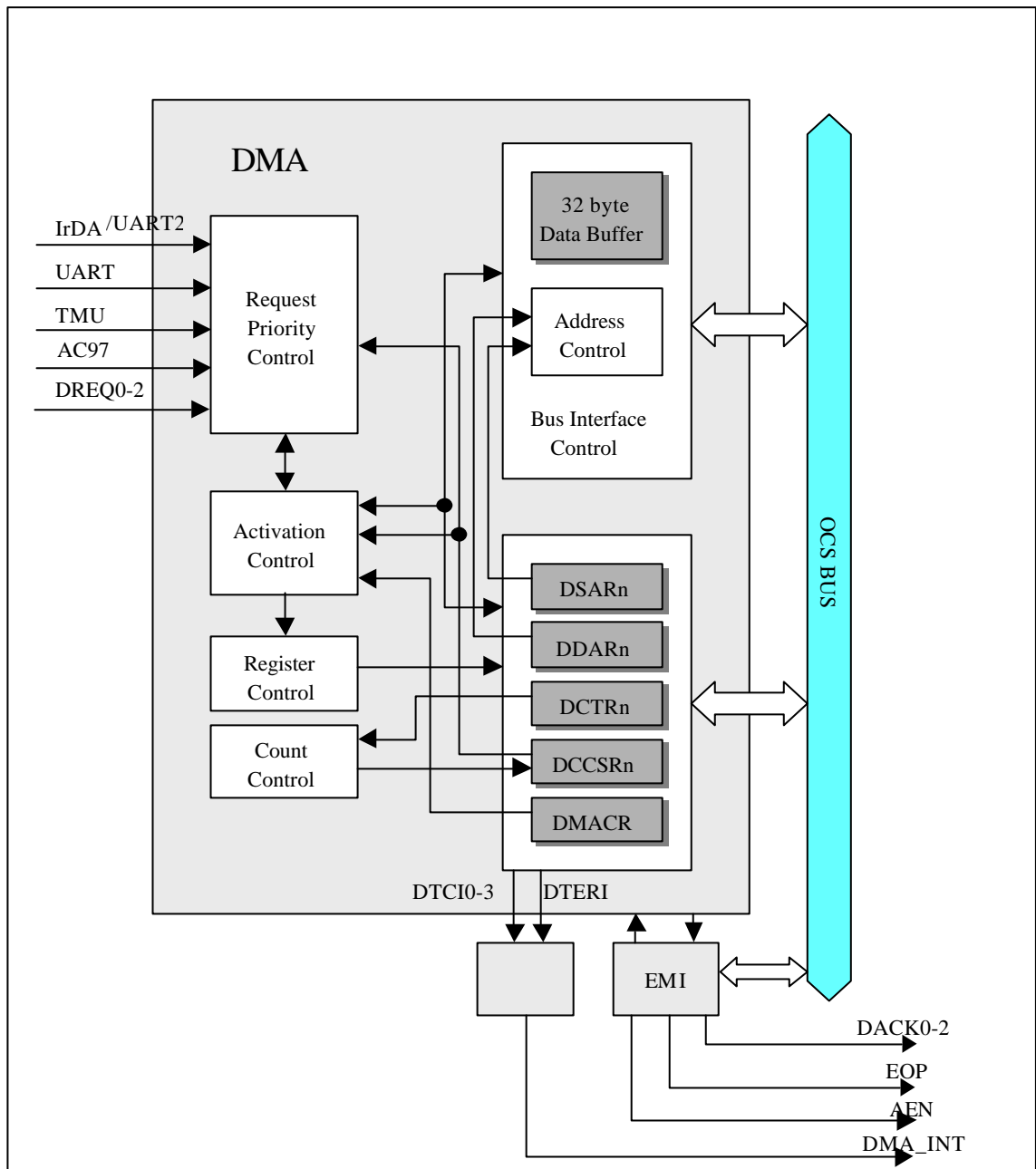


Figure 4-1 DMA Block Diagram

4.2 Pin Configuration

Table 4-1 DMA Pins

Name	Full Name	I/O	Function
DREQ0	DMA Transfer Request 0	I	Channel 0 External DMA Request
DREQ1	DMA Transfer Request 1	I	Channel 1 External DMA Request
DREQ2	DMA Transfer Request 2	I	Channel 2 External DMA Request
DACK0	DMA Transfer Acknowledge 0	O	Channel 0 External DMA Acknowledge
DACK1	DMA Transfer Acknowledge 1	O	Channel 1 External DMA Acknowledge
DACK2	DMA Transfer Acknowledge 2	O	Channel 2 External DMA Acknowledge
EOP	End of Process	O	Channel 0-2 External DMA Transfer End
AEN	Address Enable	O	Active High to Enable External DMA Transfer

4.3 Registers Configuration

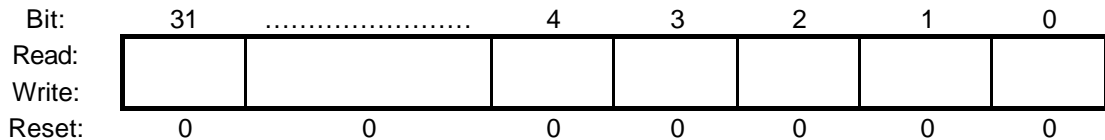
Table 4-2 DMA Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
DSAR0	DMA Source Address Register 0	R/W	H'00000000	H'E1030000	32
DDAR0	DMA Destination Address Register 0	R/W	H'00000000	H'E1030004	32
DCTR0	DMA Transfer Counter Register 0	R/W	H'00000000	H'E1030008	32
DCCSR0	DMA Channel Control/Status Register 0	R/W	H'00000000	H'E103000C	32
DSAR1	DMA Source Address Register 1	R/W	H'00000000	H'E1030010	32
DDAR1	DMA Destination Address Register 1	R/W	H'00000000	H'E1030014	32
DCTR1	DMA Transfer Counter Register 1	R/W	H'00000000	H'E1030018	32
DCCSR1	DMA Channel Control/Status Register 1	R/W	H'00000000	H'E103001C	32
DSAR2	DMA Source Address Register 2	R/W	H'00000000	H'E1030020	32
DDAR2	DMA Destination Address Register 2	R/W	H'00000000	H'E1030024	32
DCTR2	DMA Transfer Counter Register 2	R/W	H'00000000	H'E1030028	32
DCCSR2	DMA Channel Control/Status Register 2	R/W	H'00000000	H'E103002C	32
DSAR3	DMA Source Address Register 3	R/W	H'00000000	H'E1030030	32
DDAR3	DMA Destination Address Register 3	R/W	H'00000000	H'E1030034	32
DCTR3	DMA Transfer Counter Register 3	R/W	H'00000000	H'E1030038	32
DCCSR3	DMA Channel Control/Status Register 3	R/W	H'00000000	H'E103003C	32
DMACR	DMA Control Register	R/W	H'00000000	H'E1030040	32

Note: All the above registers can only be accessed with 32 bit size.

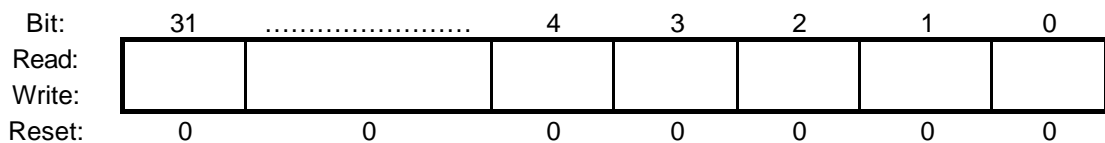
4.3.1 DMA Source Address Register (DSARn, n = 0 to 3)

DMA source address register DSARn (n = 0 to 3 which is corresponding to each channel) is used to set DMA transfer source address. It is a 32-bit readable/writable register and counts up according to the settings in DCCSRn. The initial value of the register after power on or manual reset is H'00000000.



4.3.2 DMA Destination Address Register (DDARn, n = 0 to 3)

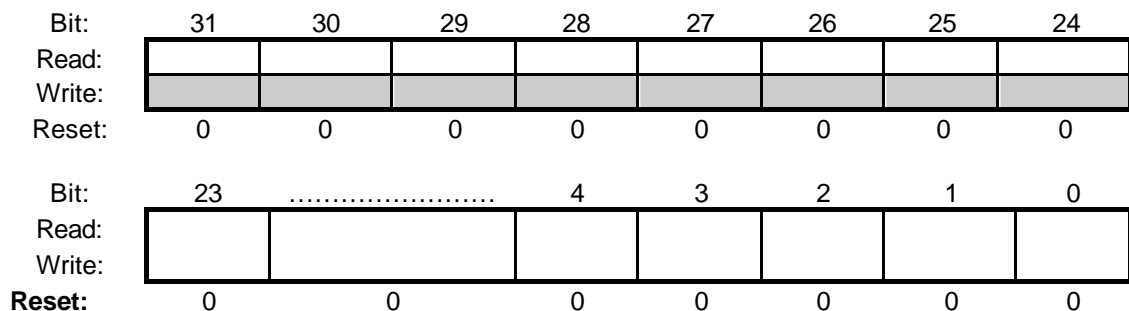
DMA destination address register DDARn (n = 0 to 3 which is corresponding to each channel) is used to set DMA transfer destination address. It is a 32-bit readable/writable register and counts up according to the settings in DCCSRn. The initial value of the register after power on or manual reset is H'00000000.



4.3.3 DMA Transfer Counter Register (DCTRn, n = 0 to 3)

The 32-bit readable/writable register DCTRn (n = 0 to 3 which is corresponding to each channel) is used to set the transfer counts for DMA channels. These registers maintain the remaining transfer count (byte count, 16-bit count, word count, 16-byte count or 32-byte count) during a DMA transfer. These registers are decremented by 1 for each transfer that is performed. Transfer ends when DCTRn counts down to 0.

The initial value of the register after power on or manual reset is H'00000000. Bits 31–24 of these registers are reserved; they are always read as 0.



4.3.4 DMA Channel Control/Status Register (DCCSRn, n = 0 to 3)

DMA channel control/status register DCCSRn (n = 0 to 3 which is corresponding to each channel) is a 32-bit readable/writable register that contains control and status bits for each channel. This register is initialized to H'00000000 by a power-on or manual reset.

Bits 31 to 29 are not present in channel 3. In DCCSR3, these bits are always read as 0.

Bit:	31	30	29	28	27	26	25	24
Read:	EACKS	EACKM	ERDM1	ERDM0	EOPM			
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:	DS2	DS1	DS0		RDIL3	RDIL2	RDIL1	RDIL0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	SAM	DAM		RS4	RS3	RS2	RS1	RS0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	TM		BT	AR	TC		TCIE	CHDE
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 26 to 24, 20, 13, 6 and 2: Reserved bits. These bits are always read as 0 and should only be written with 0.

- **External DMA Acknowledge DACKn Output Level Select (EACKS):** This bit is used to select whether DACKn is active-high or active-low.

Bit 31: EACKS	Description	
0	Active-high.	(Initial value)
1	Active-low.	

Note: DACKn (n = 0 to 2 which is corresponding to each channel except channel 3 that does not accept external DMA request).

- **External DMA Acknowledge DACKn Output Mode Select (EACKM):** This bit is used to select whether DACKn is asserted in read or write cycle.

Bit 30: EACKM	Description	
0	DACK is output in read cycle.	(Initial value)
1	DACK is output in write cycle.	

Note: DACKn (n = 0 to 2 which is corresponding to each channel except channel 3 that does not accept external DMA request).

- **External Request Detection Mode (ERDM1 to ERDM0):** These bits are used to set external request detection mode.

Bit 29: ERDM1	Bit 28: ERDM0	External Request Detection Mode	
0	0	Low level detection	(Initial value)
0	1	Falling edge detection	
1	0	High level detection	
1	1	Rising edge detection	

- **End Of Process Mode (EOPM):** This bit is used to set external transfer end signal EOP output mode.

Bit 27: EOPM	Output Mode	
0	Active-high	(Initial value)
1	Active-low	

- **Transfer Data Size 2 to 0 (DS2 to DS0):** Sets the number of data bits of a transfer unit.

Bit 23: DS2	Bit 22: DS1	Bit 21: DS0	Data Size	
0	0	0	32 bit	(Initial value)
0	0	1	8 bit	
0	1	0	16 bit	
0	1	1	16-byte	
1	0	0	32-byte	
1	0	1	Reserved	
1	1	0	Reserved	
1	1	1	Reserved	

Note: When transfer data size is set to 16-byte, 32-byte, it is a burst transfer of 4 word or 8 word. When it is a burst transfer, software should guarantee that the source and destination address are 16-byte or 32-byte boundary.

- **Request Detection Interval Length (RDIL3 to RDIL0):** These bits set the number of transfer unit (read and write cycle) between two request detection in single or block mode.

When RDIL3-0 is 0000, the request is detected every transfer unit, no matter with level or edge detection, in single mode, and is detected every transfer unit with level detection, or only once with edge detection, in block mode.

Bit 19: RDIL3	Bit 18: RDIL2	Bit 17: RDIL1	Bit 16: RDIL0	Data Size	
0	0	0	0	RDIL 3-0 is ignored	(Initial value)
0	0	0	1	2 transfer unit	
0	0	1	0	3 transfer unit	
0	0	1	1	4 transfer unit	
0	1	0	0	5 transfer unit	
0	1	0	1	6 transfer unit	
0	1	1	0	7 transfer unit	
0	1	1	1	8 transfer unit	
1	0	0	0	9 transfer unit	
1	0	0	1	10 transfer unit	
1	0	1	0	11 transfer unit	
1	0	1	1	12 transfer unit	
1	1	0	0	13 transfer unit	
1	1	0	1	14 transfer unit	
1	1	1	0	15 transfer unit	
1	1	1	1	16 transfer unit	

- **Source Address Mode (SAM):** This bit is used to enable counting up of the transfer source address for channel n (n = 0 to 3).

Bit 15: SAM	Source Address Mode	
0	Fixed	(Initial value)
1	Increment	

- **Destination Address Mode (DAM):** This bit is used to enable counting up of the transfer destination address for channel n (n = 0 to 3).

Bit 14: DAM	Destination Address Mode	
0	Fixed	(Initial value)
1	Increment	

- **Transfer Request Source 4 to 0 (RS4 to RS0):** These bits are used to set the transfer request source for channel n (n = 0 to 3).

Bit 12: RS4	Bit 11: RS3	Bit 10: RS2	Bit 9: RS1	Bit 8: RS0	Description	
0	0	0	0	0	External request with DREQn, (external address \leftrightarrow external device with DACKn)	(Initial value)
0	0	0	0	1	Reserved	
0	0	0	1	0	Reserved	
0	0	0	1	1	Reserved	
0	0	1	0	0	Reserved	
0	0	1	0	1	Reserved	
0	0	1	1	0	Reserved	
0	0	1	1	1	Reserved	
0	1	0	0	0	Auto-request (ignore RDIL3-0, external address space \rightarrow external address space)	
0	1	0	0	1	Reserved	
0	1	0	1	0	Reserved	
0	1	0	1	1	Reserved	
0	1	1	0	0	Reserved	
0	1	1	0	1	Reserved	
0	1	1	1	0	Reserved	
0	1	1	1	1	Reserved	
1	0	0	0	0	UART transmit-fifo-empty transfer request (external address space \rightarrow UARTTDR)	
1	0	0	0	1	UART receive-fifo-full transfer request (UARTRDR \rightarrow external address space)	
1	0	0	1	0	IrDA (UART2) transmit-fifo-empty transfer request (external address space \rightarrow UARTTDR2)	
1	0	0	1	1	IrDA (UART2) receive-fifo-full transfer request (UARTRDR2 \rightarrow external address space)	
1	0	1	0	0	AC97 transmit-fifo-empty transfer request	
1	0	1	0	1	AC97 receive-fifo-full transfer request	
1	0	1	1	0	Reserved	
1	0	1	1	1	Reserved	
1	1	0	0	0	Reserved	
1	1	0	0	1	Reserved	
1	1	0	1	0	Reserved	
1	1	0	1	1	Reserved	
1	1	1	0	0	TMU channel 2 (underflow interrupt, ignore RDIL3-0, external address space \rightarrow external address space)	
1	1	1	0	1	Reserved	
1	1	1	1	0	Reserved	
1	1	1	1	1	Reserved	

Note:

1. In DCCSR3, the initial value of RS4-RS0 is reserved.
2. Only auto request can be selected at the same time in all channels with different source and destination address.
3. External request 0-2 is corresponding to channel 0-2. For example, external request 1 can't be selected in channel 2.

- **Transfer Mode (TM):** This bit is used to set the transfer mode.

Bit 7: TM	Transfer Mode	
0	Single mode	(Initial value)
1	Block mode	

- **Block Transfer Type (BT):** This bit is used to set the block transfer type, continuous transfer or not.

Bit 5: BT	Description	
0	Continuous transfer is disabled. During block transfer, higher priority channel can take up the bus, the current channel transfer stops	(Initial value)
1	Continuous transfer is enabled. During block transfer, higher priority channel can not take up the bus until the current channel transfer ends	

- **Address Error (AR):** This is a status bit that indicates whether or not an address error generates during channel n (n = 0 to 3) DMA transfer. It can be only written 0 by CPU.

When performing a 16-bit, 32-bit, 16-byte or 32-byte data transfer, if a different address, which is not a 16-bit, 32-bit, 16-byte or 32-byte boundary address is specified respectively, an address error will be detected and the DMA will halt.

Bit 4: AR	Description	
0	No address error.	(Initial value)
1	An address generates.	

- **Bit 3—Terminal Count (TC):** This is a status bit that indicates whether or not DMA transfer has ended for channel n (n = 0 to 3). It is set to 1 after number of transfers specified in DCTRn when DCTRn counts down to 0.

Bit 3: TC	Description	
0	DMA transfer has not ended.	(Initial value)
1	DMA transfer has ended.	

- **Transfer End Interrupt Enable (TCIE):** This bit is to enable or disable the generation of the interrupt request when TC = 1.

Bit 1: TCIE	Description	
0	Disable interrupt	(Initial value)
1	Enable interrupt	

- **Channel DMA Transfer Enable (CHDE):** This bit is to enable or disable the DMA transfer of channel n (n = 0 to 3).

Bit 0: CHDE	Description	
0	Disable DMA transfer of the channel	(Initial value)
1	Enable DMA transfer of the channel	

4.3.5 DMA Control Register (DMACR)

DMACR contains the control and status bits used for all channels DMA transfers. It is a 32-bit readable/writable register, which is initialized to H'00000000 by a power-on or manual reset.

Bit:	31	20	19	18	17	16
Read:							
Write:							
Reset:	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:							PR1	PR0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:						AER		DME
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 31 to 10, 7 to 3 and 1: Reserved bits. These bits are always read as 0 and should only be written with 0.

- **Priority Mode 1 to 0 (PR1, PR0):** These bits are used to determine the order of priority for DMA transfer execution when transfer requests are accepted for a number of channels simultaneously.

Bit 9: PR1	Bit 8: PR0	Order of Priority	
0	0	CH0 > CH1 > CH2 > CH3	(Initial value)
0	1	CH0 > CH2 > CH3 > CH1	
1	0	CH2 > CH0 > CH1 > CH3	
1	1	Round robin mode	

- **Address Error (AER):** This is a status bit that indicates whether or not an address error has occurred during DMA transfer of all channels.

When any channel generates an address error (AR bit in DCCSRn is set to 1), this bit is set to 1, transfers on all channels are suspended, and an interrupt request (DTERI) is generated. The CPU cannot write 1 to AER. This bit can be cleared by writing 0 after reading 1 in interrupt service routine.

Bit 2: AER	Description	
0	No address error, DMA transfer enabled.	(Initial value)
1	Address error, DMA transfer disabled	

- **DMA Master Enable (DME):** Enables activation of the entire DMA. When the DME bit and the CHDE bit of the DCCSR register for the corresponding channel are set to 1, that channel is enabled for transfer. If this bit is cleared during data transfer, transfers on all channels are suspended.

Even if the DME bit has been set, transfer is not enabled when TC is 1 or CHDE is 0 in DCCSR, or when the AER bit in DMACR is 1.

Bit 0: DME	Description	
0	Operation disabled on all channels	(Initial value)
1	Operation enabled on all channels	

4.4 Operation

When a DMA transfer request is issued, the DMA starts the transfer according to the predetermined channel priority order. It ends the transfer when the transfer end conditions are satisfied. Transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request. The type of transfer is two-address transfer. Either block mode or single mode can be selected.

4.4.1 DMA Transfer Procedure

After the desired transfer conditions have been set in the DMA source address register (DSAR), DMA destination address register (DDAR), DMA transfer count register (DCTR), DMA channel control/status register (DCCSR), and DMA control register (DMACR), the DMA transfers data according to the following procedure:

1. The DMA checks to see if transfer is enabled (CHDE = 1, DME = 1, TC = 0, AER = 0).
2. In auto-request mode, the transfer begins automatically when the CHDE bit and DME bit are set to 1. In other request mode except auto-request, when a transfer request is issued and detected, the transfer has been enabled, the DMA transfers one transfer unit of data (determined by the setting of DS2-0). The DCTR value is decremented by 1 for each transfer. The actual transfer flow depends on the transfer mode.
3. When the specified number of transfers have been completed (when the DCTR value reaches 0), the transfer ends normally. If the TCIE bit in DCCSR is set to 1 at this time, a DTCl interrupt request is sent to the CPU.
4. If a DMA address error occurs, the transfer is suspended. Transfer is also suspended when the CHDE bit in DCCSR or the DME bit in DMACR is cleared to 0. In the event of an address error, a DTERI interrupt request is forcibly sent to the CPU.

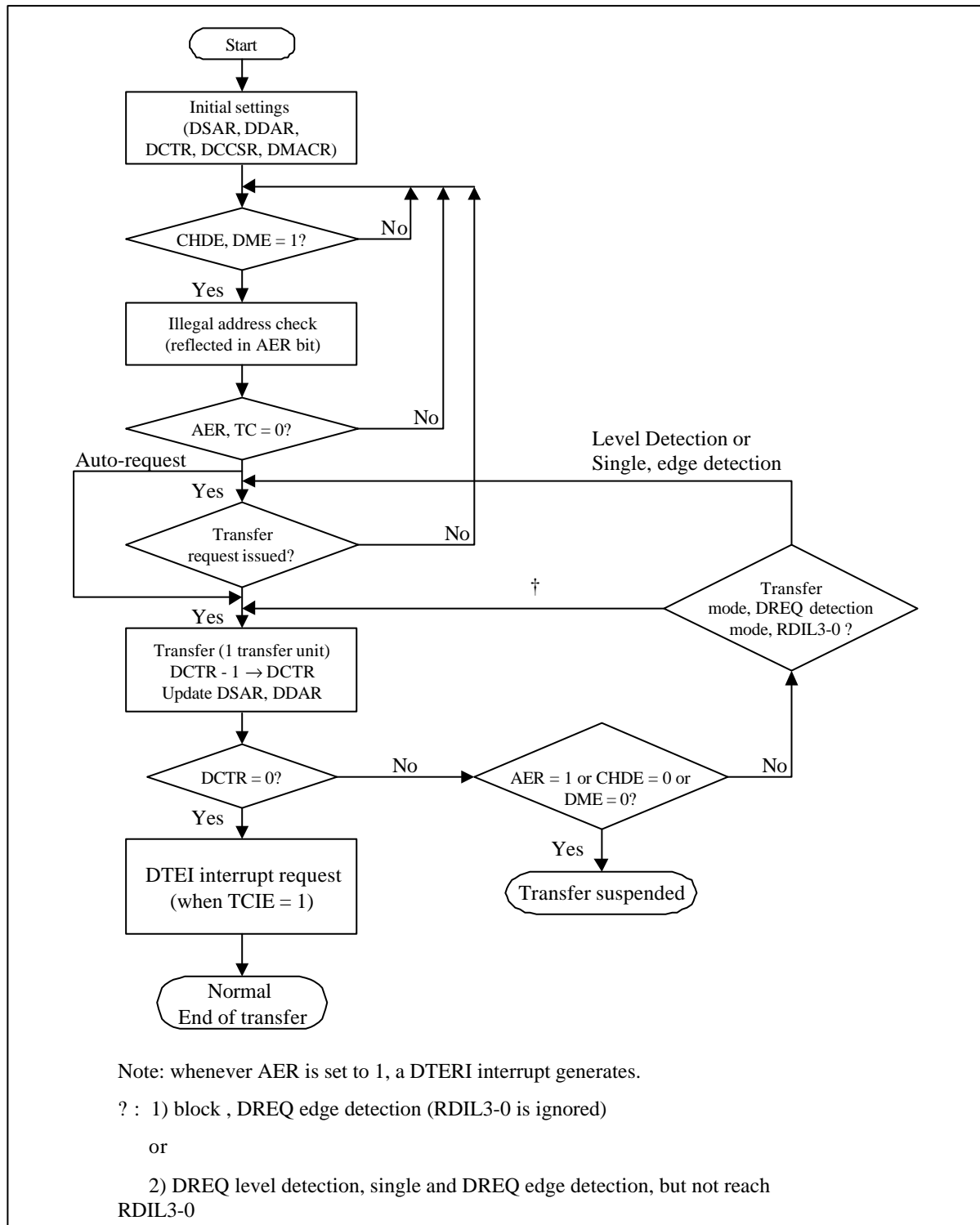


Figure 4-2 Flowchart of DMA Transfer

4.4.2 DMA Transfer Requests

DMA transfer requests are basically generated at either the data transfer source or destination, but they can also be issued by external devices or on-chip peripheral modules that are neither the source nor the destination.

DMA transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request. The transfer request source is selected by means of bits RS4–RS0 in DMA channel control/status registers 0–3 (DCCSR0–DCCSR3).

4.4.2.1 Auto Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or memory to on-chip peripheral module like GPIO, which is unable to request a transfer, the auto-request mode allows the DMA to automatically generate a transfer request signal internally. When the CHDE bit in DCCSR0–DCCSR3 and the DME bit in the DMA control register (DMACR) are set to 1, the transfer begins (so long as the TC bit in DCCSR0–DCCSR3 and AER bits in DMACR are all 0).

4.4.2.2 External Request Mode

In this mode, a transfer is performed in response to a transfer request signal DREQn from an external device by setting RS4-0(00000) in DCCSR0-2 according to the application system. If DMA transfer is enabled (CHDE = 1, DME = 1, TC = 0, AER = 0), transfer starts when DREQn is input. TM and ERDM1-0 bits in DCCSRn can determine the transfer mode and detection mode.

The transfer request (DREQn) is from either of the data transfer source or destination.

Table 4-3 RS4-0 Settings for External Transfer Request

RS4	RS3	RS2	RS1	RS0	Transfer Source	Transfer Destination
0	0	0	0	0	External memory or external device with DACKn	External device with DACKn or external memory

4.4.2.3 On-Chip Peripheral Module Request Mode

In this mode a transfer is performed in response to a transfer request signal (interrupt request signal) from an on-chip peripheral module. There are eleven such transfer request: underflow interrupt from the timer unit (TMU), and receive-fifo-full interrupts (RXI), transmit-fifo-empty interrupts (TXI) from the two serial communication interfaces (UART, IrDA/UART2) and request from AC97. If DMA transfer is enabled (CHDE = 1, DME = 1, TC = 0, AER = 0), transfer starts when a transfer request signal is input.

The source of the transfer request does not have to be the data transfer source or destination.

However, when the transfer request is set to RXI (transfer request by UART/IrDA/UART2 receive-fifo-full interrupt) according to RS4-RS0 in DCCSR, the transfer source must be the UART/IrDA/UART2's receive data register (UARTDR / UARTDR2). When the transfer request is set to TXI (transfer request by UART/IrDA/UART2 transmit-fifo-empty interrupt), the transfer destination must be the UART/IrDA/UART2 transmit data register (UARTTDR/UARTTDR2). The above transfer mode could be single mode or block mode and

the value of RDIL3-0 is set to be ignored or less than receive or transmit trigger value. It is similar to AC97.

Note: DMA doesn't handle UART timeout interrupt.

When using the DMA for transmission/reception, set and enable the DMA (RS4-RS0) before making the UART / IrDA / UART2 or TMU or AC97 settings. When using the DMA for transmission/reception, inhibit output of interrupt requests to the interrupt controller and set the DMA enable bit to enable DMA transfer.

4.4.3 Two-Address DMA Transfer

The DMA supports two-address transfer, in which both the transfer source and transfer destination addresses are output either with DACK or both not. The actual transfer operation timing depends on the transfer mode, which can be either block mode or single mode.

The transfer is used to access both the transfer source and the transfer destination by address or by both address and DACK signal. The transfer source and destination can be either on-chip peripheral module or external module accessed by address or by both address and DACK.

Data is read from the transfer source in the data read cycle, and written to the transfer destination in the data write cycle, so that the transfer is executed in two bus cycles. The transfer data is temporarily stored in the data buffer in DMA.

In a transfer between external memories such as that shown in Figure 4-3, data is read from external memory into the DMA's data buffer in the read cycle, then written to the other external memory in the write cycle.

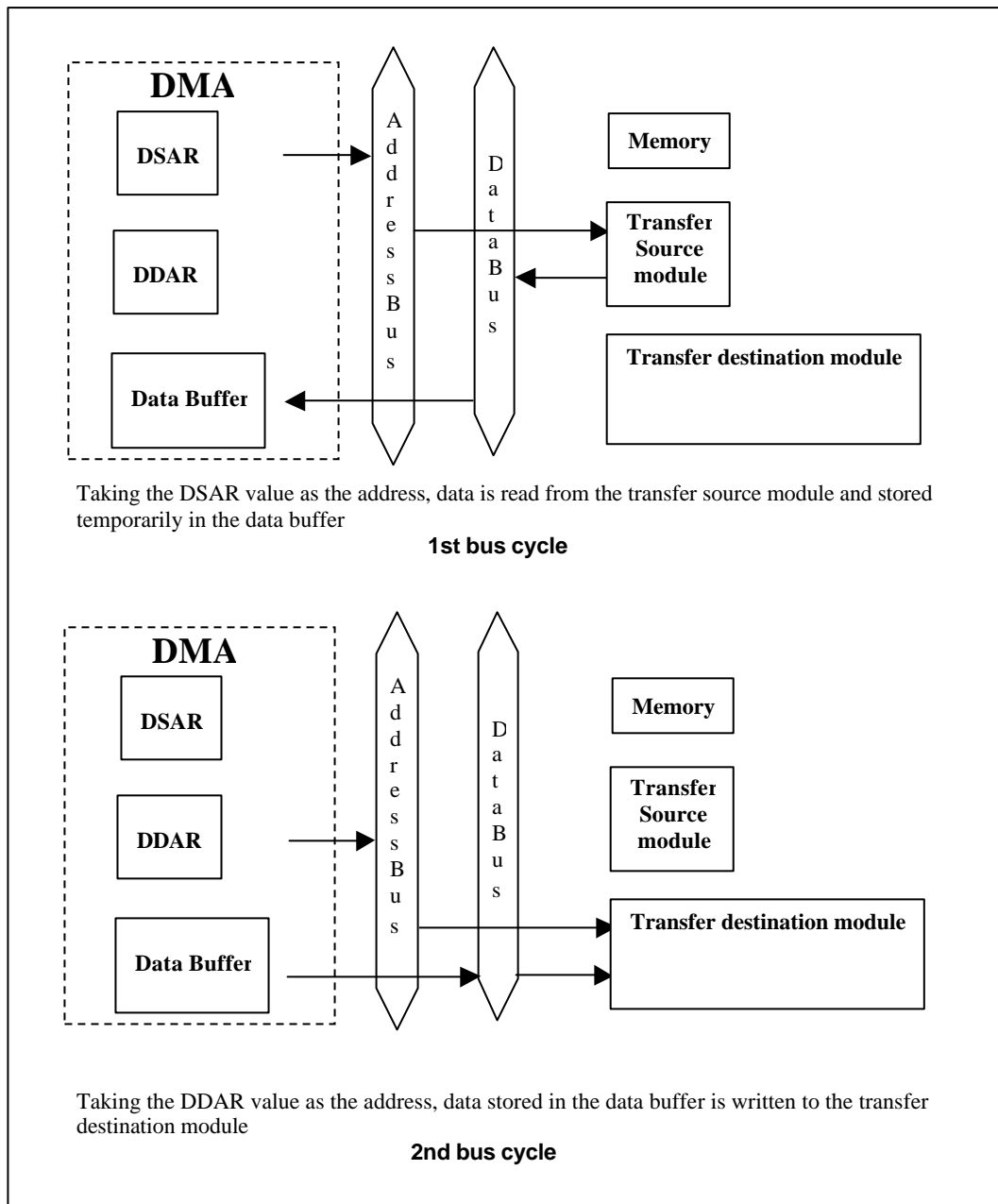


Figure 4-3 Two-address Transfer Example

Figure 4-4 shows an example of the timing for two-address transfer. In this example, DACK is output during read cycle and active-high level. Actual timing is dependent on memory type.

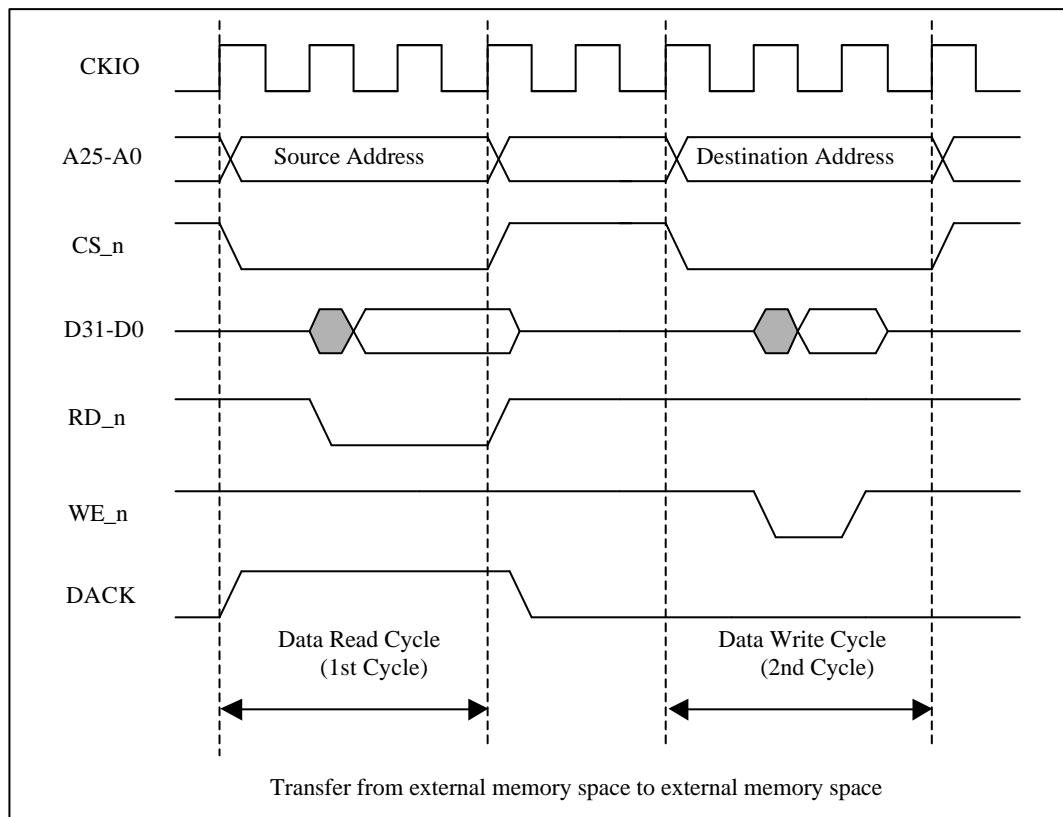


Figure 4-4 Timing in Two-address DMA Transfer

4.4.4 DMA Transfer Modes

There are two transfer modes, single mode and block mode, selected with the TM bit in DCCSR0–DCCSR3.

4.4.4.1 Single Mode

Single mode transfer needs DMA request for each transfer unit for level detection, and for edge detection when RDIL3-0 is ignored, otherwise, the detection interval can be determined by RDIL3-0, which is programmed.

In single mode, the DMA releases the bus to the CPU at the end of each transfer-unit (8-bit, 16-bit, 32-bit, 16-byte, or 32-byte) transfer unless other DMA channel requests the bus simultaneously with CPU (DMA has the higher priority to use the bus) or between the read cycle and write cycle of a transfer unit. When the next transfer request is issued, the DMA reacquires the bus from the CPU and carries out another transfer-unit transfer. If the CPU takes up the bus after the read cycle (before write cycle) of DMA, DMA will re-request the bus to finish the transfer unit. At the end of this transfer, the bus is again given to the CPU if there is no other channel transfer is suspended. This is repeated until the transfer end condition is satisfied. Single mode can be used with all categories of transfer request source, transfer source, and transfer destination.

Although there is fix or round robin priority, if a channel stops waiting for dma request, it can release the bus to another channel which requests the bus or maybe have a lower priority or another bus master like CPU even if the transfer end condition has not been satisfied. Figure 4-5 shows an example of DMA transfer timing in single mode. The transfer condition in this example is DREQ signal level detection.

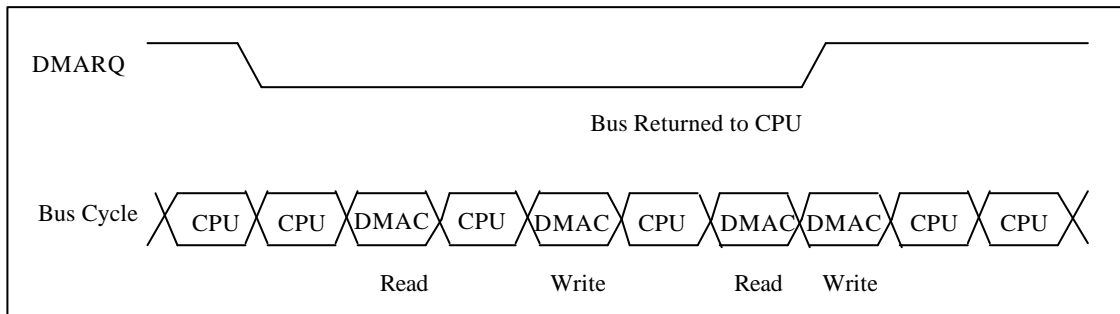


Figure 4-5 Example of DMA Transfer in Single Mode

4.4.4.2 Block Mode

In block mode, the transfer is different with BT bit setting in DCCSR.

When BT is cleared to 1, once the DMA has acquired the bus it holds the bus and transfers data continuously until the transfer end condition is satisfied unless CHDE = 0 in DCCSR, DME = 0 in DMACR, AER = 1 in DMACR. During this transfer, the higher priority channel can't take up the bus before the current channel transfer ends.

When BT is 0, once the DMA acquired the bus, it starts to transfer data and suspends when the other higher priority channel request the bus and it will resume transfer until the higher priority channel transfer end or transfer in turn with that channel depending upon PR1-0 bits setting.

There is an exception. For example, when with DREQ low level detection in external request mode, however, DREQ is driven high during transfer, it releases the bus to another channel which requests the bus or maybe have a lower priority or another bus master like CPU, even if the transfer end condition has not been satisfied.

Figure 4-6 shows an example of DMA transfer timing in block mode. The transfer conditions in this example are DREQ level detection.

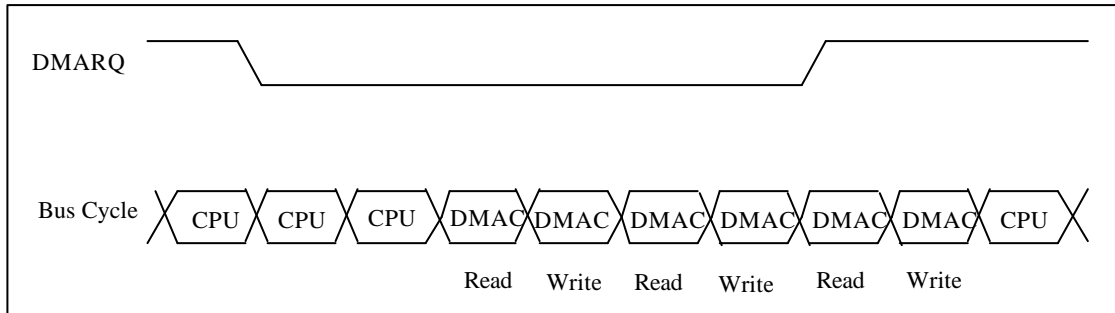


Figure 4-6 Example of DMA Transfer in Block Mode

Note: Block mode can be set regardless of the data size. A 32-byte block transfer setting can also be made. The request detection interval can be set in RDIL3-0 if the device permits the transfer request detection mode.

4.4.4.3 Relationship between DMA Transfer Source, Request Mode and Transfer Mode

Table 4-4 shows the relationship between DMA transfer source, request mode and transfer mode.

Table 4-4 Relationship between DMA Transfer Source, Request Mode and Transfer Mode

Transfer Source	Request Mode	Transfer Mode	Data Size (bits)	Channel
External memory and external memory	Auto / on-chip	Block /Single	8/16/32/16B/32B	0,1,2,3
External memory and memory-mapped external device with DACK	External	Block /Single	8/16/32/16B/32B	0,1,2
Memory-mapped external device and memory-mapped external device without DACK	Auto / on-chip	Block /Single	8/16/32/16B/32B	0,1,2,3
External memory or memory-mapped external device without DACK and on-chip peripheral module	Auto / on-chip	Block /Single	8/16/32/16B/32B	0,1,2,3

16B/32B: 16/32 bytes transfer

Notes:

1. If the transfer request source is the UART / IrDA / UART2, either the transfer source must be UARTRDR / UARTRDR2 or the transfer destination must be UARTTDR / UARTTDR2. Access size is which permitted for the on-chip peripheral module register that is the transfer source or transfer destination. It is similar to AC97.
2. When the transfer request is an external request, only channels 0, 1, 2 can be used.

4.4.5 Channel Priorities

If the DMA receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority system, either in a fixed mode or round robin mode. The mode is selected with priority bits PR1 and PR0 in the DMA control register (DMACR).

4.4.5.1 Fixed Mode

In this mode, the relative channel priorities remain fixed. The following priority orders are available in fixed mode:

- CH0 > CH1 > CH2 > CH3
- CH0 > CH2 > CH3 > CH1
- CH2 > CH0 > CH1 > CH3

The priority order is selected with bits PR1 and PR0 in DMACR.

4.4.5.2 Round Robin Mode

In round robin mode, each time the transfer of one transfer unit (byte, 16-bit, 32-bit, 16 bytes or 32 bytes) ends on a given channel, that channel is assigned the lowest priority level. The order of priority in round robin mode immediately after a reset is CH0 > CH1 > CH2 > CH3.

For example, the order of priority after a reset is:
CH0 > CH1 > CH2 > CH3

If CH0 first request and use the bus, then the next order of priority will be:
CH1 > CH2 > CH3 > CH0

If CH1 first request and use the bus, then the next order of priority will be:
CH0 > CH2 > CH3 > CH1

If CH2 first request and use the bus, then the next order of priority will be:
CH0 > CH1 > CH3 > CH2

Note: In round robin mode, if no transfer request is accepted for any channel during DMA transfer, the priority order still is CH0 > CH1 > CH2 > CH3 as that after reset.

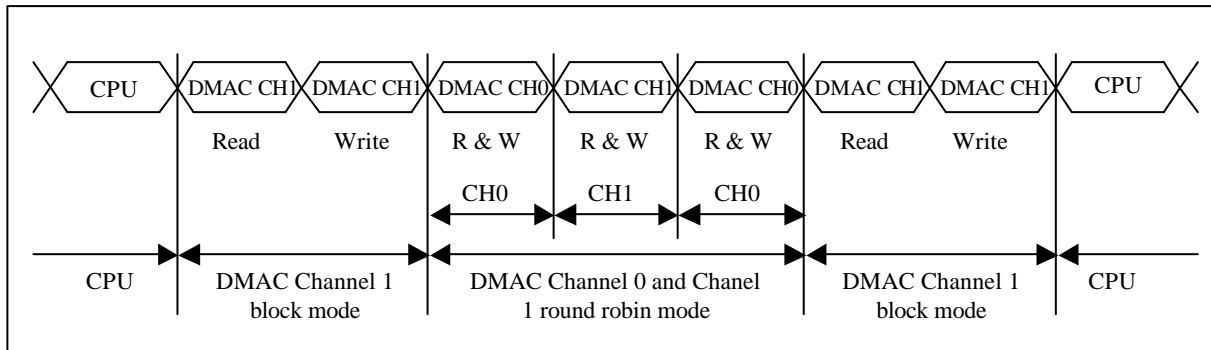
4.4.5.3 Transfer Mode and Channel Priority Order

When, for example, channel 1 is transferring data in block mode and BT in DCCSR is set to 0, and a transfer request is issued to channel 0, which has a higher priority, the channel 0 transfer is started immediately after channel 1 complete the current transfer unit.

If fixed mode has been set for the priority levels (CH0 > CH1), transfer on channel 1 is continued after transfer on channel 0 is completely finished, whether single mode or block mode is set for channel 0.

If round robin mode has been set for the priority levels, transfer on channel 1 is restarted after one transfer unit of data is transferred on channel 0, whether single mode or block mode is set for channel 0. Channel execution alternates in the order: channel 1 → channel 0 → channel 1 → channel 0.

An example of round robin mode operation is shown in Figure 4-7. The channel switch between channel 0 and channel 1 only can occur when a transfer unit (read and write cycle) is finished as the status in the figure, but in single mode, bus switch can occur during a transfer unit, such as a CPU cycle is inserted between the read and write cycle of DMA transfer. Since channel 1 is in block mode regardless of whether fixed mode or round robin mode is set for the priority order, the bus is not released to the CPU until channel 1 transfer ends unless it is request level detection and a break occurs on the request.



Note: R & W — brief case of Read and Write two cycles.

Figure 4-7 Bus Handling with Two DMA Channels Operating

4.4.6 DMA Transfer Bus Cycle States for Device with DREQ

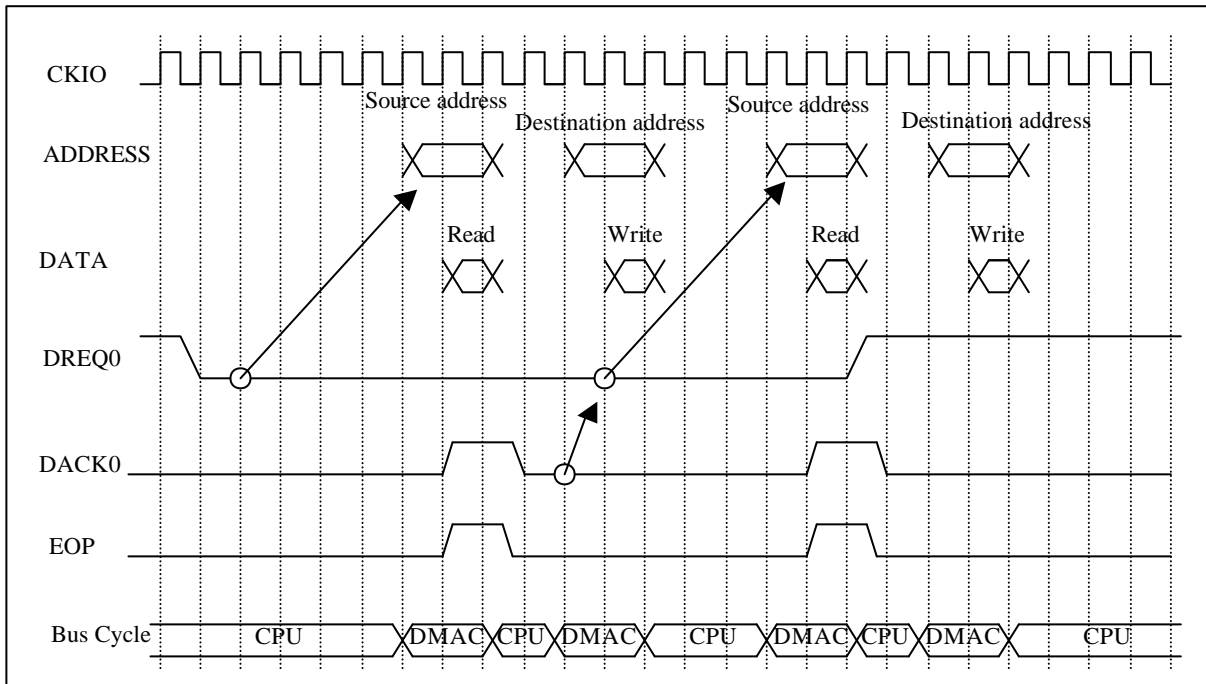
In external request mode, the DREQ pin is sampled at the rising edge of CKIO clock pulses. When DREQ input is detected, a DMA bus cycle is generated and DMA transfer is executed. The subsequent DREQ sampling operation is performed after the preceding DMA transfer bus cycle (depending on DACK output in read or write cycle) finishes and it is also dependent upon the different transfer mode (single or block) and detection mode (level or edge) and detection interval length RDIL3-0. The operation is as the following:

Note: the following figures are just examples which only show the operation bus cycles. See the actual timing for different memory in detail in EMI spec.

4.4.6.1 Single Mode and Level Detection

For example, in Figure 4-8 (single mode, DREQ level detection, DACK high active in read cycle, RDIL3-0 ignored, EOP active high), DMA transfer begins after the first sampling operation. The second sampling operation is performed after the first DMA transfer read cycle finishes. If DREQ is not detected at this time, sampling is executed in every subsequent cycle.

Note: if RDIL3-0 is not ignored, the interval of the second sampling operation is determined by RDIL3-0 if the device permits this transfer request detection mode.



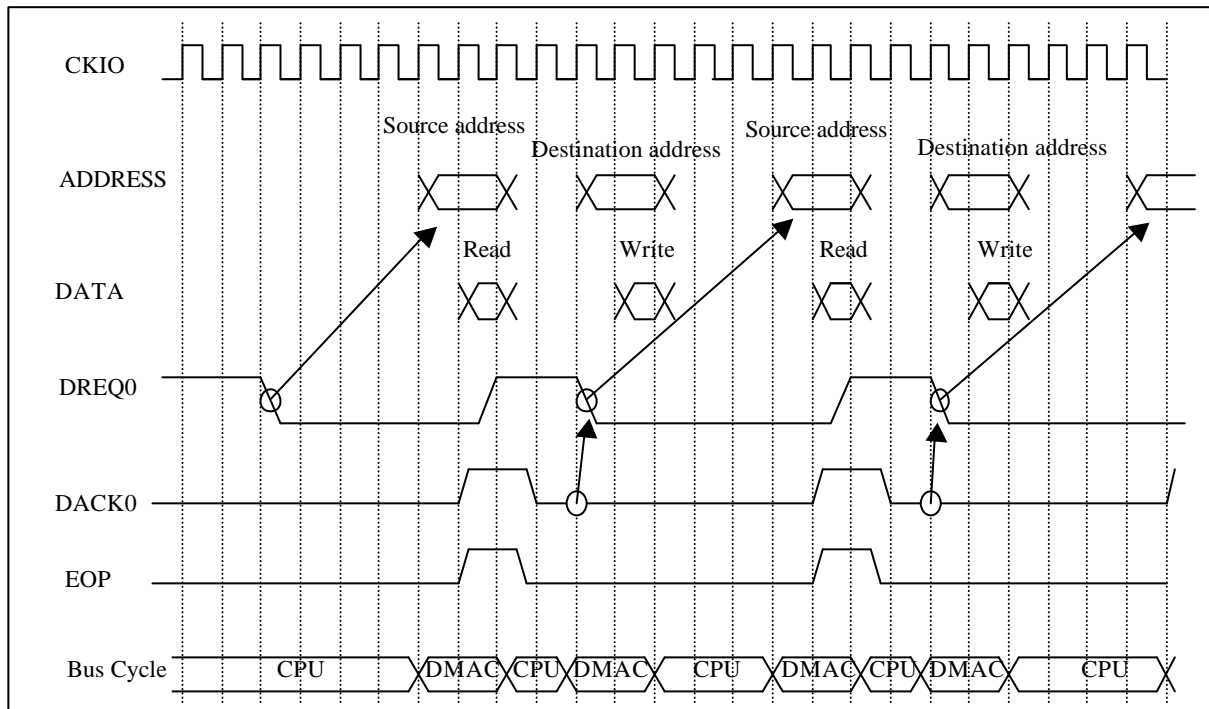
Note: In single mode, cpu cycle can be inserted into a transfer unit of dma.

Figure 4-8 Example of Single Mode, DREQ Low Level Detection

4.4.6.2 Single Mode and Edge Detection

In Figure 4-9 (single mode, DREQ falling edge detection, RDIL3-0 is ignored, DACK high active in read cycle, EOP active high), DMA transfer begins after the first sampling operation. The second sampling operation begins from the cycle in which the first DMA transfer read cycle ends. If DREQ is not detected at this time, sampling is executed in every subsequent cycle.

Note: if RDIL3-0 is not ignored, the interval of the second sampling operation is determined by RDIL3-0 if the device permits this transfer request detection mode.



Note: In single mode, cpu cycle can be inserted into a transfer unit of dma.

Figure 4-9 Example of Single Mode, DREQ Falling Edge Detection

4.4.6.3 Block Mode and Level Detection

DREQ sampling timing in block mode is virtually the same as for single mode when DREQ is level detection. The difference is that after DMA finishes each transfer unit, it doesn't release the bus to CPU unless when BT is 0 and a break on request occurs and no other DMA channel requests use of the bus.

For example, in Figure 4-10 (block mode, DREQ low level detection, DACK high active in read cycle, RDIL3-0 is ignored, EOP active high), DMA transfer begins after the first sampling operation. The second sampling operation is performed after the first DMA transfer read cycle ends.

Note: if RDIL3-0 is not ignored, the interval of the second sampling operation is determined by RDIL3-0 if the device permits this transfer request detection mode.

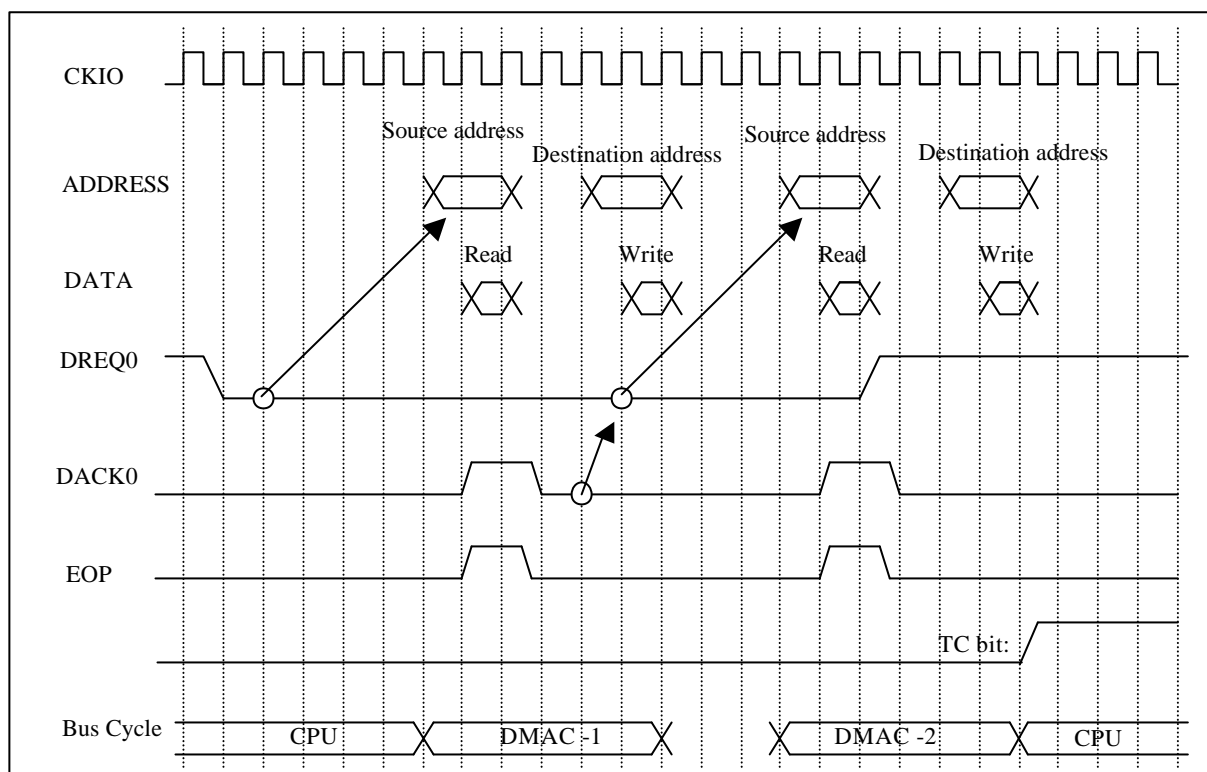


Figure 4-10 Example of Block Mode, DREQ Low Level Detection

4.4.6.4 Block Mode and Edge Detection

In block mode and DREQ edge detection, DREQ sampling is performed in the first cycle only when BT is 1 and RDIL3-0 is ignored. When BT is 0, the transfer can be suspended by the other higher priority channel and can resume transfer without sampling the request.

For example, in the Figure 4-11 (Block mode, DREQ falling edge detection, DACK high active in read cycle, RDIL3-0 is ignored, BT is 1, EOP active high), DMA transfer begins after the first sampling operation. DMA transfer then continues until the end of the number of data transfers set in DCTR and TC bit is set in DCCSR. DREQ is not sampled during this time.

Note: if RDIL3-0 is not ignored, the interval of the second sampling operation is determined by RDIL3-0 if the device permits this transfer request detection mode.

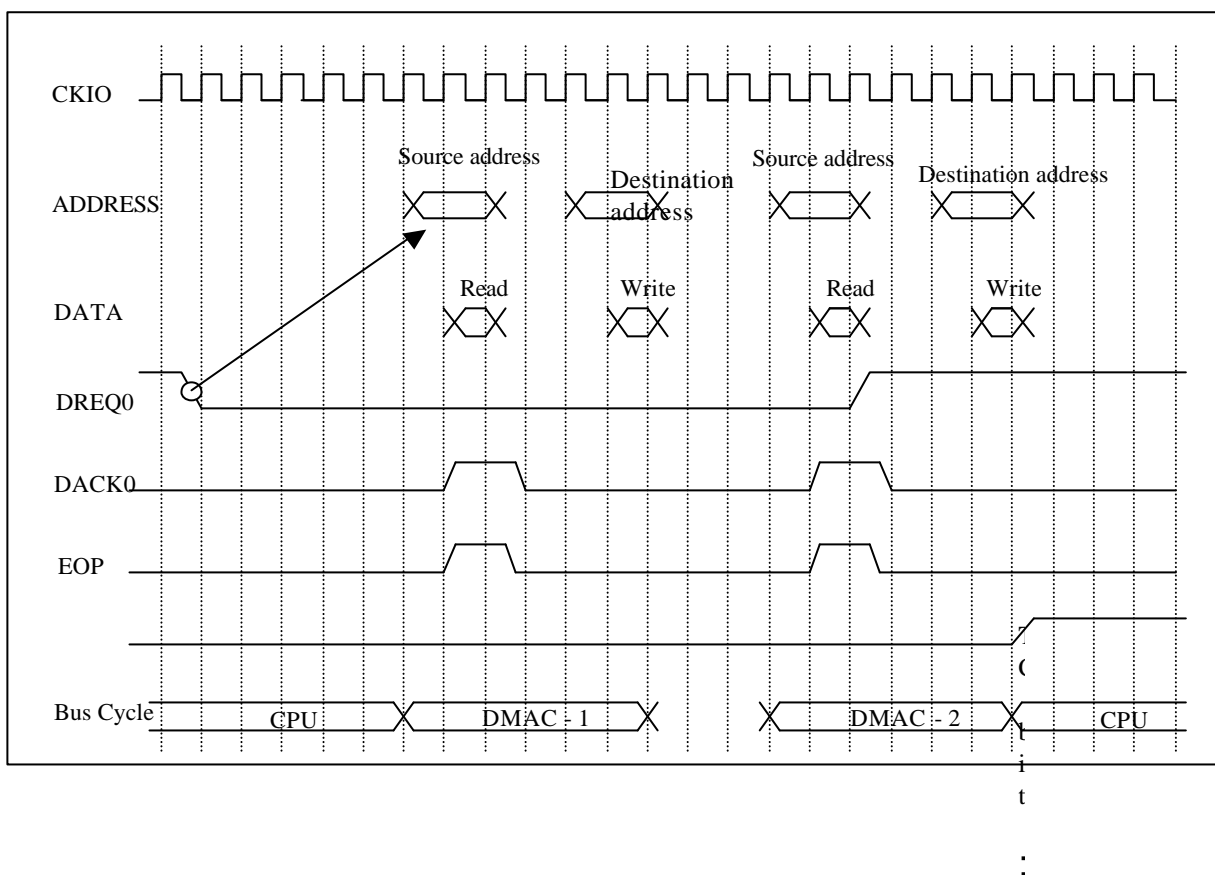


Figure 4-11 Example of Block Mode, DREQ Edge Detection

4.4.7 Normal DMA Transfer Ending

When the value in the DMA transfer count register (DCTR) reaches 0 and the transfer end flag (TC) in DCCSR is set, DMA transfer ends on the corresponding channel. If the interrupt enable bit (TCIE) is set at this time, an interrupt (DTCI) request is sent to the CPU.

4.4.8 Forced Suspension

No matter what conditions for ending DMA transfer are satisfied, transfer suspension is executed on completion of processing for one transfer unit which means write cycle processing is executed even if a transfer end condition is satisfied during the read cycle.

4.4.8.1 Suspension on Single Mode

In single mode, the operation is the same for both edge and level transfer request detection. When a transfer end condition (CHDE = 0 in DCCSR, DME = 0 in DMACR) is satisfied, acceptance of DMA transfer requests is suspended. The DMA completes transfer for the transfer requests accepted up to the point at which the transfer end condition was satisfied, then stops.

4.4.8.2 Suspension on Block Mode, Level Detection

The delay between the point at which a transfer end condition is satisfied and the point at which the DMA actually stops is the same as in single mode. The timing of stop request (CHDE = 0 in DCCSR, DME = 0 in DMACR) sampling is the same as the transfer request sampling timing. Therefore, a transfer request is regarded as having been issued until a stop request is detected, and the corresponding processing is executed before the DMA stops.

4.4.8.3 Suspension on Block Mode, Edge Detection

The delay between the point at which a transfer end condition is satisfied and the point at which the DMA actually stops is the same as in single mode.

In block mode with edge detection, only the first transfer request activates the DMA if RDIL3-0 is ignored, but the timing of stop request (CHDE = 0 in DCCSR, DME = 0 in DMACR) sampling is the same as the transfer request sampling timing during level detection transfer. Therefore, the transfer will be suspended when stop request (CHDE = 0 in DCCSR, DME = 0 in DMACR) is sampled and the corresponding processing is executed before the DMA stops.

4.4.8.4 Suspending Individual Channels

When the DMA enable bit (CHDE) in DCCSR is cleared, DMA transfer is suspended on the corresponding channel. The TC bit is not set in this case. Transfer is suspended.

4.4.8.5 Suspending Transfer Simultaneously on All Channels

Transfer ends on all channels simultaneously when either of the following conditions is satisfied:

- The address error bit (AER) in the DMA control register (DMACR) is set.
- The DMA master enable bit (DME) in DMACR is cleared to 0.

End of transfer when AER = 1 in DMACR:

If the AER bit in DMACR is set to 1 due to an address error, DMA transfer is suspended on all channels, and the bus is passed to the CPU. Therefore, when AER is set to 1, all the DMA registers remain the values for the DMA transfer to be performed next and the remaining number of transfers. The TC bit is not set in this case. Before resuming transfer, it is necessary to make a new setting for the channel that caused the address error (AR bit is set to 1 in DCCSRn), then write 0 to the AER bit in interrupt routine. Acceptance of transfer requests is suspended while AER is set to 1, so a DMA transfer request must be reissued when resuming

transfer except block mode, edge detection transfer which is not the channel generating the address error.

Acceptance of internal requests is also suspended, so when resuming transfer, the DMA transfer request enable bit (RS4-0) for the relevant on-chip peripheral module must be cleared to 0 before the new setting is made.

End of transfer when DME = 0 in DMACR:

If the DME bit in DMACR is cleared to 0, DMA transfer is suspended on all channels and the bus is passed to the CPU. The TC bit is not set in this case, all the DMA registers remain the values for the DMA transfer to be performed next and the remaining number of transfers. When resuming transfer, DME must be set to 1. Operation will then be resumed from the next transfer.

5 PCI Controller

5.1 Overview

Arca210 PCI Controller(PCIC) provides a bridge function between Arca210 memory bus and a 32-bit PCI bus. The main goal of the PCIC is to allow Arca-based systems-on-a-chip to communicate with PCI devices on PCI bus. The PCIC also provides a channel for PCI bus devices to read and write memory attached to the Arca210 EMI.

5.1.1 Feature

- Complete 33Mhz 32-bit PCI interface
- Support host or satellite operation mode
- Support both PCI master and target operation
- Target(or slave) lock support(master doesn't deliver lock)
- On-chip PCI arbiter (which is disabled in satellite mode and can be selected in host mode) supports total 5 masters include Arca210

Regarding the entire PCI 2.1 specification, refer to *PCI Local Bus Specification Revision 2.1* (available at <http://www.pcisig.com>) This document will only describe the Arca implementation and any differences from the current PCI 2.1 specification.

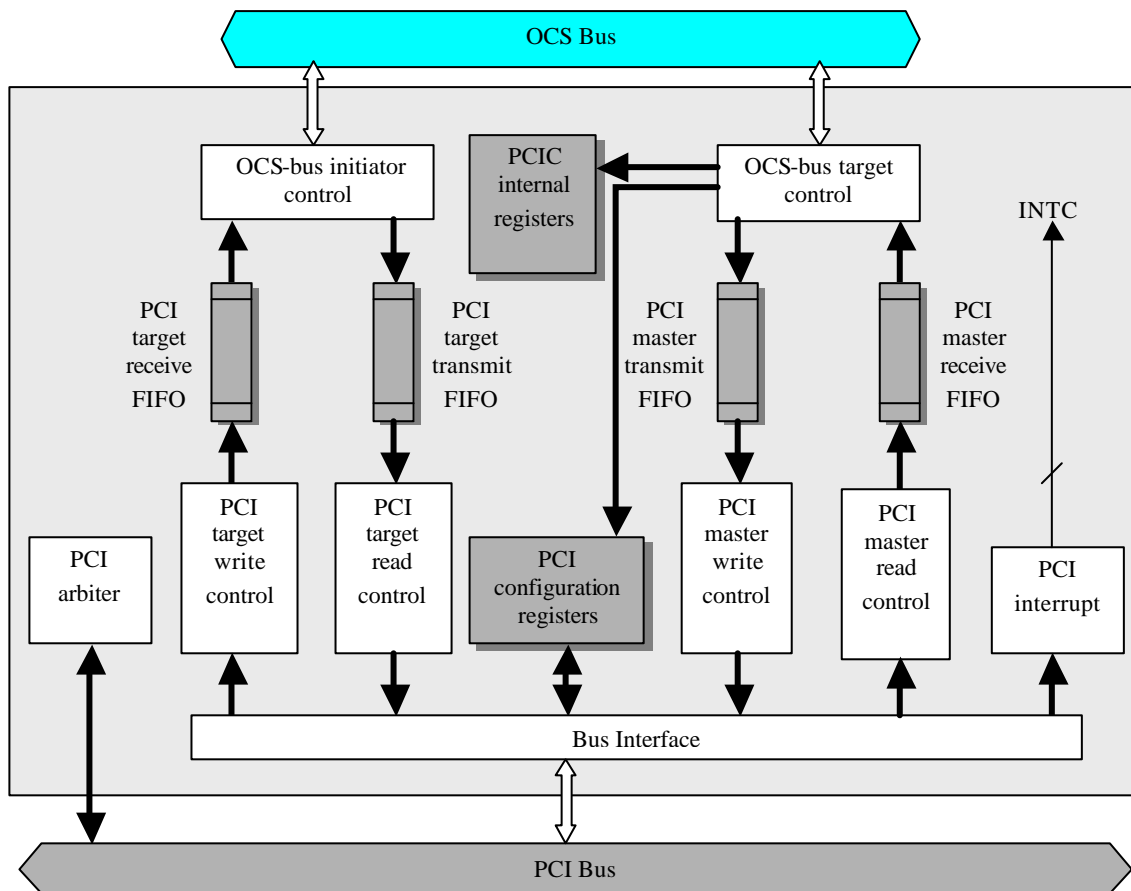


Figure 5-1 PCIC Block Diagram

The block diagram of the PCI Controller is shown in Figure 5-1.

5.1.2 Signals

Refer to PCI 2.1 specification for PCI signals.

For more background on PCI, see: *PCI Local Bus Specification Revision 2.1* (available through the pci-sig, <http://www.pcisig.com>), *PCI System Architecture* (Tom Shanley, Don Anderson, MindShare, Inc., <http://www.mindshare.com>).

5.2 Operation Mode

In most situations, PCIC will act as a PCI host bridge with the Arca210 as the host, this is described as *host mode*. In this case the Arca210 CPU is the main processor or the only processor in the system and the PCIC permits transactions between the Arca210 system bus and the PCI system bus. The Arca210 CPU is the source side of the bridge and the PCI bus is the destination side. Arca210, acting as a PCI host bridge, configures all other PCI-connected devices. The remainder of this document will assume *host mode*, except where noted.

There are also applications where Arca210 will act as a satellite PCI device in another PCI-based system. In *satellite mode*, the PCIC will be configured by another CPU connected to the PCI bus.

5.2.1 Configuration

The operation mode is controlled by pin MD_pci when system reset. Table 5-1 describes the configuration of PCI mode by pin MD_pci.

Table 5-1 PCI Mode Configuration Pin

Configuration Pin	Description	
MD_pci	0	Satellite mode. A device in a hosted PCI system. Configured by the host, for example, a PCI card in a PC.
	1	Hosts mode. The main processor in a PCI system. Responsible for configuration, and interrupt handling.

Arca210 has an internal PCI-bus arbiter that can be used when Arca210 is in host mode.

Table 5-2 PCI Arbiter Mode Configuration Pin

Configuration Pin	Description	
MD_pciarb	0	Disable the Arca210 internal PCI arbiter when in PCI host mode.
	1	Enable the Arca210 internal PCI arbiter when in PCI host mode.

5.3 PCI Bus Master Operation

Arca210 PCI Bus Master operation is defined as CPU core initiated read/write transfers between Arca210 and PCI bus to access devices attached to PCI bus. OCS-Bus read and write transactions addressed to the physical address range from **H'C0000000 to H'DFFFFFFF** are translated to PCI cycles and passed to the PCI bus.

Arca210 CPU can read/write to targets on the PCI bus through 3 PCI memory spaces and one I/O space and each space is 128-Mbyte and is mapped to a PCI address range by separate translation registers (PCI memory translation register 0-2 and PCI I/O translation register). I/O area can handle single byte and word addressing. The address map is shown in Figure 5-2.

PCIC doesn't support Cache line wrap mode defined in the PCI specification. So Arca210 PCIC master never generates a Cache line wrap mode.

Arca210 PCIC Master interface will not generate fast back-to-back transactions and deliver lock.

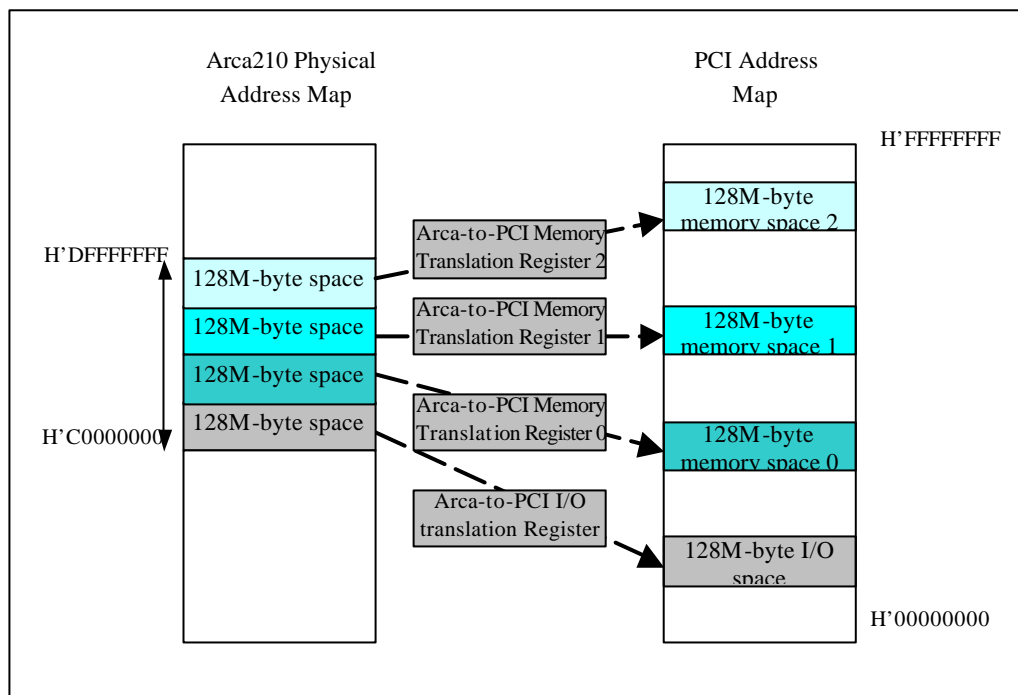


Figure 5-2 PCIC Master Address Mapping

5.4 Arca PCI Bus Target Operation

Arca210 PCIC Target operation is defined as PCI bus external master initiated read/write transfers between the PCI bus and external memory(EMI), or between the PCI bus and an external I/O. Arca210 PCIC doesn't accept I/O cycles from PCI bus and only passes memory operations to the OCS-Bus Initiator interface. Similarly, Arca210 PCIC doesn't recognize Cache line wrap mode. Figure 5-4 shows the operation as PCI target device.

5.4.1 Base Address Register 0

When other PCI bus master read/write EMI memory, PCI memory space base register 0 (PCI_MBR0) translate a PCI address to an EMI memory physical address in that the PCI address matches the address range of the memory base register 0 and the PCIC will recognize the address and assert DEVSEL to claim the transaction. The size of the memory space accessed from PCI bus is 256KB to 256MB selected by PTA_MEMATR register, thus the efficient most significant bits of the register are variable according to the size and the mapped address is from top bits of PTA_MEMATR.

If the space set by PTA_MEMATR is less than 256M-byte, the PCI address space will be shadowed to the same Arca210 memory space. For examples, we set PCI_MBR0= H'80000008, and PTA_MEMATR= H'60000005, the memory space that PCI master can access is only 8M-byte through the 256M-byte PCI memory space, Figure 5-3 shows the relationship of the two address space.

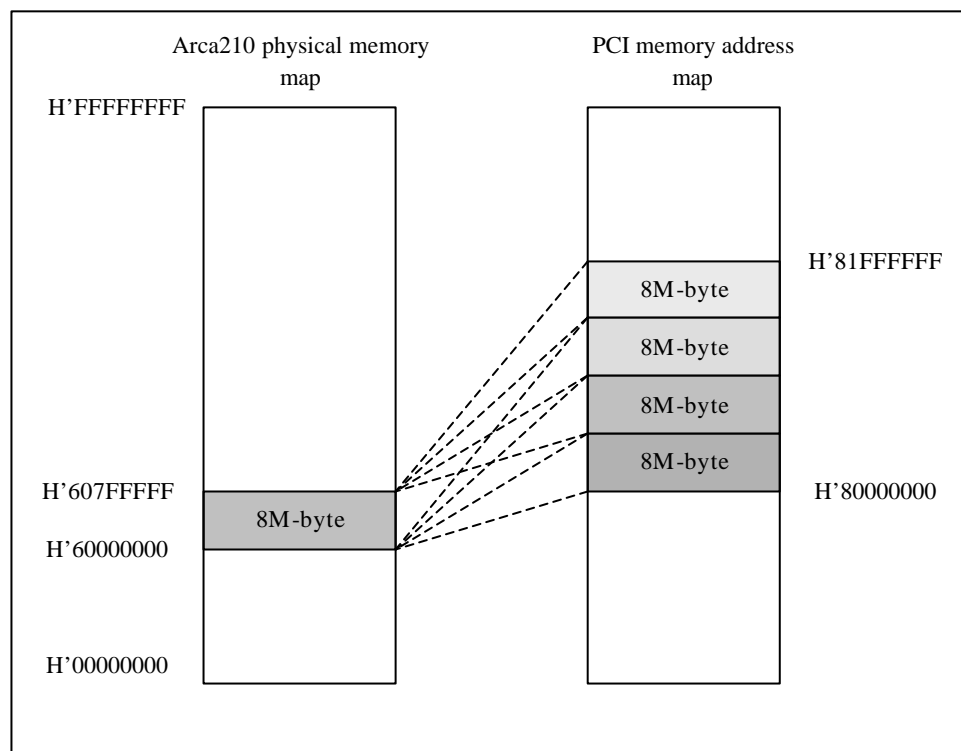


Figure 5-3 PCI Address Map to 8M-byte Arca210 Address Space

5.4.2 Base Address Register 1

PCI memory space base register 1 (PCI_MBR1) defines area of PCI memory space which maps to PCIC internal register: mailbox and mailbox interrupt registers etc.. The space of the area is 4KB.

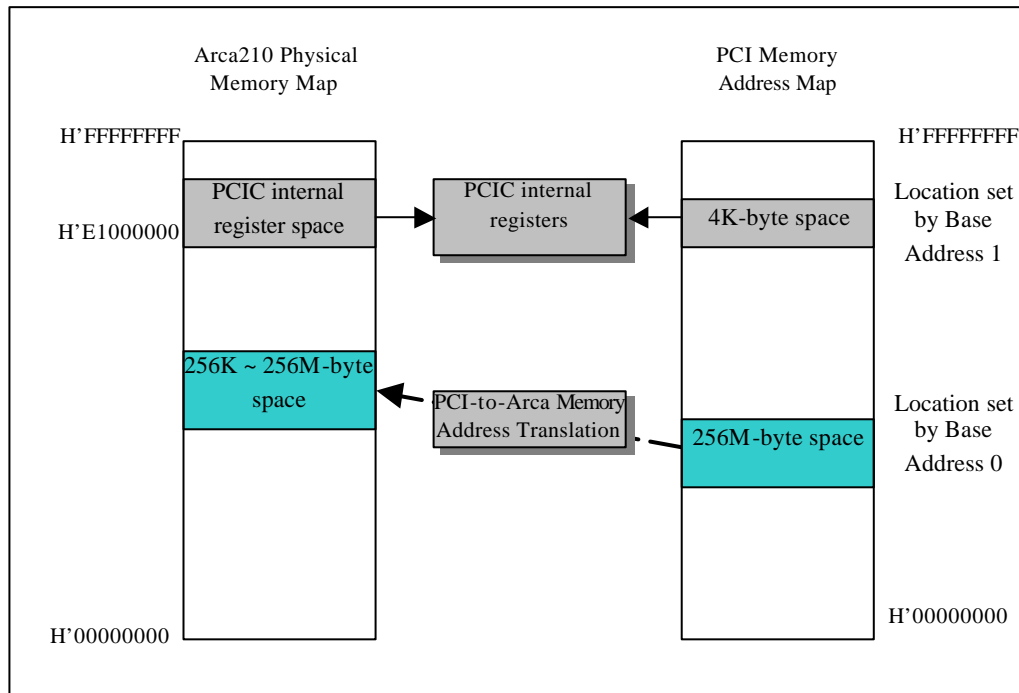


Figure 5-4 PCIC Target Address Mapping

5.5 PCI Configuration Register Access

By two registers, PCIC_CFGAR and PCIC_CFGDR, Arca210 can generate configuration cycles on the PCI bus. When accessing a PCI configuration register, first write the desired address of the configuration register to PCIC_CFGAR, and then read (or write) the PCIC_CFGDR to receive (or send) data.

Configuration operations frequently require multiple reads and writes at a single location. The address is retained in the PCIC_CFGAR register.

In host mode, Arca210 configures each device on PCI bus through these two registers. And configures its own pci configuration registers through internal registers access by CPU instruction, the area of the pci configuration registers is **H'E100_0100 – H'E100_0141**.

In satellite mode, Arca210 only can read each device's configuration on PCI bus. But it cannot access its own pci configuration registers, because it is configured by the system host device through PCI configuration command.

5.6 PCI Commands Support

Table 5-3 list the PCI commands supported by Arca210, and the corresponding action on system OCS-bus.

Table 5-3 PCI Commands Support

CBEn[3:0]	Command	As a Master	As a Target
0000	Interrupt Acknowledge	Yes	Ignored
0001	Special cycle	No	Ignored
0010	I/O read	Yes	Ignored
0011	I/O write	Yes	Ignored
010x	Reserved	No	Ignored
0110	Memory read	Yes, single or 4/8 burst on OCS-bus	Yes, 4 burst read on OCS-bus
0111	Memory write	Yes, single or 4/8 burst on OCS-bus	Yes, single word write on OCS-bus
100x	Reserved	No	Ignored
1010	Configuration read	Yes	Yes
1011	Configuration write	Yes	Yes
1100	Memory read multiple	Yes	Yes, 4 burst read on OCS-bus
1101	Dual address cycle	No	No
1110	Memory read line	No	Yes, 4 burst read on OCS-bus
1111	Memory write and invalidate	No	Yes, as Memory Write

5.7 PCIC Registers

Arca210 PCIC has two groups of internal registers which can be accessed by the Arca210 CPU. One group is associated with mailbox, mailbox interrupt, status and control registers. The other group is associated with PCI configuration space.

5.7.1 PCIC Control Registers

PCIC contains several internal control registers, as shown in the following table. Some of these control registers can be accessed both by another PCI master device and by the Arca210 CPU. These registers are used to pass messages and status between the Arca210 CPU and other PCI master devices.

The base address for accessing all of these PCIC registers by Arca210 is H'E100_00xx. The offset of every register is list in Table 5-4. And we will describe the details of each register below.

Table 5-4 PCIC Control Registers

Accessed from PCI Side			Description
Register Address: PCI_MBR1 + Offset			
Offset	R/W	Register Name	
H'00	W	PTA_MAILTR0	PCI-to-Arca Mailbox Transmit Register 0
H'04	W	PTA_MAILTR1	PCI-to-Arca Mailbox Transmit Register 1
H'08	W	PTA_MAILTR2	PCI-to-Arca Mailbox Transmit Register 2
H'0C	W	PTA_MAILTR3	PCI-to-Arca Mailbox Transmit Register 3
H'10	R	ATP_MAILRR0	Arca-to-PCI Mailbox Receive Register 0
H'14	R	ATP_MAILRR1	Arca-to-PCI Mailbox Receive Register 1
H'18	R	ATP_MAILRR2	Arca-to-PCI Mailbox Receive Register 2
H'1C	R	ATP_MAILRR3	Arca-to-PCI Mailbox Receive Register 3
H'20	R	PCI_MAILSR	PCI Mailbox Status Register
H'24	RW	PTA_DB	PCI-to-Arca Doorbell Register
H'28	RW	ATP_DB	Arca-to-PCI Doorbell Register
H'2C		No	
H'30		No	
H'34	R	PCI_SR	PCI Status Register
H'38		No	
H'3C	RW	PCI_IER	PCI Interrupt Enable Register
H'40		No	
H'44		No	
H'48		No	
H'4C		No	
H'50		No	
H'54		No	
Access from Arca Side			Description
Register Address: H'FD000000 + Offset			
Offset	R/W	Register Name	
H'00	R	PTA_MAILRR0	PCI-to-Arca Mailbox Receive Register 0
H'04	R	PTA_MAILRR1	PCI-to-Arca Mailbox Receive Register 1
H'08	R	PTA_MAILRR2	PCI-to-Arca Mailbox Receive Register 2
H'0C	R	PTA_MAILRR3	PCI-to-Arca Mailbox Receive Register 3
H'10	W	ATP_MAILTR0	Arca-to-PCI Mailbox Transmit Register 0
H'14	W	ATP_MAILTR1	Arca-to-PCI Mailbox Transmit Register 1
H'18	W	ATP_MAILTR2	Arca-to-PCI Mailbox Transmit Register 2
H'1C	W	ATP_MAILTR3	Arca-to-PCI Mailbox Transmit Register 3
H'20	R	PCIC_MAILSR	PCIC Mailbox Status Register
H'24	RW	PTA_DB	PCI-to-Arca Doorbell Register
H'28	RW	ATP_DB	Arca-to-PCI Doorbell Register
H'2C	RW	PCIC_CFGAR	PCIC Configuration Address Register
H'30	RW	PCIC_CFGDR	PCIC Configuration Data Register
H'34	R	PCIC_SR	PCIC Status Register
H'38	RW	PCIC_IER	PCIC Interrupt Enable Register
H'3C	-	-	No
H'40	RW	PTA_MEMATR	PCI-to-Arca Memory Address Translation Register
H'44	RW	ATP_MTR0	Arca-to-PCI Memory Translation Register 0
H'48	RW	ATP_MTR1	Arca-to-PCI Memory Translation Register 1
H'4C	RW	ATP_MTR2	Arca-to-PCI Memory Translation Register 2
H'50	RW	ATP_IOTR	Arca-to-PCI IO Translation Register
H'54	RO	PCIC_INTACK	PCIC Interrupt Acknowledge Register

There are eight 32-bit dual-port mailbox registers are provided for communication between Arca210 and external PCI device. Four of these registers (H'10~H'1C) can be written by the Arca210 CPU and read by PCI bus masters. The other four mailbox registers (H'00-H'0C) can be written by a PCI bus master and read by the Arca210 CPU. From the PCI-side, they are PCI Internal registers, that is, they exist in the memory address space assigned by Base Address Register 1. From the Arca210 OCS-bus side, the registers are normal registers which exist in the control register space assigned to the PCIC.

In addition to the mailbox registers, one common mailbox status register is accessible from each side. Figure 5-5 describes the operation to these registers from side of OCS-bus and PCI-bus.

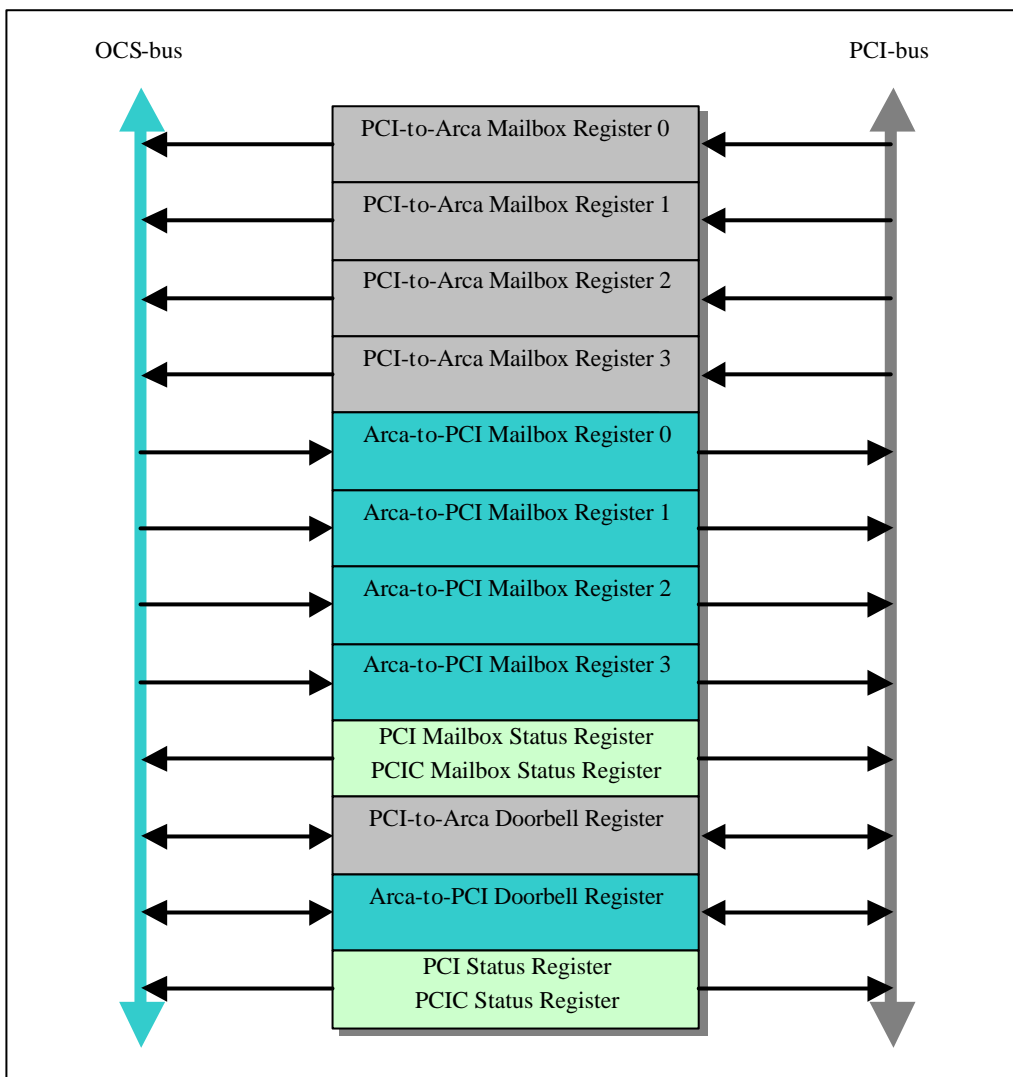


Figure 5-5 Mailbox and Doorbell Registers Access

5.7.1.1 PCI-to-Arca Mailbox Transmit Register

The registers are used to transmit information from a PCI master to Arca210. It can be written from PCI side and each byte of the register can be independently written. When read, it ignores. Each bit value update will automatically update the corresponding status bit value mailbox status register.

PTA_MAILTR 0 — PCI-to-Arca Mailbox Transmit Register 0 PCI Address Offset: H'00
PTA_MAILTR 1 — PCI-to-Arca Mailbox Transmit Register 1 Offset: H'04
PTA_MAILTR 2 — PCI-to-Arca Mailbox Transmit Register 2 Offset: H'08
PTA_MAILTR 3 — PCI-to-Arca Mailbox Transmit Register 3 Offset: H'0C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.2 PCI-to-Arca Mailbox Receive Register

The registers are used for Arca210 to receive information from a PCI master. It can be read from Arca and each byte of the register can be independently read. When read, they return current value and automatically update the corresponding status bit value in mailbox status register. When write, they ignore.

PTA_MAILRR 0 — PCI-to-Arca Mailbox Receive Register 0 Arca Address Offset: H'00
PTA_MAILRR 1 — PCI-to-Arca Mailbox Receive Register 1 Offset: H'04
PTA_MAILRR 2 — PCI-to-Arca Mailbox Receive Register 2 Offset: H'08
PTA_MAILRR 3 — PCI-to-Arca Mailbox Receive Register 3 Offset: H'0C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.3 Arca-to-PCI Mailbox Transmit Register

The registers are used to transmit information from Arca to PCI bus master. It can be written from Arca and each byte of the register can be independently written. When read, it ignores. Each bit value update will automatically update the corresponding status bit value in mailbox status register.

ATP_MAILTR 0 — Arca-to-PCI Mailbox Transmit Register 0 Arca Address Offset: H'10
ATP_MAILTR 1 — Arca-to-PCI Mailbox Transmit Register 1 Offset: H'14
ATP_MAILTR 2 — Arca-to-PCI Mailbox Transmit Register 2 Offset: H'18
ATP_MAILTR 3 — Arca-to-PCI Mailbox Transmit Register 3 Offset: H'1C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.4 Arca-to-PCI Mailbox Receive Register

The registers are used for PCI master to receive information from Arca. It can be read from PCI master and each byte of the register can be independently read. When read, they return current value and automatically update the corresponding status bit value in mailbox status register. When write, they ignore.

ATP_MAILRR 0 — Arca-to-PCI Mailbox Receive Register 0 PCI Address Offset: H'10
ATP_MAILRR 1 — Arca-to-PCI Mailbox Receive Register 1 Offset: H'14
ATP_MAILRR 2 — Arca-to-PCI Mailbox Receive Register 2 Offset: H'18
ATP_MAILRR 3 — Arca-to-PCI Mailbox Receive Register 3 Offset: H'1C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.5 PCIC Mailbox Status Register

Each bit of this register indicates whether each byte of the corresponding mailbox register contains unread data, Table 5-5 describes the details. When write, it ignores.

PCIC_MAILSR — PCIC Mailbox Status Register

Arca Address Offset: H'20

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0 = Mailbox byte is empty

1 = Mailbox byte is full

Table 5-5 PCIC_MAILSR Bit Meaning

Bit	Meaning	Bit	Meaning
[0]	ATP_MAILTR0/ATP_MAILRR0 Byte 0	[16]	PTA_MAILTR0/PTA_MAILRR0 Byte 0
[1]	ATP_MAILTR0/ATP_MAILRR0 Byte 1	[17]	PTA_MAILTR0/PTA_MAILRR0 Byte 1
[2]	ATP_MAILTR0/ATP_MAILRR0 Byte 2	[18]	PTA_MAILTR0/PTA_MAILRR0 Byte 2
[3]	ATP_MAILTR0/ATP_MAILRR0 Byte 3	[19]	PTA_MAILTR0/PTA_MAILRR0 Byte 3
[4]	ATP_MAILTR1/ATP_MAILRR1 Byte 0	[20]	PTA_MAILTR1/PTA_MAILRR1 Byte 0
[5]	ATP_MAILTR1/ATP_MAILRR1 Byte 1	[21]	PTA_MAILTR1/PTA_MAILRR1 Byte 1
[6]	ATP_MAILTR1/ATP_MAILRR1 Byte 2	[22]	PTA_MAILTR1/PTA_MAILRR1 Byte 2
[7]	ATP_MAILTR1/ATP_MAILRR1 Byte 3	[23]	PTA_MAILTR1/PTA_MAILRR1 Byte 3
[8]	ATP_MAILTR2/ATP_MAILRR2 Byte 0	[24]	PTA_MAILTR2/PTA_MAILRR2 Byte 0
[9]	ATP_MAILTR2/ATP_MAILRR2 Byte 1	[25]	PTA_MAILTR2/PTA_MAILRR2 Byte 1
[10]	ATP_MAILTR2/ATP_MAILRR2 Byte 2	[26]	PTA_MAILTR2/PTA_MAILRR2 Byte 2
[11]	ATP_MAILTR2/ATP_MAILRR2 Byte 3	[27]	PTA_MAILTR2/PTA_MAILRR2 Byte 3
[12]	ATP_MAILTR3/ATP_MAILRR3 Byte 0	[28]	PTA_MAILTR3/PTA_MAILRR3 Byte 0
[13]	ATP_MAILTR3/ATP_MAILRR3 Byte 1	[29]	PTA_MAILTR3/PTA_MAILRR3 Byte 1
[14]	ATP_MAILTR3/ATP_MAILRR3 Byte 2	[30]	PTA_MAILTR3/PTA_MAILRR3 Byte 2
[15]	ATP_MAILTR3/ATP_MAILRR3 Byte 3	[31]	PTA_MAILTR3/PTA_MAILRR3 Byte 3

5.7.1.6 PCI Mailbox Status Register

Each bit of this register indicates whether each byte of the corresponding mailbox register contains unread data, Table 5-6 describes the details. When write, it ignores.

PCI_MAILSR — PCI Mailbox Status Register

Address Offset: H'20

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0 = Mailbox byte is empty

1 = Mailbox byte is full

Table 5-6 PCI_MAILSR Bit Meaning

Bit	Meaning	Bit	Meaning
[0]	PTA_MAILTR0/PTA_MAILRR0 Byte 0	[16]	ATP_MAILTR0/ATP_MAILRR0 Byte 0
[1]	PTA_MAILTR0/PTA_MAILRR0 Byte 1	[17]	ATP_MAILTR0/ATP_MAILRR0 Byte 1
[2]	PTA_MAILTR0/PTA_MAILRR0 Byte 2	[18]	ATP_MAILTR0/ATP_MAILRR0 Byte 2
[3]	PTA_MAILTR0/PTA_MAILRR0 Byte 3	[19]	ATP_MAILTR0/ATP_MAILRR0 Byte 3
[4]	PTA_MAILTR1/PTA_MAILRR1 Byte 0	[20]	ATP_MAILTR1/ATP_MAILRR1 Byte 0
[5]	PTA_MAILTR1/PTA_MAILRR1 Byte 1	[21]	ATP_MAILTR1/ATP_MAILRR1 Byte 1
[6]	PTA_MAILTR1/PTA_MAILRR1 Byte 2	[22]	ATP_MAILTR1/ATP_MAILRR1 Byte 2
[7]	PTA_MAILTR1/PTA_MAILRR1 Byte 3	[23]	ATP_MAILTR1/ATP_MAILRR1 Byte 3
[8]	PTA_MAILTR2/PTA_MAILRR2 Byte 0	[24]	ATP_MAILTR2/ATP_MAILRR2 Byte 0
[9]	PTA_MAILTR2/PTA_MAILRR2 Byte 1	[25]	ATP_MAILTR2/ATP_MAILRR2 Byte 1
[10]	PTA_MAILTR2/PTA_MAILRR2 Byte 2	[26]	ATP_MAILTR2/ATP_MAILRR2 Byte 2
[11]	PTA_MAILTR2/PTA_MAILRR2 Byte 3	[27]	ATP_MAILTR2/ATP_MAILRR2 Byte 3
[12]	PTA_MAILTR3/PTA_MAILRR3 Byte 0	[28]	ATP_MAILTR3/ATP_MAILRR3 Byte 0
[13]	PTA_MAILTR3/PTA_MAILRR3 Byte 1	[29]	ATP_MAILTR3/ATP_MAILRR3 Byte 1
[14]	PTA_MAILTR3/PTA_MAILRR3 Byte 2	[30]	ATP_MAILTR3/ATP_MAILRR3 Byte 2
[15]	PTA_MAILTR3/PTA_MAILRR3 Byte 3	[31]	ATP_MAILTR3/ATP_MAILRR3 Byte 3

5.7.1.7 PCI-to-Arca Doorbell Register

The register is used by a PCI master to communicate with Arca. If mailbox interrupts are enabled, an Arca interrupt is generated if any PTA_DB bit is set. Each bit is set when written 1 from the PCI master and is cleared when written 1 from the Arca. When read by PCI master or Arca, it returns the value. This register is used both in host and satellite mode.

PTA_DB — PCI-to-Arca Doorbell Register

PCI Address Offset: H'24

Arca Address Offset: H'24

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.8 Arca-to-PCI Doorbell Register

The register is used for Arca to communicate with PCI master. If mailbox interrupts are enabled, A '1' in any bit of this register asserts the pci_inta_n interrupt request signal of the PCI bus, this function is used only in satellite mode. Each bit is set when written 1 from the Arca and is cleared when written 1 from the PCI master. When read by Arca or PCI master, it returns the value.

ATP_DB — Arca-to-PCI Doorbell Register

Arca Address Offset: H'28

PCI Address Offset: H'28

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.9 PCIC Configuration Address Register

PCIC_CFGAR — PCIC Configuration Address Register

Arca Address Offset: H'2C

Bit:	31	30	29	28	27	26	25	24
Read:	Enable	0	0	0	0	0	0	0
Write:								
Reset:	1	0	0	0	0	0	0	0
Bit:	23	22	21	20	19	18	17	16
Read:	Bus number							
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	15	14	13	12	11	10	9	8
Read:	Device number					Function number		
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	7	6	5	4	3	2	1	0
Read:	Register number						0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 30~24 These bits are always read as 0 and written are ignored.

- **Enable (Enable bit):**
0 = Disable configuration
1 = Enable configuration
- **Bus number (PCI Bus Number):** It is used to identify one of the 256 possible PCI bus segments in the target system.
- **Device number (PCI Device Number):** It is used to identify one of the 32 possible PCI physical devices on a PCI bus segment. It is an encoded value used to select one of 32 devices on a given bus. (There are only 21 devices that can be selected by tying the **IDSEL** to an **AD (AD[31::11])** line.)
- **Function number (PCI Function Number):** It is used to identify one of the 8 possible functions within the target. Each function has a separate configuration space.
- **Register number (PCI Configuration Register Address):** It is used to identify one of the 64 possible 32-bit registers in the target's configuration space.

5.7.1.10 PCIC Configuration Data Register

PCIC configuration data register contains the data value of configuration read/write access.

PCIC_CFGDR — PCIC Configuration Data Register

Arca Address Offset: H'30

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.11 PCIC Status Register

PCIC_SR — PCIC Status Register

Arca Address Offset: H'34

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	INTD	INTC	INTB	INTA	APER	DPER	MA	TA
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	MTRS				MRRS			
Write:								
Reset:	1	1	1	1	0	0	0	0

Bits 31~16 These bits are always read as 0 and written are ignored.

- **INTD (PCI Bus Interrupt D Status):**
0 = No interrupt
1 = Interrupt D is asserted (it is only set when Arca is in host mode)
- **INTC (PCI Bus Interrupt D Status):**
0 = No interrupt
1 = Interrupt C is asserted (it is only set when Arca is in host mode)
- **INTB (PCI Bus Interrupt D Status (INTB)):**

- 0 = No interrupt
- 1 = Interrupt B is asserted (it is only set when Arca is in host mode)
- **INTA (PCI Bus Interrupt D Status):**
 - 0 = No interrupt
 - 1 = Interrupt A is asserted (it is only set when Arca is in host mode)
- **APER (Address Parity Error Status):**
 - 0 = No address parity error
 - 1 = Address parity error
- **DPER (Data Parity Error Status):**
 - 0 = No data parity error
 - 1 = Data parity error
- **MA (Master Abort):**
 - 0 = No master abort
 - 1 = Master abort generates
- **TA (Target Abort):**
 - 0 = No target abort
 - 1 = Target abort generates (a received target abort when as a bus master or a signaled target abort when as a bus target)
- **MTRS (Mailbox Transmit Register Status):** Each bit of the field indicates the status of ATP_MAILTR0-3, respectively.
 - 0 = Mailbox is not empty, there have valid data in the mailbox.
 - 1 = Mailbox is empty, every byte of the mailbox have not valid data.
- **MRRS (Mailbox Receive Register Status):** Each bit of the field indicates the status of PTA_MAILRR0-3, respectively.
 - 0 = Mailbox is not full, there may be one to three bytes have not valid data.
 - 1 = Mailbox is full, all the four bytes of the mailbox contain valid data.

5.7.1.12 PCI Status Register

PCI_SR — PCI Status Register

PCI Address Offset: H'34

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:	0	0	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:					APERP	DPERP	MAP	TAP
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	MTRSP				MRRSP			
Write:								
Reset:	1	1	1	1	0	0	0	0

Bits 31~12: These bits are always read as 0 and written are ignored.

- **APERP (Address Parity Error Status from PCI side):**
0 = No address parity error
1 = Address parity error
- **DPERP (Data Parity Error Status from PCI side):**
0 = No data parity error
1 = Data parity error
- **MAP (Master Abort from PCI side):**
0 = No master abort
1 = Master abort generates
- **TAP (Target Abort from PCI side):**
0 = No target abort
1 = Target abort generates (a received target abort when as a bus master or a signaled target abort when as a bus target)
- **MTRSP (Mailbox Transmit Register Status from PCI side):** Each bit of the field indicates the status of PTA_MAILTR0-3, respectively.
0 = Mailbox is not empty, there have valid data in the mailbox.
1 = Mailbox is empty, every byte of the mailbox have not valid data.
- **MRRSP (Mailbox Receive Register Status from PCI side):** Each bit of the field indicates the status of ATP_MAILRR0-3, respectively.
0 = Mailbox is not full, there may be one to three bytes have not valid data.
1 = Mailbox is full, all the four bytes of the mailbox contain valid data.

5.7.1.13 PCIC Interrupt Enable Register

PCIC_IER — PCIC Interrupt Enable Register

Arca Address Offset: H'38

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:						MIE	IE	PMD
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 31~3: These bits are always read as 0 and written are ignored.

- **PMD (PCIC Mode):** This bit determines whether PCIC is operating in host mode or satellite mode. When write, it ignores and it is read-only and the value after reset is determined by mode pin of the chip.
0 = Satellite mode
1 = Host mode
- **IE (Interrupt Enable):** This bit determines whether to enable or disable PCIC interrupt function.
0 = Disable any interrupt
1 = Enable interrupt
- **MIE (Mailbox Interrupt Enable):** This bit determines whether to enable or disable PCIC mailbox interrupt function.
0 = Disable mailbox interrupt
1 = Enable mailbox interrupt

5.7.1.14 PCI Interrupt Enable Register

PCI_IER — PCI Interrupt Enable Register

PCI Address Offset: H'3C

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:							MIEP	IEP
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 31~2: These bits are always read as 0 and written are ignored.

- **IEP (Interrupt Enable):** This bit determines whether to enable or disable PCI interrupt function.
0 = Disable any interrupt
1 = Enable interrupt
- **MIEP (Mailbox Interrupt Enable):** This bit determines whether to enable or disable PCI mailbox interrupt function.
0 = Disable mailbox interrupt
1 = Enable mailbox interrupt

5.7.1.15 PCI-to-Arca Memory Address Translation Register

This register determines the size and the location of the local memory accessed by other PCI master. This register is initialized by power on or manual reset.

PTA_MEMATR — PCI-to-Arca Memory Address Translation Register

Arca Address Offset: H'40

Bit:	31	30	29	28	27	26	25	24
Read:	ML							
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	23	22	21	20	19	18	17	16
Read:	ML							
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	7	6	5	4	3	2	1	0
Read:					MS			
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 17~4: these bits are always read as 0 and written are ignored.

- **MS (Memory Size):** This bits indicates the size of the local memory defined by PCI Memory Base Register 0. When in satellite mode, this value must be loaded into by the Arca CPU during power-on initialization, prior to the system processor reading the PCIC configuration space.

Table 5-7 Memory Size As PCI Slave

Value	Memory Size
0000	256Mbyte
0001	128Mbyte
0010	64Mbyte
0011	32Mbyte
0100	16Mbyte
0101	8Mbyte
0110	4Mbyte
0111	2Mbyte
1000	1Mbyte
1001	512Kbyte
1010	256Kbyte
1011~1111	Reserved

- **ML (Memory Location):** These bits indicate the Arca local memory base address (physical address) defined by Base Address Register 0. This field is mapped to top bits of Arca bus physical address. The number of bits mapped depends on memory size defined by MS.

5.7.1.16 Arca-to-PCI Memory Translation Register (Master access)

When generating memory read/write command, Arca-to-PCI memory translation register contents are mapped to the high order 5 bits of OCS-Bus address concatenated with the left 27-bits offset to form the 32-bit PCI address.

ATP_MTR0 is used when access **H'C8000000 – H'FFFFFFF** physical address space.

ATP_MTR1 is used when access **H'D0000000 – H'D7FFFFFFF** physical address space.

ATP_MTR2 is used when access **H'D8000000 – H'DFFFFFFF** physical address space.

ATP_MTR 0 — Arca-to-PCI Memory Translation Register 0 Arca Address Offset: H'44

ATP_MTR 1 — Arca-to-PCI Memory Translation Register 1 Arca Address Offset: H'48

ATP_MTR 2 — Arca-to-PCI Memory Translation Register 2 Arca Address Offset: H'4C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.17 Arca-to-PCI IO Translation Register (Master access)

When generating IO read/write command, Arca-to-PCI IO translation register contents are mapped to the high order 5 bits of OCS-Bus address concatenated with the left 27-bits offset to form the 32-bit PCI address.

ATP_IOTR is used when access **H'C0000000 – H'C7FFFFFFF** physical address space.

ATP_IOTR — Arca-to-PCI IO Translation Register Arca Address Offset: H'50

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.1.18 PCIC Interrupt Acknowledge Register

When in *host mode* Arca210 generates an interrupt acknowledge cycle on the PCI bus by a read to the PCI_INTACK register and this register contains the interrupt vector number read from PCI bus. This register is read-only and initialized by power on or manual reset.

The read-data phase of the corresponding OCS-Bus read transaction is delayed until the PCI cycle completes and data is returned.

PCIC_INTACK — PCIC Interrupt Acknowledge Register

Arca Address Offset: H'54

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:	Interrupt Vector Number															
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:	Interrupt Vector Number															
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.2 PCI Configuration Registers

The PCIC has the configuration registers required by the PCI specification. For operation as a satellite, these registers will be configured by the other host processor. When Arca210 is operating as a PCI Host, the registers will be configured by the Arca CPU. The base address for accessing all of these PCI configuration registers by Arca210 is **H'E100_01xx**.

Table 5-8 PCI Configuration Registers

31	16		15	0	Address
Device ID (H'0001)			Vendor ID (H'8888)		00h
Status			Command		04h
Class Code (H'02 80 00)				Revision ID (1)	08h
BIST (0)		Header Type (0)	Latency Timer	Cache Line Size	0Ch
Base Address Register 0 (Arca memory space)					10h
Base Address Register 1 (PCIC control registers)					14h
Reserved Base Address Register					18h
Reserved Base Address Register					1Ch
Reserved Base Address Register					20h
Reserved Base Address Register					24h
Reserved					28h
SubSystem ID			SubSystem Vendor ID		2Ch
Expansion ROM Base Address, not used (0)					30h
Reserved					34h
Reserved					38h
Max_Lat (H'F0)		Min_Gnt (1)	Interrupt Pin(1)	Interrupt Line (1)	3Ch
			Retry_Timer (H'80)	Trdy_Timer (H'80)	40h

5.7.2.1 Vendor ID Register

This register specifies the vendor of the device.

Arca Address Offset: H'00

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:	Vendor ID															
Write:																
Reset:	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

5.7.2.2 Device ID Register

This read/write register specifies the ID to identify this device.

Arca Address Offset: H'02

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:	Device ID															
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

5.7.2.3 PCI Command Register

The PCI command register is a read/write register that provides protocol control to generate and respond to PCI cycles.

Arca Address Offset: H'04

Bit:	15	14	13	12	11	10	9	8
Read:							Fast_Ba ck_to_B ack_En	System_ Error_En
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	Wait_C ycle	Parity_ Error_ En	VGA_P alette	MWINV	Special _Cycle	Master _En	Mem_A ccess_ En	IO_Acce ss_En
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 15~10: reserved. Writing is invalid. Always read as 0.

- **IO_Access_En:** This bit is fixed to 0 which means Arca doesn't response to I/O commands from PCI bus.
- **Mem_Access_En:** This bit specifies whether Arca responses to memory commands from PCI bus.
0 = Memory access is disabled
1 = Memory access is enabled

- **Master_En:** This bit controls whether Arca can be a master on PCI bus.
0 = Master mode is disabled
1 = Master mode is enabled
- **Special_Cycle:** This bit specifies whether Arca can respond the special cycle commands from PCI bus. It is fixed 0 which means Arca doesn't respond special cycle command.
- **MWINV (Memory Write and Invalidate Enable):** This bit specifies whether Arca can generate the memory write and invalidate command to PCI bus. It is fixed 0 which means Arca doesn't generate memory write and invalidate command.
- **VGA_Palette (VGA Palette Snoop Enable):** This bit specifies how Arca can respond PCI command to VGA palette. It is fixed 0 which means Arca doesn't have snoop capability.
- **Parity_Error_En (Parity Error Enable):** This bit specifies whether Arca asserts PERR# when a parity error is detected during commands from another PCI master. It is fixed with a value of 0 which means Arca does not assert PERR# when a parity error is detected.
- **Wait_Cycle:** This bit specifies whether Arca when as a bus master, is capable of performing address/data stepping. It is fixed with a value of 0 which means Arca does not do address/data stepping.
- **System_Error_En (System Error Enable):** This bit specifies whether Arca asserts SERR# when a system error is detected. It is fixed with a value of 0 which means Arca does not assert SERR#.
- **Fast_Back_to_Back_En (Fast back-to-back Enable):** This bit specifies whether Arca when as a master perform fast back-to-back operation. It is fixed with a value of 0 which means doesn't perform this operation.

5.7.2.4 PCI Status Register

Arca Address Offset: H'06

Bit:	15	14	13	12	11	10	9	8
Read:	Detect_Error	Signaled_System_Error	Master_Abort_Status	Receive_Target_Abort_Status	Signaled_Target_Abort_Status	Device_Select_Timing		Data_Parity_Error_Detected
Write:								
Reset:	0	0	0	0	0	0	1	0

Bit:	7	6	5	4	3	2	1	0
Read:	Fast_Back_to_Back	UDF_Support	66Mhz_Capable					
Write:								
Reset:	1	0	0	0	0	0	0	0

Bits 4~0: reserved. Writing is invalid. Always read as 0.

- **66Mhz_Capable:** This bit specifies whether Arca is capable 66Mhz. It is fixed 0 which means Arca is not capable of 66Mhz.

- **UDF_Support:** This bit is fixed 0 which means Arca doesn't support user-definable PCI feature.
- **Fast_Back_to_Back:** This bit is fixed 1 which means Arca as a target can accept fast back-to-back operation.
- **Data_Parity_Error_Detected:** This bit is fixed 0 which means Arca as a master doesn't assert or detect PERR#.
- **Device_Select_Timing:** These bits indicate the timing of PCI_DEVSEL# when Arca as a target device. The fixed value 01 indicates Arca can only at Medium respond speed.
- **Signal_Target_Abort_Status:** When Arca as a PCI target, this bit is set when Arca terminates a transaction with a target abort.
- **Received_Target_Abort_Status:** When Arca as a PCI master, this bit is set when the transaction is terminated by the current-addressed target.
- **Master_Abort_Status:** When Arca as a PCI master, this bit is set when Arca is terminated by a master abort.
- **Signaled_System_Error:** This bit is fixed to 0 which means Arca can't assert SERR# signal.
- **Detect_Error:** This bit is set when means a parity error is detected.

5.7.2.5 Device Revision ID Register

This read-only register specifies the current revision Id for the device.

								Arca Address Offset: H'08
Bit:	7	6	5	4	3	2	1	0
Read:	Revision ID							
Write:								
Reset:	0	0	0	0	0	0	0	1

5.7.2.6 Class-Code Register

The class code register is read-only and is used to identify the generic function of the device and if possible, a specific register-level programming interface. The register has three byte-size fields.

								Arca Address Offset: H'09
Bit:	31	30	29	28	27	26	25	24
Read:	Class Code							
Write:								
Reset:	0	0	0	0	0	0	1	0
Bit:	23	22	21	20	19	18	17	16
Read:	Sub-Class Code							
Write:								

Reset:	1	0	0	0	0	0	0	0
Bit:	15	14	13	12	11	10	9	8
Read:	Programming Interface							
Write:								
Reset:	0	0	0	0	0	0	0	0

5.7.2.7 Cache Line Size Register

The cacheline size register specifies the system cacheline size in units of 32-bit words. It's fixed to 0 which means Arca doesn't support Memory Write and Invalidate command.

Arca Address Offset: H'0C

Bit:	7	6	5	4	3	2	1	0
Read:	Cache Line Size							
Write:								
Reset:	0	0	0	0	0	0	0	0

5.7.2.8 Master Latency Timer Register

This register is initialized by PCI reset. The Master Latency Time Register is an 8-bit read/write register that controls the amount of time that the Arca, when as a PCI bus Master, can perform burst transfers if another Master requests the bus.

Arca Address Offset: H'0D

Bit:	15	14	13	12	11	10	9	8
Read:	Master Latency Timer Count Value							
Write:								
Reset:	0	0	0	0	0	0	0	0

5.7.2.9 Header Type

Refer to PCI2.1 specification for Header Type. This field specifies the format of the device's configuration space registers. Arca uses a fixed value of H'00. Arca has a function type fixed to 0 which means it is a single-function device.

Arca Address Offset: H'0E

Bit:	23	22	21	20	19	18	17	16
Read:	Function Type	Header Type						
Write:								
Reset:	0	0	0	0	0	0	0	0

5.7.2.10 BIST (Built-In Self Test)

This optional register is used for initiating a BIST and for monitoring the result of executing a BIST sequence. Arca does not support BIST.

Arca Address Offset: H'0F

Bit:	31	30	29	28	27	26	25	24
Read:	BIST							
Write:								
Reset:	0	0	0	0	0	0	0	0

5.7.2.11 PCI Memory Base Address Register 0

This register contains the base address through which the PCI memory space is accessed and defines PCI address range assigned to Arca memory.

[The Arca PCIC has two 32-bit base address registers both requesting memory-mapped resources. One defines the PCI address range used by the mailbox and mailbox interrupt registers and the other defines the PCI slave address range assigned to Arca memory. These registers are initialized by PCI reset.]

PCI_MBR0 — PCI Memory Base Address Register 0

Arca Address Offset: H'10

Bit:	31	30	29	28	27	26	25	24
Read:	Memory Base							
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:					Prefetchable	Memory_Type		Memory_Space
Write:								
Reset:	0	0	0	0	1	0	0	0

Bits 27~4: reserved. These bits are always read as 0 and written are ignored.

- **Memory_Base:** This field defines the actual location in PCI address space of memory connected to Arca which is accessible by the PCI master.
- **Prefetchable:** This bit is fixed to 1 which means the memory space defined by this base address register is prefetchable.
- **Memory_Type:** These bits are fixed to 0 which means the memory space defined by this base address register can be located anywhere in 32-bit address space.
- **Memory_Space:** This bit is fixed to 0 which means the memory space defined by this base address register is memory not I/O.

5.7.2.12 PCI Memory Base Address Register 1

This register contains the base address through which the PCI memory space is accessed and defines PCI address range assigned to Arca PCIC control registers.

PCI_MBR1— PCI Memory Base Address Register 1

Arca Address Offset: H'14

Bit:	31	30	29	28	27	26	25	24
Read:	Memory Base							
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	23	22	21	20	19	18	17	16
Read:	Memory Base							
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	15	14	13	12	11	10	9	8
Read:	Memory Base							
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	7	6	5	4	3	2	1	0
Read:					Prefetchable	Memory_Type		Memory_Space
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 11~4: reserved. These bits are always read as 0 and written are ignored.

- **Memory_Base:** This field defines the actual location in PCI address space of memory connected to Arca which is accessible by the PCI master.
- **Prefetchable:** This bit is fixed to 0 which means the memory space defined by this base address register is not prefetchable.
- **Memory_Type:** These bits are fixed to 0 which means the memory space defined by this base address register can be located anywhere in 32-bit address space.
- **Memory_Space:** This bit is fixed to 0 which means the memory space defined by this base address register is memory not I/O.

5.7.2.13 SubSystem Vendor ID Register

The read-only register identifies the vendor of the subsystem where the PCI device resides.

Arca Address Offset: H'2C

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:	Subsystem Vendor ID															
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.2.14 Subsystem ID

The read-only register identifies the subsystem where the PCI device resides.

Arca Address Offset: H'2E

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read:	Subsystem Vendor ID															
Write:																
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.7.2.15 Interrupt Line Register

When Arca as a satellite, this register is used to specify which of the system interrupt request lines Arca's interrupt pin is connected to.

Arca Address Offset: H'3C

Bit:	7	6	5	4	3	2	1	0
Read:	Interrupt Line Register							
Write:								
Reset:	0	0	0	0	0	0	0	0

5.7.2.16 Interrupt Pin Register

This register is fixed to H'01 which means Arca uses the INTA# interrupt pin when as a satellite.

Arca Address Offset: H'3D

Bit:	15	14	13	12	11	10	9	8
Read:	Interrupt Pin Register							
Write:								
Reset:	0	0	0	0	0	0	0	1

5.7.2.17 Minimum Grant Register

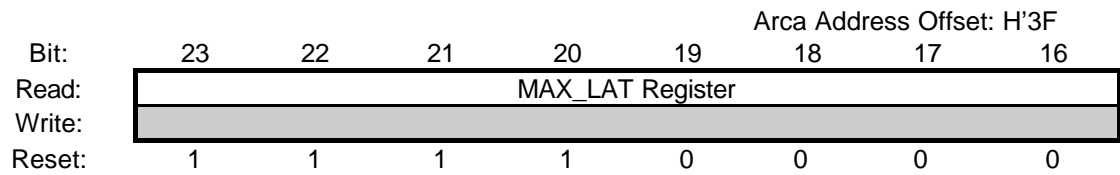
This register specifies how long a burst period the device needs. It is fixed to 1 in Arca.

Arca Address Offset: H'3E

Bit:	23	22	21	20	19	18	17	16
Read:	MIN_GNT Register							
Write:								
Reset:	0	0	0	0	0	0	0	1

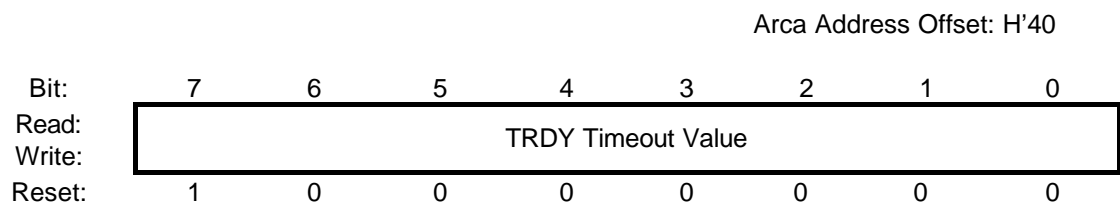
5.7.2.18 Maximum Latency Register

This register specifies how often the device needs to gain access to the PCI bus.



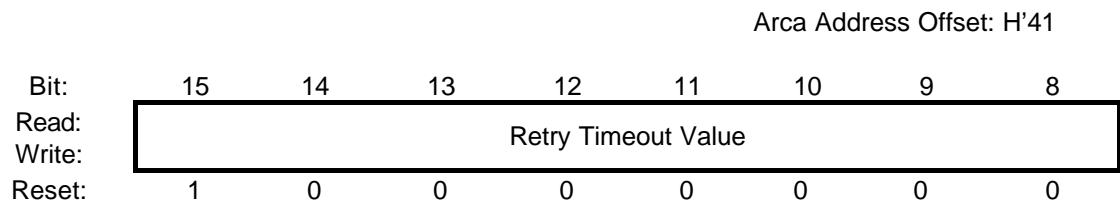
5.7.2.19 TRDY Timeout Value

This field specifies the maximum number of PCI clock periods which Arca, operating as a bus master, will wait for a TRDY response from an addressed target.



5.7.2.20 Retry Timeout Value

This field specifies the maximum number of consecutive retries that Arca, operating as a bus master, will perform when attempting to transfer data to/from a target.



5.8 PCI Bus Arbiter

Arca210 PCI bus arbiter is an on-chip PCI bus arbiter which could be used or not according to the choice of external pin, MD_pciarb. It selects which of the requesting bus masters will be granted use of the PCI bus by monitoring and arbitrating the pending requests of all the masters connected to the PCI bus arbiter and grants the highest priority master to use the bus.

5.8.1 Features and Block Diagram

- Support arbitrating up to 5 pending requests of masters, request 0, 1, 2, 3, 4, four external sources and one from Arca210 which is fixed to Request 0.
- Support fairness algorithm: rotational within group and fixed with groups.
- Totally two groups, request 0-1 belong to group A, request 2-4 belong to group B. Group A has higher priority and thus group B has lower priority.
- No lock function.

Figure 5-6 shows block diagram of the arbiter.

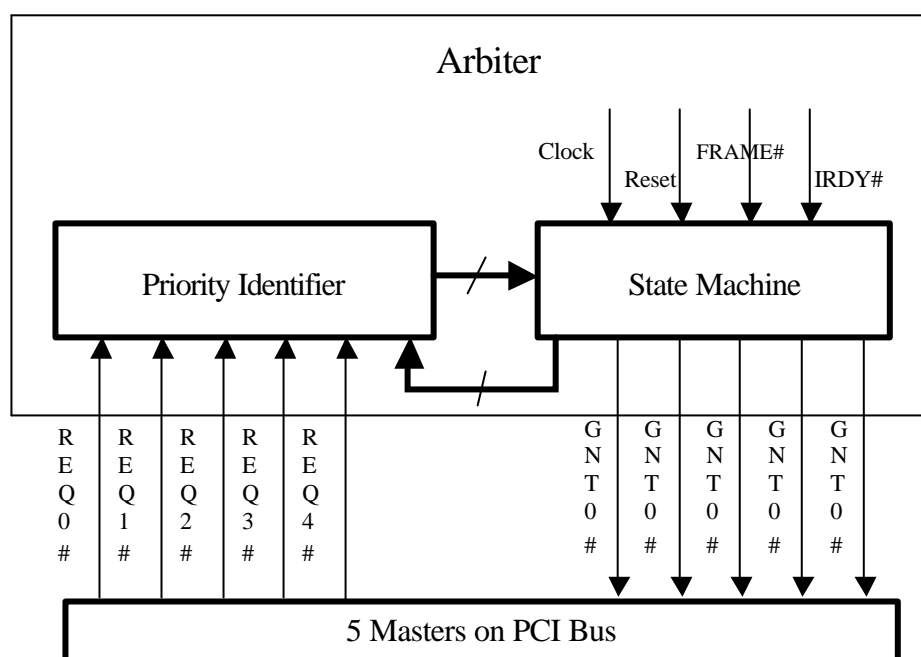


Figure 5-6 PCI Arbiter Diagram

5.8.2 Arbitration Algorithm

5.8.2.1 Priority

To avoid deadlock on PCI bus, Arca210 PCI Arbiter supports fairness algorithm in which the priority is rotational within a group and fixed between groups. Up to 5 pairs of request/ grant can be connected to the arbiter. The 5 requests are divided into 2 groups, and group A contains request 0 (master 0) and request 1 (master 1) and group B contains request 2-3 (master 2-3). Group A has a higher priority than group B. After power on reset, the default master of group A is master 0, and the default master of group B is master 2.

Within one group, the priority is rotational and last master that just uses the bus has the lowest priority. For example, in group A, if last master is master 0, the next master should be master 1 if it requests to use the bus; if last master is master 1, the next master should be a master of group B if it requests to use the bus.

Figure 5-7 shows the arbitration scheme.

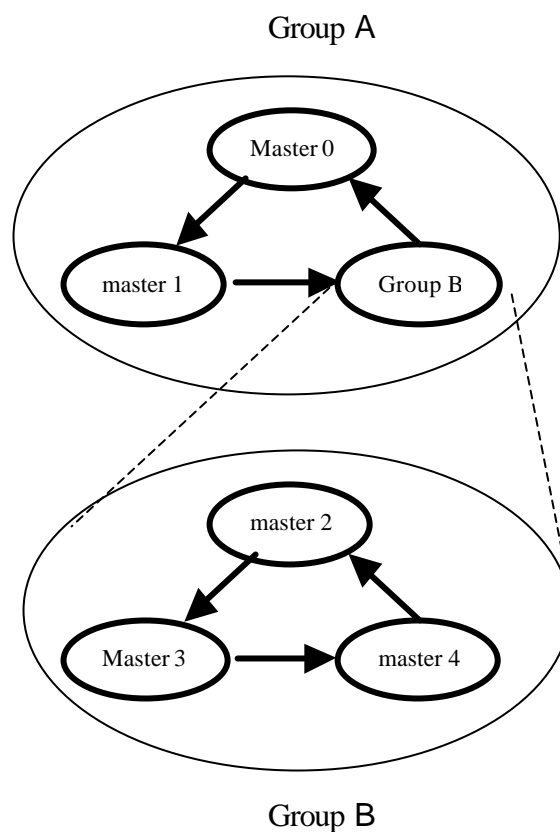


Figure 5-7 Arbitration Scheme

But between these two groups, the priority is fixed. That is, group A has higher priority than group B. For example, if the last master is a master of group B, the next master should be a master of group A if it requests to use the bus, no matter whether there is any other master of group B is requesting to use the bus.

5.8.2.2 Bus Parking

Arca210 PCI Arbiter supports bus parking function and the default bus master is the last master to use the bus. It means when a transaction initiated by a master ends and no other request is pending at that time, the arbiter asserts grant signal to it even though it doesn't assert its request signal and then it is the default master. So if the default master requests to use the bus and no other higher priority request is pending, the bus is immediately available to it when it samples the bus idle, furthermore, it doesn't need to assert its request signal for subsequently requiring access to the bus. Figure 5-8 shows the function of bus parking.

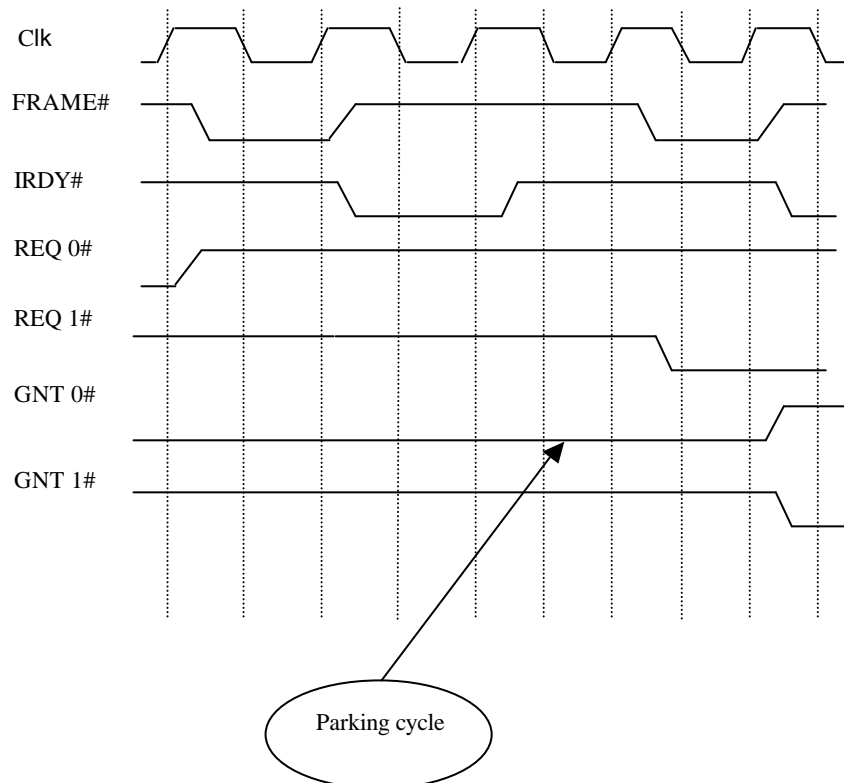


Figure 5-8 Bus Parking

After power on, the overall default master is master 0 although that of group B is master 2.

5.8.2.3 Bus Lock

Arca210 PCI bus arbiter doesn't support lock function.

5.8.3 Request/ Grant Timing

When the arbiter determines that it is a master's turn to use the bus, it asserts the master's GNT#, but may deassert it according to state of bus and other requests. Once asserted by the arbiter, GNT# may be deasserted under the following circumstances:

1. If GNT# is deasserted and FRAME# is asserted at the same PCI clock rising edge, the transfer is valid and will continue. The deassertion of GNT# by the arbiter indicates that the master will no longer own the bus at the completion of the transaction currently in progress, of course after subsequent transactions, it may own the bus again according to the arbitration of the arbiter.
2. The GNT# to one master can be deasserted simultaneously with the assertion of another master's GNT# **if the bus isn't in the idle state**. When the bus is in idle state and there is a new REQ# which doesn't match the default master, thus has a higher priority, the arbiter will pull up all GNT# to high level for one cycle and assert GNT# to the new master after the cycle of GNT# all high level.
3. Arbitration can take place while the current initiator is performing a data transfer in which either FRAME# or IRDY# is valid. When the master is in the midst of a burst transaction and the arbiter removes its GNT# granting the bus ownership to the other master, if the current master can not complete its burst transfer before expiration of its latency timer or its latency timer is zero, the master is permitted to complete one more data transfer and must then yield the bus.
4. If a master receives the GNT# from the arbiter by requesting the use of the bus, that means its GNT# and REQ# are all valid, the master should initiate a transaction within 16 PCI clocks, otherwise, the arbiter will remove its GNT# and the master become the last master that means its priority is the lowest.

5.8.4 Examples of Arbiter Operation

5.8.4.1 Multiple Requests in Bus Idle After Power On

REQ 0#, REQ 1#, REQ 2# are all valid at the same cycle when the bus is in the idle state after power on reset. Because the default master is master 0, its priority is the lowest and master 1 has a higher priority than master 2. Then master 1 is granted to use the bus. If master 0 and master 2 still need to use the bus, they keep their REQ# asserted. There is one cycle of all GNT# high level during transition because the bus is in idle state.

Refer to Figure 5-9.

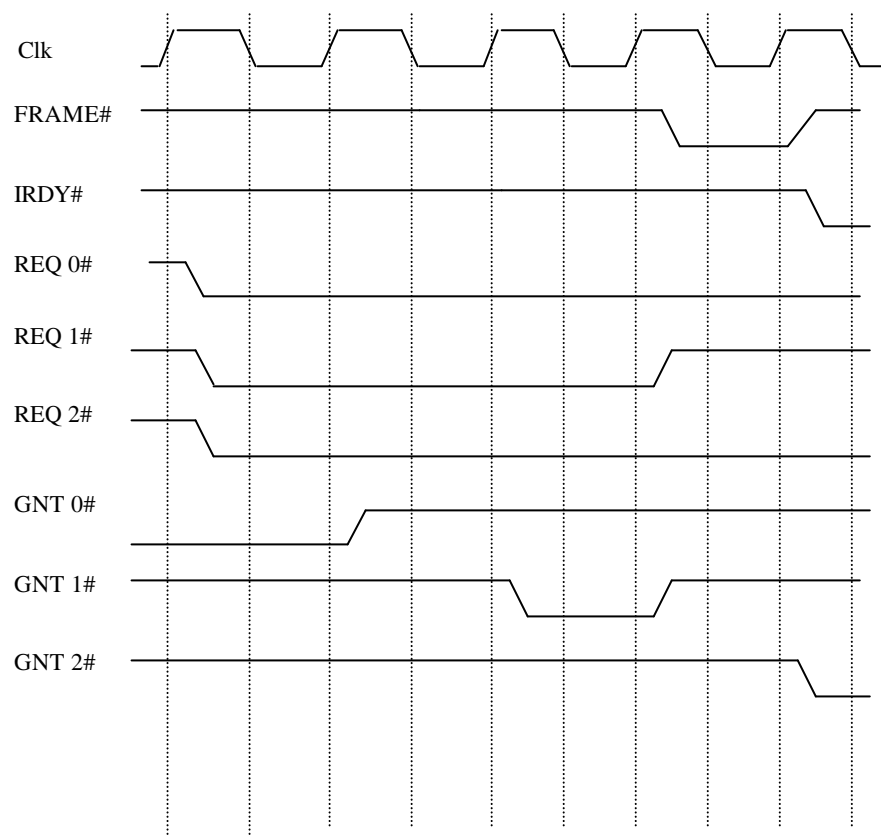


Figure 5-9 Arbitration in Bus Idle

5.8.4.2 Group A preempts Bus Ownership During Group B Data Transfer

The current master is master 2 and its burst data transfer is in progress, at the same time master 0, master 1 and master 3 are requesting the bus ownership during the data transfer. Because group A has a higher priority than group B, the priorities of master 0 and master 1 are higher than master 3 in this case. Because the master 0's priority is higher than master 1 according to rotational rule, the master 0 is granted. There is no one cycle of all GNT# high level during transition because the bus is not in idle state.

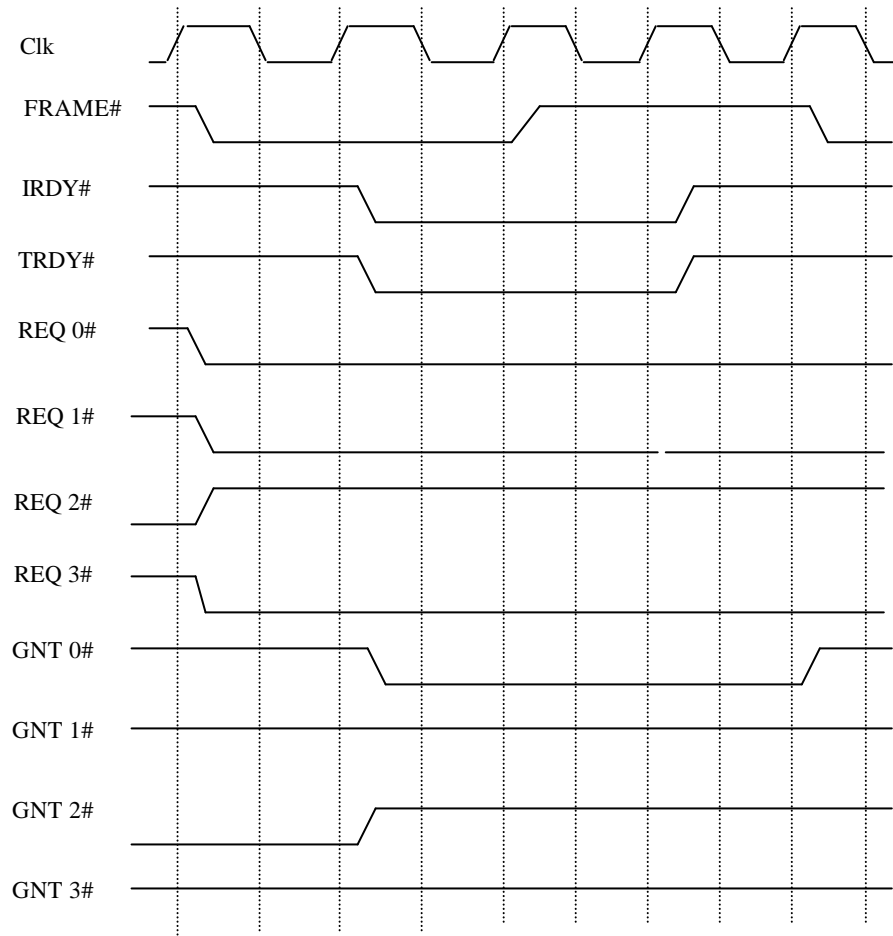


Figure 5-10 Master of Group B Preempted

6 Ethernet Controller

6.1 Overview

The Ethernet Controller (ETHC) provides the interface between the host application and the PHY layer through the Media Independent Interface (MII). The PHY layer devices are external to Arca and selected by the customers.

The Ethernet Controller Operates at either 100 Mbps or 10 Mbps in half-duplex or full-duplex mode. In half-duplex mode, the controller supports the IEEE802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. In full-duplex mode, it supports the IEEE802.3 MAC Control Layer, including the Pause operation for flow control.

6.1.1 Features

The ETHC has the following features:

- Compliant with IEEE802.3, 802.3u Specification
- 10/100 Mbps data transfer rates
- IEEE802.3 compliant MII interface to talk to an external PHY
- VLAN support
- Full and half duplex modes
- Support CSMA/CD protocol in half duplex mode
- Supports flow-control for full-duplex operation
- Collision detection and auto retransmission on collisions in half-duplex mode
- Automatic 32 bit CRC generation and checking
- Options to insert PAD/CRC32 on transmit
- Options to Automatic Pad stripping on the receive packets
- Provides External and internal loop back capability on the MII Interface
- Contains a variety of flexible address filtering modes on the Ethernet side:
 - One 48 bit Perfect address
 - 64 hash-filtered multicast addresses
 - Pass all multicast addresses
 - Promiscuous Mode
 - Pass all incoming packets with a status report
 - Support MII Management
- Tx/Rx buffers (2K bytes/2K bytes)
- DMA engine using burst mode
- Supports dual buffer and linked list Descriptor Chaining
- Descriptor architecture allows large blocks of data transfer with minimum CPU intervention

6.1.2 Block Diagram

Following figure illustrates a block diagram of ETHC.

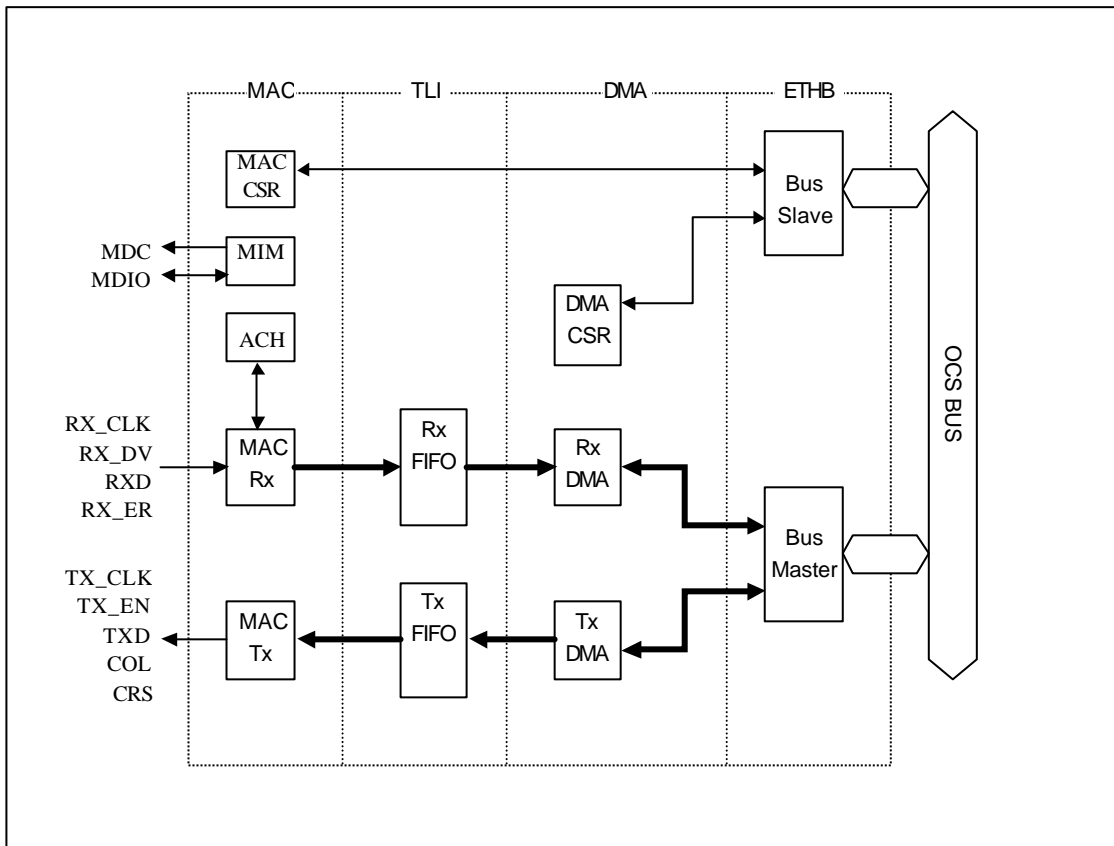


Figure 6-1 ETHC Block Diagram

ETHC is composed of four layers, a System Bus Bridge, a DMA layer, a Transaction layer Interface (TLI), and a Media Access Controller layer (MAC).

The DMA is the first layer of the Ethernet Subsystem, and it is responsible for exchanging data between the TLI layer, and the System Memory. A set of registers (DMA CSR) can be accessed by the host for controlling the DMA operation.

The TLI consists of 2 sets of FIFO, a Transmit (TX) FIFO with dynamic threshold, and a Receive (RX) FIFO with a fixed Threshold.

The MAC interfaces to the external physical layer (PHY) through a media independent interface (MII) bus. It handles all the Ethernet Protocol for both Full and Half duplex modes. The MAC also contains a set of registers (MAC CSR) for controlling and checking the status of the transmissions.

All Layers are VCI compliant that is, they use the VCI bus as the protocol to transfer data from one layer to another. VCI is the Virtual Component Interface. It is an emerging peripheral and system on a chip bus.

6.2 Pin Configuration

Table 6-1 ETHC Pin Configuration

Pin Name	I/O	Signal	Description
Receive Clock	I	RX_CLK	RX_CLK is a continuous clock that provides the timing reference for the transfer of the RX_DV, RX_ER, and RXD signals from the Ethernet PHY Controller to the MAC. RX_CLK is driven by the Ethernet PHY Controller chip. The RX_CLK shall have a frequency equal to 25% of the data rate of the received signal on the Ethernet Cable.
Receive Data Valid	I	RX_DV	RX_DV is driven by the external Ethernet PHY Controller to indicate the MAC that it is presenting the recovered and decoded nibbles on the RXD[3:0] bundle and that the data on RXD[3:0] is synchronous to RX_CLK. RX_DV shall be transferred synchronously with respect to the RX_CLK. RX_DV shall remain be asserted continuously from the first recovered nibble of the frame through the final recovered nibble, and shall be negated prior to the first RX_CLK that follows the final nibble. When asserted the RX_DV will be at logic '1' and it will be at logic '0' while de-asserted.
Receive Error	I	RX_ER	RX_ER is driven by the Ethernet PHY Controller chip. RX_ER shall be asserted for one or more RX_CLK periods to indicate to the MAC that an error (e.g., a coding error, or any error that the PHY is capable of detecting, and that otherwise be undetectable by the MAC) was detected some where in the frame presently being transferred from the PHY to the MAC. RX_ER shall be transferred synchronously with respect to RX_CLK. While RX_DV is de-asserted, RX_ER will have no effect on the MAC. When asserted the RX_ER will be at logic '1' and it will be at logic '0' while de-asserted.
Receive Data	I	RXD [3:0]	RXD is a bundle of four data signals RXD [3:0] that transition synchronously with respect to the RX_CLK. RXD[3:0] are driven by the Ethernet PHY Controller chip. For each RX_CLK period in which RX_DV is asserted, RXD [3:0] transfer four bits of recovered data from the PHY to the ETHC. RXD [0] is the least significant bit. While RX_DV is de-asserted, RXD [3:0] will have no effect on the ETHC.

Table 6-1 (continued)

Transmit Clock	I	TX_CLK	TX_CLK is a continuous clock that provides for the timing reference for the transfer of the TX_EN, TX_ER, and TXD signals from the MAC to the Ethernet PHY Controller. TX_CLK is driven by the Ethernet PHY Controller chip. The operating frequency of the TX_CLK is 25 MHz when operating at 100-Mb/s and 2.5 MHz when operating at 10-Mb/s.
Transmit Enable	O	TX_EN	TX_EN indicates that the MAC is presenting nibbles on the MII for transmission. It will be asserted by the Mac with the first nibble of the preamble and will remain asserted while all nibbles to be transmitted are presented to the MII. TX_EN will be negated prior to the first TX_CLK following the final nibble of the frame. TX_EN is driven by the MAC and will be transferred synchronously with respect to the TX_CLK. When asserted the TX_EN will be at logic '1' and it will be at logic '0' while de-asserted.
Transmit Data	O	TXD [3:0]	TXD is a bundle of 4 data signals TXD[3:0] that are driven by the MAC. TXD[3:0] will be transferred synchronously with respect to the TX_CLK. For each TX_CLK period in which TX_EN is asserted, TXD[3:0] will have the data to be accepted by the Ethernet PHY Controller chip. TXD[0] is the least significant bit. While TX_EN is de-asserted the data presented on TXD[3:0] should be ignored.
Carrier Sense	I	CRS	CRS shall be asserted by the Ethernet PHY Controller Chip when either transmit or receive medium is non idle. CRS shall be de-asserted by the PHY when both the TX medium and the RX medium are idle. The PHY shall ensure that CRS remains asserted throughout the duration of a collision condition. The transitions on the CRS signal are not synchronous to either the TX_CLK or the RX_CLK.
Collision	I	COL	COL shall be asserted by the Ethernet PHY Controller chip upon detection of a collision on the medium, and shall remain asserted while the collision condition persists. The transitions on the COL signal are not synchronous to either the TX_CLK or the RX_CLK. The MAC ignores the signal COL during Full Duplex Operation.

Table 6-1 (continued)

Management Clock	O	MDC	The MAC sources the MDC signal to the Ethernet PHY Controller as the timing reference for transfer of information on the MDIO signal. MDC is an aperiodic signal that has no maximum high or low times. The minimum high and low times for MDC will be 160 ns each, and the minimum period for MDC will be 400 ns, regardless of the nominal period of TXClk and RXClk.
Management Data Input/Output	IO	MDIO	<p>When input, MDIO is the data input signal from the Ethernet PHY. The Read Data is driven by the PHY synchronously with respect to the MDC clock during the read cycles.</p> <p>When output, MDIO is the data output signal from the MAC that is used to drive the control information during the Read/Write cycles to the Ethernet PHY. The MAC drives the MDIO signal synchronously with respect to the MDC.</p>

6.3 Register Configuration

Table 6-2 ETHC Register Configuration

Name	Full Name	R/W	Initial Value	Address	Access Width
CSR0	Bus Mode Register	R/W	H'00000000	H'E1041000	32
CSR1	Transmit Poll Demand Register	W	H'xxxxxxxx	H'E1041004	32
CSR2	Receive Poll Demand Register	W	H'xxxxxxxx	H'E1041008	32
CSR3	Receive list Base Address Register	R/W	H'00000000	H'E104100C	32
CSR4	Transmit list Base Address Register	R/W	H'00000000	H'E1041010	32
CSR5	Status Register	R/W	H'00000000	H'E1041014	32
CSR6	Control (Operation Mode) Register	R/W	H'00000000	H'E1041018	32
CSR7	Interrupt Enable Register	R/W	H'00000000	H'E104101C	32
CSR8	Missed Frame Counter (receive only)	R	H'00000000	H'E1041020	32
	Reserved			H'E1041024 ~ H'E104104F	
CSR20	Current Host Transmit Buffer Address Register	R	H'00000000	H'E1041050	32
CSR21	Current Host Receive Buffer Address Register	R	H'00000000	H'E1041054	32
	Reserved			H'E1041058 ~ H'E1041FFF	
MCR	MAC Control Register	R/W	H'00040000	H'E1040000	32
MAH	MAC Address High Register	R/W	H'0000FFFF	H'E1040004	32
MAL	MAC Address Low Register	R/W	H'FFFFFFFF	H'E1040008	32
HTH	Multi Cast Hash Table High Register	R/W	H'00000000	H'E104000C	32
HTL	Multi Cast Hash Table Low Register	R/W	H'00000000	H'E1040010	32
MIA	MII Address Register	R/W	H'00000000	H'E1040014	32
MID	MII Data Register	R/W	H'00000000	H'E1040018	32
FCR	Flow Control Register	R/W	H'00000000	H'E104001C	32
VTR1	VLAN1 Tag Register	R/W	H'0000FFFF	H'E1040020	32
VTR2	VLAN2 Tag Register	R/W	H'0000FFFF	H'E1040024	32
	Reserved			H'E1040028 ~ H'E1040FFF	

6.3.1 Bus Mode Register (CSR0)

The Bus Mode Register (CSR0) is a 32-bit read/write register that establishes the bus operating modes for the DMA controller. It is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:				DBO				
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:			PBL					
Write:								
Reset:	0	0	0					

Bit:	7	6	5	4	3	2	1	0
Read:	BLE	DSL						SWR
Write:								
Reset:	0	0					0	0

- **Bits 31~21,19~14, 1:** Reserved bits. Writes to these bits have no effect and always read as 0.
- **DBO (Descriptor Byte Ordering):** When set, the DMA controller operates in the big endian mode for descriptors. In the big endian mode descriptor byte lanes 0-3 and 1-2 are swapped. CSR bit positions within each byte are not changed. On reset this bit defaults to little endian mode.
- **PBL (Programmable Burst Length):** Specifies the maximum number of WORDs to be transferred in one DMA transaction. This will be the maximum value that is used in a single block read/write. The DMA controller will always attempt to burst length specified in PBL each time it starts a burst transfer on the host bus. PBL can be programmed with permissible values of 1 and 4. Any other value will result in unpredictable behavior. (default: B'000000)
- **BLE (Big/Little Endian):** Specifies byte order mode. When set, the DMA controller operates in big endian mode when accessing host data buffers. Buffer byte lanes 0-3 and 1-2 are swapped. On reset this bit defaults to little endian mode.
- **DSL (Descriptor Skip Length):** Specifies the number of WORDs to skip between two unchained descriptors. (default: B'000000)
- **SWR (Software Reset):** When set, the DMA controller resets all the internal hardware. This bit is automatically cleared by hardware. (default: B'0)

6.3.2 Transmit Poll Demand Register (CSR1)

The Transmit Poll Demand Register (CSR1) enables the Transmit DMA to check for any new descriptors.

It is a 32-bit WRITE ONLY register. When written with any value, the controller checks for frames to be transmitted. If no descriptor is available, the transmit process returns to the suspended state and CSR_5[2] is asserted. If the descriptor is available, the transmit process resumes. The Write is effective only if the transmit process is in the suspended state.

6.3.3 Receive Poll Demand Register (CSR2)

The Receive Poll Demand Register (CSR2) enables the Receive DMA to check for any new descriptors.

It is a 32-bit WRITE ONLY register. When written with any value, the checks for receive descriptors. If no descriptor is available, the receive process returns to the suspended state and CSR_5[7] is asserted. If the descriptor is available, the receive process resumes. The Write is effective only if the receive process is in the suspended state.

6.3.4 Receive/Transmit Descriptor List Base Address Registers (CSR3, CSR4)

The Receive Descriptor List Base Address Register (CSR3) is used for receive buffer descriptors and the Transmit Descriptor List Base Address Register (CSR4) is used for transmit buffer descriptors. In both cases, the registers are used to point to the start of the appropriate descriptor list.

The descriptor lists reside in host's physical memory space and must be WORD aligned. The DMA controller behaves UNPREDICTABLY when the lists are not WORD aligned.

Writing to CSR3 or CSR4 is permitted only when its respective process is in the stopped state. When stopped, the CSR3 and CSR4 registers must be written before the respective START command is given.

They are initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.

Receive Descriptor List Base Address Register (CSR3):

Bit:	31~2																													1	0
Read:	Start of Receive Descriptor List																														
Write:																															
Reset:	0																													0	0

Transmit Descriptor List Base Address Register (CSR4):

Bit:	31~2	1	0
Read:	Start of Transmit Descriptor List		
Write:			
Reset:	0	0	0

6.3.5 DMA Status Register (CSR5)

The Status Register (CSR5) is a 32-bit register that contains all the status bits that the DMA controller reports to the host. CSR5 is usually read by the driver during interrupt service routine or polling. Most of the fields in this register cause the host to be interrupted. CSR5 bits are not cleared when read. CSR5[31:17] are Read Only. Writing '1' to (non-Reserved) bits in CSR5[15:0] clears them and writing '0' has no effect. Each field can be masked by masking the appropriate bit in the CSR7.

CSR5 is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.

Bit:	31	30	29	28	27	26	25	24
Read:							EB	
Write:								
Reset:	0	0	0	0	0	0	0	

Bit:	23	22	21	20	19	18	17	16
Read:	EB	TS			RS			NIS
Write:								
Reset:	0			0			0	

Bit:	15	14	13	12	11	10	9	8
Read:	AIS	ERI	FBE			ETI	RWT	RPS
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	RU	RI	UNF		TJT	TU	TPS	TI
Write:								
Reset:	0	0	0	0	0	0	0	0

- **Bits 31~26, 12, 11, 4:** Reserved bits Writes to these bits have no effect and always read as 0.

- **EB (Error Bits):** Indicates the type of error that is caused by bus error. Valid only when fatal bus error, FBE bit of CSR5 is set. This field does not generate interrupt. This field is read only.

Bit 25~23: EB	Description	
000	No error	(Initial value)
001	Tx Abort: Host VCI received ABORT during TX process (descriptor or buffer access)	
010	Rx Abort: Host VCI received ABORT during RX process (descriptor or buffer access)	
1xx	Reserved	

- **TS (Transmit Process State):** Indicates the state of the transmit process. This field does not generate interrupt. This field is read only.

Bit 22~20: TS	Description	
000	Stopped - RESET command or STOP command issued	(Initial value)
001	Running - Fetching the transmit descriptor	
010	Running - Waiting for end of transmission	
011	Running - Reading the data from memory and queuing the data into the transmit buffer	
100	Reserved	
101	Reserved	
110	Suspended - Transmit buffer underflow, or an unavailable transmit descriptor	
111	Running - Closing transmit descriptor	

- **RS (Receive Process State):** Indicates the state of the receive process. This field does not generate interrupt. This field is read only.

Bit 19~17: RS	Description	
000	Stopped - RESET command or STOP RECEIVE command	(Initial value)
001	Running - Fetching the receive descriptor	
010	Running - Checking for end of receive packet before prefetch of next descriptor	
011	Running - Waiting for Receive Packet	
100	Suspended - Unavailable receive buffer	
101	Running - Closing receive descriptor	
110	Running - Flushing the current frame from the receive buffer because of unavailable receive buffer	
111	Running - Queuing the receive frame from the receive buffer into the host memory	

- **NIS (Normal Interrupt Summary):** Normal Interrupt Summary bit. Its value is the logical OR of:

CSR5[0] - Transmit Interrupt
 CSR5[2] - Transmit buffer unavailable
 CSR5[6] - Receive interrupt
 CSR5[14] - Early Receive interrupt

Only unmasked bits affect this field.

Note: This is a sticky bit and must be cleared each time a corresponding bit that causes NIS to be set is cleared. During the CSR access to clear this bit, the bit will be masked, once the bit is cleared, it will be allowed to set again if there is another pending interrupt.

Bit 16: NIS	Description	
0	No normal interrupt	(Initial value)
1	Normal interrupt occurs	

- **AIS (Abnormal Interrupt Summary):** Abnormal Interrupt Summary bit. Its value is the logical OR of:

CSR5[1] - Transmit Process Stopped
 CSR5[3] - Transmit jabber timeout
 CSR5[5] - Transmit Underflow
 CSR5[7] - Receive Buffer Unavailable
 CSR5[8] - Receive Process Stopped
 CSR5[9] - Receive Watchdog Timeout
 CSR5[10] - Early Transmit Interrupt
 CSR5[13] - Fatal Bus Error.

Only unmasked bits affect this field.

Note: This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared. During the CSR access to clear this bit, the bit will be masked. Once the bit is cleared, it will be allowed to set again if there is another pending interrupt.

Bit 15: AIS	Description	
0	No abnormal interrupt	(Initial value)
1	Abnormal interrupt occurs	

- **ERI (Early Receive Interrupt):** Indicates that the DMA controller had filled the first data buffer of the packet. Receive interrupt CSR5[6] automatically clears this bit.

Bit 14: ERI	Description	
0	No early receive interrupt	(Initial value)
1	Early receive interrupt occurs	

- **FBE (Fatal Bus Error):** Indicates that a bus error occurred. EB field (bit 25~23) reflect the error cause. When this bit is set, the DMA controller disables all its bus accesses.

Bit 13: FBE	Description	
0	No fatal bus error	(Initial value)
1	Fatal bus error occurs	

- **ETI (Early Transmit Interrupt):** Indicates that the packet to be transmitted was fully transferred into the destination port. Transmit interrupt of CSR5 (bit 0) clears this bit automatically.

Bit 10: ETI	Description	
0	No early transmit error	(Initial value)
1	Early transmit error occurs	

- **RWT (Receive Watchdog Timeout):** Reflects the line status and indicates that the receive watchdog timer has expired while another node is still active on the network.

Bit 9: RWT	Description	
0	No receive watchdog timeout	(Initial value)
1	Receive watchdog timeout occurs	

- **RPS (Receive Process Stopped)**

Bit 8: RPS	Description	
0	Receive process doesn't enter stopped state	(Initial value)
1	Receive process enters stopped state	

- **RU (Receive Buffer Unavailable):** Indicates that the next descriptor in the receive list is owned by the host and cannot be acquired by the DMA controller. The reception process is suspended. To resume processing of receive descriptors, the host should change the ownership of the descriptor and issue a receive poll demand command. If no receive poll demand is issued, the reception process resumes when the next recognized incoming frame is received. After the first assertion, this field is not asserted for any subsequent "not owned" receive descriptor fetches. This field is set only when the previous receive descriptor was owned by the DMA controller.

Bit 7: RU	Description	
0	Receive buffer is available	(Initial value)
1	Receive buffer is unavailable	

- **RI (Receive Interrupt):** Indicates the completion of the frame reception. Specific frame status information has been posted in the descriptor. The reception process remains in the running state.

Bit 6: RI	Description	
0	No receive interrupt	(Initial value)
1	Receive interrupt occurs	

- **UNF (Transmit Underflow):** Indicates that the transmit Buffer had an underflow condition during the packet transmission. The transmit process is placed in the suspended state and underflow error TDES0[1] is set.

Bit 5: UNF	Description	
0	No transmit underflow	(Initial value)
1	Transmit underflow occurs	

- **TJT (Transmit Jabber Timeout):** Indicates that the transmit jabber timer expired, meaning that the transmitter had been activated excessively. The transmission process is aborted and set to the stopped state. This even causes the transmit jabber timeout TDES0[14] flag to assert.

Bit 3: TJT	Description	
0	No transmit jabber timeout	(Initial value)
1	Transmit jabber timeout occurs	

- **TU (Transmit Buffer Unavailable):** Indicates that the next descriptor in the transmit list is owned by the host and cannot be acquired by the DMA Controller. The transmission process is suspended. TS field explains the transmit process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a transmit poll demand command, unless transmit automatic polling is enabled.

Bit 2: TU	Description	
0	Transmit buffer is available	(Initial value)
1	Transmit buffer is unavailable	

- **TPS (Transmit Process Stopped)**

Bit 1: TPS	Description	
0	Transmit process doesn't enter stopped state	(Initial value)
1	Transmit process enters stopped state	

- **TI (Transmit Interrupt):** Indicates that a frame transmission was completed and TDES1[31] is set in the first descriptor.

Bit 0: TI	Description	
0	No transmit interrupt	(Initial value)
1	Transmit interrupt occurs	

6.3.6 Control (Operation Mode) Register (CSR6)

The Control Register (CSR6) is a 32-bit register that establishes the RX and TX operating modes and commands. CSR6 should be the last CSR to be written as part of initialization.

CSR6 is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:		TTM	SF					
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	TR		ST					
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:						OSF	SR	
Write:								
Reset:	0	0	0	0	0	0	0	0

- **Bits 31~23, 19~16, 12~3, 0:** Reserved bits. Writes to these bits have no effect and always read as 0.

- **SF (Store and Forward):** When set, it instructs the application to store a frame of transmit data in application buffer before forwarding to final destination.

Bit 21: SF	Description	
0	Start transmission to MAC level once the Tx FIFO level crosses the threshold	(Initial value)
1	Start transmission to MAC level until one full frame is available in the Tx FIFO	

- **TTM, TR (Transmit Threshold Mode, Threshold Control bits):** These three bits control the transmit threshold values that the application may use. These bits are not used by the DMA controller and are application specific. These bits are used when SF bit is set. The intent is to allow the application to transfer data to the final destination only after the threshold value is met.

Bit 22, 15~14: TTM, TR	Tx FIFO Threshold	
000	32 words (100 Mbps)	(Initial value)
001	64 words (100 Mbps)	
010	128 words (100 Mbps)	
011	256 words (100 Mbps)	
100	18 words (10 Mbps)	
101	24 words (10 Mbps)	
110	32 words (10 Mbps)	
111	40 words (10 Mbps)	

- **ST (Start/Stop Transmission Command):** When set, the transmission process is set to the running state, and the DMA controller checks the transmit list at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the transmit list base address set by CSR4, or from the position retained when the transmit process was previously stopped. If no descriptor can be acquired, the transmit process enters the suspended state. If the current descriptor is not owned by the DMA controller, the transmission process enters the suspended state and the transmit buffer unavailable CSR5[2] is set. The start transmission command is affected only when the transmission process is stopped. If the command is issued before setting CSR4, then the DMA controller will behave UNPREDICTABLY.

When cleared, the transmission process is set to the stopped state after completing the transmission of the current frame. The next descriptor position in the transmit list is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only when the transmission process is in either running or suspended state.

- **OSF (Operate on Second Frame):** When set, the DMA will prefetch the second frame and put it into transmit FIFO before the first frame is completed. If this bit is not set, the next frame will be fetched from the memory only after the MAC has completely processed the frame and the DMA has released the descriptors.
- **SR (Start/Stop Receive Command):** When set, the receive process is set to the running state. The DMA controller attempts to acquire the descriptor from the receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in

the list, which is the address set by the CSR3 or the position retained when the receive process was previously stopped. If no descriptor is owned by the DMA Controller, the receive process enters the suspended state and the receive buffer unavailable (CSR5[7]) sets. The start reception command is affected only when the reception process has stopped. If the command was issued before setting CSR3, the DMA controller behaves UNPREDICTABLY.

When cleared, the receive process enters the stopped state after completing the reception of the current frame. The next descriptor position in the receive list is saved, and becomes the current position after the receive process is restarted. The stop reception command is effective only when the receive process is in running or suspended state.

6.3.7 Interrupt Enable Register (CSR7)

The Interrupt Enable Register (CSR7) is a 32-bit register that enables the interrupts reported by CSR5. Setting a bit to 1 enables a corresponding interrupt. After a hardware or software reset, all interrupts are disabled.

CSR7 is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								NI
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	AI	ERE	FBE			ET	RWE	RS
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	RU	RI	UN		TJ	TU	TS	TI
Write:								
Reset:	0	0	0	0	0	0	0	0

- **Bits 31~17, 12~11, 4:** Reserved bits. Writes to these bits have no effect and always read as 0.
- **NI (Normal Interrupt Summary Enable):** When set, normal interrupt is enabled. When reset, no normal interrupt is enabled. This bit enables the following bits:

CSR5[0] - Transmit Interrupt

CSR5[2] - Transmit buffer unavailable

CSR5[6] - Receive Interrupt
 CSR5[14] - Early Receive Interrupt.

Bit 16: NI	Description	
0	Normal interrupt is disabled	(Initial value)
1	Normal interrupt is enabled	

- **AI (Abnormal Interrupt Summary Enable):** When set, abnormal interrupt is enabled. When reset, no abnormal interrupt is enabled. This bit enables the following bits:

CSR5[1] - Transmit process stopped
 CSR5[3] - Transmit jabber timeout
 CSR5[5] - Transmit underflow
 CSR5[7] - Receive buffer unavailable
 CSR5[8] - Receive process stopped
 CSR5[9] - Receive watchdog timeout
 CSR5[10] - Early transmit interrupt
 CSR5[13] - Fatal bus Error

Bit 15: AI	Description	
0	Abnormal interrupt is disabled	(Initial value)
1	Abnormal interrupt is enabled	

- **ERE (Early Receive Interrupt Enable):** When set together with normal interrupt summary enable (CSR7[16]), the early receive interrupt (CSR5[14]) is enabled. When reset the interrupt is disabled.

Bit 14: ERE	Description	
0	Early receive interrupt is disabled	(Initial value)
1	When set together with normal interrupt summary enable (CSR7[16]), early receive interrupt is enabled	

- **FBE (Fatal Bus Error Enable):** When set together with abnormal interrupt summary enable (CSR7[15]), the fatal bus error interrupt (CSR5[13]) is enabled. When reset fatal bus error interrupt (CSR5[13]) is disabled.

Bit 13: FBE	Description	
0	Fatal bus error interrupt is disabled	(Initial value)
1	When set together with abnormal interrupt summary enable (CSR7[15]), fatal bus error interrupt is enabled	

- **ETE (Early Transmit Interrupt Enable):** When set together with abnormal interrupt summary enable (CSR7[15]), the early transmit interrupt (CSR5[10]) is enabled. When reset and early transmit interrupt (CSR5[10]) is disabled.

Bit 10: ETE	Description	
0	Early transmit interrupt is disabled	(Initial value)

1	When set together with abnormal interrupt summary enable (CSR7[15]), early transmit interrupt is enabled	
---	--	--

- **RWE (Receive Watchdog Timeout Enable):** When this bit and the abnormal interrupt summary enable bit CSR7[15] are set, the receive watchdog timeout interrupt (CSR5[9]) is enabled. When this bit is reset the receive watchdog timeout interrupt is disabled.

Bit 9: RWE	Description	
0	Receive watchdog timeout interrupt is disabled.	(Initial value)
1	When set together with abnormal interrupt summary enable (CSR7[15]), receive watchdog timeout interrupt is enabled	

- **RS (Receive Stopped Enable):** When set together with abnormal interrupt summary enable (CSR7[15]), the receive stopped interrupt (CSR5[8]) is enabled. When reset the receive stopped interrupt (CSR5[8]) is disabled.

Bit 8: RS	Description	
0	Receive stopped interrupt is disabled	(Initial value)
1	When set together with abnormal interrupt summary enable (CSR7[15]), receive stopped interrupt is enabled	

- **RU (Receive Buffer Unavailable Enable):** When set together with abnormal interrupt summary enable (CSR7[15]), the receive buffer unavailable interrupt (CSR5[7]) is enabled. When reset the receive buffer unavailable interrupt (CSR5[7]) is disabled.

Bit 7: RU	Description	
0	Receive buffer unavailable interrupt is disabled	(Initial value)
1	When set together with abnormal interrupt summary enable (CSR7[15]), receive buffer unavailable interrupt is enabled	

- **RI (Receive Interrupt Enable):** When set together with normal interrupt summary enable (CSR7[16]) the receive interrupt (CSR5[6]), is enabled. When reset the receive interrupt (CSR5[6]) is disabled.

Bit 6: RI	Description	
0	Receive interrupt is disabled	(Initial value)
1	When set together with normal interrupt summary enable (CSR7[16]), receive interrupt is enabled	

- **UN (Underflow Interrupt Enable):** When set together with abnormal interrupt summary enable (CSR7[15]), the transmit underflow interrupt (CSR5[5]) is enabled. When reset the transmit underflow interrupt (CSR5[5]) is disabled.

Bit 5: UN	Description	
0	Transmit underflow interrupt is disabled	(Initial value)
1	When set together with abnormal interrupt summary enable (CSR7[15]), transmit underflow interrupt is enabled	

- **TJ (Transmit Jabber Timeout Enable):** When set together with abnormal interrupt summary enable (CSR7[15]) the transmit jabber timeout interrupt (CSR5[3]) is enabled. When reset the transmit jabber timeout interrupt (CSR5[3]) is disabled.

Bit 3: TJ	Description	
0	Transmit jabber timeout interrupt is disabled	(Initial value)
1	When set together with abnormal interrupt summary enable (CSR7[15]), transmit jabber timeout interrupt is enabled	

- **TU (Transmit Buffer Unavailable Enable):** When set together with normal interrupt summary enable (CSR7[16]) the transmit buffer unavailable interrupt (CSR5[2]) is enabled. When reset the transmit buffer unavailable interrupt (CSR5[2]) is disabled.

Bit 2: TU	Description	
0	Transmit buffer unavailable interrupt is disabled	(Initial value)
1	When set together with normal interrupt summary enable (CSR7[16]), transmit buffer unavailable interrupt is enabled	

- **TS (Transmit Stopped Enable):** When set together with abnormal interrupt summary enable (CSR7[15]), the transmission stopped interrupt (CSR5[1]) is enabled. When reset the transmission stopped interrupt (CSR5[1]) is disabled.

Bit 1: TS	Description	
0	Transmit stopped interrupt is disabled	(Initial value)
1	When set together with abnormal interrupt summary enable (CSR7[15]), transmit stopped interrupt is enabled	

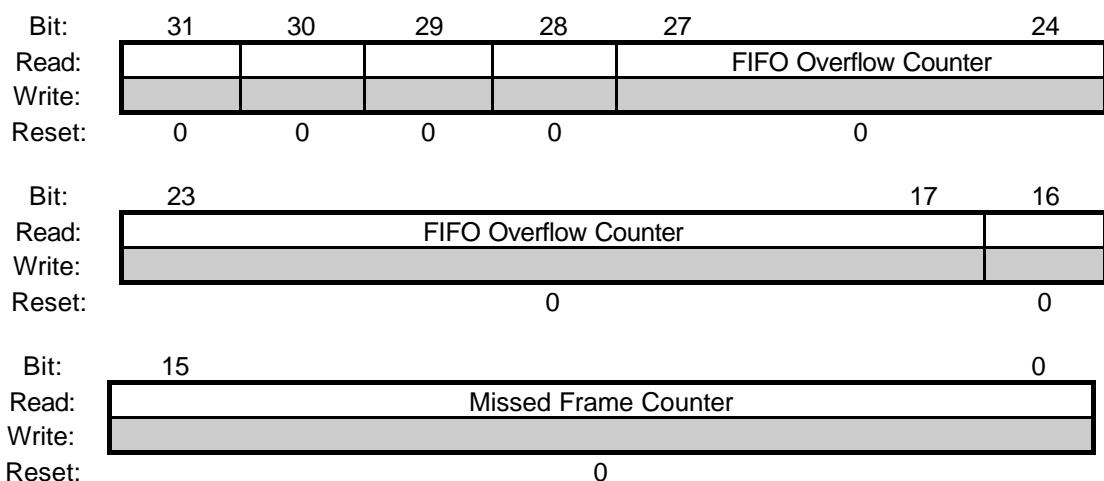
- **TI (Transmit Interrupt Enable):** When set together with normal interrupt summary enable (CSR7[16]) the transmit interrupt (CSR5[0]), is enabled. When reset the transmit interrupt (CSR5[0]) is disabled.

Bit 0: TI	Description	
0	Transmit interrupt is disabled	(Initial value)
1	When set together with normal interrupt summary enable (CSR7[16]), transmit interrupt is enabled	

6.3.8 Missed Frame and Buffer Overflow Counter Register

The Missed Frame and Buffer Overflow Counter Register (CSR8) is a 32 bit counter to track number of missed frames during a receive operation. This register reports the current value of the counter. The counter is used for diagnostic purposes. Bits[15:0] indicate missed frames due to host buffer being unavailable. Bits[27:17] indicate missed frames due to local buffer unavailable conditions in the application interface.

CSR8 is initialized to H'00000000 by power-on reset or manual reset. It is not initialized by ETHC software reset. Read on CSR8 will clear CSR8 to H'00000000 also.



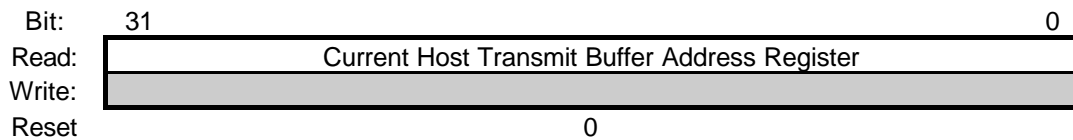
- **Bits 31~29:** Reserved bits. Writes to these bits have no effect and always read as 0.
- **Bits 28, 16:** Reserved bits. Writes to these bits have no effect and read data should be ignored.

- **FIFO Overflow Counter:** Indicates number of frames missed by the controller due to host receive FIFO overflow. This counter is incremented each time the controller discards an incoming frame. The counter is cleared when this register is read.
- **Missed Frame Counter:** This counter is incremented every time the receive DMA gets a frame and the host's descriptor is not available. The counter is cleared when this register is read.

6.3.9 Current Host Transmit Buffer Address Register (CSR20)

The Current Host Transmit Buffer Address Register (CSR20) is a 32-bit read only register that points to the current transmit buffer address being read by the DMA controller.

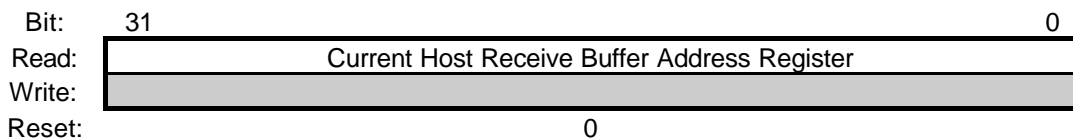
CSR20 is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.



6.3.10 Current Host Receive Buffer Address Register (CSR25)

The Current Host Receive Buffer Address Register (CSR25) is a 32-bit read only register that points to the current receive buffer address being written by the DMA controller.

CSR25 is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.



6.3.11 MAC Control Register

The MAC Control Register is a 32-bit read/write register that establishes the RX and TX operating modes and controls for address filtering and packet filtering.

It is initialized to H'00040000 by power-on reset, manual reset or ETHC software reset.

Bit:	31	30	29	28	27	26	25	24
Read:	RA			HBD	PS			
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:	DRO		OM	F	PM	PR	IF	PB
Write:								
Reset:	0		0	0	0	1	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	HO		HP	LCC	DBF	DTRY		ASTP
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:	BOLMT		DC		TE	RE		
Reset:	0		0	0	0	0	0	0

- **Bits 30, 29, 26~24, 14, 9, 4, 1, 0:** Reserved bits. Writes to these bits have no effect and always read as 0.
- **RA (Receive All):** When set, all incoming packets will be received, regardless of the destination address.
- **HBD (Heart Beat Disable):** When set, the heartbeat signal quality (SQE) generator function is disabled. This bit should be set in the MII Mode.
- **PS (Port Select):** When reset, the MII port is selected and when set, the SRL (ENDEC) port is selected for transmit/receive operations on the Ethernet side.
- **DRO (Disable Receive Own):** When set, the MAC disables the reception of frames when the TXEN is asserted. The MAC will ignore any loop-backed receive packets. When reset, the MAC receives all the packets that are given by the PHY. This bit should be reset when the Full Duplex Mode bit is set or the Operating Mode is set to other than 'Normal Mode'.

- **OM (LoopBack Operating Mode):** Selects the loopback operation modes for the MAC110. This is only for FULL Duplex Mode.

In the Internal Loopback mode, the Tx frame is received by the MII, and turned around back to the MAC. In the External mode however, the TX frame is sent up to the PHY. The PHY will then turn that TX frame back to be received by the MAC. Note that in the External mode, the application has to set the PHY in LoopBack mode by setting bit-14 in the register space of the PHY.

Bit 22, 21: OM	Loopback Mode	
00	Normal: No feedback	(Initial value)
01	Internal: Through MII	
10	External: Through PHY	
11	Reserved	

- **F (Full Duplex Mode):** When Set, the MAC operates in a full-duplex mode where it can transmit and receive simultaneously. While in full-duplex mode: heart-beat check is disabled, heartbeat fail status should be ignored, and internal loop-back is not allowed.

Bit 20: F	Description	
0	Half-duplex mode	(Initial value)
1	Full-duplex mode	

- **PM (Pass All Multicast):** When set, indicates that all the incoming frames with a multicast destination address (first bit in the destination address field is '1') are received. Incoming frames with physical address destinations are filtered only if the address matches with the MAC Address.
- **PR (Promiscuous Mode):** When set, indicates that any incoming valid frame is received regardless of its destination address.
- **IF (Inverse filtering):** When set, Address Check block operates in the inverse filtering mode. This is valid only during perfect filtering mode.
- **PB (Pass Bad Frames):** When set, all incoming frames that passed the address filtering are received, including runt frames, collided frames, or truncated frames caused by Buffer underflow.
- **HO (Hash Only Filtering Mode):** When set, the Address Check block operates in the imperfect address filtering mode both for physical and multicast addresses.
- **HP (Hash/Perfect Filtering Mode):** When reset, the Address Check block does a perfect address filter of incoming frames according to the address specified in the MAC Address register.

When set, the Address Check block does imperfect address filtering of multicast incoming frames according to the hash table specified in the multicast Hash Table Register. If the Hash Only (HO) is set, then physical addresses are imperfect filtered too. If Hash Only bit

(HO) is reset, then physical addresses are perfect address filtered according to the MAC Address Register.

- **LCC (Late Collision Control):** When set, enables the retransmission of the collided frame even after the collision period (late collision). When reset, the MAC disables the frame transmission on a late collision. In any case the Late Collision Status is appropriately updated in the Transmit Packet Status.
- **DBF (Disable Broadcast frames):** When set, disables the reception of broadcast frames. When reset, forwards all the broadcast frames to the memory.
- **DTRY (Disable Retry):** When set, the MAC will attempt only one transmission. When a collision is seen on the bus, the MAC will ignore the current frame and goes to the next frame and a retry error is reported in the Transmit Status. When reset, the MAC will attempt 16 transmissions before signaling a retry error.
- **ASTP (Automatic Pad Stripping):** When set, the MAC will strip the pad field on all the incoming frames, if the length field is less than 46 bytes. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has had the pad characters stripped. Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified (FCS is not stripped). When reset, the MAC will pass all the incoming frames to the host unmodified.
- **BOLMT (BackOff Limit):** The BLOMT bits allow the user to set its BackOff limit in a relaxed or aggressive mode. According to IEEE 802.3, the MAC has to wait for a random number [r] of Slot-Times** after it detects a collision, where:

$$0 < r < 2^K \dots\dots\dots(\text{eq.1})$$

The number K is dependent on how many times the current Frame to be transmitted have been retried, as follows:

$$K = \min(n, 10) \dots\dots\dots(\text{eq.2})$$

where n is the current number of retries.

If a frame has been retried for 3 times, then K = 3 and r = 8 Slot-Times maximum.

If it has been retried for 12 times, then K = 10, and r = 1024 Slot-Times maximum.

A LFSR (linear feedback shift register) 20-bit counter is used to emulate a 20bit random number generator of which [r] is obtained. Once a collision is detected, the number of the current retry of the current frame is used to obtain K (eq.2). This value of K translates into the number of bits to use from the LFSR counter. If the value of k is 3, the MAC100 will take the value in the first 3 bits of the LFSR counter, and use it to count down till Zero on every Slot-Time. This will effectively causes the MAC to wait 8 Slot-Times.

To add more flexibility to the user the Value of the BOLMT will force the number of bits to be used from the LFSR counter to a predetermined value as in the table below.

Bit 7, 6: BOLMT	BackOff Limit	
00	10	(Initial value)
01	8	
10	4	
11	1	

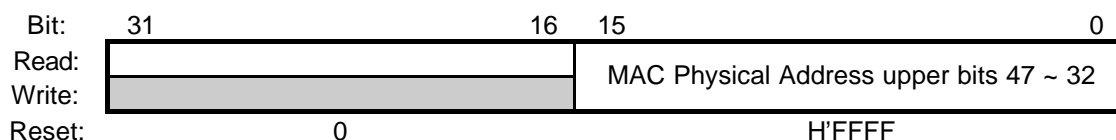
Thus if the value of $k = 10$, then the MAC will look at the BOLMT if it is 00 then it will use the lower 10bits of the LFSR counter for the wait countdown. If BOLMT is 10 then it will only use the value in the first 4bits for its wait countdown, and so on...

** Slot-Time = 512 bit times.

- **DC (Deferral Check):** When set, the deferral check is enabled in the MAC. The MAC will abort the transmission attempt if it has deferred for more than 24,288 bit times. Deferring starts when the transmitter is ready to transmit, but is prevented from doing so because CRS is active. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of BackOff, the deferral timer resets to 0 and restarts. When reset, the deferral check is disabled in the MAC and the MAC defers indefinitely.
- **TE (Transmitter Enable):** When set, the MAC's transmitter is enabled and it will transmit frames from the buffer on to the cable. When reset, the MAC's transmitter is disabled and will not transmit any frames.
- **RE (Receiver Enable):** When set, the MAC's receiver is enabled and will receive frames from the MII interface. When reset, the MAC's receiver is disabled and will not receive any frames from the MII interface.

6.3.12 MAC Address Hi Register

The MAC Address Hi Register is a 32-bit read/write register that contains the upper 16 bits of the physical address of the MAC. It is initialized to H'0000FFFF by power-on reset, manual reset or ETHC software reset.



- **Bits 30~16:** Reserved bits. Writes to these bits have no effect and always read as 0.

6.3.13 MAC Address Lo Register

The MAC Address Lo Register is a 32-bit read/write register that contains the lower 32 bits of the physical address of the MAC. It is initialized to H'FFFF_FFFF by power-on reset, manual reset or ETHC software reset.



6.3.14 Multicast Address Hi/Lo Register

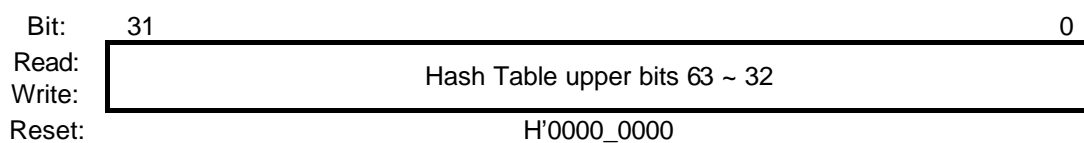
The 64-bit multicast table is used for group address filtering. For hash filtering, the contents of the destination address in the incoming frame is passed through the CRC logic and the upper 6 bits of the CRC register are used to index the contents of the Hash table. The most significant

bit determines the register to be used (Hi/Low), while the other five bits determine the bit within the register. A value of '00000' selects the bit 0 of the selected register and a value of '11111' selects the bit 31 of the selected register.

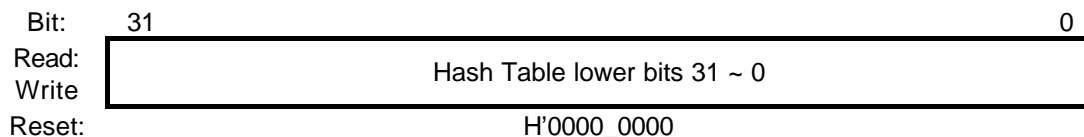
If the corresponding bit is '1', then the multicast frame is accepted else it is rejected. If the Pass All Multicast is set, then all multi-cast frames are accepted regardless of the multi-cast hash values.

The Multi Cast Hash Table Hi Register contains the higher 32 bits of the hash table and the Multi Cast Hash Table Low Register contains the lower 32 bits of the hash table. They are initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.

Multi Cast Hash Table Hi Register:

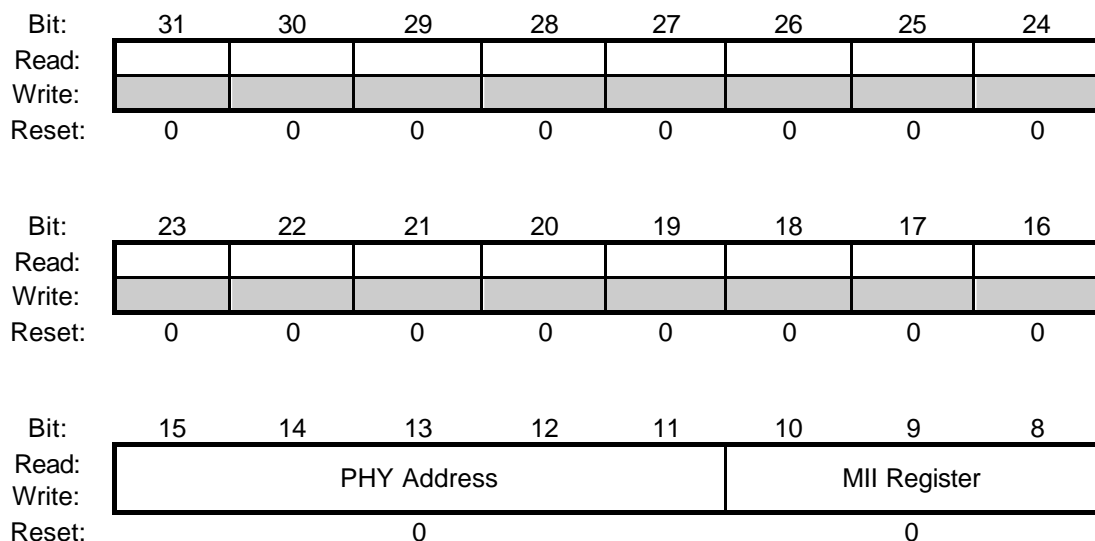


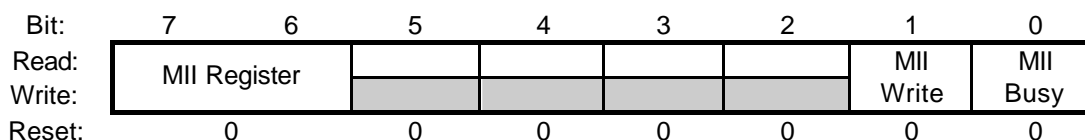
Multi Cast Hash Table Low Register:



6.3.15 MII Address Register

The MII Address Register is a 32-bit read/write register that is used to control the management cycles to the external PHY controller chip. It is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.

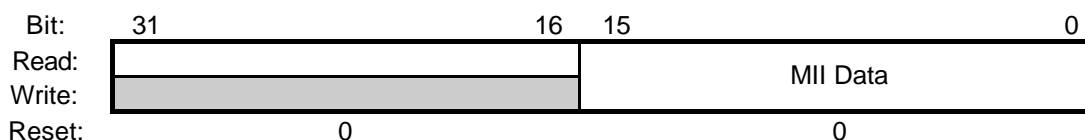




- **Bits 31~16, 5~2:** Reserved bits. Writes to these bits have no effect and always read as 0.
- **PHY Address:** These bits tell which of the 32 possible PHY devices are being accessed.
- **MII Register:** These bits select the desired MII register in the selected PHY device.
- **MII Write:** Setting this bit tells the PHY that this will be a write operation using the MII data register. If this bit is not set, this will be a read operation, placing the data in the MII data register.
- **MII Busy:** This bit should read a logic 0 before writing to the MII address and MII data registers. This bit must also be set to 0 during write to the MII address register. During a MII register access, this bit will be then set to signify that a read or write access is in progress. The MII data register should be kept valid until the MAC clears this bit during a PHY write operation. The MII data register is invalid until the MAC clears this bit during a PHY read operation. The MII address register should not be written to until this bit is cleared.

6.3.16 MII Data Register

The MII Data Register is a 32-bit read/write register that contains the data to be written to the PHY register specified in the MII address register, or it contains the read data from the PHY register whose address is specified in the MII address register. It is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.

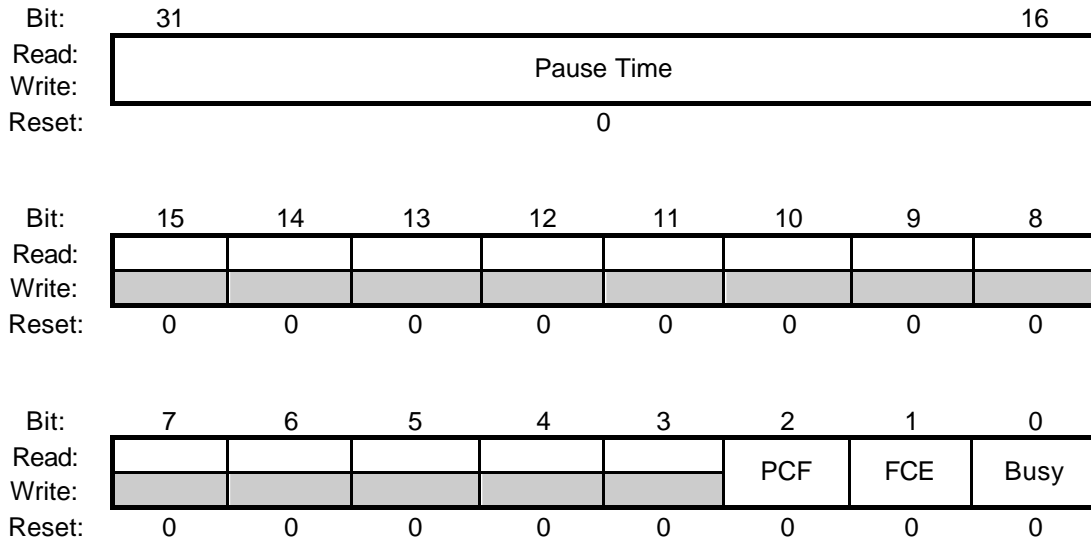


- **Bits 31~16:** Reserved bits. Writes to these bits have no effect and always read as 0.

6.3.17 Flow Control Register

This register is a 32-bit read/write register that is used to control the generation and reception of the Control (PAUSE Command) frames by the MAC's Flow control block. A write to register with busy bit set to '1' triggers the Flow Control block to generate a Control frame. The fields of the control frame are selected as specified in the 802.3x specification and Pause Time value from this register is used in the "Pause Time" field of the control frame. The Busy bit is set until the control frame is transferred onto the cable. The Host has to make sure that the Busy bit is cleared before writing the register. The Pass Control Frames bit indicates the MAC whether to pass the control frame to the Host or not and Flow Control Enable bit enables the receive portion of the Flow Control block.

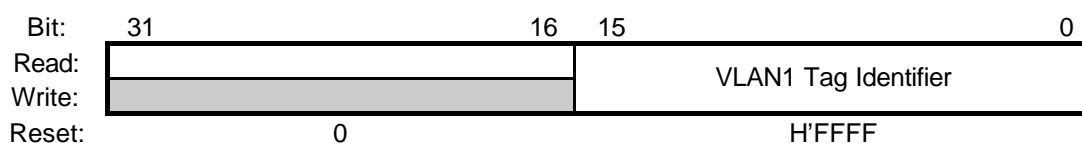
It is initialized to H'00000000 by power-on reset, manual reset or ETHC software reset.



- **Bits 15~3:** Reserved bits. Writes to these bits have no effect and always read as 0.
- **Pause Time:** This field tells the value that is to be used in the PAUSE TIME field in the control frame.
- **PCF (Pass Control Frames):** When set, the control frames are passed to the Host. The MAC will decode the control frame (PAUSE), disables the transmitter for the specified amount of time. When reset, the MAC will decode the control frames but will not pass the frames to the Host.
- **FCE (Flow Control Enable):** When set, the MAC is enabled for flow control operation and it will decode all the incoming frames for control frames. If the MAC receives a valid control frame (PAUSE command), it will disable the transmitter for the specified time. When reset, the flow control operation in the MAC is disabled and the MAC does not decode the frames for control frames.
- **Flow Control Busy:** This bit should read a logic 0 before writing to the Flow Control register. To initiate a PAUSE control frame the host must set this bit to '1'. During a transfer of Control Frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of the control frame (PAUSE) transmission, the MAC will reset to '0'. The Flow Control register should not be written to until this bit is cleared.

6.3.18 VLAN1 Tag Register

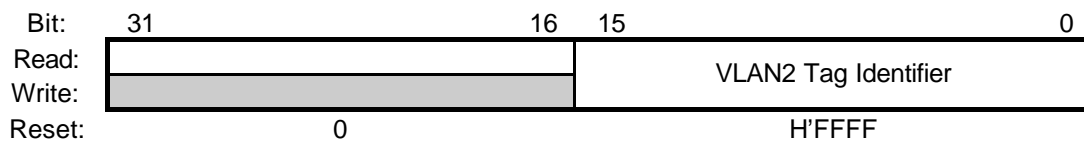
This register contains the Length/Type field to identify the VLAN1 frames. The MAC compares the 13th and 14th bytes of the incoming frame (Length/Type) field and if a match occurs, it sets the VLAN1 tag to the received frame. The legal length of the frame is increased from 1518 bytes to 1522 bytes. It is initialized to H'0000FFFF by power-on reset, manual reset or ETHC software reset.



- **Bits 31~16:** Reserved bits. Writes to these bits have no effect and always read as 0.

6.3.19 VLAN2 Tag Register

This register contains the Length/Type field to identify the VLAN2 frames. The MAC compares the 13th and 14th bytes of the incoming frame (Length/Type) field and if a non-zero match occurs, it sets the VLAN2 tag to the received frame. The legal length of the frame is increased from 1518 bytes to 1522 bytes. It is initialized to H'0000FFFF by power-on reset, manual reset or ETHC software reset.



- **Bits 31~16:** Reserved bits. Writes to these bits have no effect and always read as 0.

6.4 Data Descriptors and Buffers

The DMA controller and the driver communicate through two data structures:

- Control and Status registers (CSR), as described in Chapter 6.3.5.
- Descriptor lists and data buffers, described in this chapter.

The DMA controller transfers RX data frames to the RX buffers in the host memory and transmits data from the TX buffers in the host memory. Descriptors that reside in the host memory contain pointers to these buffers.

There are two descriptor lists, one for Receive and one for Transmit. The base address of each list is written into DMA CSR3 and CSR4, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both receive and transmit descriptors (RDES_1[24] and TDES_1[24]). The descriptor lists reside in the host's physical memory address space. Each descriptor can point to maximum of two buffers. This enables two physically addressed buffers, and not contiguous buffers in memory, to be used.

A data buffer reside in host physical memory space, and consists of an entire frame, but it cannot exceed a single frame. Buffers contain only data; buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot scan multiple frames. The DMA controller will skip to the next frame buffer when end of frame is detected. Data chaining can be enabled or disabled.

Notes: Descriptors of a Zero Buffer length are not supported at the initial and Final Descriptor of a Chain.

Following figure illustrates the descriptor organization.

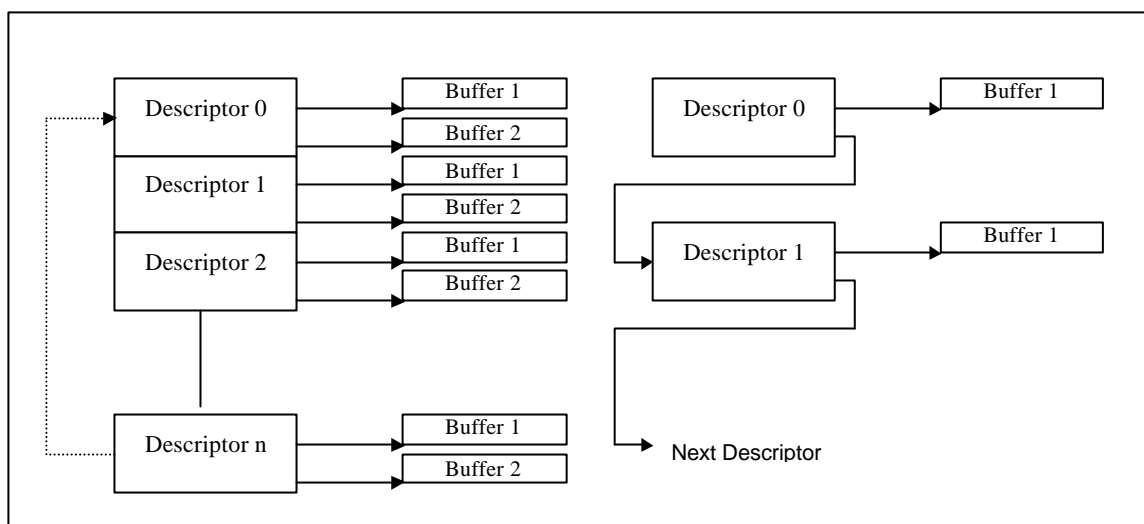


Figure 6-2 Descriptor Ring and Chain Structure Examples

6.4.1 Receive Descriptors

Following figure shows the Receive Descriptor. The Descriptors and the RX buffers' addresses must be WORD (32-bit) aligned.

Providing two buffers, two byte-count buffers, and two address pointers in each descriptor enables the adapter port to be compatible with various types of memory-management schemes.

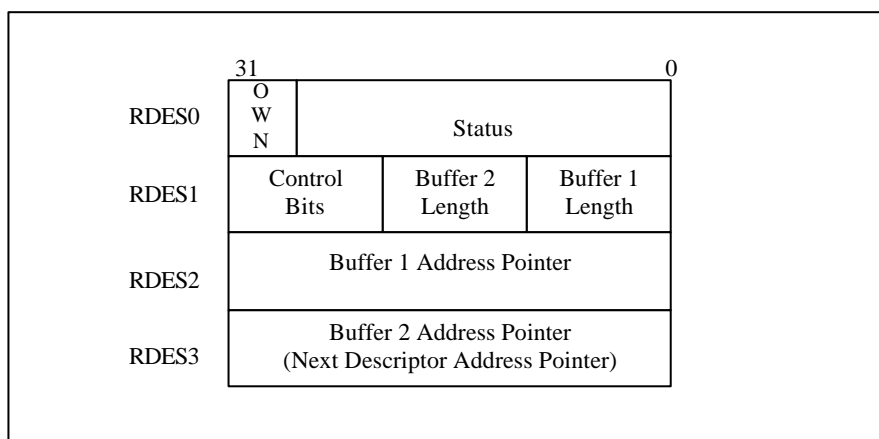


Figure 6-3 Receive Descriptor Format

6.4.1.1 Receive Descriptor 0 (RDES0)

RDES_0 contains the received frame status, the frame length, and the descriptor ownership information. Following table describes the bit fields of the RDES_0.

Table 6-3 RDES0 Receive Descriptor 0 Description

Field	Description
31	OWN - Own Bit When set, indicates that the descriptor is owned by the DMA controller. When reset, indicates that the descriptor is owned by the host. The DMA controller clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	Filtering Fail When set, it indicates that the destination address field in the current frame failed the address filtering. This bit is reset when it passes the address filtering selected in the MAC control register.
29:16	FL - Frame Length Indicates the length in bytes, of the received frame that was transferred to host memory, including the CRC. This field is valid only when the last descriptor (RDES_0[8]) is set and the descriptor error (RDES_0[14]) is reset.
15	ES-Error Summary Indicates the logical OR of the following bits: [1] CRC Error [6] Collision Seen [7] Frame too long [11] Runt Frame [14] Descriptor Error This bit is the generic Error Status bit set by the DMA controller. This field is valid only when the last descriptor (RDES_0[8]) is set.
14	DE-Descriptor Error When set, indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA controller does not own the next descriptor. The frame is truncated. This field is valid only when the last descriptor (RDES_0[8]) is set.
13	Reserved
12	Length Error When set, indicates that for the current frame the Length value is inconsistent with the total number of bytes received in the current frame. This is valid when the Frame Type is set to '0' (802.3 Frame).
11	Runt Frame When set, indicates that this frame was damaged by a collision or premature termination before the collision window (64 bytes default) passed. Runt frames are passed on to the host only if the pass bad frames bit CSR6[3] is set.
10	Multicast Frame When set, it indicates that the Destination Address in the current frame is a multicast address, i.e., the first bit of the DA is 1.

9	<p>FS-First Descriptor</p> <p>When set, indicates that this descriptor contains the first buffer of the frame. If the buffer size of the first buffer is 0, the second buffer contains the beginning of the frame. If the buffer size of the second is also 0, the second descriptor contains the beginning of the frame.</p>
8	<p>LS-Last Descriptor</p> <p>When set, indicates that the buffers pointed to by this descriptor are the last buffers of the frame.</p>
7	<p>Frame too Long</p> <p>When set, indicates that the frame length exceeds the maximum Ethernet specified size of 1518 bytes. Frame too long is only a frame length indication and does not cause any frame truncation.</p>
6	<p>Collision Seen</p> <p>When set, indicates that the frame was damaged by a collision that occurred after the 64 bytes following the start of frame delimiter (SFD). This is a late collision.</p>
5	<p>Frame Type</p> <p>When set, indicates that the frame is an Ethernet-type frame (frame length field is greater than 1500 bytes). When clear, indicates the frame is an IEEE 802.3 frame. This bit is not valid for runt frames of less than 14 bytes.</p>
4	<p>Watchdog Timeout.</p> <p>The watchdog timer monitors the time of each packet reception. If a single packet reception exceeds (2048 bytes) the watchdog circuitry disables the receive path.</p>
3	<p>MII Error</p> <p>When set, indicates that the MII RX_ER was asserted during the reception of this frame.</p>
2	<p>Dribbling Bit</p> <p>When set, indicates that the frame contained a non-integer multiple of 8 bits. This bit is not valid if either collision seen or runt frame bits are set. If set, and the CRC error is reset, then the packet is valid.</p>
1	<p>CRC Error</p> <p>When set, indicates that a cyclic redundancy check (CRC) error occurred on the received frame. This bit is also set when the MII RX_ER pin is asserted during the reception of a receive packet even though the CRC may be correct.</p>
0	Reserved

6.4.1.2 Receive Descriptor 1 (RDES1)

Following table describes the bit fields of the RDES1.

Table 6-4 RDES1 Receive Descriptor 1 Description

Field	Description
31:26	Reserved
25	RER-Receive End of Ring When set, indicates that the descriptor list reached its final descriptor. The DMA controller returns to the base address of the list, creating a descriptor ring.
24	RCH-Second Address Chained When set-indicates that the second address in the descriptor is the next descriptor address, rather than the second buffer address. When this field is set RBS_2 (RDES_1[20:10]) should be all zeros. RDES_1[25] takes precedence over RDES_1[24].
23:22	Reserved
21:11	RBS2-Receive Buffer 2 Size Indicates the size, in bytes, of the second data buffer. The buffer size must be multiple of 4. This field is not valid if RDES_1[24] is set.
10:0	RBS1-Receive Buffer 1 Size Indicates the size, in bytes, of the first data buffer. The buffer size must be multiple of 4. If this field is 0, the DMA controller ignores this buffer and uses buffer 2. (This field cannot be zero if the descriptor chaining is used – Second Address Chained (RDES_1[24]) is set).

6.4.1.3 Receive Descriptor 2 (RDES2)

Following table describes the bit fields of the RDES2.

Table 6-5 RDES2 Receive Descriptor 2 Description

Field	Description
31:0	Buffer 1 Address Pointer Indicates the physical address of buffer 1. The buffer must be WORD (32-bit) aligned (RDES_2[1:0]=00)

6.4.1.4 Receive Descriptor 3 (RDES3)

Following table describes the bit fields of the RDES3.

Table 6-6 RDES3 Receive Descriptor 3 Description

Field	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of buffer 2 when the descriptor chaining is used. If the Second Address Chained (RDES_1[24]) bit is set, then this address contains the pointer to the physical memory where the next descriptor is present. The buffer (next descriptor) must be WORD (32-bit) aligned (RDES_3[1:0]=00).

6.4.2 Transmit Descriptors

Following Figure shows the transmit descriptor format. The Descriptors addresses must be WORD (32-bit) aligned.

Providing two buffers, two byte-count buffers, and two address pointers in each descriptor enables the adapter port to be compatible with various types of memory-management schemes.

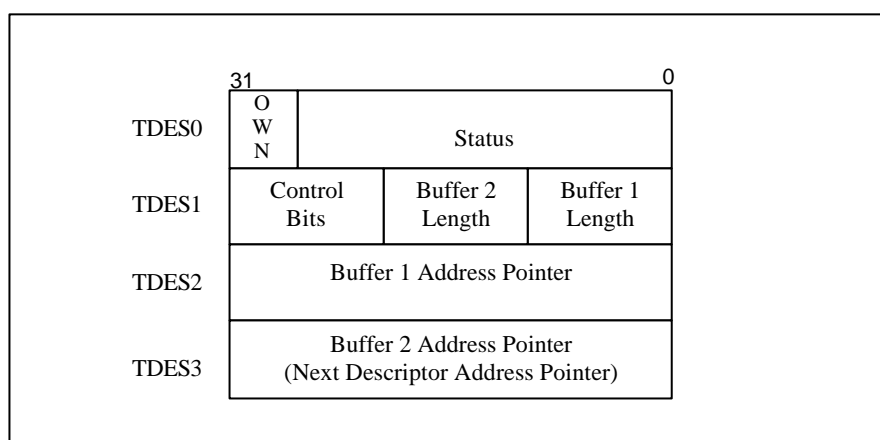


Figure 6-4 Transmit Descriptor Format

6.4.2.1 Transmit Descriptor 0 (TDES0)

TDES0 contains the transmitted frame status and the descriptor ownership information. Following table describes the bit fields of the TDES0.

Table 6-7 TDES0 Receive Descriptor 0 Description

Field	Description
31	<p>OWN-Own Bit</p> <p>When set, indicates that the descriptor is owned by the DMA controller. When reset, indicates that the descriptor is owned by the host. The DMA controller clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the first descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.</p>
30:16	Reserved
15	<p>Frame Aborted</p> <p>When set, it indicates that the Host has aborted the transmission of the current frame due to one of the following conditions:</p> <ul style="list-style-type: none"> – No Carrier – Loss of Carrier – Excessive Deferral – Late Collision – Retry Count exceeds the attempt Limit – Data under run <p>If this bit is reset, it indicates that the current frame was successfully transmitted onto the MII Interface.</p>
14:12	Reserved
11	<p>Loss of Carrier</p> <p>When set, indicates that the loss of carrier occurred during the frame's transmission (i.e., the MII CRS input was inactive for one or more bit times when the frame is being transmitted). This is valid only for the frames transmitter without collision and when the MAC is operating in half duplex mode.</p>
10	<p>No Carrier</p> <p>When set, indicates that the carrier signal from the transceiver was not present during transmission.</p>
9	<p>Late Collision</p> <p>When set, indicates that the frame transmission was aborted due to collision occurring after the collision window of 64 bytes. Not valid if under run error is set.</p>
8	<p>Excessive Collisions</p> <p>When set, indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DTRY (bit 10 of MAC Control Register) is asserted, this bit is set after the first collision and the transmission of the frame will be aborted.</p>

7	Heart Beat Fail This is effective only in 10-Mb/s operation mode and the Serial port is selected. When set, this bit indicates a heartbeat collision check failure (the transceiver failed to return a collision pulse as a check after the transmission). This bit is not valid if underflow error bit is set.
6:3	Collision Count The 4-bit counter indicates the number of collisions that occurred before the frame was transmitted. Not valid when the Excessive Collisions bit is set.
2	Excessive Deferral When set, indicates that the transmission has ended because of excessive deferral of over 24,288 bit times during the transmission, if the defer bit is set high in the control register.
1	UF-Underflow Error When set, indicates that the transmitter aborted the frame because data arrived late from memory. Underflow error indicates that the DMA controller encountered an empty transmit buffer while transmitting the frame. The transmission process enters the suspended state and sets both transmit underflow (CSR_5[5]) and transmit interrupt (CSR_5[0])
0	Deferred When set, indicates that the transmitter had to defer while ready to transmit a frame because the carrier was asserted.

6.4.2.2 Transmit Descriptor 1 (TDES1)

Following table describes the bit fields of the TDES1.

Table 6-8 TDES1 Receive Descriptor 1 Description

Field	Description
31	IC-Interrupt on Completion When set, the sets transmit interrupt (CSR_5[0]) after the present frame has been transmitted.
30	LS-Last Segment When set, indicates that the buffer contains the last segment of the frame.
29	FS-First Segment When set, indicates that the buffer contains the first segment of a frame.
28:27	Reserved
26	AC-Add CRC Disable When set, the does not append the cyclic redundancy check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES_1[29]) is set.
25	TER-Transmit End of Ring When set, indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.

24	TCH-Second Address Chained When set-indicates that the second address in the descriptor is the next descriptor address, rather than the second buffer address. When this bit is set, the TBS2 (TDES_1[2:11]) should be all zeros. TDES_1[25] takes precedence over TDES_1[24].
23	DPD-Disable Padding When set, the does not automatically add a padding field, to a packet shorter than 64 bytes. When reset, the DMA controller automatically adds a padding field and also a CRC field to a packet shorter than 64 bytes and the CRC field is added despite the state of the add CRC disable (TDES_1[26]) flag. This is valid only when the first segment (TDES_1[2]) is set.
22	Reserved
21:11	TBS2-Transmit Buffer 2 Size Indicates the size, in bytes, of the second data buffer. This field is not valid if TDES_1[24] is set.
10:0	TBS1-Transmit Buffer 1 Size Indicates the size, in bytes, of the first data buffer. If this field is 0, the DMA controller ignores this buffer and uses buffer 2.

6.4.2.3 Transmit Descriptor 2 (TDES2)

Following table describes the bit fields of the TDES2.

Table 6-9 TDES2 Transmit Descriptor 2 Description

Field	Description
31:0	Buffer 1 Address Pointer Indicates the physical address of buffer 1. There are no limitations on the buffer address alignment.

6.4.2.4 Transmit Descriptor 3 (TDES3)

Following table describes the bit fields of the TDES3.

Table 6-10 TDES3 Transmit Descriptor 3 Description

Field	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of buffer 2 when the descriptor chaining is used. If the Second Address Chained (TDES_1[24]) bit is set, then this address contains the pointer to the physical memory where the next descriptor is present. There are no limitations on the buffer address alignment.

6.5 Operation

6.5.1 Data Flow

The ETHC itself interfaces to the Ethernet medium through a Medium Independent interface module (MII) and to system bus. Thus, the Data flow is as follows:

- Transmit

1. DMA VCI Master to TLI VCI Slave (DMD port.)
2. TLI VCI Master to MAC VCI Slave (APD port.)
3. MAC TX To MII Rx (MCS port.)
4. MII TX to PHY Rx (MIM port.)
5. PHY TX to Medium. (not shown in diagram)

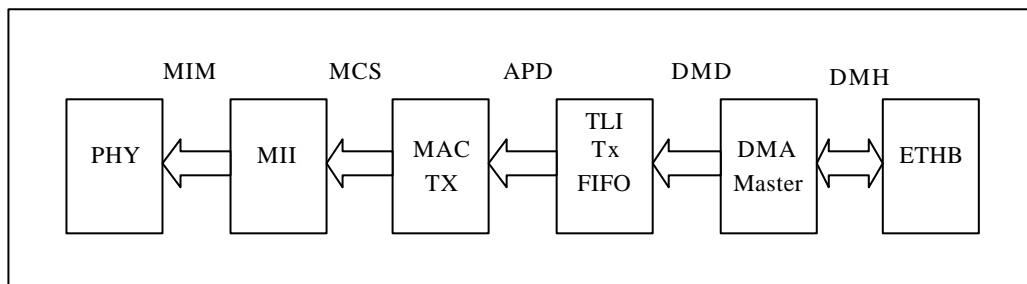


Figure 6-5 Transmit Data Flow

- Receive

1. PHY TX to MII Rx (MIM port.)
2. MII TX to MAC Rx (MCS port.)
3. MAC VCI Master to TLI VCI Slave (APS port.)
4. TLI VCI Master to DMA VCI Slave (DMS port.)
5. DMA VCI Master to System Memory (DMH port.)

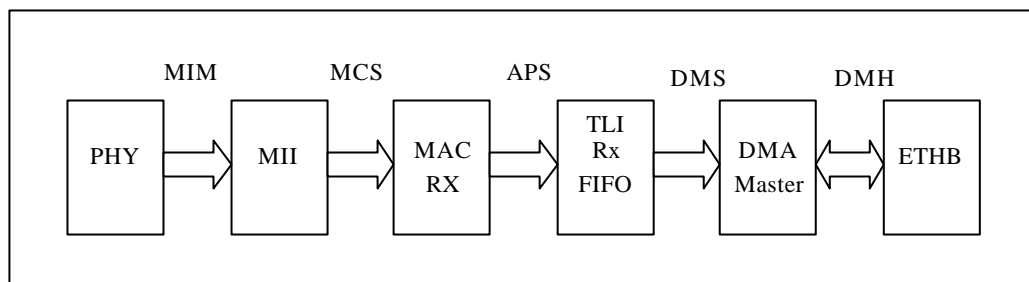


Figure 6-6 Receive Data Flow

The Ethernet Subsystem engages in two types of activities Transmit, and Receive, each of which performs the following steps:

1. Initialization
2. Transfer

3. Status

In the following section Transmit and Receive Operation are discussed.

6.5.2 Transmit Operation

6.5.2.1 Initialization

The Host Driver residing on the System is responsible for the initialization of the Ethernet Controller. Initialization is directed to the CSR space of each of the Components of the Ethernet Controller, and setting up the “Transmit” descriptors for each frame.

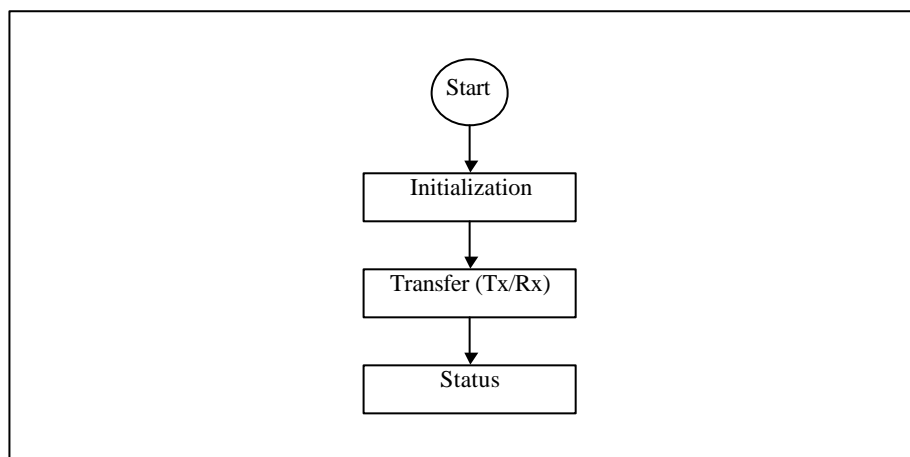


Figure 6-7 Ethernet Controller Operation

The registers affecting the Transmit operation are the following:

- | | |
|-------------------------------|---|
| 1. MAC Control Register [3] | Enable Transmit (TE) |
| 2. MAC Control Register [5] | Deferral Check (DC) |
| 3. MAC Control Register [6-7] | Set Back-off limit (BOLMT) |
| 4. MAC Control Register [12] | Late Collision Control (LCC) |
| 5. MAC Control Register [28] | Heart Beat Disable (HBD) |
| 6. DMA CSR-6 [21] | Data Mode: Store and Forward (SF) |
| 7. DMA CSR-6 [22,15:14] | Data Mode: Transmit Threshold Mode (TTM) |
| 8. DMA CSR-4 [31:2] | Start of Transmit List: Address of First Descriptor |
| 9. DMA CSR-0 [13:8] | DMA Burst Length |

After setting the mode of operation of the Ethernet Controller, the driver creates the “Transmit” descriptors for the transmit frame, (see DMA descriptors), and resets the “OWN” bit in the descriptors’ status field. Then it initiates the transmit process by setting the DMA CSR-6 [13] Start/Stop (ST).

6.5.2.2 Transmit Process

Once the ST bit is set, the DMA will obtain the descriptor located in the host memory at the address written in its CSR-4[31:2]. If the OWN bit in the descriptor is not set, the DMA will assume that the Host is still busy with this descriptor, and will go into suspend state. The DMA stays in the suspend state until the host performs a read or a write on the DMA CSR-1.

However, if OWN bit was found to be set, then the DMA, will obtain the pointer to the frame data from the descriptor, and download that data from the Host memory to the TLI TX FIFO. The TLI will initiate the transfer from its TX FIFO to the MAC depending on the Data Mode programmed in the DMA CSR-5[22], [15:14].

As an example, if the TLI is set to be in the SF mode, then it will not initiate the transaction unto the MAC, until one whole frame has been completely transferred into its TX FIFO from the DMA. Once this is completed, the TLI will transfer the data to the MAC until Frame transfer is completed and a status is reported to the host.

On Completion, the MAC and DMA update their CSR spaces and the TDES0 Status, and the host is interrupted to check the status of the transfer.

6.5.2.3 Status

Transmission is completed either normally, or abnormally. In both cases the TLI informs the DMA of the status by writing to CSR-5. The DMA will then report the status in the TDES-0 status field, and then interrupts the Host by writing to DMA CSR-7, depending on whether the corresponding Interrupt enable bit is set in CSR-7.

In normal completion, the following bits set to zero:

1. DMA TDES-0 [15] Frame Abort

In abnormal completion, the following bits are set:

1. DMA TDES-0 [15] Frame Abort
2. MAC TDES-0 [0,2,8-11] Cause of Abort
3. MAC TDES-0 [7] Heart Beat Fail (for Serial PHY use Only)
4. DMA CSR-5 [25-23] Error Bits (ER)
5. DMA CSR-5 [15] Abnormal Interrupt Summary (AIS)
6. DMA CSR-5 [13] Fatal Bus Error (FBE)
7. DMA CSR-5 [10] Early Transmit Interrupt (ETI)
8. DMA CSR-5 [5] Transmit Underflow (UNF)
9. DMA CSR-5 [3] Transmit Jabber Timeout (TJT)
10. DMA CSR-5 [2] Transmit Buffer Unavailable (TU)
11. DMA CSR-5 [1] Transmit Process Stopped (TPS)

Notes:

- All Error conditions can be enabled to generate an interrupt to the host through the interrupt Enable Register DMA CSR-7.
- If the current descriptor happened to be part of link list of descriptors, then the DMA traverses the list of descriptors down to bottom of the descriptor list, and writes the status in its Status field. The status is valid in the TDES0 field of the last segment descriptor.

6.5.2.4 Transmit Error handling

1. Error conditions 64 bytes are transferred: E.g.: Collision before 64 bytes. If the TLI receives an abort before 64 bytes have been transferred to the MAC, it returns the pointer back to start of frame and transmits the entire frame again. It retains all the data so that the DMA does not have to re-fetch the data from system memory.

2. Error conditions after 64 bytes reached: E.g.: late collision, jabber timeout or transmit underflow: When the MAC abort the transaction, the TLI gets the status from the MAC, sends it to the DMA and flushes the FIFO without sending it to the MAC. It continues to flush the FIFO until it reaches the end of the frame. If the FIFO is empty before end of the frame is reached (if the DMA has not completed fetching the packet yet), it keeps flushing any new words that the DMA puts into the FIFO until the end of frame is encountered. If the end of frame is encountered and if there is a second frame available, the TLI starts the new packet to the MAC.
3. When the DMA starts a transaction or a burst transaction is half way, if the FIFO becomes full, the TLI issues error signal. At this point, the DMA will close current cycle and retry the same data transaction after some time. DMA keeps doing the retry till it completes the transaction successfully. This error is not a fatal error condition.
4. When a frame is aborted on the MAC interface because of any error on the Ethernet, the TLI starts the status phase by a read transaction. It then asserts a signal to notify DMA once it receives the Status from the MAC and flushes the FIFO till the end of frame is reached. If the TLI is receiving the remaining data of the same frame from the DMA, which is aborted by the MAC, the TLI will keep receiving data from the DMA and flush internally. If the FIFO empty condition occurs, it waits for the data from the DMA and keep flushing them as and when acquired in the FIFO, until the end of frame is reached.

6.5.3 Receive Operation

Similar to the transmit mode, the Receive mode consists of the following phases:

6.5.3.1 Initialization

The Host Driver residing on the System is responsible for the initialization of the Ethernet subsystem. Initialization is directed to the CSR space of each of the Components of the Ethernet subsystem, and setting up the "Receive" descriptors for each frame.

The Registers involved in the receive transaction are:

1. MAC Address Hi/Lo Register
2. Multicast Hash Table Hi/Lo Register
3. VLAN1 Tag Register
4. VLAN2 Tag Register
5. MAC Control Register [31] Receive All (RA)
6. MAC Control Register [23] Disable Receive Own (DRO)
7. MAC Control Register [19] Pass All Multicast Frames (PM)
8. MAC Control Register [18] Promiscuous mode (PR)
9. MAC Control Register [17] Inverse Filtering (IF)
10. MAC Control Register [16] Pass Bad Frames (PB)
11. MAC Control Register [15] Hash Only (HO)
12. MAC Control Register [13] Hash/Perfect (HP)
13. MAC Control Register [11] Disable Broadcast Frames (DBF)
14. MAC Control Register [8] Automatic Pad Stripping (ASTP)
15. MAC Control Register [2] Receive Enable (RE)
16. DMA CSR3 DMA Receive Descriptor Base Address
17. DMA CSR0 [6-2] Descriptor Skip Length

After setting the mode of operation of the Ethernet subsystem, the driver creates the "Receive"

descriptors for the Receive frame, (see DMA descriptors), and resets the “OWN” bit in the “Receive” descriptor s' status field. It then concludes initialization by setting the DMA CSR-6[1] Start/Stop (ST).

6.5.3.2 Receive Process

Once the ST bit is set, the DMA is in the running mode, and will always acquire an extra descriptor for it received frame. If the “OWN” bit is not set in the Descriptor's Status field, the DMA Receive engine will go into suspend mode until the Driver writes to the DMA CSR-2 (Receive Poll Demand Resiter).

However, if OWN bit was found to be set, then the DMA will transfer the data from the TLI RX FIFO to the “Receive” Descriptors' data buffer.

The TLI will initiate the transfer to the DMA immediately after receiving 64Bytes of data from the MAC. The MAC filters the frames on the medium according to the setting of its MAC Control Register.

As an example, if the MAC Control Register [13] set to do perfect filtering, then the Frame received on the MAC is transferred to the TLI layer. The TLI RX FIFO thresholds at the first 64bytes of data and thus, initiates the transaction on the DMA. The DMA transfers the data to the descriptors' buffers until the Frame transfer is completed.

On Completion, the MAC and DMA update their CSR spaces and the RDES0 status, and the host is interrupted to check the status of the transfer.

6.5.3.3 Status

The Receive process is completed either normally, or abnormally. In both cases the DMA and the host are informed, and the status is reported in the descriptors' status field, and in the CSR of the DMA.

In normal completion, the following bits are reset:

1. DMA RDES-0 [15] Error Summary

In abnormal completion, the following bits are set:

- | | |
|-----------------------|----------------------------------|
| 1. RDES-0 [15] | Error Summary |
| 2. RDES-0 [14] | Descriptor Error |
| 3. RDES-0 [12] | Length Error |
| 4. RDES-0 [11] | Runt Frame |
| 5. RDES-0 [7] | Frame Too Long |
| 6. RDES-0 [6] | Collision Seen |
| 7. RDES-0 [4] | Watchdog timeout |
| 8. RDES-0 [3] | MII Error |
| 9. RDES-0 [2] | Dribbling bit |
| 10. RDES-0 [1] | CRC Error |
| 11. DMA CSR-5 [25-23] | Error Bits (EB) |
| 12. DMA CSR-5 [13] | Fatal Bus Error (FBE) |
| 13. DMA CSR-5 [15] | Abnormal Interrupt Summary (AIS) |
| 14. DMA CSR-5 [8] | Receive Process Stopped (RPS) |
| 15. DMA CSR-5 [7] | Receive Buffer Unavailable |

Notes: All Error conditions can be enabled to generate an interrupt to the host through the interrupt Enable Register DMA CSR-7.

The status is returned by the MAC through an interrupt to the DMA and to the Host. On interrupt, the DMA reports an error or a normal transaction in the descriptors 'status filed (see DMA RDES0).

6.5.3.4 Receive Error handling

1. If the receive FIFO becomes full causing an overflow, MAC will close current cycle and start a status phase to TLI. MAC will retry this status phase until success.
2. There may be conditions that cause bad data to be sent by the MAC, for e.g., an incorrect address or a runt frame. The TLI trashes the data and reverts the pointer to the tail of the previous frame received into the FIFO. The DMA controller has no knowledge of having received this frame.
2. After 64 bytes have been received in the FIFO, the TLI starts sending data to the DMA controller. If an abort is received by the TLI (for e.g. If a descriptor was unavailable), it starts flushing the current frame until end of frame is reached. If there are more frames behind this flushed frame, it attempts to start a new transfer. If it receives an abort again, it flushes the full frame. This process is repeated until the FIFO is empty.
3. Unusually long frames (greater than 2048 bytes on Ethernet) will be truncated by MAC.

6.5.4 Host Bus Burst Access

The DMA controller will always attempt to execute burst transfers on the host bus. The maximum burst length is limited by the PBL field (bits [18:8] is CSR0). Each time the controller starts a burst cycle on the host bus to access data buffers it will start from count 0 and go up to PBL. RX and TX descriptors are always accessed in 4 WORD bursts. If a burst is terminated (due to host bus being busy and a RETRY issued on VCI), the DMA controller will attempt to restart the cycle after being in the idle state for one clock. There is no mechanism to introduce delay between bursts.

6.5.5 Host Buffer Data Alignment

Receive data buffers always have to be WORD aligned (LSB address bits [1:0] must be 00).

Transmit data buffers do not have any restriction on data alignment. However, the DMA controller will access the host VCI bus on a WORD boundary for non-WORD aligned data. This would typically happen in the beginning and ending of the transfer. Example: If the host buffer address is 0x00000FF1, and 7 bytes needs to transferred then the controller will read 2 full WORD from address 0x00000FF0 by asserting all byte enables, but when transferring data to the destination data port, the extra byte will not be transferred.

6.5.6 Buffer Size Calculations

The DMA controller does not update the size fields in the TX and RX descriptors. Only the status field in each descriptor is modified by the controller. Software has to do size calculations. If a descriptor is not marked as last, then its buffer(s) are full and the amount of valid data in a

buffer is accurately indicated by its buffer size field. If a descriptor is marked as last, then the buffer may not be full. To compute the amount of valid data in this final buffer, software must read the frame length, and subtract the sum of the buffer sizes of the preceding buffers in this frame. CSR registers CSR20 (CUR_TX_BUF_ADDR) and CSR21 (CUR_RX_BUF_ADDR) may also be used to calculate size of data transferred.

6.5.7 DMA Suspended State Behavior

6.5.7.1 Transmit

The transmit engine enters the suspended state when either of these conditions occur:

- The DMA controller detects a descriptor owned by the host (TDES_0[31]=0). To resume, the driver must give the descriptor ownership to the DMA controller and then issue a poll demand command.
- A DMA transmission was aborted due to a local error.

In both cases the abnormal interrupt summary (CSR_5[15]) and the transmit interrupt (CSR_5[0]) are set and the appropriate status bit in TDES_0 is set. The position in the transmit list is retained. The retained position is that of the descriptor following the descriptor that was last closed.

Notes: The DMA controller does not poll the transmit descriptor list automatically. The driver must issue a transmit poll demand after rectifying the suspension cause explicitly.

6.5.7.2 Receive

The receive engine enters the suspended state when a receive buffer is unavailable. If a frame arrives when the receiver is in the suspended state the receive engine re-fetches the descriptor and if now owned by the DMA controller reenters the running state and starts frame reception. Receive polling resumes from the last list position.

The DMA controller generates a Receive Buffer Unavailable interrupt only once - when entering the suspended state from the running state. In the suspended state if a new frame is received and a descriptor is still not available the frame is discarded. Only in the suspended state does the controller respond to a Receive Poll Demand (i.e. a buffer is available before the next incoming frame) and enter the running state.

6.5.8 Interrupts

As described in earlier sections, there are a number of events that cause interrupt. CSR_5 contains all the bits that might cause interrupt. CSR_7 contains an enable bit for each of the events that can cause interrupt. Interrupt signal is asserted if any of the interrupt conditions are satisfied.

There are two groups of interrupts, normal and abnormal interrupts as outlined in CSR_5. Interrupts are cleared by writing a '1' to the bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both of the summary bits are cleared the interrupt signal is de-asserted.

Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, receive interrupt (CSR_5[6]) indicates that

one or more frames were transferred to the host buffer. The driver must scan all descriptors, from the last recorded position to the first one owned by the DMA controller.

An interrupt is generated only once for simultaneous, multiple events. The driver must scan CSR_5 for the interrupt cause. The interrupt is not generated again, unless a new interrupting event occurs after the driver has cleared the appropriate CSR_5 bit. For example, the controller generates a receive interrupt (CSR_5[6]) and the driver begins reading CSR_5. Next, a receive buffer unavailable (CSR_5[7]) occurs.

The driver clears the receive interrupt. The interrupt signal gets de-asserted for at least one cycle and then asserted again for the RX buffer unavailable interrupt.

6.6 MAC Operation

The heart of the Ethernet Controller is the Medium Access Controller (MAC). It performs a number of functions that are medium independent:

- Package data into Frames
- Calculate Cyclic Redundancy Check (CRC) and send with packet
- Transmit frames through the Physical Layer
- Monitor Physical Layer for any Collision Detection (CD)
- Handles “Backoff” procedure if collision detected
- Receive frames from Physical Layer
- Perform calculate CRC and compare with CRC received with from
- Send data up stream to Logic Link Layer if CRC is valid

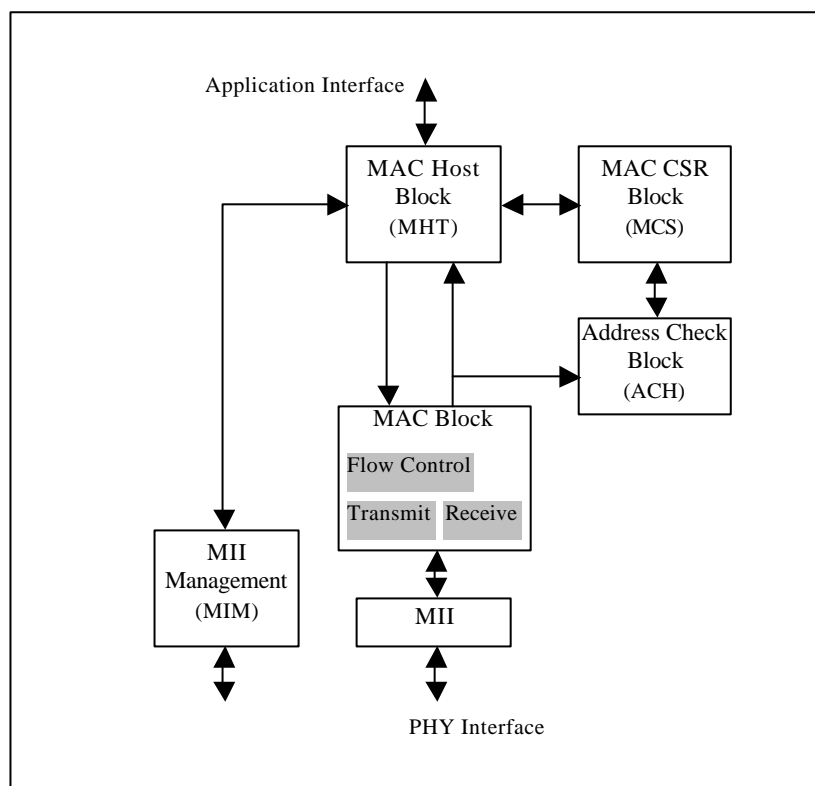


Figure 6-8 Block Diagram of MAC Block

10/100Mbps MAC Block (MAC): This block handles all the functionality of the Ethernet MAC Layer for both transmit and receive operations. CSMA/CD protocol is implemented for both half-duplex and full-duplex mode.

MAC CSR (MCS) Block: This block contains the control and status registers for the operation of the MAC. The registers/counters in this block can be accessed through the Host Interface on the application bus.

MAC-Host (MHT) Block: This block handles the synchronization of control and data signals between the Host bus and the MAC Interface.

Address Check (ACH) Block: This block checks the destination address field of all the incoming packets. Based on the type of address decoding selected this will indicate the MAC RX Interface block the result of the address checking.

MII Management Block (MIM): This block controls the read/write transactions to the PHY Registers that are in the external MII based PHY Controller ASIC, through the MII Management Interface using MDC and MDI/MDO signals.

In the following the functionality of the MAC is described.

6.6.1 Transmit Operation

A transmit operation is initiated by TLI. The transmitter starts transmitting the frame on to the MII interface until the transmitter successfully sends the packet or until transmit is abort because of excessive collisions, excessive deferral, late collision or any other abort conditions.

Transmission begins when the InterGap timer has expired. If the timer has reached 9.6 us (0.96 us at 100 Mbps) prior to the start indication of TLI, packet transmission begins immediately. If the start indication is given before 0.64 us (6.4 us at 10 Mbps) of the InterGap and the MAC was the station transmitting, a new packet begins transmission at 0.96 us (9.6 us at 10 Mbps) regardless of MII CRS signal. If the timer is greater than 0.64 us (6.4 us at 10 Mbps) and the MII CRS is high, the transmission defers until MII CRS is deasserted, at which time the 0.96 us (9.6 at 10 Mbps) timer activates and transmission starts after timeout.

The transmission of the frame starts with the MAC sending 7 bytes of preamble followed by an 8-bit SFD (10101011). The MAC at this point reads a byte at a time from internal buffer until the end of packet or a collision is seen on the bus or the packet is aborted. The MAC after transmitting the last nibble of the data from buffer, will append the PAD and FCS fields appropriately and completes the transmission. The appending of the PAD and FCS fields depend on the TDES1[26] and TDES1[23]. After successful transmission of the frame, the MAC returns the status information.

The MAC handles the collision situations in according with IEEE802.3. In the default mode the packet is retried for 16 times and after every collision the standard binary BackOff algorithm is implemented to wait for attempting to retransmit again. The MAC block contains a 20-bit LFSR (linear feedback shift register) counter. When a collision is sensed, a 32-bit jam pattern (5555_5555) is transmitted. After the jam pattern completes, a number of N (N depends on the collision number) is dumped in to a 10-bit counter, which in turn decrements by the turn over of 5.12 us (51.2 us at 10 Mbps) timer. Backoff lasts until the 10-bit counter reaches 0. Transmission is retried if 0.96us (9.6 us at 10 Mbps) timer expires. The deferral check helps to eliminate the "capture effect" by aborting a frame which had to defer for greater than 24,288 bit time. The capture effect is the unfair advantage one station gets in relation to other stations experiencing the same collision. This is further explained by understanding the following scenario. Station A and Station B both attempt to transmit simultaneously with a slot time of 52.2 us and each station has an initial collision count set to 0. The stations experience a collision and both stations increment their collision count to 1. Each station will then pick a Backoff time value uniformly distributed from 0 to 2^n-1 . It's to say, that station B selects a Backoff of 1 and station A selects a Backoff of 0. Station A successfully transmits its frame while station B waits for the data transmission by station A before it begins its re-transmission attempt. Collision count at station B remains at 1, while collision count of station A is reset to 0. If station A has another packet to transmit then both station contend for the line again. If these stations collide, the Backoff value for station A is 0 or 1 slots. The Backoff value for station B is 0, 1, 2 or 3 slots because the collision count is now at 2 (station A's collision count 1). Station A is more likely to succeed in getting the line while station B begins deferral of completing it's Backoff interval. This process could continue until station B reaches the maximum number of collisions,

16, while attempting to transmit its data. If deferral lasts longer than 24,288 bit times, the MAC aborts the transmission only if Deferral Check (bit 5 of MAC Control Register) is set. If a packet cannot be transmitted after making the selected number of attempts, the transmission is aborted and the Excessive Collisions flag is set in the status register. If a collision occurs during preamble, the preamble-SFD sequence is completed prior to jamming. If a later collision (after 512 bit times, including preamble and SFD) occurs, the MAC aborts the transmission and the late collision flag is set in the status register. If the Late Collision Control (bit 12 of MAC Control Register) is set, the MAC does not abort the transmission after the late collision but continues up to the maximum retry limit (16).

6.6.2 Receive Operation

The Receiver continuously monitors the MII RX_DV line for incoming packets. Once the MII RX_DV is asserted, the receiver senses for SFD (10101011) sequence in the incoming stream. From this point the receiver assembles a byte of information from the incoming frame and forwards the byte to the MAC-Host. Normally a byte is assembled every other clock. This process continues until the MII RX_DV signal is de-asserted at which point the receiver checks the CRC calculated up to the last full byte. If MII CRS falls on a byte boundary, the packet is either good or has a CRC error. If MII RX_DV fails on a non-byte boundary, but the last full byte received has a good CRC, it is a good packet with dribble bits. The receiver informs the MAC-Host of the end of the packet and presenting packet's status.

Other status information presented to the MAC-Host include the Jabber timeout (packet with > 1518 bytes and a bad CRC or alignment error), long packet (> 1518 bytes with good CRC), and others which are described in the receive packet status.

If the MAC block is operating in the half-duplex mode, the receiver ignores any packets that start while the TX_EN is asserted, to avoid buffering one's own transmitted packet. In order to prevent glitches on the MII CRS signal during a collision situation from affecting the receiver, the receiver ignores high-to-low transitions of MII CRS is a packet reception is in progress and the MII COL signal is present. In full-duplex mode, the receiver ignores the MII COL signal.

6.6.3 Address Check

Ethernet address consist of two 6-byte fields: one field for the destination address and another one for source address. The first bit of the destination address signifies whether it is a physical address or a multicast address.

Table 6-11 Destination Address Bit 0

Field	Description
0	Station Address (Physical)
1	Multicast Address

The address check block filters the frame based on the Ethernet receive filter mode that has been enabled. The filter modes are specified based on the state of the control signals and following table shows the various filtering modes that are used by the Address Check Block (ACH). If the frame fails the filter, the host has an option to either accept the packet or ignore the packet.

Table 6-12 Address Filtering Mode

Pass All Multicast	Promiscuous Mode	Inverse Filtering	Hash Only	Hash/ Perfect	Description
0	0	0	0	0	MAC Address Perfect filtering only for all addresses.
0	0	0	0	1	MAC Address Perfect filtering for physical address and Hash filtering for multicast addresses
0	0	0	1	1	Hash Filtering for physical and multicast addresses
0	0	1	0	0	Inverse Filtering
X	1	0	X	X	Promiscuous
1	0	0	0	X	Pass all multicast frames. Frames with physical addresses are perfect filtered
1	0	0	1	1	Pass all multicast frames. Frames with physical addresses are hash filtered

6.6.3.1 Perfect Filtering

This type of filtering mode, passes only frames with the exact match of the destination address field of the incoming frame with the value programmed into the MAC Address High Register and MAC Address Low Register in the CRC module. The MAC address is formed by the concatenation of the above two registers in the MAC CSR block.

6.6.3.2 Hash Only Filtering

This type of filtering checks for incoming receive packet with destination address which is either multicast or physical, and executes an imperfect address filtering against the hash table.

During imperfect hash filtering, the destination address in the incoming frame is passed through the CRC logic and the upper 6 bits of the CRC register are used to index the contents of the Hash Table. The Hash Table is formed merging the registers Multicast Hash Table high and Multicast Hash Table Low in the MAC CSR block to form a 64-bit hash table. The most significant bit determines the register to be used (High/Low), while the other five bits determine the bit within the register. A value of '000000' selects the bit '0' of Multicast Hash Table Low Register and a value of '111111' selects the bit 31 of the Multicast Hash Table High Register.

6.6.3.3 Hash Perfect Filtering

In this type of filtering, if the received frame is a physical address, a perfect filtering of the destination field of the incoming frame with the value programmed into the MAC Address High Register and the MAC Address Low Register is performed. But in the case when the incoming frame is a multicast frame, an imperfect address filtering against the hash table is performed. The imperfect filtering against the hash table is same imperfect filtering process described for the hash only filtering explained in the section above.

6.6.3.4 Inverse Filtering

In this type of filtering the incoming frames with the destination address not matching the perfect address (value programmed into the MAC Address High Register and the MAC Address Low Register) and rejects the frames with the destination addresses matching the perfect address.

6.6.4 Flow Control

The MAC allows full-duplex using the PAUSE operation and Control Frame.

The PAUSE operation is used to inhibit data transmission of data frames for a specified period of time. A PAUSE operation consists of a frame containing the globally assigned multicast address (01-80-C2-00-00-01), The PAUSE opcode, and a parameter indicating the quanta of slot times (512 bit times) to inhibit data transmissions. The PAUSE parameter may range from 0 to 65,535 slot times. The MAC block on receiving a frame with the reserved multicast address and PAUSE opcode will inhibit data frame transmissions of the length of time indicated. If a PAUSE request is received while a transmission is in progress, then PAUSE will take effect after the transmission is complete. Control frames are received and processed by MAC and are passed on. The MRI Interface can use the Pass Control Frame bit in the CSR6 to block the control frames from being transferred to the host memory.

The MAC also transmits Control frames (PAUSE command) on the request from the host. The Host requests the MAC to transmit a control frame by setting Flow Control Busy bit of Flow Control Register. The MAC block will construct a Control frame with the appropriate values set in Flow Control Register and transmits the frame to the MII interface. The transmission of the control frame is not affected by the current state of the PAUSE timer value that is set because of a recently received control frame. Upon the completion of control frame transmission, the MAC will clear the Control Frame Busy bit of Control Frame Register.

6.6.5 Virtual Local Area Network (VLAN)

VLAN is a means to form a broadcast domain without the restriction on the physical or geographical location on the members of that domain. VLAN can be implemented in any number of different factors such as

Physical Port
 MAC Address
 Layer-3 unicast address
 Multicast Address
 Date/Time in combination with MAC Address, etc.

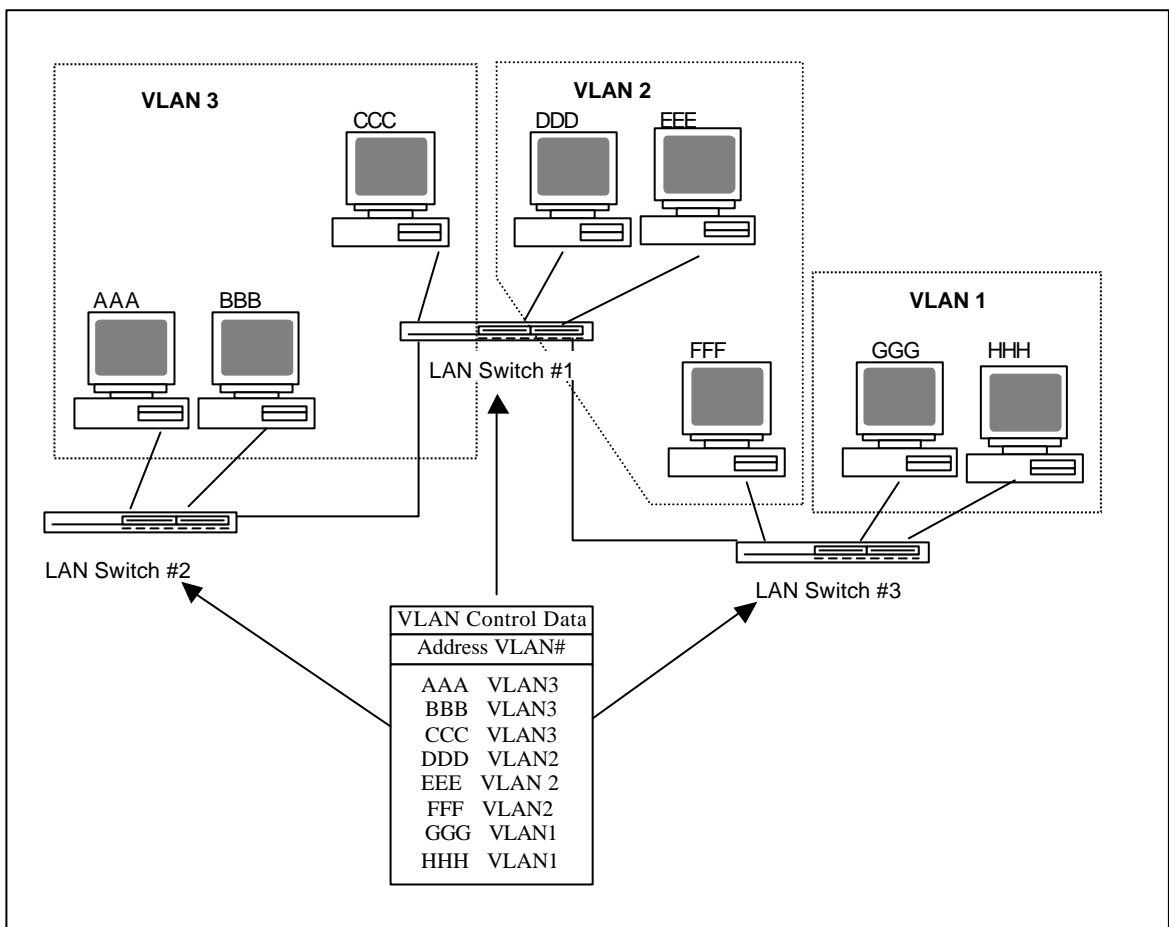


Figure 6-9 VLAN Topology

An example of a VLAN is depicted in above figure. It demonstrates the freedom from physical constraint on the network, and the ability to divide a single switched network into a smaller broadcast domain.

Moreover, VLAN offers a number of other advantages such as:

Configuration: Changes to an existing VLAN can be made on the network administrative level, rather than on the hardware level. A member of a VLAN can thus, change its MAC Address, or its port and still be a member of the same VLAN. Extra routing is not necessary.

Security: VLAN can improve security by demanding a predefined authentication before joining a new member to the domain.

Network efficiencies: Allow shielding one system resource from traffic not meant for that resource. A workstation in on VLAN is shielded from traffic on another VLAN, which increases the efficiency of that workstation.

Broadcast containment: Leakage of broadcast frames from one VLAN to another is prevented.

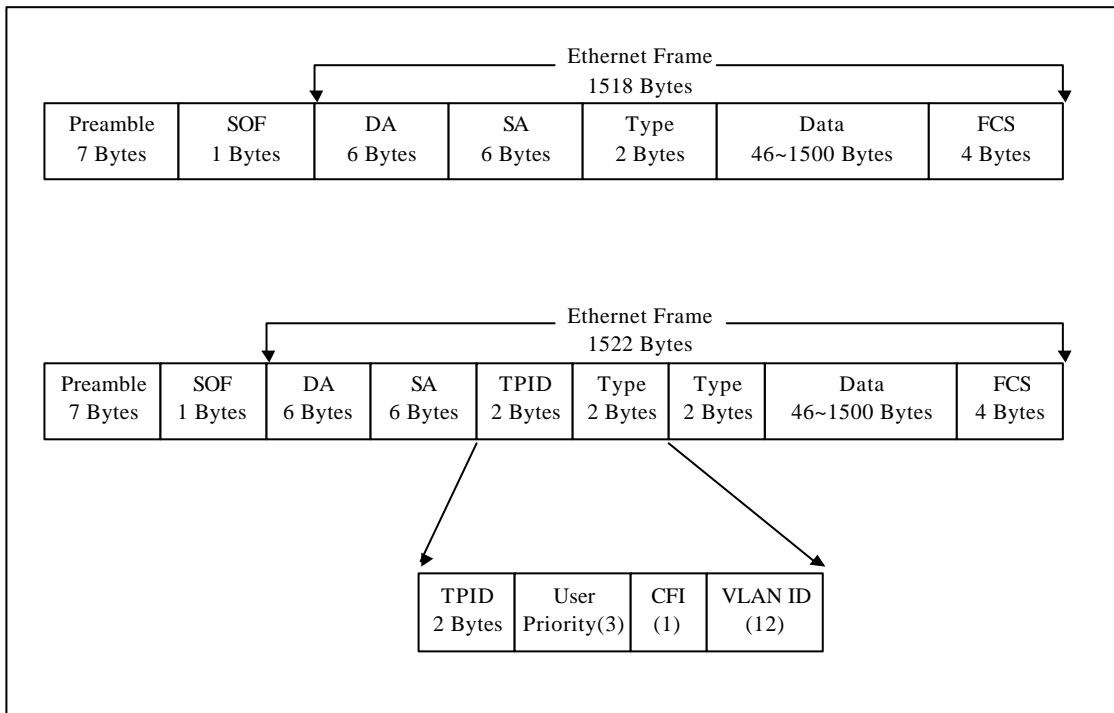


Figure 6-10 VLAN Frame

When the members of a VLAN are not located on the same physical medium, the VLAN uses a Tag to help it determined how to forward the frame from one member to another. The Tag structure has been proprietary until IEEE has released a supplement to the 802.3 frames where the VLAN Frame Structure including the tag has been defined. This new frame structure for VLAN is depicted in above figure.

The MAC block recognizes transmit and received frames that are tagged with either one-level or two-level VLAN Ids. The MAC compares the thirteenth and fourteenth bytes of transmit and receive frames to the contents of both the one-level VLAN tag register and two-level VLAN tag register. If a match is found, the MAC block identifies the frame as either a one-level or two-level VLAN frame, depending on where the match was found. Upon recognizing that a frame has a VLAN tag, counter thresholds are adjusted to account for the extra bytes that the VLAN tag adds to the frame. That is the maximum length of the good packet is changed from 1518 bytes to 1522 bytes.

6.6.6 MII Management

6.6.6.1 MII Management Write Operation

To write to any register in the PHY, the host sets the PHY address, MII register address, MII Write bit in MII Address Register and write data in MII Data Register, then set the MII Busy bit to 1 in MII Address Register. The MIM block at this point starts a write operation on the MII Management Interface using the Management Frame Format specified in the MII Specification. The host should not change these registers while the transaction is going on. After the write operation is completed, the MIM block will clear the MII Busy bit to 0.

6.6.6.2 MII Management Read Operation

To read from any register in the PHY, the host sets the PHY address, MII register address, and clear MII Write bit in MII Address Register, then set the MII Busy bit to 1 in MII Address Register. The MIM block at this point starts a read operation on the MII Management Interface using the Management Frame Format specified in the MII Specification. The host should change these registers while the transaction is going on. After the read operation is completed, the MIM block will clear the MII Busy bit and presents the read data in MII Data Register.

7 USB Host Controller

7.1 Overview

The USB Host Controller(UHC) is Open Host Controller Interface (OHCI)-compatible and USB Revision 1.1-compatible. It supports both low-speed (1.5 Mbps) and full-speed (12 Mbps) USB devices. Two downstream ports are provided. The interrupt from the USB host controller is active low.

See the *Universal Serial Bus Specification Revision 1.1* and the *Open HCI – Open Host Controller Specification for USB* for details of the interface operation.

7.1.1 Block Diagram

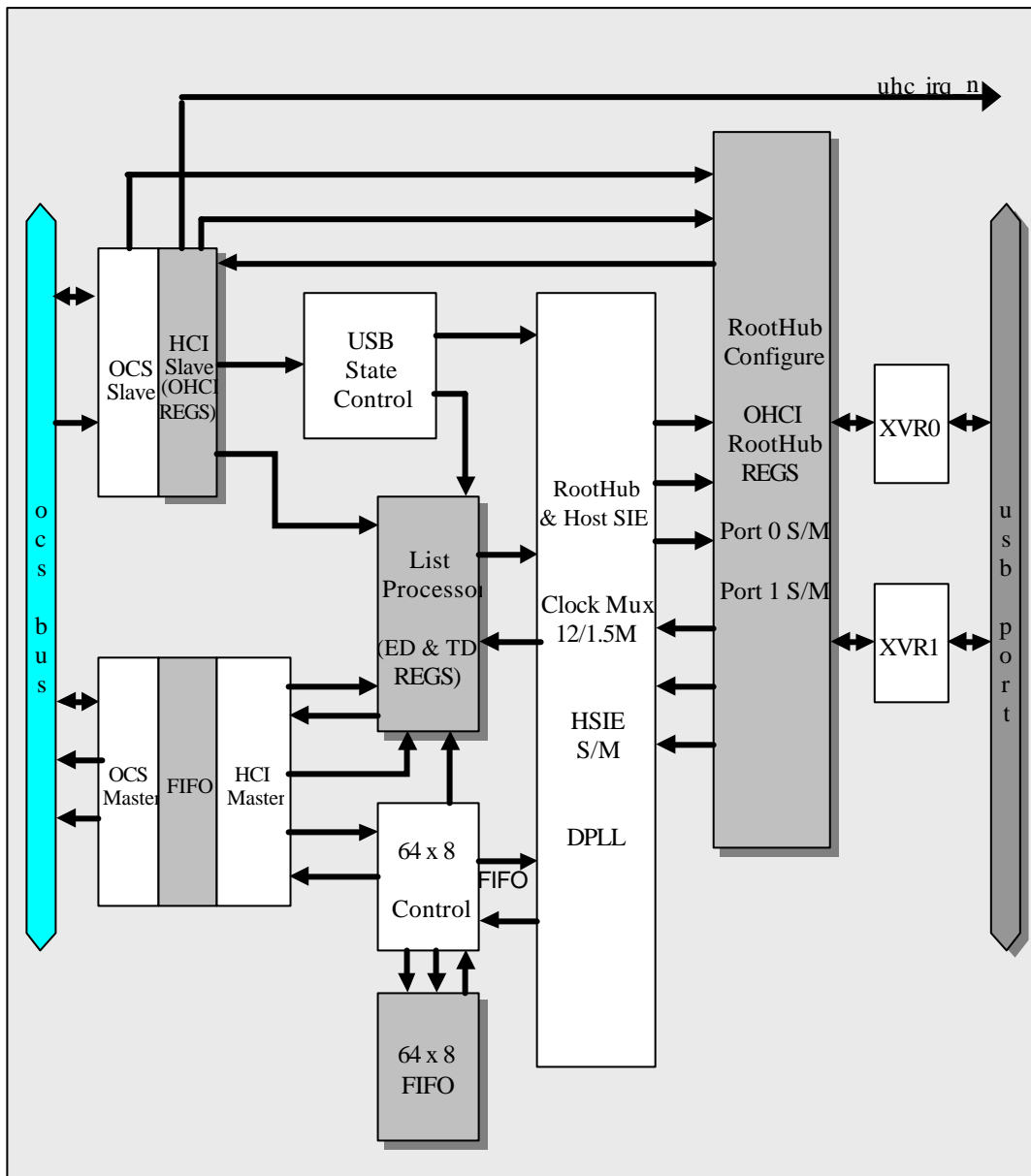


Figure 7-1 UHC Block Diagram

7.2 Pin Configuration

Following table list the UHC pins of Arca210.

Table 7-1 UHC Pin Configuration

Pin Name	I/O	Signal	Function
USB Port 0 Data+	I/O	DPLS0	Downstream Plus USB Port 0 I/O
USB Port 1 Data+	I/O	DPLS1	Downstream Plus USB Port 1 I/O
USB Port 0 Data-	I/O	DMNS0	Downstream Minus USB Port 0 I/O
USB Port 1 Data-	I/O	DMNS1	Downstream Minus USB Port 1 I/O
USB Port 0 Power Status	O	PPWR0	USB Port 0 Power Switch used to enable or disable the external voltage supplying power to the port and is de-asserted when a power supply problem is detected at OVC0 pin
USB Port 1 Power Status	O	PPWR1	USB Port 1 Power Switch used to enable or disable the external voltage supplying power to the port and is de-asserted when a power supply problem is detected at OVC1 pin
USB Port 0 OverCurrent Status	I	OVC0	USB Port 0 Overcurrent to indicate there is a power supply problem with the port
USB Port 1 OverCurrent Status	I	OVC1	USB Port 1 Overcurrent to indicate there is a power supply problem with the port

7.3 Register Configuration

The base of the Open HCI register block is at physical address H'E1050000. Only 32-bit accesses are allowed to the Open HCI registers.

See *Open HCI – Open Host Controller Specification for USB* for details of the each register.

Table 7-2 UHC Registers

Name	Full Name	R/W	Initial value	Address	Access Size
HcRevision	HcRevision Register	R	H'00000010	H'E1050000	32
HcControl	HcControl Register	R/W	H'00000000	H'E1050004	32
HcCommandStatus	HcCommandStatus Register	R/W	H'00000000	H'E1050008	32
HcInterruptStatus	HcInterruptStatus Register	R/W	H'00000000	H'E105000C	32
HcInterruptEnable	HcInterruptEnable Register	R/W	H'00000000	H'E1050010	32
HcInterruptDisable	HcInterruptDisable Register	R/W	H'00000000	H'E1050014	32
HcHCCA	HcHCCA Register	R/W	H'00000000	H'E1050018	32
HcPeriodCurrentED	HcPeriodCurrentED Register	R	H'00000000	H'E105001C	32
HcControlHeadED	HcControlHeadED Register	R/W	H'00000000	H'E1050020	32
HcControlCurrentED	HcControlCurrentED Register	R/W	H'00000000	H'E1050024	32
HcBulkHeadED	HcBulkHeadED Register	R/W	H'00000000	H'E1050028	32
HcBulkCurrentED	HcBulkCurrentED Register	R/W	H'00000000	H'E105002C	32
HcDoneHead	HcDoneHead Register	R	H'00000000	H'E1050030	32
HcFmInterval	HcFmInterval Register	R/W	H'00002EDF	H'E1050034	32
HcFmRemaining	HcFmRemaining Register	R	H'00000000	H'E1050038	32
HcFmNumber	HcFmNumber Register	R	H'00000000	H'E105003C	32
HcPeriodicStart	HcPeriodicStart Register	R/W	H'00000000	H'E1050040	32
HcLSThreshold	HcLSThreshold Register	R/W	H'00000628	H'E1050044	32
HcRhDescriptorA	HcRhDescriptorA Register	R/W *1	H'02000902	H'E1050048	32
HcRhDescriptorB	HcRhDescriptorB Register	R/W	H'00060000	H'E105004C	32
HcRhStatus	HcRhStatus Register	R/W *2	H'00000000	H'E1050050	32
HcRhPortStatus 1	HcRhPortStatus 1 Register	R/W	H'00000100	H'E1050054	32

HcRhPortStatus 2	HcRhPortStatus 2 Register	R/W	H'00000100	H'E1050058	32
------------------	------------------------------	-----	------------	------------	----

Note:

*1 Bit 10 and Bit 7-0 are read only.

*2 Bit 1 is read only and Bit 31 is write only.

8 Clock Generation Unit (CGU)

8.1 Overview

The Clock Generation Unit (CGU) has one input clock and supplies four clocks for internal modules and four clocks to the external of Arca.

The input clocks an oscillator circuit output in CGU or an external clock directly input.

The output clocks are:

- ICLK, Internal clock
- PCLK, peripheral clock
- SCLK, System clock
- RTCCLK, Real time clock
- CKO0, CKO1, CKO2 and CKO3, which are equal to system clock, is output to the external of Arca

Following table shows the clock sources of internal blocks.

Table 8-1 Arca210 Clock Source

Block name	Clock source
CPU	iclk
Cache	iclk
MMU	iclk
EMI	sclk
DMA	sclk
UHC	sclk
ETHC	sclk
PCIC	sclk, pci_clk
INTC	pclk, rtcclk
UART	pclk
UART2	pclk
IRDA	pclk
AC97	pclk
I2CI	pclk
RTC	pclk, rtcclk
TMU	pclk, rtcclk
WDT	pclk
CGU	pclk, rtcclk

8.1.1 CGU Functional Feature

- Generates four clocks: internal clock (iclk) for CPU core, system clock (sclk) for system bus devices, peripheral clock (pclk) and real time clock (rtcclk) for peripheral devices.
- 7 clock modes can be selected by MD0 ~ MD2.
- On-chip oscillator circuit.

- On-chip phase-locked loop (PLL) with programmable multiple-ratio. An internal counter is used to ensure PLL stabilization time.
- The PLL on/off is programmable by software.
- iclk, pclk and sclk frequency can be changed separately for software by setting division ratio.
- Input clock source can be selected from oscillator or an external clock input.
- Clock output function.

8.1.2 CGU Block Diagram

Following figure illustrates a block diagram of CGU.

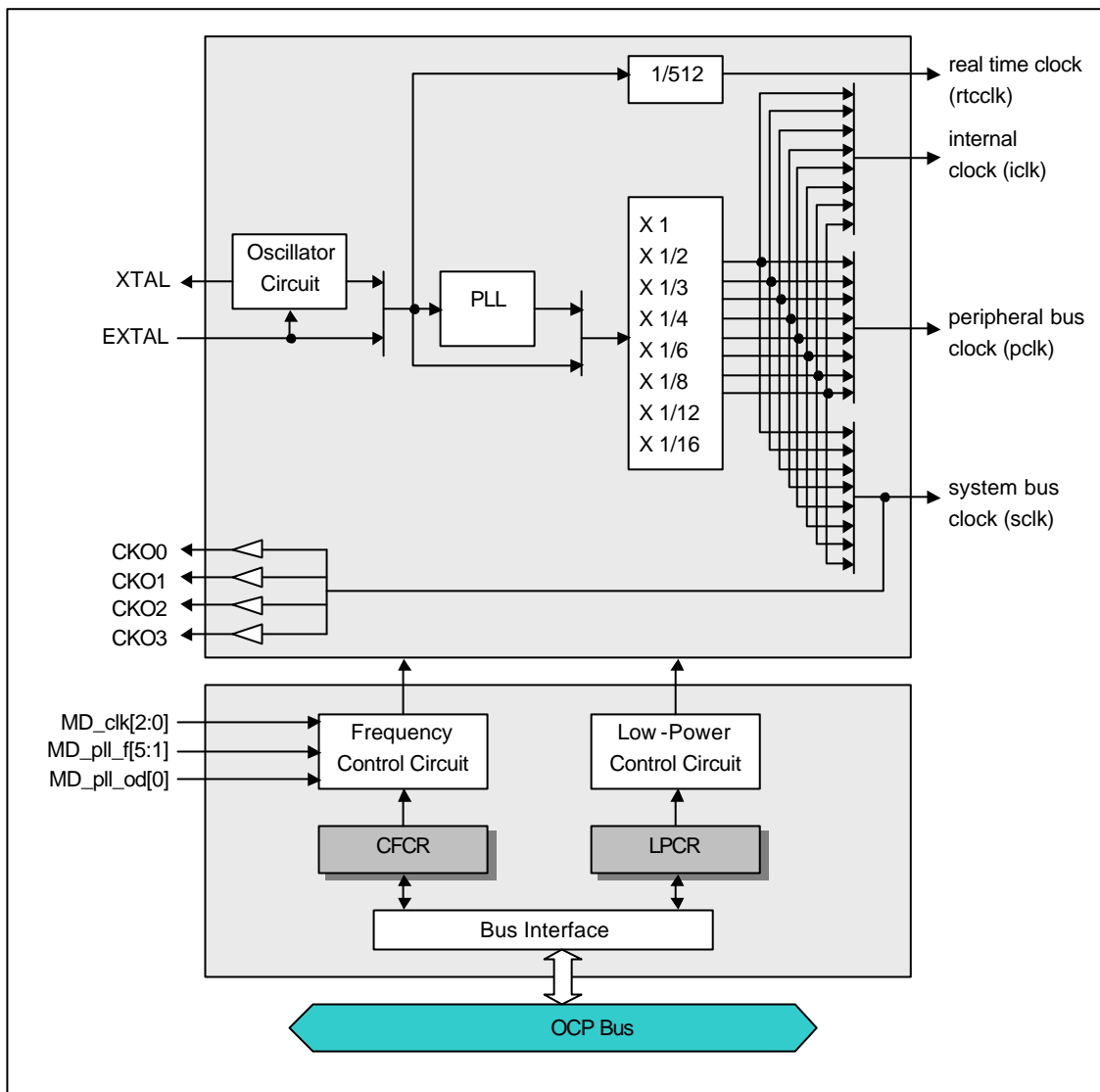


Figure 8-1 CGU Block Diagram

8.1.3 CGU Pin Configuration

Table 8-2 CGU Pin Configuration

Pin Name	I/O	Signal	Description
Clock mode control pins	I*	MD_CLK [2:0]	Set clock modes
PLL multiple-ratio control pins	I*	MD_PLL_F [5:1]	Set PLL feed-back divider
	I*	MD_PLL_OD [0]	Set PLL output divider
Oscillator pins	O	XTAL	Connects crystal resonator
	I	EXTAL	Connects crystal resonator or used as an external clock input pin.
Clock output pin	O	CKO0	Output sclk to external.
	O	CKO1	
	O	CKO2	
	O	CKO3	

Note: * These pins are registered during power-on reset

8.2 Clock Modes

Following table shows PLL status and iclk, sclk, and pclk division ratio, CFCR initial value, input and output clock frequency range in different clock mode.

Table 8-3 CGU Clock Modes

Clock mode	MD_clk	PLL	CFCR initial value	CPU clock (iclk) divider	System clock (sclk) divider	Device clock (pclk) divider
0	000	ON	H'xxxx 0800	x 1	x 1	x 1
1	001	ON	H'xxxx 1848	x 1	x 1/2	x 1/2
2	010	ON	H'xxxx 2890	x 1	x 1/3	x 1/3
3	011	ON	H'xxxx 38D8	x 1	x 1/4	x 1/4
4	100	OFF	H'xxxx 4000	x 1	x 1	x 1
5	101	ON	H'xxxx 5920	x 1	x 1/6	x 1/6
6	110	ON	H'xxxx 6968	x 1	x 1/8	x 1/8

Following table specifies CFCR setting and clock frequency with different CFCR setting. These settings are available in all clock modes.

Table 8-4 CFCR Setting and Clock Frequency

Frequency division ratio			PFR2~0	SFR2~0	IFR2~0
iclk	sclk	pclk			
1	1	1	000	000	000
		1/2	001		
		1/3	010		
		1/4	011		
		1/6	100		
		1/8	101		
		1/12	110		
		1/16	111		
		1/2	1/2		
	1/4		011		
	1/6		100		
	1/8		101		
	1/16		111		
	1/3	1/3	010	010	
		1/6	100		
		1/12	110		
	1/4	1/4	011	011	
		1/8	101		
		1/16	111		
	1/6	1/6	100	100	
		1/12	110		
	1/8	1/8	101	101	
		1/16	111		
	1/12	1/12	110	110	
	1/16	1/16	111	111	
1/2	1/2	1/2	001	001	001
		1/4	011		
		1/6	100		
		1/8	101		
		1/12	110		
		1/16	111		
	1/4	1/4	011	011	
		1/8	101		
		1/12	110		
		1/16	111		
	1/6	1/6	100	100	
		1/12	110		
	1/8	1/8	101	101	
		1/16	111		
	1/12	1/12	110	110	
	1/16	1/16	111	111	

Table 8-4 (Continued)

Frequency division ratio			PFR2~0	SFR2~0	IFR2~0
iclk	sclk	pclk			
1/3	1/3	1/3	010	010	010
		1/6	100		
		1/12	110		
	1/6	1/6	100	100	
		1/12	110		
	1/12	1/12	110	110	
1/4	1/4	1/4	011	011	011
		1/8	101		
		1/16	111		
	1/8	1/8	101	101	
		1/16	111		
	1/12	1/12	110	110	
	1/16	1/16	111	111	
1/6	1/6	1/6	100	100	100
		1/12	110		
	1/12	1/12	111	111	
1/8	1/8	1/8	101	101	101
		1/16	111		
	1/16	1/16	111	111	

- Take PLL output clock frequency value as 1.
- Don't set CFCR value outside of this table.

8.3 CGU Registers

Following table shows CGU registers

Table 8-5 CGU Registers Configuration

Name	Full Name	R/W	Initial Value	Address	Access Size
CFCR	Clock Frequency Control Register	R/W	Undefined*	H'E0000100	32

Note: * - Depend on the clock mode control pins and PLL control pins. Please refer to Table 8-2.

8.3.1 Clock Frequency Control Register

The Clock Frequency Control Register (CFCR) is a 32-bit read/write register, which controls PLL multiple-ratio, clock division ratios and PLL on/off. It is initialized only by power-on reset. The initial value depends on the clock mode and PLL control pins (refer to Table 8-2). Only word access can be used on CFCR.

Bit:	31	30	29	28	27	26	25	24
Read:	PLL	PLL	PLL	PLL	PLL	PLL	PLL	PLL
Write:	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1
Reset:	0	0	0	*	*	*	*	*

Bit:	23	22	21	20	19	18	17	16
Read:	PLL	PLL	PLL	PLL	PLL	PLL	PLL	PLL
Write:	FD0	RD4	RD3	RD2	RD1	RD0	OD1	OD0
Reset:	0	0	0	0	0	0	0	*

Bit:	15	14	13	12	11	10	9	8
Read:		MD2	MD1	MD0	PLLEN			PFR2
Write:								
Reset:	0	*	*	*	*	0	0	*

Bit:	7	6	5	4	3	2	1	0
Read:	PFR1	PFR0	SFR2	SFR1	SFR0	IFR2	IFR1	IFR0
Write:								
Reset:	*	*	*	*	*	0	0	0

Note: * - Initial value depends on input pins.

- **Bit 31~23 PLLFD:** PLL Feedback Division Ratio. Controls the PLL feedback divider. The MD_pll_f[5:1] pins are registered into PLLFD5~1 during power-on. PLLFD8~6 and PLLFD0 are initialized to 0 by a power-on reset.
- **Bit 22~18 PLLRD:** PLL Input Division Ratio. Controls the PLL input divider. PLLRD is initialized to 2 by a power-on reset.
- **Bit 17~16 PLLOD:** PLL Output Division Ratio. Controls the PLL output divider. The MD_pll_od[0] pin is registered into PLLOD0 during power-on reset.

Bit 17: PLL0D1	Bit 16: PLL0D0	Divided by
0	0	1
0	1	2
1	0	2
1	1	4

- **Bit 14~12 MD2~0:** Clock Mode. Specifies the clock mode. The MD_clk pins are registered into this field during power-on reset, respectively. This field is read only.
- **Bit 15, 10~9:** Reserved bits. Writes to these bits have no effect and always read as 0.
- **Bit 11 PLEN:** PLL Enable. Control the PLL on/off state.

Bit 11: PLEN	Description	Initial
0	PLL is off	Clock mode 4
1	PLL is on	Clock mode 0, 1, 2, 3, 5, 6

- **Bit 8~6 PFR:** Peripheral Clock Frequency Division Ratio. Specified the pclk division ratio.

Bit 8: PFR2	Bit 7: PFR1	Bit 6: PFR0	Description	Initial
0	0	0	X 1	Clock mode 0, 4
0	0	1	X 1/2	Clock mode 1
0	1	0	X 1/3	Clock mode 2
0	1	1	X 1/4	Clock mode 3
1	0	0	X 1/6	Clock mode 5
1	0	1	X 1/8	Clock mode 6
1	1	0	X 1/12	
1	1	1	X 1/16	

- **Bit 5~3 SFR:** System Bus Clock Frequency Division Ratio. Specifies the sclk division ratio.

Bit 5: SFR2	Bit 4: SFR1	Bit 3: SFR0	Description	Initial
0	0	0	X 1	Clock mode 0, 4
0	0	1	X 1/2	Clock mode 1
0	1	0	X 1/3	Clock mode 2
0	1	1	X 1/4	Clock mode 3
1	0	0	X 1/6	Clock mode 5
1	0	1	X 1/8	Clock mode 6
1	1	0	X 1/12	
1	1	1	X 1/16	

- **Bit 2~0 IFR:** Internal Clock Frequency Division Ratio. Specifies the iclk division ratio.

Bit 2: IFR2	Bit 1: IFR1	Bit 0: IFR0	Description	
0	0	0	X 1	(initial value)
0	0	1	X 1/2	
0	1	0	X 1/3	
0	1	1	X 1/4	
1	0	0	X 1/6	
1	0	1	X 1/8	
1	1	0	X 1/12	
1	1	1	X 1/16	

8.4 CGU Operation

8.4.1 PLL

The programmable phase-locked-loop (PLL) generates the overall system operating frequency in multiples of the input clock frequency. The PLL reference clock can be generated from the oscillator circuit or be input from external of the chip directly.

The main purpose of the PLL is to generate a stable reference frequency by multiplying the frequency and eliminating the clock skew. The PLL allows the processor to operate at a high internal clock frequency using low frequency clock input, providing two advantages. First, lower frequency clock input reduces the overall electromagnetic interference generated by the system. Second, the programmability of the oscillator enables the system to operate at a variety of frequencies with only a single external clock source. The PLL block diagram is shown in following figure.

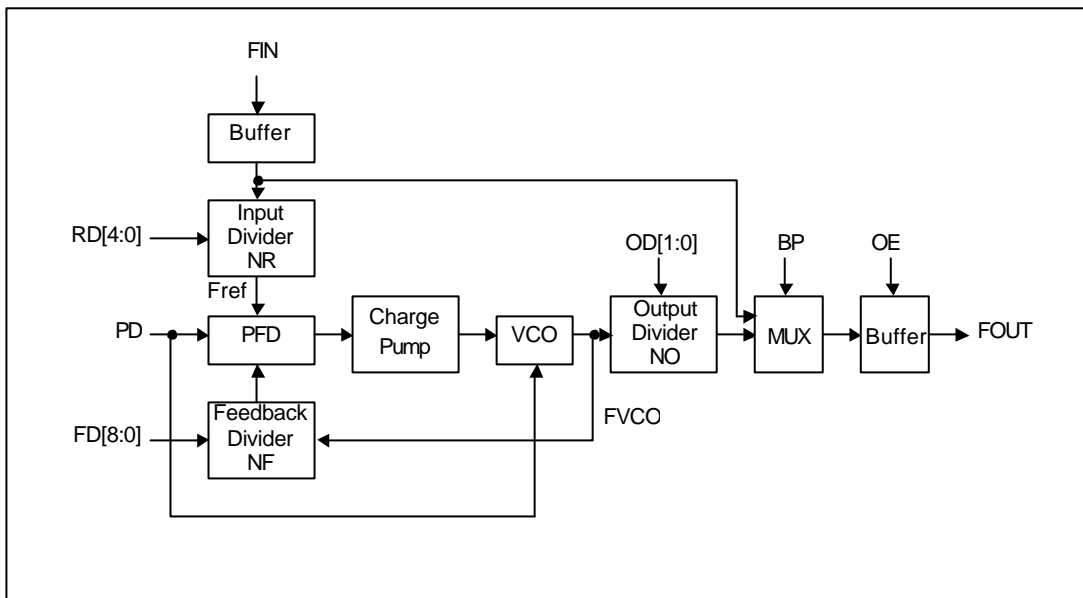


Figure 2 Block Diagram of PLL

- PLL Divider Value Setting

There are 3 divider values (NR, NF and NO) to set the PLL output clock frequency FOUT:

- ❑ Input Divider Value NR

$$NR = RD + 2$$

- ❑ Feedback Divider Value NF

$$NF = FD + 2$$

- ❑ Output Divider Value NO

Output Divider Setting (OD) *	Output Divider Value (NO)
0	1
1	2
2	2
3	4

Note: When OD0=1 or OD1=1, FOUT would have around 50% duty-cycle.

- PLL Output Clock Frequency Setting

$$\mathbf{FOUT = (FIN * NF) / (NR * NO)}$$

Meanwhile, the following **constraints** must be followed:

- ❑ **800KHz < Fref < 8MHz**
- ❑ **200MHz < FVCO < 400MHz**, FVCO > 250MHz is preferred

Where

PLL's comparison frequency **Fref = FIN/(NR*2)**.

PLL's VCO frequency **FVCO = FIN*(NF*2)/(NR*2)**.

For example, if we want to get a 266MHz output frequency clock (when using the 14.318 MHz crystal oscillator as input), we need the NF/NR=130/7, NO=1, So, we get

NF = 130, FD8:FD0 = 128 ----- 0 0 1 0 0 0 0 0

NR = 7, RD4:RD0 = 5 ----- 0 0 1 0 1

NO = 1, OD1:OD0 = 0 ----- 0 0

8.4.2 Change Clock Frequency

Internal clock frequency can be changed with three methods: changing PLL on/off, changing PLL multiple-ratio and changing clock division ratio. These controls are performed by software to set Clock Frequency Control Register.

Note: When PLEN is 0, Any PLLFD/PLLRD/PLLOD change has no effect on clock frequency.

8.4.2.1 Change PLL from Off to On

When PLL is changed from off to on, an on-chip PLL stabilization time is required. All internal clocks will be stopped for several clock cycles. Clock output via CKO pin is stopped also.

1. Set PLEN bit in CFCR to 1. Clock division ratios (IFR, SFR and PFR) and PLL multiple ratio (PLLFD, PLLRD and PLLOD) can be changed simultaneously.
2. The on-chip PLL starts to lock phase. Divider clock source is still external clock input. Stable clocks are output during this period.

3. After on-chip PLL stabilization, CGU will send stop request to OCS Bus arbiter and EMI. The OCS Bus arbiter will not grant any following bus request. EMI will drive DRAM/SDRAM from auto-refresh mode to self-refresh mode. Then all clocks are stopped.
4. Divider clock source is switched from external clock input to PLL output.
5. Clock supply is enabled after several clock cycles. And then the processor resumes operation.

Note: In step 1, clocks are still running. Instructions are executed successively. Don't re-change PLLLEN during this period (PLL stabilization time: about 1ms). Otherwise, unpredictable error may occur.

8.4.2.2 Change PLL from On to Off

When PLL is changed from on to off, no stabilization time is needed. PLL will be shut off immediately. The divider clock source will be switched from PLL output directly to external clock input. All internal clocks will be stopped for several clock cycles. Clock output via CKO pin is stopped also.

1. Set the PLLLEN bit in CFCR to 0. Clock division ratios (IFR, SFR and PFR) can be changed simultaneously.
2. First, CGU will send stop request to OCS Bus arbiter and EMI. The OCS Bus arbiter will not grant any following request after current access complete. EMI will drive SDRAM from auto-refresh mode to self-refresh mode. Then All clocks are stopped.
3. PLL is shut off and clock switching is performed.
4. Clock supply is re-enabled after several clock cycles. And then the processor resumes operation.

8.4.2.3 Change PLL Multiple Ratio

When PLL multiple ratio is changed, a on-chip PLL stabilization time is required. All internal clocks will be stopped during PLL stabilization time. Clock output via CKO pin is stopped also.

1. Set the PLLFD, PLLRD or PLLOD separately or simultaneously to the desired value. Clock division ratios (IFR, SFR and PFR) may be changed simultaneously.
2. First, CGU will send stop request to OCS Bus arbiter and EMI. The OCS Bus arbiter will not grant any following request. EMI will drive SDRAM from auto-refresh mode to self-refresh mode. Then All clocks are stopped.
3. The on-chip PLL start to re-lock phase and clock switching is performed.
4. After on-chip PLL stabilization, clock supply is re-enabled. And then the processor resumes operation.

8.4.2.4 Change Clock Division Ratio

Clock (iclk, sclk and pclk) frequencies can be changed separately or simultaneously by changing division ratio. All or some clocks will be stopped for several cycles. Which clock is stopped according to following table. Clock output via CKO pin is stopped when sclk is stopped.

iclk frequency	sclk frequency	pclk frequency	iclk	sclk	pclk
changed	unchanged	unchanged	stop	supply	supply
X	changed	unchanged	stop	stop	supply
X	X	changed	stop	stop	stop

1. Set the IFR0~2 or SFR0~2 or PFR0~2 separately or simultaneously to the desired value.
2. First, CGU will send stop request to OCS Bus arbiter. The OCS Bus arbiter will not grant any following request. Then clocks are stopped according whose frequency is changed.
3. Clock switching is performed.
4. Clock supply is re-enabled after several clock cycles. And then the processor resumes operation.

9 Power/Reset Management Controller (PMC)

9.1 Overview

In the Low-Power mode, part or whole processor is halted. This will reduce power consumption. The Power/Reset Management Controller contains low-power mode control and reset sequence control.

9.1.1 Low-Power Modes and Function

Arca supports five low-power modes and function:

- Normal mode

When the processor starts operating after reset, system is in normal mode and all clocks run continuously.

- Doze mode

Doze mode is entered by setting DOZE bit of LPCR to 1. In doze mode, clock is bursted to CPU core and the clock duty is set by DUTY field of LPCR. For various core resource requirements, you can program the clock-duty cycles of any value between 0/31 and 31/31. This effectively provides a variable clock frequency (and power dissipation) between 0% and 100% of the clock frequency in 3% incremental steps. Doze mode is canceled by reset, interrupt or clearing DOZE bit to 0. Continuous clock is supplied immediately after doze mode is canceled. Clocks of OCS bus and OCP bus run continuously in doze mode.

- Sleep mode

Sleep mode is entered by executing SLEEP instruction when LPM bit of LPCR is 0. In sleep mode, clock supply to CPU core is disabled until reset or interrupt occurs. Clocks of OCS bus and OCP bus run continuously in sleep mode.

- Standby mode

Standby mode is entered by executing SLEEP instruction when LPM bit of LPCR is 1. In standby mode, all clocks except RTC clock are disabled. PLL is disabled also. Standby mode is canceled by reset or interrupt. When standby mode is canceled, PLL is restarted and all clocks start operating after approximate 1msec.

- Module-stop function

Module-stop function is used to stop specified on-chip module when it is not used. Set specified MSTP0~5 bits in LPCR will enter specified module-stop function. Clock supplies to these modules are stopped. Module-stop function is canceled by clearing specified MSTP0~5 to 0.

Following table shows the status of different block of the processor in low-power mode and conditions of entering and exiting low-power mode.

Table 9-1 Low-Power Modes Status

	Normal	Module Stop	Doze	Sleep	Standby
CPU, CC, MMU	Running	Running	Running ^{*3}	Halted	Halted
On-chip devices	Running	Specified module halted	Running ^{*1}	Running ^{*1}	Halted ^{*2}
DMA	Running	Halted if MSTP3 bit in LPCR is 1.	Running	Running	Halted
External memory	Refreshing	Refreshing	Refreshing	Refreshing	Self-Refreshing
PLL	Running	Running	Running	Running	Halted
Entering conditions		Set the specified MSTP bit in LPCR to 1.	Set DOZE bit in LPCR to 1	SLEEP instruction is executed when LPM bit in LPCR is 0.	SLEEP instruction is executed when LPM bit in LPCR is 1.
Exiting methods		Clear the specified MSTP bit in LPCR to 0. reset	Clear DOZE bit in LPCR to 0. Interrupt	Interrupt reset	Interrupt reset

Note:

- *1: Specified modules, whose MSTP bit in LPCR is set to 1, are halted.
- *2: The TMU channels, which select RTCCLK as count clock, are still running.
- *3: Clock is bursted to CPU core and the clock duty is set by DUTY field of LPCR.

9.2 Register Description

Following table shows the registers of Power/Reset Management Controller. System programmer can control the power consumption of the system by access Low-Power Control register. Get the reset information from Reset Status register.

Table 9-2 Power/Reset Management Controller Registers Configuration

Name	Full Name	R/W	Initial Value	Address	Access Size
LPCR	Low-Power Control Register	R/W	H'001F001E	H'E0000104	32
RSTR	Reset Status Register	R/W	Undefined	H'E0000108	8

9.2.1 Low-Power Control Register

The Low-Power Control Register (LPCR) is a 32-bit read/write register that controls low-power mode status. It is initialized to H'001F001E only by power-on reset. Only word access can be operated on LPCR.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	

Bit:	23	22	21	20	19	18	17	16
Read:	DOZE							DUTY
Write:								
Reset:	0	0	0					B'11111

Bit:	15	14	13	12	11	10	9	8
Read:	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:				CKO3	CKO2	CKO1	CKO0	LPM
Write:				EN	EN	EN	EN	
Reset:	0	0	0	1	1	1	1	0

– **Bit 31~24, 22~21, 7~5:** Reserved bits. Writes to these bits have no effect and always read as 0.

- **Bit 0 LPM:** Low Power Mode. Specifies which low-power mode will be entered when SLEEP instruction is executed.

Bit 0: LPM	Description	
0	Sleep mode will be entered when SLEEP instruction is executed.	(Initial value)
1	Standby mode will be entered when SLEEP instruction is executed.	

- **Bit 1 CKO0EN:** Clock Output 0 Enable. Controls the output of CKO0

Bit 1: CKO0EN	Description	
0	Disable CKO0 output, CKO0 is Hi-z state	
1	Enable CKO0 output	(Initial value)

- **Bit 2 CKO1EN:** Clock Output 1 Enable. Controls the output of CKO1

Bit 2: CKO1EN	Description	
0	Disable CKO1 output, CKO1 is Hi-z state	
1	Enable CKO1 output	(Initial value)

- **Bit 3 CKO2EN:** Clock Output 2 Enable. Controls the output of CKO2

Bit 3: CKO2EN	Description	
0	Disable CKO2 output, CKO2 is Hi-z state	
1	Enable CKO2 output	(Initial value)

- **Bit 4 CKO3EN:** Clock Output 3 Enable. Controls the output of CKO3

Bit 4: CKO3EN	Description	
0	Disable CKO3 output, CKO3 is Hi-z state	
1	Enable CKO3 output	(Initial value)

- **Bit 8 MSTP0:** Module Stop 0. Control the clock supply to UART. Clock supply to UART is stopped when this bit is set to 1. UART Registers can not be accessed when clock is stopped.

Bit 8: MSTP0	Description	
0	UART running	(Initial value)
1	Clock supply to UART is stopped	

- **Bit 9 MSTP1:** Module Stop 1. Control the clock supply to TMU. PCLK supply to TMU is stopped when this bit is set to 1. TMU Registers can not be accessed when clock is stopped.

Bit 9: MSTP1	Description	
0	TMU running	(Initial value)
1	Clock supply to TMU is stopped	

- **Bit 10 MSTP2:** Module Stop 2.
- **Bit 11 MSTP3:** Module Stop 3. Control the clock supply to DMA. Clock supply to DMA is stopped when this bit is set to 1. DMAC Registers can not be accessed when clock is stopped.

Bit 11: MSTP3	Description	
0	DMAC running	(Initial value)
1	Clock supply to DMAC is stopped	

- **Bit 12 MSTP4:** Module Stop 4. Control the clock supply to WDT. Clock supply to WDT is stopped when this bit is set to 1. WDT Registers can not be accessed when clock is stopped.

Bit 12: MSTP4	Description	
0	WDT running	(Initial value)
1	Clock supply to WDT is stopped	

- **Bit 13 MSTP5:** Module Stop 5. Control the clock supply to I2CI. Clock supply to I2CI is stopped when this bit is set to 1. I2CI Registers can not be accessed when clock is stopped.

Bit 13: MSTP5	Description	
0	I2CI running	(Initial value)
1	Clock supply to I2CI is stopped	

- **Bit 14 MSTP6:** Module Stop 6. Control the clock supply to UART2. Clock supply to UART2 is stopped when this bit is set to 1. UART2 Registers can not be accessed when clock is stopped.

Bit 14: MSTP6	Description	
0	UART2 running	(Initial value)
1	Clock supply to UART2 is stopped	

- **Bit 15 MSTP7:** Module Stop 7. Control the clock supply to AC97. Clock supply to AC97 is stopped when this bit is set to 1. AC97 Registers can not be accessed when clock is stopped.

Bit 15: MSTP7	Description	
0	AC97 running	(Initial value)
1	Clock supply to AC97 is stopped	

- **Bit 16~20 DUTY:** CPU Clock Duty. Control the CPU clock duty in doze mode. While this bits is set to 1 and doze mode is enabled, the clock is bursted to the CPU at a duty-cycle of 1/31. When the DUTY field is 1F(hex), the clock is always on and when it is zero, the clock is always off. Set the DUTY field to 0 when the CPU will be disabled for an extended amount of time. You can immediately wake it up again without waiting for the PLL to re-lock.

This field is not affected by the DOZE bit. When an interrupt disables the doze, these bits are not changed.

00000 = 0/31 duty-cycle

00001 = 1/31 duty-cycle

00010 = 2/31 duty-cycle

.

.

.

11111 = 31/31 duty-cycle

- **Bit 23 DOZE:** Doze Mode. Control the doze mode. When doze mode is canceled, this bit is cleared to 0 automatically.

Bit 23: DOZE	Description	
0	Doze mode is off	(Initial value)
1	Doze mode is on	

9.2.2 Reset Status Register (RSTR)

The Reset Status Register (RSTR) is an 8-bit read/write register which records last cause of reset. Each RSTS bit is set by a different source of reset. Please refer to Reset Sequence Control for reset sources description.

Bit:	7	6	5	4	3	2	1	0
Read:				PCIR	WMR	WPR	EMR	EPR
Write								
Reset:	0	0	0	—	—	—	—	—

- **Bit 5~7:** Reserved bits. Writes to these bits have no effect and always read as 0.
- **Bit 0:** External Power-on Reset (EPR). When an external power-on reset via RESETP pin is detected, EPR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored.

Bit 0: EPR	Description	
0	External power-on reset has not occurred since the last time the software clears this bit.	
1	External power-on reset has occurred since the last time the software clears this bit.	

- **Bit 1:** External Manual Reset (EMR). When an external manual reset via RESETM is detected, EMR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored.

Bit 1: EMR	Description	
0	External manual reset has not occurred since the last time the software clears this bit.	
1	External manual reset has occurred since the last time the software clears this bit.	

- **Bit 2:** WDT Power-on Reset (WPR). When a WDT power-on reset is detected, WPR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored.

Bit 2: WPR	Description	
0	WDT power-on reset has not occurred since the last time the software clears this bit.	
1	WDT power-on reset has occurred since the last time the software clears this bit.	

- **Bit 3:** WDT Manual Reset (WMR). When a WDT manual reset is detected, WMR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored.

Bit 3: WMR	Description	
0	WDT manual reset has not occurred since the last time the software clears this bit.	
1	WDT manual reset has occurred since the last time the software clears this bit.	

- **Bit 4:** PCI Input Reset (PCIR). When Arca runs in satellite mode and a PCI input reset is detected, PCIR is set and remains set until software clears it or another reset occurs. When Arca runs in host mode, this bit is fixed at 0. This bit can only be written with 0. Write with 1 will be ignored.

Bit 4: PCIR	Description	
0	PCI input reset has not occurred since the last time the software clears this bit.	
1	PCI input reset has occurred since the last time the software clears this bit.	

9.3 Doze Mode

9.3.1 Entering Doze Mode

Firstly, software should set the DUTY bits of LPCR. Then set DOZE bit of LPCR to 1 to enter doze mode. When slot controller of PMC indicates that the CPU clock's time-slot has expired, CPU bus grant will be disabled and then clock is stopped. Other on-chip devices are still running. During doze mode, program can modify clock duty-cycle according to core resource requirement. Clock control is in increments of approximately 3% (1/31).

Following figure shows how doze mode operates. In this example, the clock bursts at about a 30% duty-cycle, so CPU is active about 30% of the time. The remainder of the time, CPU is stopped. The burst period is 31 RTC clock cycles.

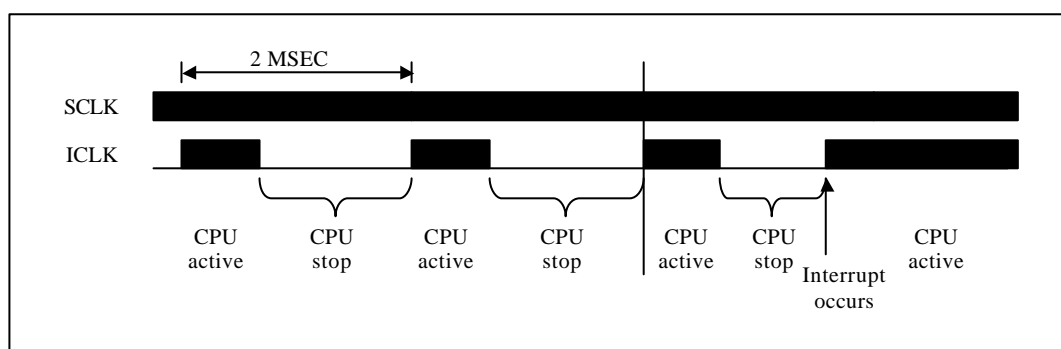


Figure 9-1 Doze Mode Operation

9.3.2 Exiting Doze Mode

If an interrupt occurs, DOZE bit of LPCR is cleared to 0 automatically, and clock supply and CPU bus grant are enabled immediately. Software can clear DOZE to 0 and exit doze mode also.

Doze mode can also be canceled by power-on or manual reset.

9.4 Sleep Mode

9.4.1 Entering Sleep Mode

In normal mode, when LPM bit in LPCR is 0 and SLEEP instruction is executed, the processor enters sleep mode. CPU is halted but its register contents are retained. All on-chip devices continue to run unless they are in module stop function. Clock continues to be output from CKO pin.

The procedure of entering sleep mode is shown blow:

1. Set LPM bit in LPCR to 0.
2. Executes SLEEP instruction.
3. When current access on OCS Bus complete, the arbiter stop granting any request. When OCS Bus is idle state, iclk is stopped.

9.4.2 Exiting Sleep Mode

Sleep mode is exited by an interrupt (FIQ, IRQ or on-chip devices) or a reset.

Exit by Interrupt: When a FIQ, IRQ or on-chip devices interrupt is detected, clock supply is re-enabled to entire chip. The chip start to run and the interrupt handling is then executed.

Following figure illustrates the sequence of exit from sleep mode by interrupt.

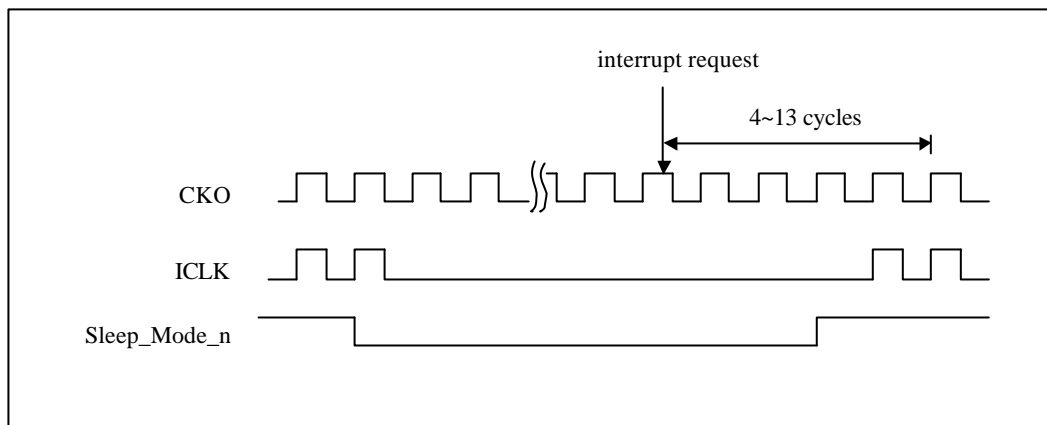


Figure 9-2 Sleep Mode Cancelled By Interrupt Sequence

Exit by Reset: When a power-on reset via RESETP pin or manual reset via RESETM pin, or power-on reset or manual reset generated by WDT, Sleep mode is exited.

Following figure illustrates the sequence of exit from sleep mode by manual reset.

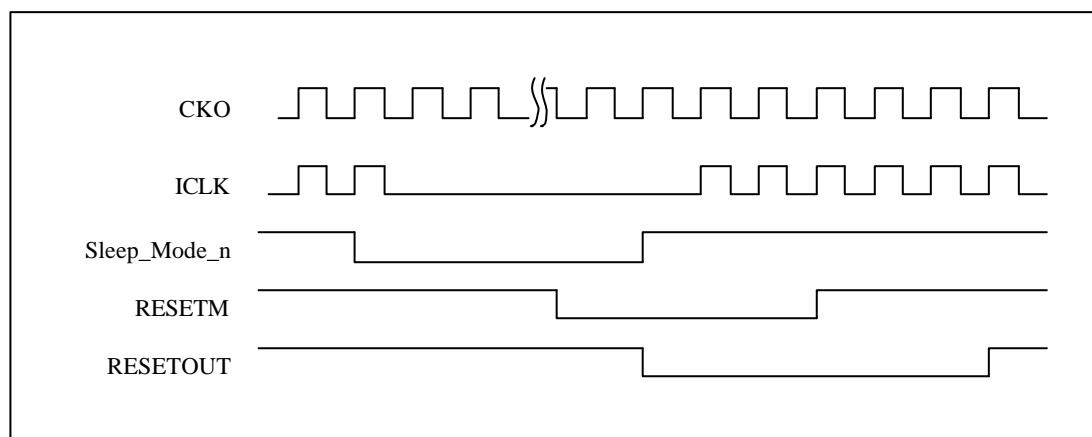


Figure 9-3 Sleep Mode Cancelled By Manual Reset

Following figure illustrates the sequence of exit from sleep mode by power-on reset.

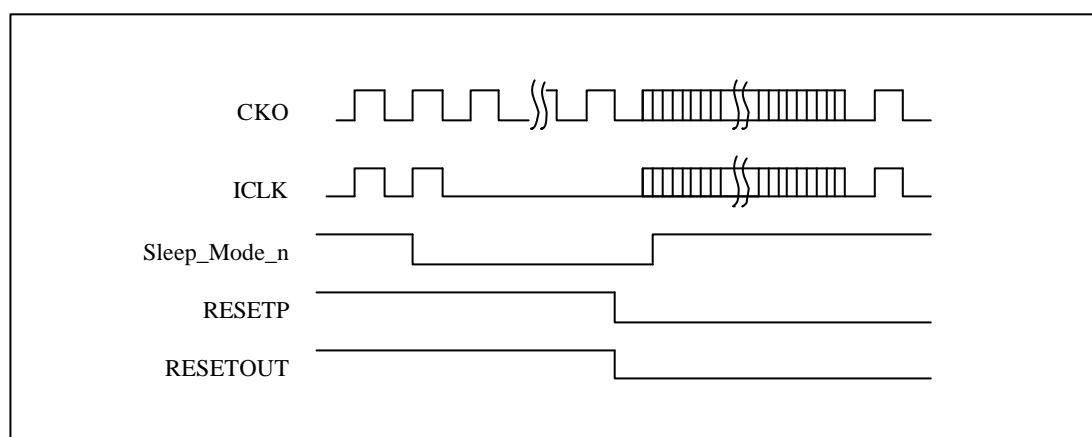


Figure 9-4 Sleep Cancelled By Power-on Reset

9.5 Standby Mode

9.5.1 Entering Standby Mode

In normal mode, when LPM bit in LPCR is 1 and SLEEP instruction is executed, the processor enter standby mode. On-chip devices and CPU are halted. PLL is shut off. Clock output from CKO pin is also stopped. CPU registers and on-chip devices registers contents are retained.

Only rtcclk is operating in standby mode.

The procedure of entering standby mode is shown blow:

1. Set LPM bit in LPCR to 1.
2. Execute a SLEEP instruction.
3. When current access on OCS BUS complete, the arbiter will not grant any following request. EMI will drive SDRAM from auto-refresh mode to self-refresh mode.
4. When OCS BUS is idle state and SDRAM is self-refresh mode, iclk, bclk and dclk supplies are stopped. If PLLEN=1, PLL is shut off.

Note: DMA transfer should be terminated before entering standby mode. Transfer results are not guaranteed if standby mode is entered during transfer.

9.5.2 Exiting Standby Mode

Standby mode can be exited by an interrupt (FIQ, IRQ or on-chip devices) or a power-on reset via the RESETP pin or a manual reset via the RESETM pin.

Exit by Interrupt (PLLLEN=1): When a FIQ, IRQ or on-chip device interrupt is detected, the on-chip PLL starts to lock phase. After on-chip PLL stabilize, then clock supplies are re-enabled. The chip start to run and the interrupt handling is then executed. Clock output is always enabled throughout the whole process. The output is unstable during on-chip PLL stabilization time.

Following figure illustrates the sequence of exit from standby mode by interrupt.

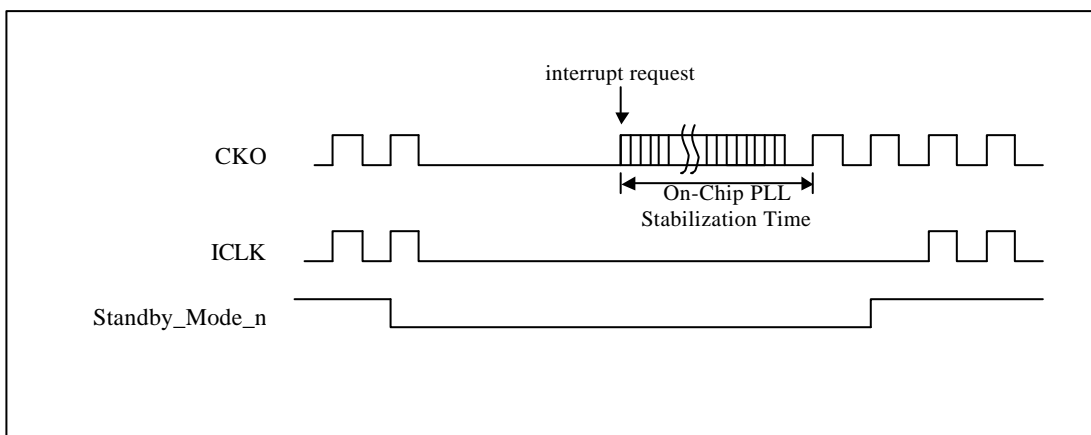


Figure 9-5 Standby Mode Cancelled by Interrupt (PLLEN=1)

Exit by Reset (clock mode 0, 1, 2 and 3, PLLEN=1): Standby mode can be exited by power-on reset via RESETP pin or manual reset via RESETM pin. Clock supply and clock output is re-enabled immediately after reset signal is detected. RESETP and RESETM pin should be held until clock stabilizes. Unstable clock is output on CKO pin during PLL lock phase.

Following figure illustrates the sequence of exit from standby mode by power-on reset.

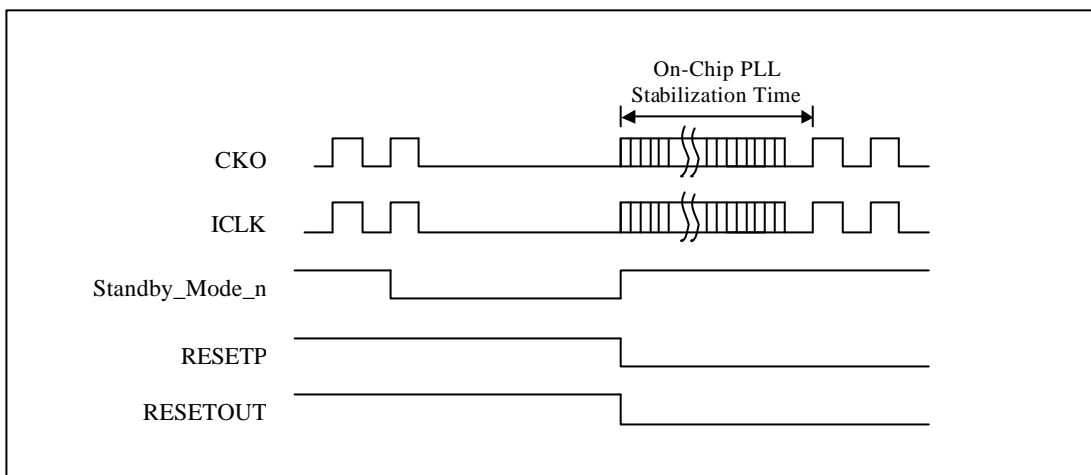


Figure 9-6 Standby Mode Cancelled by Power-on Reset (clock mode 0, 1, 2 and 3, PLLEN=1)

Following figure illustrates the sequence of exit from standby mode by manual reset.

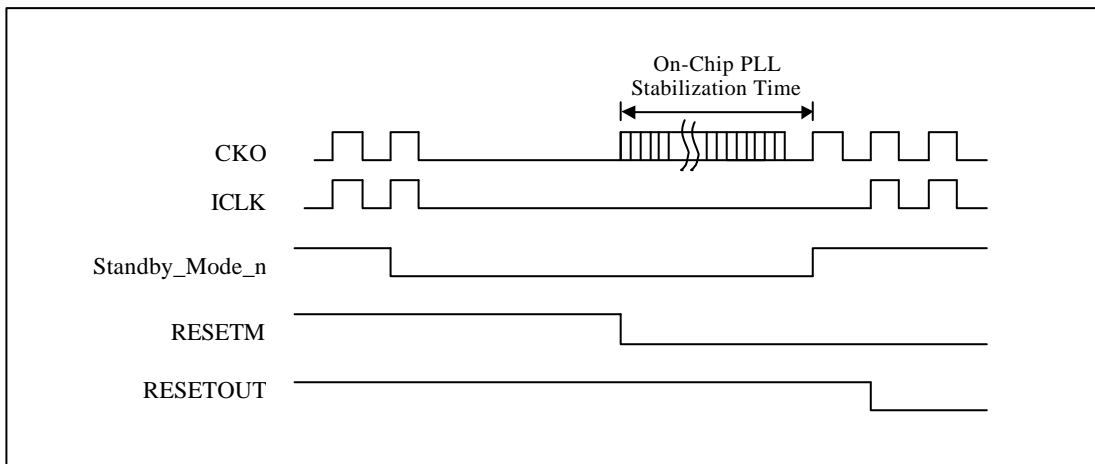


Figure 9-7 Standby Mode Cancelled by Manual Reset (clock mode 0, 1, 2 and 3, PLLN=1)

9.6 Module Stop Function

9.6.1 Entering Module Stop Function

Set the specified MSTP0~5 bits in LPCR to 1 will enter specified module stop function. Clock supplies to these modules are stopped. Power consumption can be reduced by using module stop function in running mode. In sleep mode, using module stop function can further reduce power consumption.

In module stop function, the halted modules retain its register contents.

Bit in LPCR		Description
MSTP0	0	UART running
	1	Clock supply to UART is stopped.
MSTP1	0	TMU running
	1	Clock supply to TMU is stopped ^{*1} .
MSTP3	0	DMA running
	1	Clock supply to DMA is stopped
MSTP4	0	WDT running
	1	Clock supply to WDT is stopped
MSTP5	0	I2CI running
	1	Clock supply to I2CI is stopped
MSTP6	0	UART2 running
	1	Clock supply to UART2 is stopped
MSTP7	0	AC97 running
	1	Clock supply to AC97 is stopped

Note: *1: If rtclk is selected as count clock, the counter is still running.

9.6.2 Exiting Module Stop Function

Module stop function can be exited by clearing specified MSTP bit in LPCR to 0, or a power-on reset via RESETP pin, or a manual reset via RESETM pin, or a reset generated by WDT. Clock supply is then re-enabled.

9.7 Reset Control

9.7.1 Reset Sources and Pins

The Arca has five reset sources and four reset-related pins. These five resets are classed to two types: power-on reset and manual reset. On reset, the core and all other on-chip modules are programmed to predetermined states.

- External power-on reset via RESETP pin

When RESETP is detected active, an internal power-on reset sequence is started immediately. This reset is propagated to the RESETOUT pin. When the Arca runs in host mode, this reset is propagated to the PCI_RESET pin also.

The RESETP pin is low active. This input should be hold until on-chip PLL stabilization.

- External manual reset via RESETM pin

When RESETM is detected active, an internal manual reset sequence is started after current transaction on OCS BUS has finished. The OCS BUS arbiter will not respond any following request until the manual reset sequence end. This reset is propagated to the RESETOUT pin. When the Arca runs in host mode, this reset is propagated to the PCI_RESET pin also.

The RESETM pin is low active. This pin should be hold until on-chip PLL stabilization.

- WDT power-on reset

When WDT counter overflow and RSTS bit in WTCSR is 0, an internal power-on reset sequence is started immediately. This reset is propagated to the RESETOUT pin but not propagated to PCI_RESET pin.

- WDT manual reset

When WDT counter overflow and RSTS bit in WTCSR is 1, an internal manual reset sequence is start after current transaction on OCS BUS has finished. The OCS BUS arbiter will not respond any following request until the manual reset sequence end. This reset is propagated to the RESETOUT pin but not propagated to the PCI_RESET pin.

- PCI reset

When Arca runs in satellite mode, the PCI_RESET pin is used as an input pin and acts as a power-on reset. An internal power-on sequence is started immediately after the PCI_RESET pin is detected active. This reset is propagated to the RESETOUT pin. The PCI_RESET should hold until on-chip PLL stabilization.

When Arca runs is host mode, the PCI_RESET pin is used as an output pin. The PCI_RESET pin is asserted during external power-on reset and external manual reset. The PCI_RESET is low active.

- Reset output

The RESETOUT pin is a low active pin used to reset external devices. It is asserted during external power-on/manual reset, WDT power-on/manual reset or PCI reset.

9.7.2 Reset Sequence

The internal reset sequence has the same timing as RESETOUT. In host mode, the PCI_RESET has the same timing as RESETOUT also.

Power-On Reset via RESETP Pin

Internal reset is asserted at same time of RESETP pin assertion. The on-chip PLL start to lock phase. The RESETP pin should keep long enough to ensure the on-chip PLL stabilization time. Internal reset will keep assertion 100 ms after RESETP deassertion.

Power-On Reset via PCI_RESET Pin

Internal reset is asserted at same time of PCI_RESET pin assertion. The on-chip PLL start to lock phase. The PCI_RESET pin should keep long enough to ensure the on-chip PLL stabilization time. Internal reset will keep assertion 100 ms after PCI_RESET deassertion.

WDT Power-On Reset

Internal reset is asserted when WDT overflow and RSTS bit in WTCSR is 0. The on-chip PLL start to lock phase. An internal counter is used to ensure the PLL stabilization time. Internal reset will be asserted for 100 ms.

Following figure illustrates the sequence of power-on reset via RESETP pin.

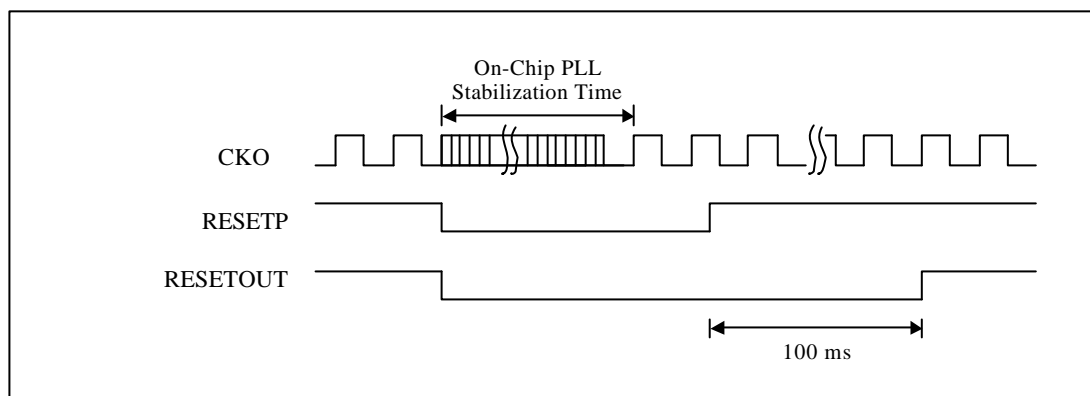


Figure 9-8 Power-On Reset via RESETP Pin (clock mode 0, 1, 2, 3)

In clock mode 4, the on-chip PLL is bypassed and the external clock input is used directly. No PLL stabilization time is needed.

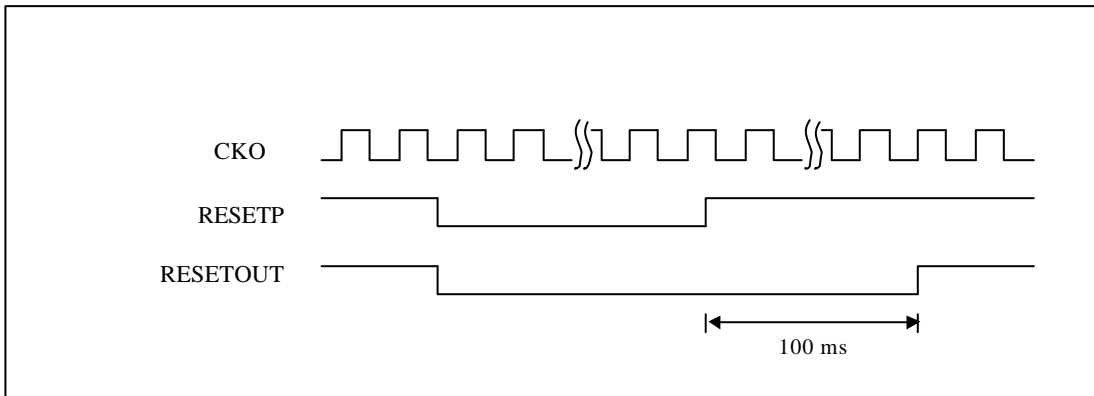


Figure 9-9 Power-On Reset via RESETP Pin (clock mode 4)

Manual Reset via RESETM pin

When RESETM is asserted, CGU will send stop request to OCS Bus arbiter. The OCS Bus arbiter will not grant any following request after complete current access. When SBUS is idle, internal reset is asserted. Internal reset will keep assertion for 3 to 4 rtccclk cycles after RESETM deassertion.

Following figure illustrates the timing of external manual reset.

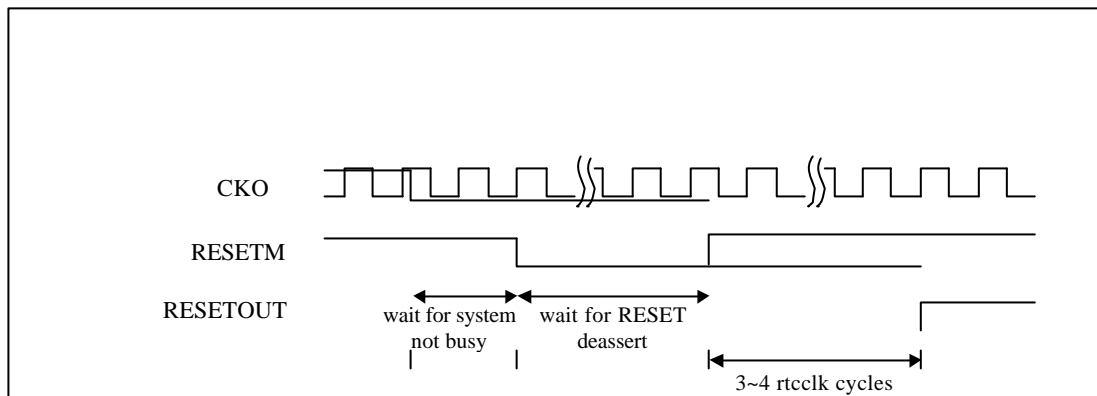


Figure 9-10 Manual Reset via RESETM Pin

When manual reset occurs in standby mode, the manual reset signal should ensure the on-chip PLL stabilization time. Please refer to section 9.5 for detail.

WDT Manual Reset

When WDT overflow and RSTS bit in WTCSR is 1, CGU will send stop request to OCS Bus arbiter. The OCS Bus arbiter will not grant any following request after complete current access. When SBUS is idle, internal reset will be asserted for 3 to 4 rtcclk cycles.

Following figure illustrates the timing of WDT manual reset.

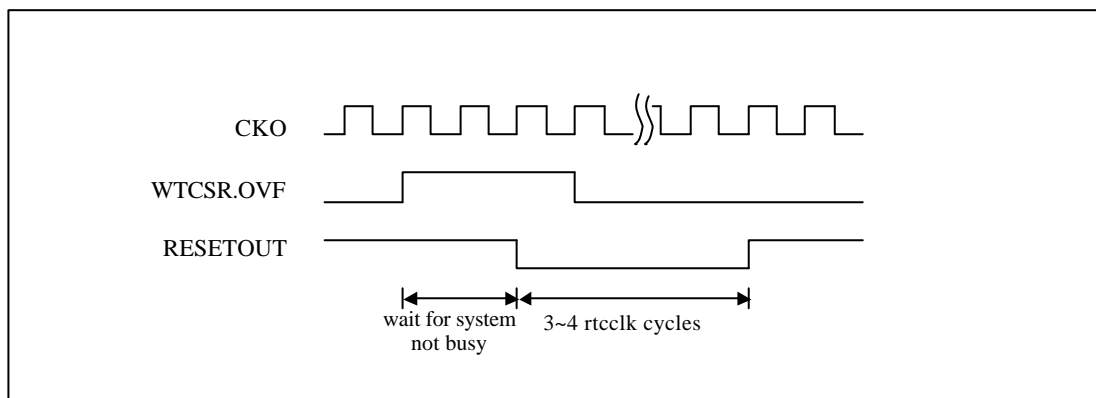


Figure 9-11 WDT Manual Reset

10 Interrupt Controller (INTC)

10.1 Overview

The interrupt controller uses a bit position for each different interrupt source. All the interrupt sources can send interrupt request to CPU via INTC. CPU responds the interrupt request, breaks current instruction executing, and jumps to the interrupt service routine. By this means, devices can get service by CPU in an interrupted manner.

10.1.1 Features

INTC features:

- Total 14 interrupt sources
- Individual masking of sources.
- Interrupt source register and interrupt pending register give user sufficient information.
- All interrupt source inputs are level sensitive and active LOW (External edge interrupts are translated into level interrupts by GPIO module).
- All the registers are accessed from OCP-BUS.
- Unmasked interrupts can wake up the chip in sleep mode or in standby mode.

10.1.2 Block Diagram

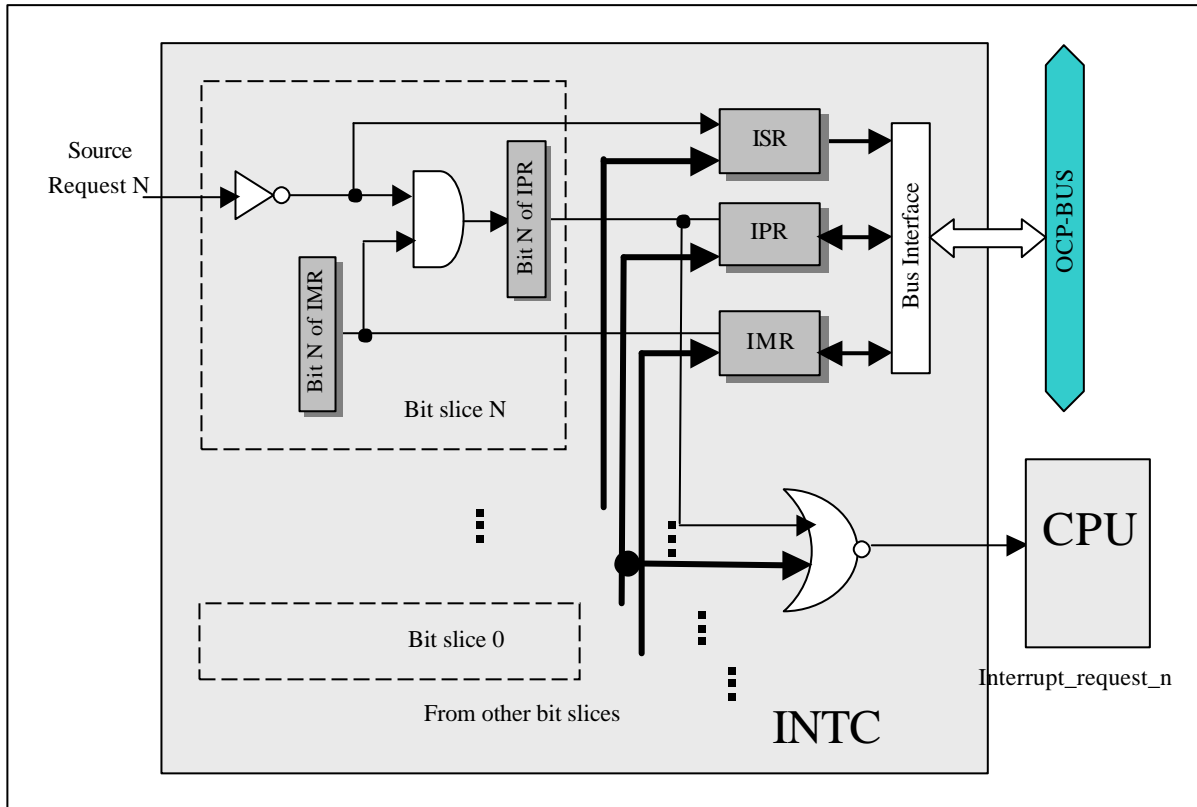


Figure 10-1 INTC Block Diagram

Note:

ISR: Interrupt source register
 IPR: Interrupt pending register
 IMR: Interrupt mask register

Bit slice N generates bit N of ISR, IPR, IMR registers.

10.2 Pin configuration

Table 10-1 INTC Pins

Name	Source	Destination	Description
IRQ0/GPIO0	External Devices	INTC	External interrupt request0
IRQ1/GPIO1	External Devices	INTC	External interrupt request1
IRQ2/GPIO2	External Devices	INTC	External interrupt request2
IRQ3/GPIO3	External Devices	INTC	External interrupt request3
IRQ4/GPIO4	External Devices	INTC	External interrupt request4
IRQ5/GPIO5	External Devices	INTC	External interrupt request5
IRQ6/GPIO6	External Devices	INTC	External interrupt request6
IRQ7/GPIO7	External Devices	INTC	External interrupt request7

Note: All 8 IRQ pins are multiplexed as general purpose inputs/outputs in GPIO port B.

10.3 Registers Configuration

Table 10-2 lists the registers of Interrupt Controller. All of these registers are 32bit, and each bit of the register represents or controls one interrupt source that list in Table 10-3.

Table 10-2 INTC Register

Name	Full Name	R/W	Initial Value	Address	Access Size
ISR	Interrupt Source Register	R	H'00000000	H' E0000000	32
IMR	Interrupt Mask Register	R/W	H'00000000	H' E0000004	32
IMSR	Interrupt Mask Set Register	W	Undefined	H' E0000008	32
IMCR	Interrupt Mask Clear Register	W	Undefined	H' E000000C	32
IPR	Interrupt Pending Register	R/W	H'00000000	H' E0000010	32

Note: Initial value is the register value after power on reset or manual reset.

10.3.1 Interrupt Source Register (ISR)

This register contains all the interrupts' status from GPIO port B. A "1" indicates that the corresponding interrupt is active (at LOW level) now. A "0" indicates that the interrupt is not active (at HIGH level) now. The register is read only.

Bit of ISR	Description	
0	The corresponding interrupt source is not active.	(Initial value)
1	The corresponding interrupt source is active.	

10.3.2 Interrupt Mask Register (IMR)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. Its value can be changed either by writing IMSR and IMCR or by writing itself.

Bit of IMR	Description	
0	The corresponding interrupt is not masked.	(Initial value)
1	The corresponding interrupt is masked.	

10.3.3 Interrupt Mask Set Register (IMSR)

This register is used to set bits in the interrupt mask register. This register is write only.

Bit of IMSR	Description	
0	Ignore	(Initial value)
1	Will set the corresponding interrupt mask bit	

10.3.4 Interrupt Mask Clear Register (IMCR)

This register is used to clear bits in the interrupt mask register. This register is write only.

Bit of IMCR	Description	
0	Ignore	(Initial value)
1	Will clear the corresponding interrupt mask bit	

10.3.5 Interrupt Pending Register (IPR)

This register contains the status of the interrupt sources after masking. Writing “1” to a bit can clear it. Writing “0” has no effect.

Bit of IPR	Description	
0	The corresponding interrupt is not active or is masked.	(Initial value)
1	The corresponding interrupt is active and is not masked to the processor.	

10.3.6 Bits Definition for IMR, IMSR, IMCR, ISR, IPR

Table 10-3 Interrupt Sources

Bit/Bits	Interrupt Source
Bit 31	---- Reserved
Bit 30	---- Reserved
Bit 29	---- IRQ0/GPIO0
Bit 28	---- IRQ1/GPIO1
Bit 27	---- IRQ2/GPIO2
Bit 26	---- IRQ3/GPIO3
Bit 25	---- Reserved
Bit 24	---- Reserved
Bit 23	---- TMU
Bit 22	---- RTC
Bit 21	---- DMA
Bit 20	---- PCI
Bit 19	---- ETHC (Ethernet Controller)
Bit 18	---- Reserved
Bit 17	---- IRQ4/GPIO4
Bit 16	---- IRQ5/GPIO5
Bit 15	---- IRQ6/GPIO6
Bit 14	---- IRQ7/GPIO7
Bit 13	---- UHC
Bit 12	---- Reserved
Bit 11	---- Reserved
Bit 10	---- Reserved
Bit 9	---- UART2/IrDA
Bit 8	---- UART
Bit 7	---- SARB
Bit 6	---- CPU-core
Bit 5	---- Reserved
Bit 4	---- Reserved

Bit 3	---- AC97
Bit 2	---- Reserved
Bit 1	---- I2CI
Bit 0	---- Reserved

Note: Reserved bits in IMR, IMSR and IMCR are normal bits to be written into and read out. Reserved bits in ISR and IPR are read-only and always 0.

10.4 INTC Operation

The interrupt controller is an interface representing the status of interrupt sources in the peripheral and the status of software masking.

If the prioritizing the responding order of concurrently unmasked interrupts is necessary, software should perform the task - determine the interrupt source from the list of pending status in IPR. In Arca210, pending interrupts have two levels in structure. Interrupting device that contains more than one interrupt sources need software to determine how to service it by reading interrupt status registers within it.

In the interrupt handler, the serviced interrupt source needs to be cleared in the interrupting device. In order to make certain the cleared source request status has been reflected at the corresponding IPR bit, software should wait enough time before executing "RTE" instruction.

External interrupts have their programmable detect manner choices in GPIO port B they connected.

The flowchart of procedure is shown in the following figure:

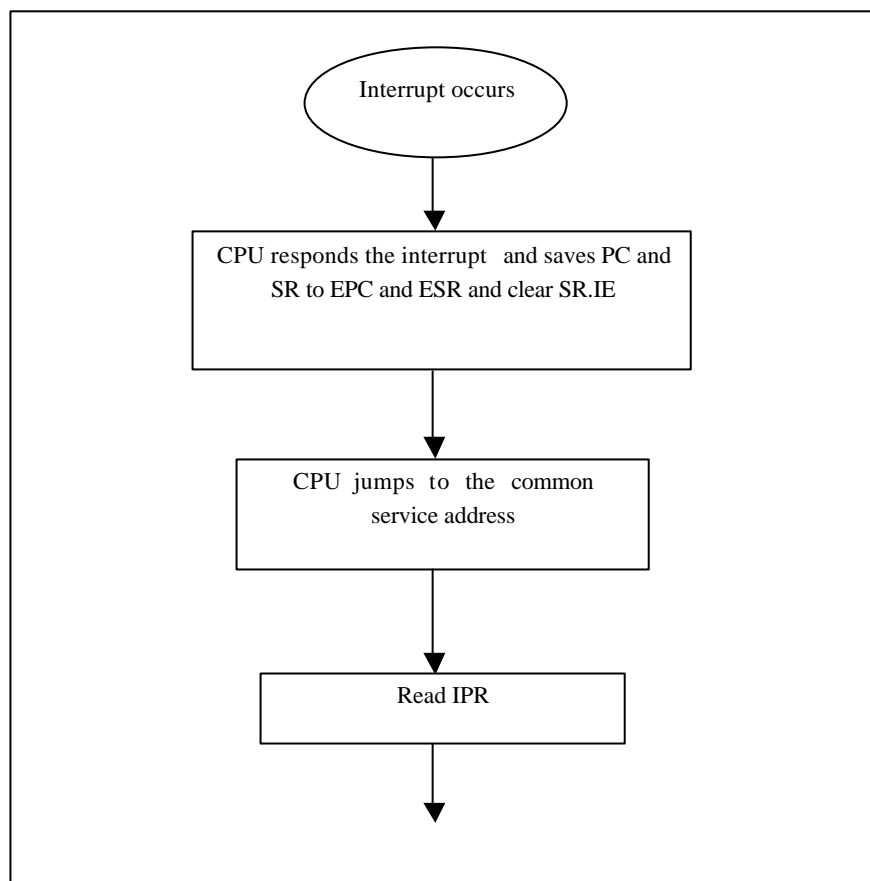


Figure 10-2 Flowchart

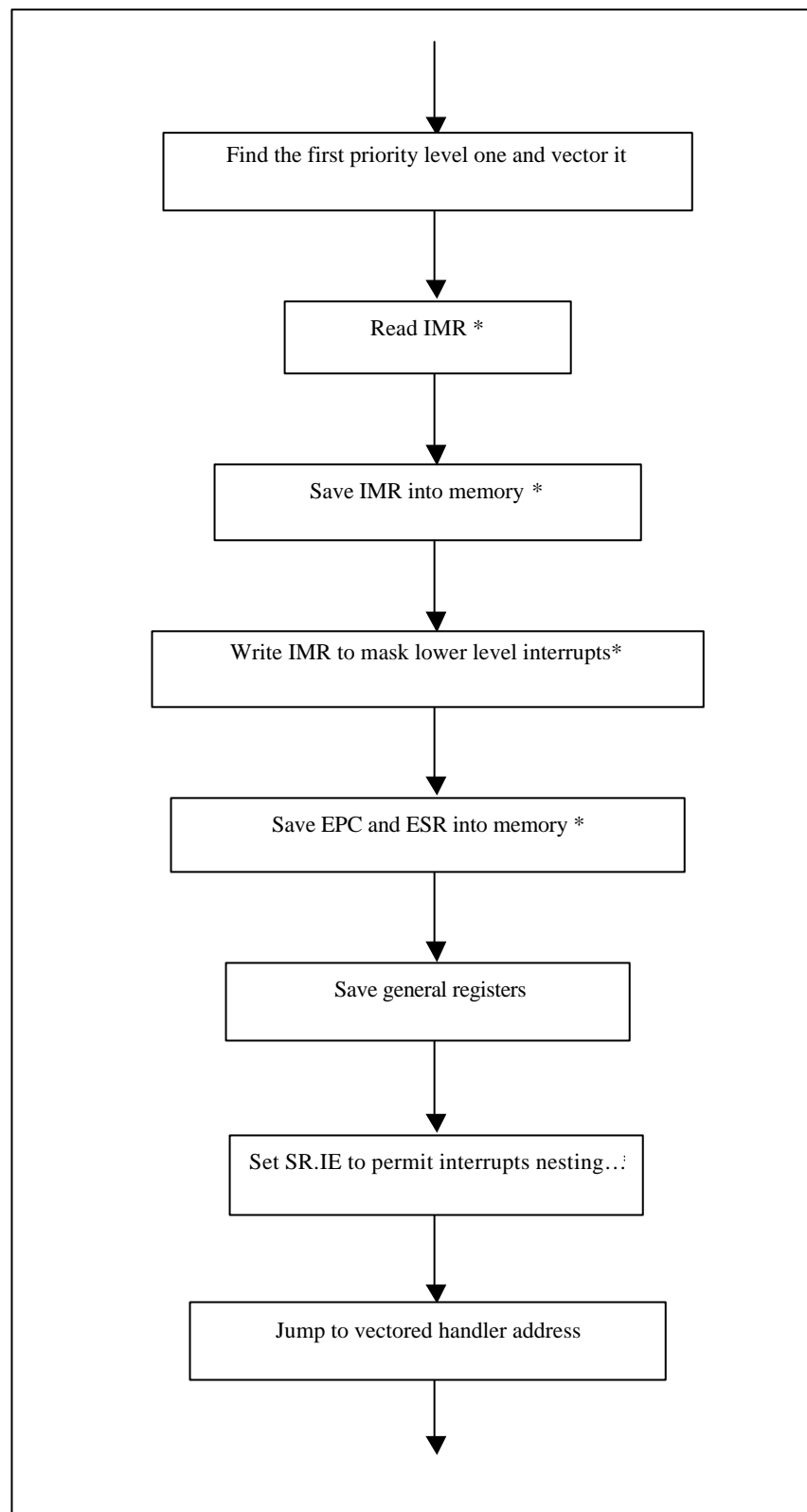
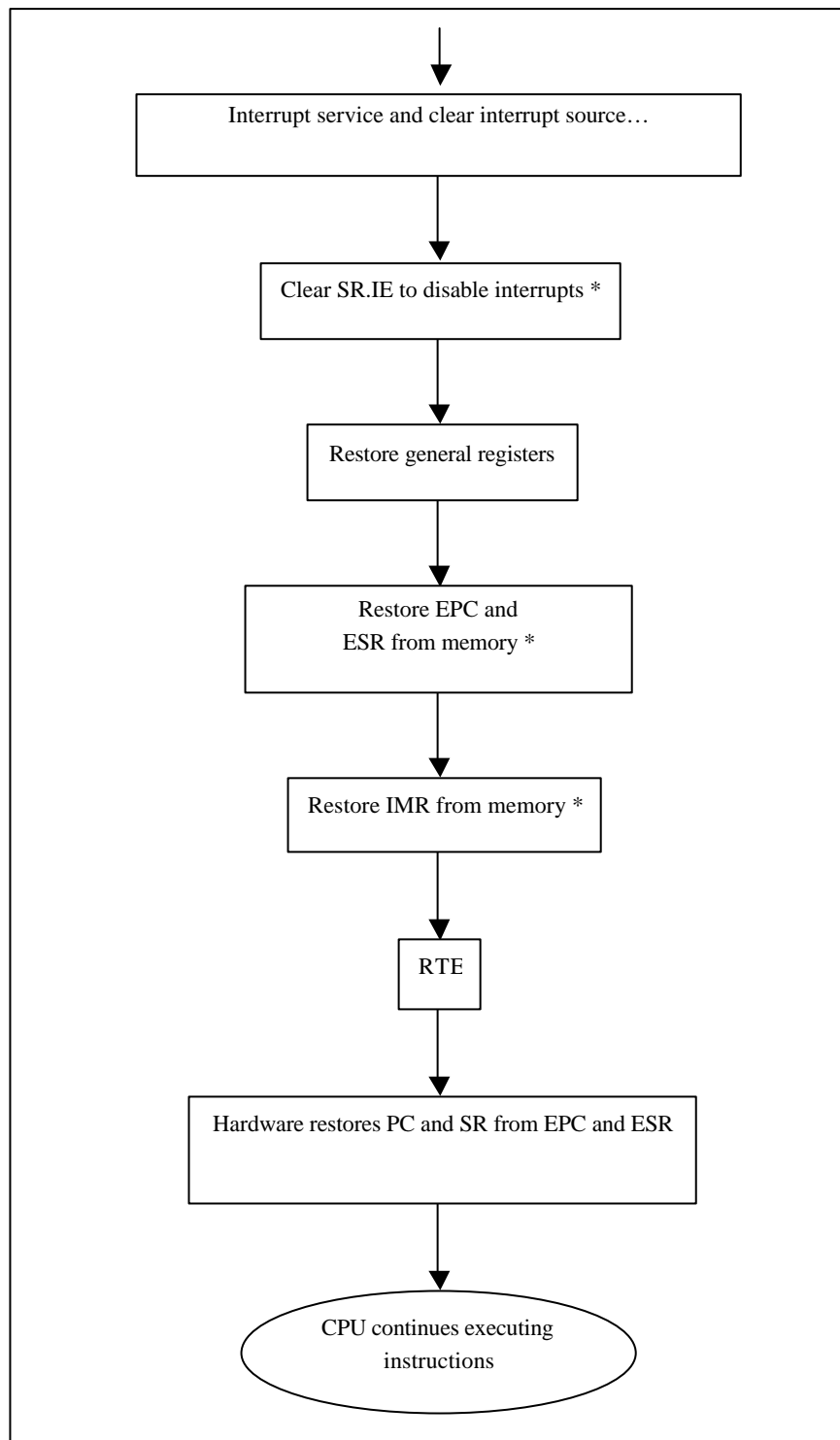


Figure 10-2 Flowchart (continue 1)

**Figure 10-2 Flowchart (continue 2)**

Note: These steps are necessary when interrupt nesting is enabled.

11 Timer Unit (TMU)

11.1 Overview

The timer unit (TMU) of Arca has three channels, and each channel can be programmed counting on different clock.

11.1.1 Features

Some of the key features are as the following:

- Three separate channels are provided.
- Each channel is comprised of a 32-bit down counter and a 32-bit constant register.
- Auto-reload function is provided for each channel.
- One interrupt is generated for all channels when the down counter underflows (H'00000000 → H'FFFFFFFF).
- The down counter working clock can be selected from 5 input clocks: RTCCLK (real time clock), $\phi/4$, $\phi/16$, $\phi/64$, $\phi/256$. (ϕ is the internal clock for on-chip peripheral)
- All channels can operate when the system is in standby mode: When the RTCCLK is used as count clock. The timer is still able to count in standby mode.
- Bus interface with On-chip Peripheral Bus.

11.1.2 Block Diagram

Following figure shows the block diagram of TMU.

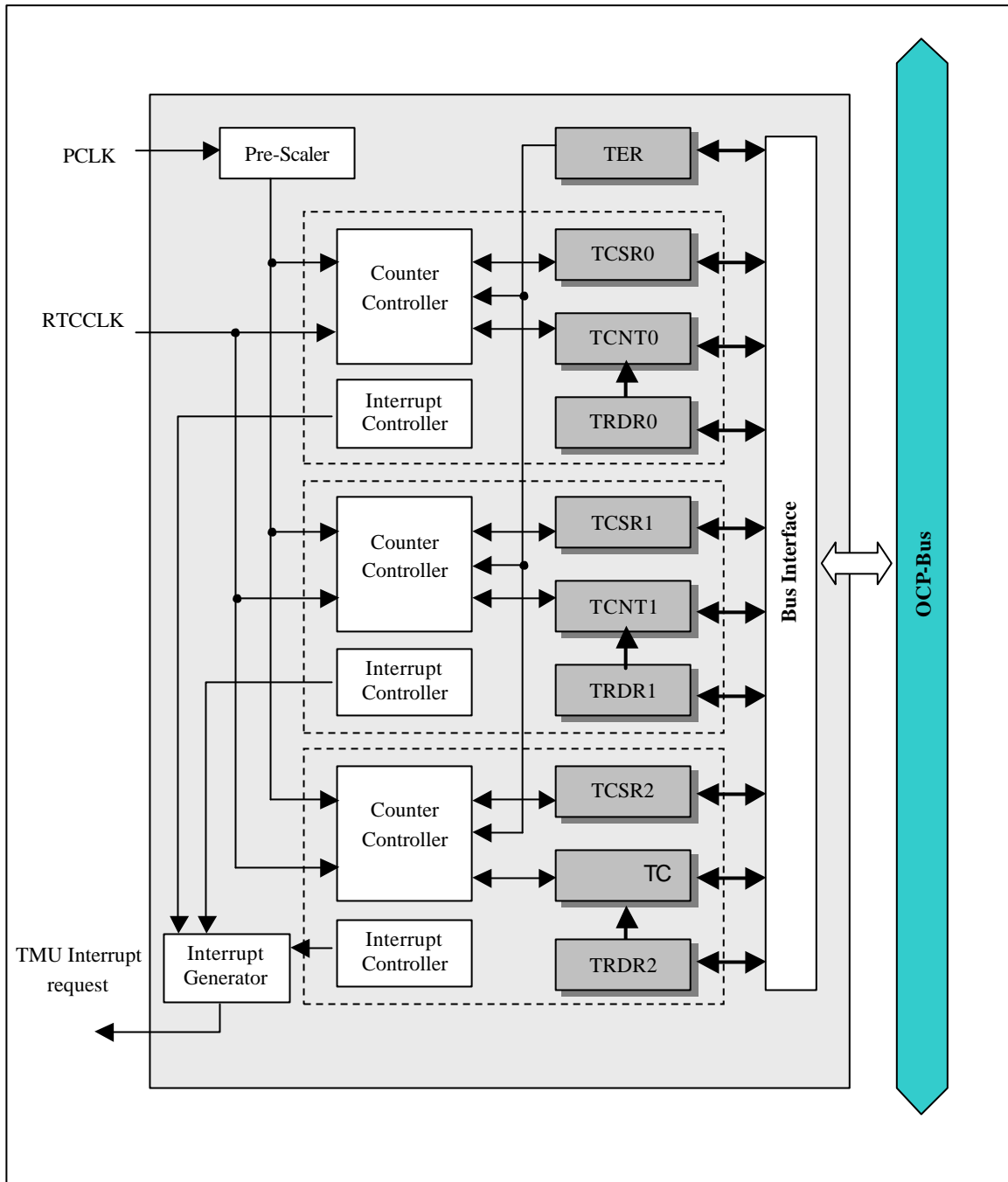


Figure 11-1 TMU Block Diagram

11.2 Register Configuration

In this section, we will describe the registers in TMU. Following table lists all the registers definition. And detailed function of each register will be described below.

Table 11-1 TMU Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
TER	Timer Enable Register	R/W	H'00	H'E0000200	8
TRDR0	Timer Reload Data Register 0	R/W	H'FFFFFFFF	H'E0000204	32
TCNT0	Timer Counter 0	R/W	H'FFFFFFFF	H'E0000208	32
TCSR0	Timer Control/Status Register 0	R/W	H'0000	H'E000020C	16
TRDR1	Timer Reload Data Register 1	R/W	H'FFFFFFFF	H'E0000210	32
TCNT1	Timer Counter 1	R/W	H'FFFFFFFF	H'E0000214	32
TCSR1	Timer Control/Status Register 1	R/W	H'0000	H'E0000218	16
TRDR2	Timer Reload Data Register 2	R/W	H'FFFFFFFF	H'E000021C	32
TCNT2	Timer Counter 2	R/W	H'FFFFFFFF	H'E0000220	32
TCSR2	Timer Control/Status Register 2	R/W	H'0000	H'E0000224	16

11.2.1 Timer Enable Register (TER)

TER is an 8-bit read/write register that selects whether to run or halt the timer counters (TCNT) for channels 0–2. TER is initialized to H'00 by a power-on reset or manual reset, and retains its content in standby mode.

Bit:	7	6	5	4	3	2	1	0
Read:						TE2	TE1	TE0
Write:								
Reset:	0	0	0	0	0	0	0	0

- **Bit 3-7:** Reserved bits. These bits always read 0. Write data to these bits are ignored.
- **TE0 :** Channel 0 Enable. Halt or run timer channel 0.

Bit 0: TE0	Description	
0	Halts channel 0 count.	(Initial value)
1	Starts channel 0 count.	

- **TE1:** Channel 1 Enable. Halt or run timer channel 1.

Bit 1: TE1	Description	
0	Halts channel 1 count.	(Initial value)
1	Starts channel 1 count.	

- **TE2:** Channel 2 Enable. Halt or run timer channel 2.

Bit 2: TE2	Description	
0	Halts channel 2 count.	(Initial value)
1	Starts channel 2 count.	

11.2.2 Timer Control/Status Register (TCSR)

The Timer Control/Status Register controls count clock selection and interrupt issuance when the counter underflow flag has been set to 1. Each channel has a control register and they have the same configuration. The Timer Control Register is initialize to H'0000 by a power-on reset and manual reset. They are not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:		UF	UIE			CKS2	CKS1	CKS0
Write:								
Reset:	0	0	0	0	0	0	0	0

- **Bit 3, 4, 10-15:** Reserved bits. These bits always read 0. Data written to these bits are ignored.
- **CKS0-CKS2:** Clock Select. These bits select the TCNT count clock.

Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	0	Internal clock: $\phi/4$	(Initial value)
0	0	1	Internal clock: $\phi/16$	
0	1	0	Internal clock: $\phi/64$	
0	1	1	Internal clock: $\phi/256$	
1	0	0	RTCCLK	
1	0	1	Reserved	
1	1	0	Reserved	
1	1	1	Reserved	

- **UIE:** Underflow Interrupt Enable. Controls interrupt issuance when underflow flag (UF) has been set to 1.

Bit 5: UIE	Description	
0	Underflow interrupt is disabled.	(Initial value)
1	Underflow interrupt is enabled.	

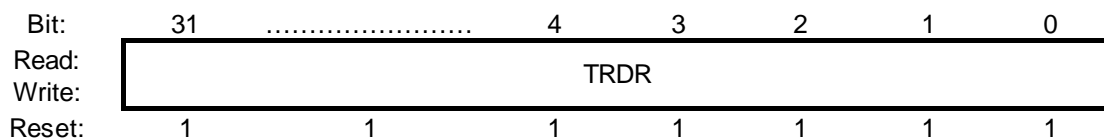
- **UF:** Underflow Flag. Indicates the status of TCNT underflow.

Bit 6: UF	Description	
0	TCNT underflow has not occurred. Clearing condition: When 0 is written to UF.	(Initial value)
1	TCNT underflow has occurred. Setting condition: when TCNT underflows *.	

Note: Contents do not change when 1 is written to UF.

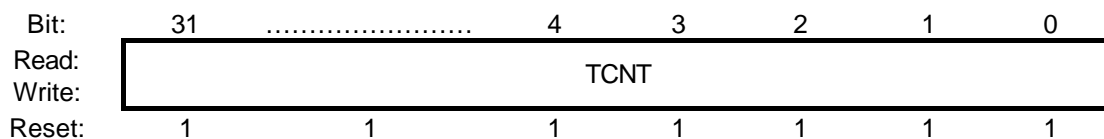
11.2.3 Timer Reload Data Register (TRDR)

Each channel has a Timer Reload Data Register (TRDR). They have the same configuration. TRDR is a 32-bit read/write register used to save the data for reload function. The counter (TCNT) starts from its initial value, and when it countdown results in an underflow, the value of TRDR is set in TCNT and continues decreasing from that value, this is called reload function. TRDR is initialized to H'FFFFFFF by a power-on reset or manual reset; it is not initialized in standby mode.



11.2.4 Timer Counter (TCNT)

Each channel has a Timer Counter (TCNT). They have the same configuration. TCNT is a 32-bit read/write register. When a TCNT countdown results in an underflow (H'00000000 → H'FFFFFFF), the underflow flag (UF) in the Timer Control/Status Register (TCSR) of the relevant channel is asserted. The TRDR value is simultaneously set in TCNT, and the countdown of TCNT continues from that value.



11.3 TMU Operation

11.3.1 Basic Functions

Counter Operation: set the enable bit in the Timer Enable Register (TER) to start the corresponding timer counter (TCNT) decreasing from its current value. When a TCNT underflows (H'00000000 → H'FFFFFFF), its corresponding UF flag of the Timer Control/Status Register (TCSR) is asserted, and if the UIE bit in TCSR is 1, an interrupt is sent to the INTC. At this time, TCNT reload the value TRDR as its new initial value, and continues down-counting from the value.

The count operation is set as follows:

1. Select the counter clock with the CKS2–CKS0 bits in the Timer Control/Status Register (TCSR). If the external clock is selected, select its edge with the CES1 and CES0 bits in TCSR.
2. Use the UIE bit in TCSR to set whether to generate an interrupt when TCNT underflows.
3. Set a value in the Timer Reload Data Register (TRDR) (the cycle is the set value plus 1).
4. Set the initial value in the timer counter (TCNT).
5. Set the TE bit in the Timer Enable Register (TER) to 1 to start operation.

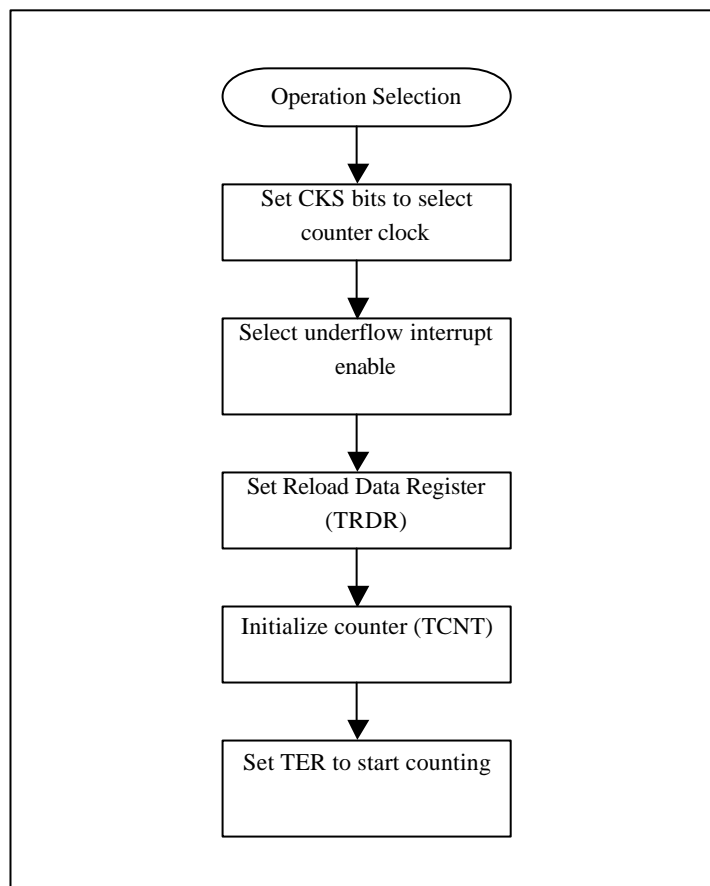


Figure 11-2 TMU Operation

11.3.2 TMU counter decrease and reload timing

Following figure illustrates the counter decrease and reload timing.

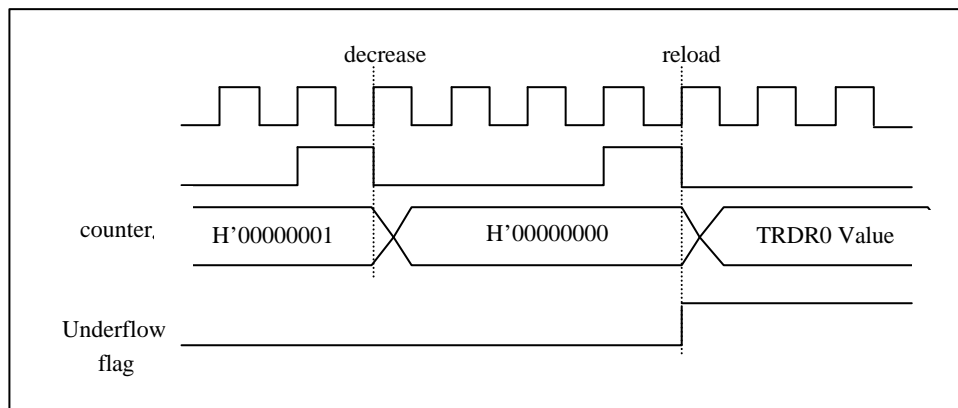


Figure 11-3 TMU Channel 0 Decrease and Reload Timing

11.4 Usage Notes

11.4.1 Interrupt handling

Only one interrupt request is used for all three channels. The interrupt is asserted whichever channel underflow and UIE is 1. When an interrupt has been generated, interrupt handler should read TCSR of all three channels and judge which channel assert the interrupt. Then clear the UF bit in TCSR. If interrupts are enabled without clearing the flag, another interrupt will be generated.

11.4.2 Standby mode

In standby mode, all registers and counters retain its content. Software can clear the start bits in TER to stop counting before standby mode. If the start bits are not cleared, the counters continue to count after standby mode.

11.4.3 Interrupt enabling during counting

Before set UIE bit to 1, always clear the TE bit in TER to stop the channel count. Then set UIE. Clear UF bit at the same time. Start count after setting.

12 Watchdog Timer (WDT)

12.1 Overview

When software gets trapped in dead loop or system lockup, the watchdog timer provides a means of escape. Once started, the Watchdog Timer must be serviced by software periodically. If servicing does not take place, the watchdog times out and generates a reset signal.

12.1.1 Features

- Generates power-on reset or manual reset.
- 16-bit counter
- Counter clock can be selected from eight clocks: ϕ , $\phi/4$, $\phi/16$, $\phi/32$, $\phi/64$, $\phi/256$, $\phi/1024$, $\phi/4096$. (ϕ is the internal clock for on-chip device)

12.1.2 Block Diagram

Following figure illustrates WDT block diagram.

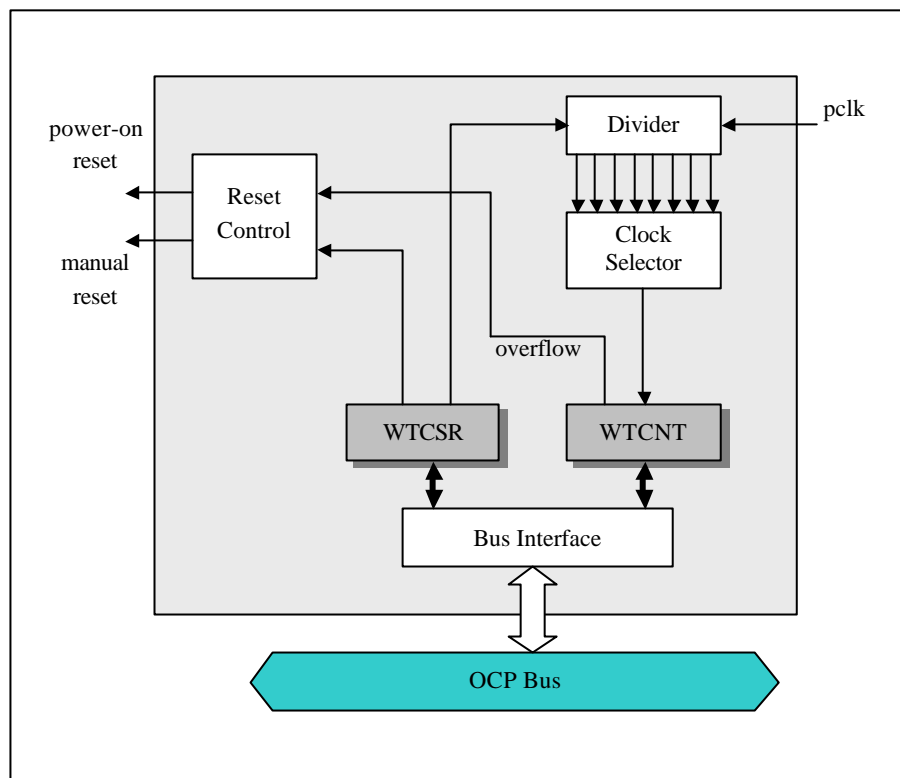


Figure 12-1 WDT Block Diagram

12.2 Register Configuration

In this section, we will describe the registers in WDT. Following table lists all the registers definition. And detailed function of each register will be described below.

Table 12-1 WDT Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
WTCNT	Watchdog Timer Counter	R/W	H'0000	H'E0000404	Read: 16 Write: 16
WTCSR	Watchdog Timer Control/Status Register	R/W	H'00	H'E0000400	Read: 8 Write: 8

12.2.1 Watchdog Timer Control/Status Register (WTCSR)

The WTCSR is an 8-bit read/write register. It is initialized to H'00 by power-on reset or manual reset. The WTCSR is also initialized by reset signal generated by WDT.

Bit:	7	6	5	4	3	2	1	0
Read:			OVF	START	RSTS	CS2	CS1	CS0
Write:								
Reset:	0	0	0	0	0	0	0	0

- **Bit 6~7:** Reserved bits. These bits always read 0, and written are ignored.
- **CS0—CS2:** Clock Selection. Select counter clock division ratio.

Bit 2: CS2	Bit1: CS1	Bit0: CS0	Description	
0	0	0	dclk	(Initial value)
0	0	1	dclk /4	
0	1	0	dclk /16	
0	1	1	dclk /32	
1	0	0	dclk /64	
1	0	1	dclk /256	
1	1	0	dclk /1024	
1	1	1	dclk /4096	

- **RSTS:** Reset Mode Selection. Select reset mode.

Bit 3: RSTS	Description	
0	Generates Power-on reset when overflow.	(Initial value)
1	Generates manual reset when overflow.	

- **START:** Timer Start. Start or stop the timer counting.

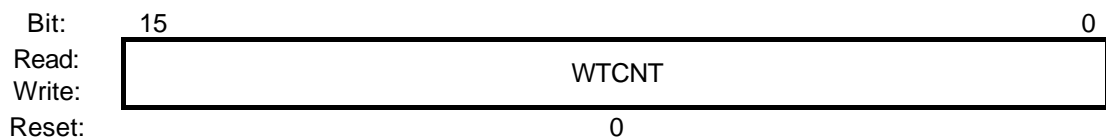
Bit 4: START	Description	
0	Timer counting is disabled.	(Initial value)
1	Timer counting is enabled.	

- **OVF:** Overflow. Indicates that the timer has overflow. This can only be written with 0. Write with 1 will be ignored.

Bit 5: OVF	Description	
0	Timer has not overflowed.	(Initial value)
1	Timer has overflowed.	

12.2.2 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 16-bit read/write counter working on the selected clock setting by CS0-CS2 in WTCSR. When the counter overflows, a power-on reset or manual reset is generated due to the RSTS bit in WTCSR. The WTCNT is initialized to H'0000 by a power-on reset or manual reset.



12.3 Usage Notes

12.3.1 Watchdog Timer Function

1. Select the type of reset desired by setting RSTS bit in WTCSR, and set count clock division ratio by setting CS0~2 bits.
2. Set START bit in WTCSR to 1. The counter begins to count.
3. Software should clear the WTCNT to H'0000 periodically so that the counter can not overflow.
4. If counter overflows, OVF bit in WTCSR is set to 1, and a reset of specified type will be asserted.

Note: Before count clock division ratio is changed, the counter should be stopped.

13 General Purpose Input/Output Port (GPIO)

13.1 Overview

The General Purpose Input/Output (GPIO) ports are used for control and handshake functions between Arca210 and external circuitry by generating output signals and capturing input signals specific for applications. Arca210 supports six GPIO ports – A, B, C, D, E and F. The general-purpose input/output function and alternate internal chip function are configurable. All the pins in port B can be configured as interrupts and sent to INTC. UART, AC97, SCC, UART2, DMA and ETHC share their pins with GPIO in port C, D, E and F.

13.1.1 Features

GPIO features:

- Total pin number is 46 and total port number is 6.
- Each pin can be configured as general purpose input or output.
- Port A is general-purpose I/O port with no alternate functions.
- Signals of Port B have interrupt input alternate function.
- Port C is multiplexed with UART and AC97.
- Port D is multiplexed with SCC, UART2 and DMA.
- Port E is multiplexed with ETHC and DMA.
- Port F is multiplexed with ETHC.
- After power-on reset, all the ports set their pins to input direction and function as general purpose I/O.

13.2 Port A

13.2.1 Features

- Total pin number of port A is 8.
- Port A has no alternate functions.

13.2.2 Block Diagram

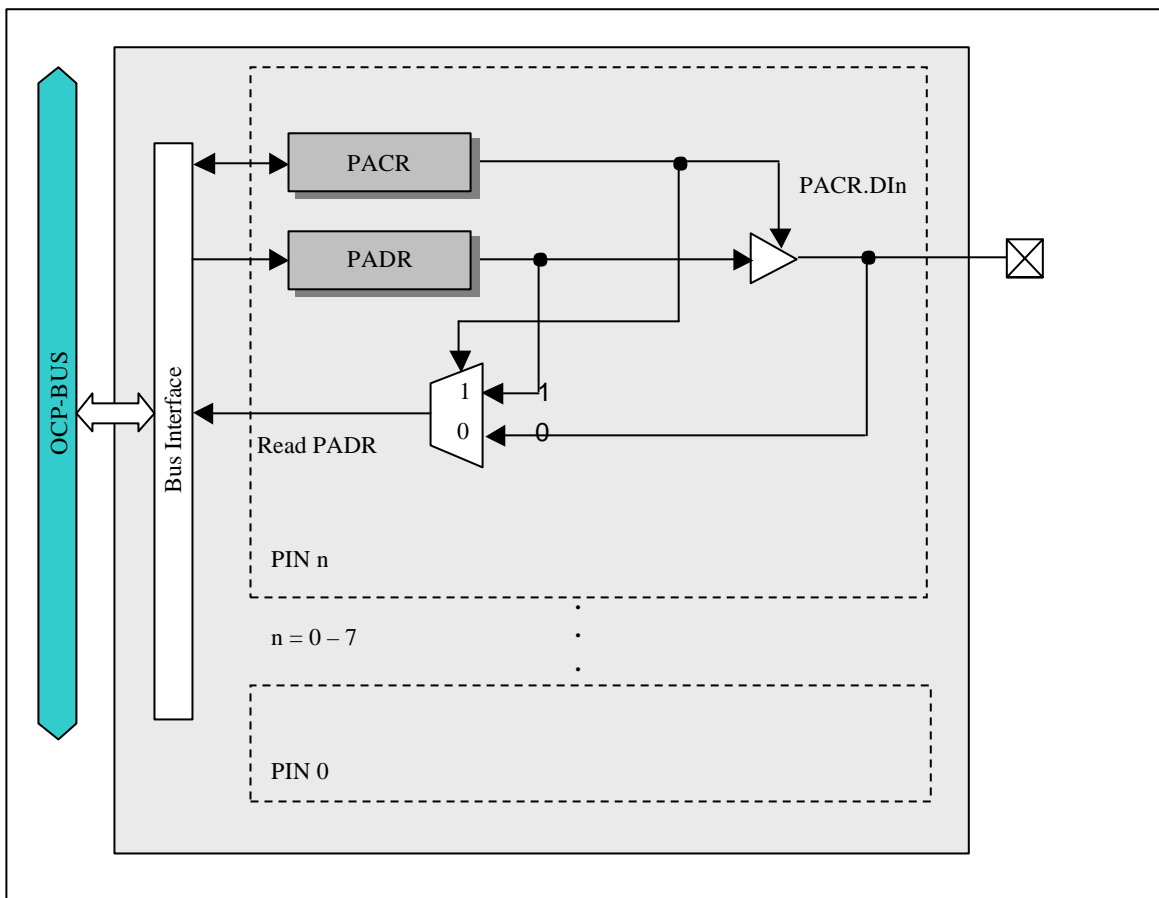


Figure 13-1-1 Port A Block Diagram

13.2.3 Registers Configuration

Table 13-1 Port A Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
PACR	Control Register	R/W	H'00000000	H'E0000500	32
PADR	Data Register	R/W	H'00	H'E0000504	8

13.2.4 Port A Data Register (PADR)

Port A data register (PADR) is an 8-bit register that always stores the data written into via OCP-bus interface. When a bit is configured as a general purpose input pin, the pin level will be read out instead of the bit value stored in PADR. Otherwise, the bit value stored in PADR will be read out. This register can always be written into regardless of its direction and function.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

PADR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

13.2.5 Port A Control Register (PACR)

Port A control register (PACR) is a 32-bit register that controls the input/output direction.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Write:								
Reset:	0	0	0	0	0	0	0	0

PACR is initialized to H'00000000 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

Bit 31~8: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **Direction Control (DI):** The control bit specifies input/output direction for corresponding pin.

Bit n: DI	Description	
0	Bit n of port A is input.	(Initial value)
1	Bit n of port A is output.	

Note: n = 0~7

13.3 Port B

13.3.1 Features

- Total pin number of port B is 8.
- Pull-up resistors are inserted for individual pins.
- Each pin has interrupt input alternate function.
- Interrupt detect manner can be level or edge.
- Each input signal is Schmitt triggered.

13.3.2 Block Diagram

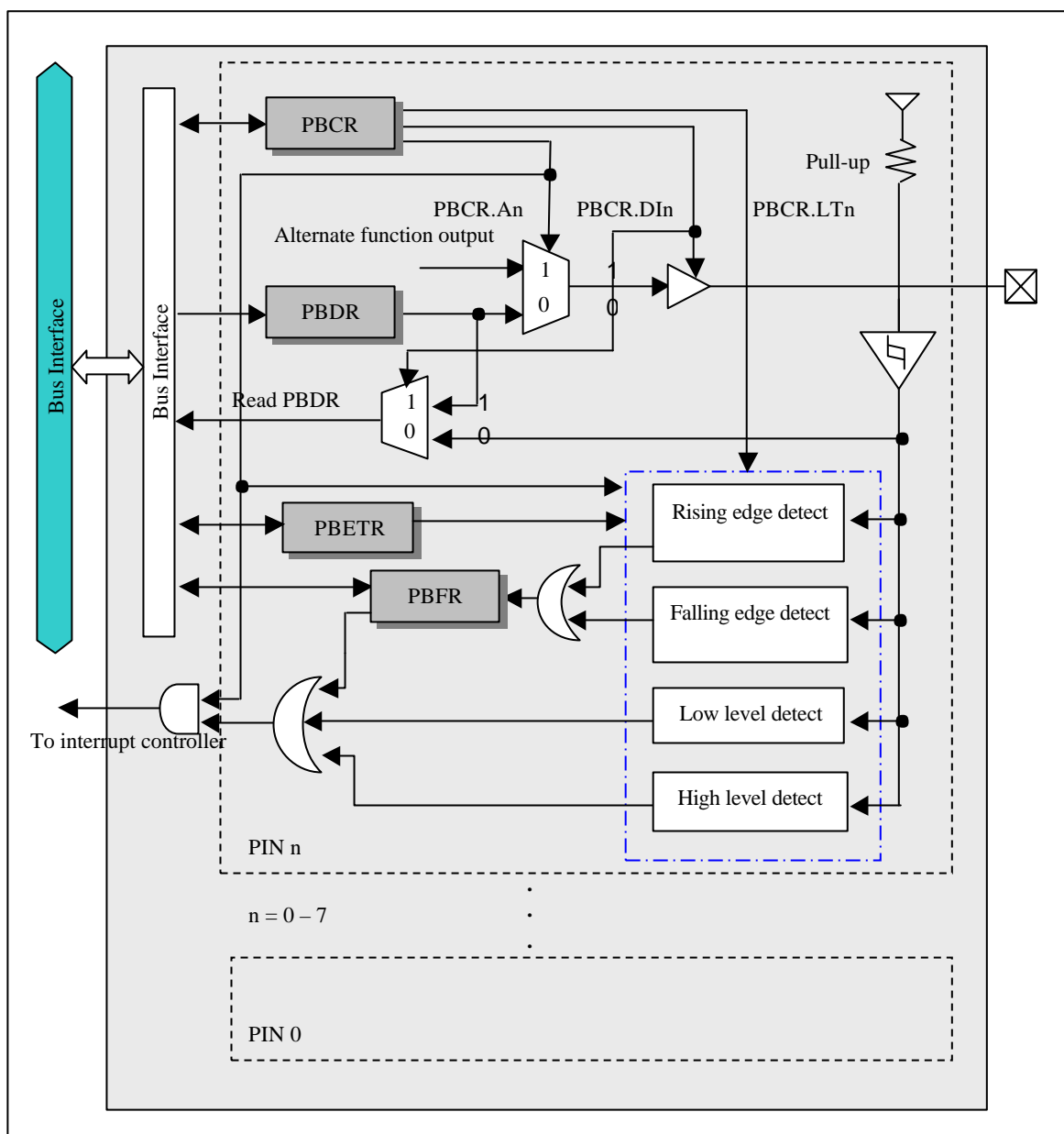


Figure 1-13-3 Port B Block Diagram

13.3.3 Registers Configuration

Table 13-2 Port B Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
PBCR	Control Register	R/W	H'00000000	H'E0000510	32
PBDR	Data Register	R/W	H'00	H'E0000514	8
PBETR	Edge Detect Register	R/W	H'0000	H'E0000518	16
PBFR	Edge Flag Register	R/W	H'00	H'E000051C	8

13.3.4 Port B Data Register (PBDR)

Port B data register (PBDR) is an 8-bit register that always stores the data written into via OCP-bus interface. When a bit is configured as a general purpose input pin, the pin level will be read out instead of the bit value stored in PBDR. Otherwise, the bit value stored in PBDR will be read out. This register can always be written into regardless of its direction and function.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

PBDR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

13.3.5 Port B Control Register (PBCR)

13.3.5.1 Register Description

Port B control register (PBCR) is a 32-bit register that controls the input/output direction, set the function of individual bits to interrupt or to general-purpose I/O port, and select the effective voltage when the corresponding bit is set to interrupt function manner and level sensitive.

Only when interrupt function is selected, the signal from port B to interrupt controller is got from the ORed result on the results of different effective detect manners (level and edge).

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:	LT7	LT6	LT5	LT4	LT3	LT2	LT1	LT0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	A7	A6	A5	A4	A3	A2	A1	A0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Write:								
Reset:	0	0	0	0	0	0	0	0

PBCR is initialized to H'00000000 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

Bit 31~24: Reserved. These bits always read as 0. Write data to these bits are ignored.

Effective Level for Detecting (LT): The control bit specifies the sensitive voltage level for the corresponding pin *n* when it functions as external interrupt (PBCR.An is 1) and interrupt detect manner is level sensitive (PBETR.ETn is 0).

Bit n: LT	Description	
0	Bit <i>n</i> of port B is low-level sensitive when PBETR.ETn is 0.	(Initial value)
1	Bit <i>n</i> of port B is high-level sensitive when PBETR.ETn is 0.	

Note: *n* = 0~7

Alternate Function Control (A): The control bit specifies the corresponding pin functions as general purpose input/output port or external interrupt.

Bit n: A	Description	
0	Bit <i>n</i> of port B functions as general-purpose I/O port.	(Initial value)
1	Bit <i>n</i> of port B functions as external interrupt.	

Note: *n* = 0~7

Direction Control (DI): The control bit specifies input/output direction for corresponding pin.

Bit n: DI	Description	
0	Bit <i>n</i> of port B is input.	(Initial value)
1	Bit <i>n</i> of port B is output.	

Note: *n* = 0~7

13.3.5.2 Alternate Function

Table 1-3 Alternate Function When PBCR.An = 1

Bit	7	6	5	4	3	2	1	0
Name	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
I/O	I	I	I	I	I	I	I	I

Note: When any bit (bit *n*) is configured to alternate (interrupt) function (**An** = 1), regardless the value of corresponding **DIn** bit in PCCR, the direction of that pin will be automatically set to input according to above table.

13.3.6 Port B Edge Detect Register (PBETR)

Port B edge detect register (PBETR) is a 16-bit register. It is implemented to enable/disable rise/fall edge detecting. When edge detect manner is active (disregard whether the port B pins

function as interrupt input), any rise/fall edge on the input signal will generate a logic one being latched into edge flag register PBFR.

Bit:	15	14	13	12	11	10	9	8
Read:	ET7		ET6		ET5		ET4	
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	ET3		ET2		ET1		ET0	
Write:								
Reset:	0	0	0	0	0	0	0	0

PBETR is initialized to H'0000 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

Effective Edge for Detecting (ET): The control bit specifies interrupt edge-detect manner for the corresponding pin n when it functions as external interrupt (PBCR.An is 1).

Bit n: ET	Description	
00	Bit n of port B is level detected.	(Initial value)
01	Bit n of port B is rise-edge detected.	
10	Bit n of port B is fall-edge detected.	
11	Bit n of port B is rise-edge or fall-edge detected.	

Note: n = 0~7

13.3.7 Port B Edge Flag Register (PBFR)

Port B edge flag register (PBFR) is an 8-bit register used to reflect the edge latched for each bit. Writing "1" into a bit will clear it, writing "0" has no effect. The change of detect manner of a pin does not make hardware automatically clear the corresponding bit in PBFR.

Bit:	7	6	5	4	3	2	1	0
Read:	EDGE7	EDGE6	EDGE5	EDGE4	EDGE3	EDGE2	EDGE1	EDGE 0
Write:								
Reset:	0	0	0	0	0	0	0	0

PBFR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

Edge Status (EDGE): The bit shows the edge latched status for the corresponding pin n regardless whether it functions as external interrupt or general purpose input/output port.

Bit n: EDGE	Description	
0	Bit n of port B has not detected an edge.	(Initial value)
1	Bit n of port B has detected an edge.	

Note: n = 0~7

13.4 Port C

13.4.1 Features

- Total pin number of port C is 6.
- Only bit 7 is connected with pull-up resistor.
- Bits 7-6 are multiplexed with UART.
- Bits 3-0 are multiplexed with AC97.

13.4.2 Block Diagram

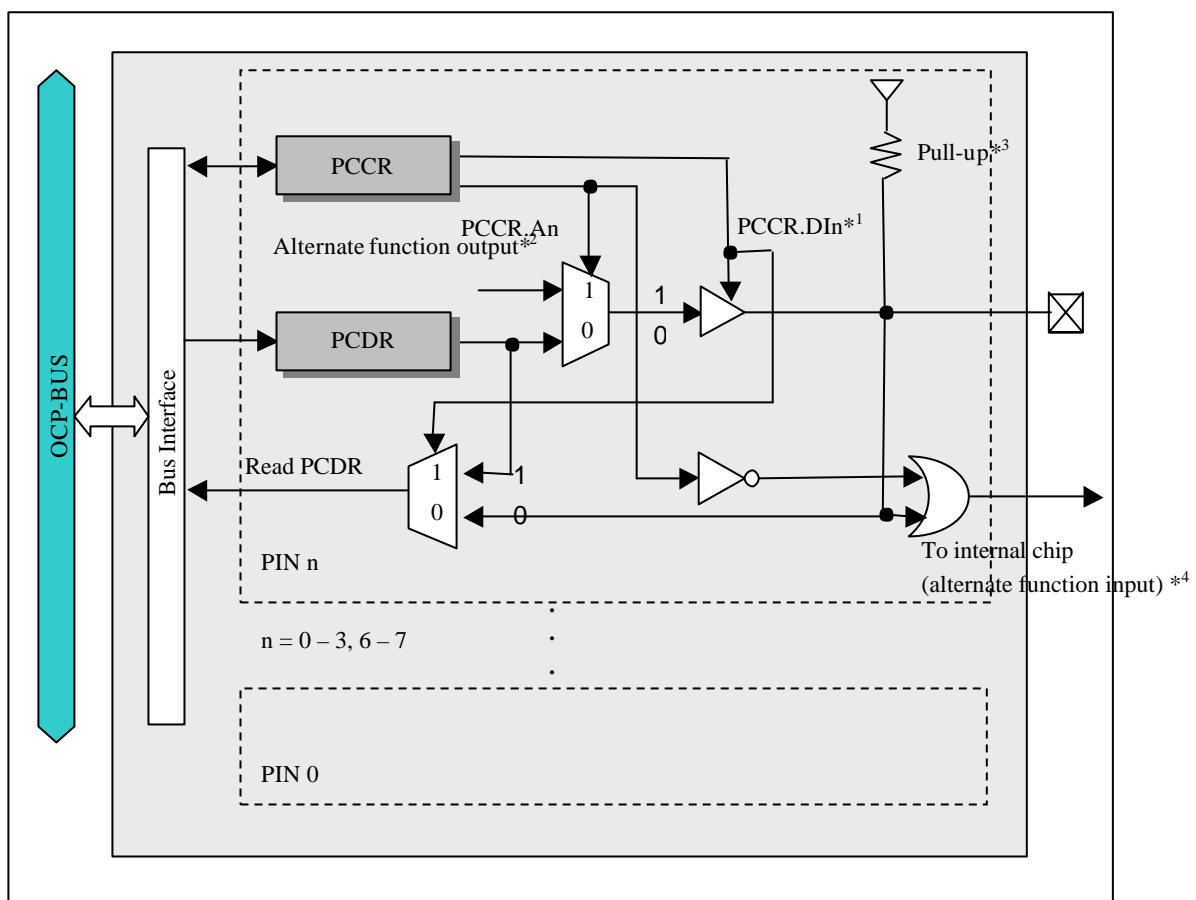


Figure 1-13-4 Port C Block Diagram

Note:

1. $n = 0-3, 6-7$
2. Bit 6, 3, 2, 0
3. Bit 7
4. Bit 7, 1

13.4.3 Registers Configuration

Table 13-4 Port C Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
PCCR	Control Register	R/W	H'00000000	H'E0000520	32
PCDR	Data Register	R/W	H'00	H'E0000524	8

13.4.4 Port C Data Register (PCDR)

Port C data register (PCDR) is an 8-bit register that always stores the data written into via OCP-bus interface. When a bit is configured as a general purpose input pin, the pin level will be read out instead of the bit value stored in PCDR. Otherwise, the bit value stored in PCDR will be read out. This register can always be written into regardless of its direction and function.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

PCDR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

13.4.5 Port C Control Register (PCCR)

13.4.5.1 Register Description

Port C control register (PCCR) is a 32-bit register that controls the input/output direction, selects pins' function between internal chip function and general-purpose I/O port function.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	A7	A6			A3	A2	A1	A0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	DI7	DI6			DI3	DI2	DI1	DI0
Write:								
Reset:	0	0	0	0	0	0	0	0

PCCR is initialized to H'00000000 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

Bit 31~16, 13~12, 5~4: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **Alternate Function Control (A):** The control bit specifies the corresponding pin functions as general purpose input/output port or chip function pin.

Bit n: A	Description	
0	Bit n of port C functions as general-purpose I/O port.	(Initial value)
1	Bit n of port C alternates to its internal chip function.	

Note: n = 0~3, 6~7

- **Direction Control (DI):** The control bit specifies input/output direction for corresponding pin.

Bit n: DI	Description	
0	Bit n of port C is input.	(Initial value)
1	Bit n of port C is output.	

Note: n = 0~3, 6~7

13.4.5.2 Alternate Function

Table 1-5 Alternate Function When PCCR.An = 1

Bit	7	6	5	4	3	2	1	0
Name	UART. RxD	UART. TxD	Reserved	Reserved	AC97. RESET_	AC97. SYNC	AC97. SDATA_IN	AC97. SDATA_OUT
I/O	I	O			O	O	I	O

Note: When any bit (bit n) is configured to alternate function (**An** = 1), regardless the value of corresponding **DIn** bit in PCCR, the direction of that pin will be automatically set to input or output according to above table.

13.5 Port D

13.5.1 Features

- Total pin number of port D is 8.
- Bit 3 and bit 4 are connected with pull-up resistors.
- Bits 7-6 are multiplexed with DMA.
- Bits 5-4 are multiplexed with SCC.
- Bits 3-0 are multiplexed with UART2.

13.5.2 Block Diagram

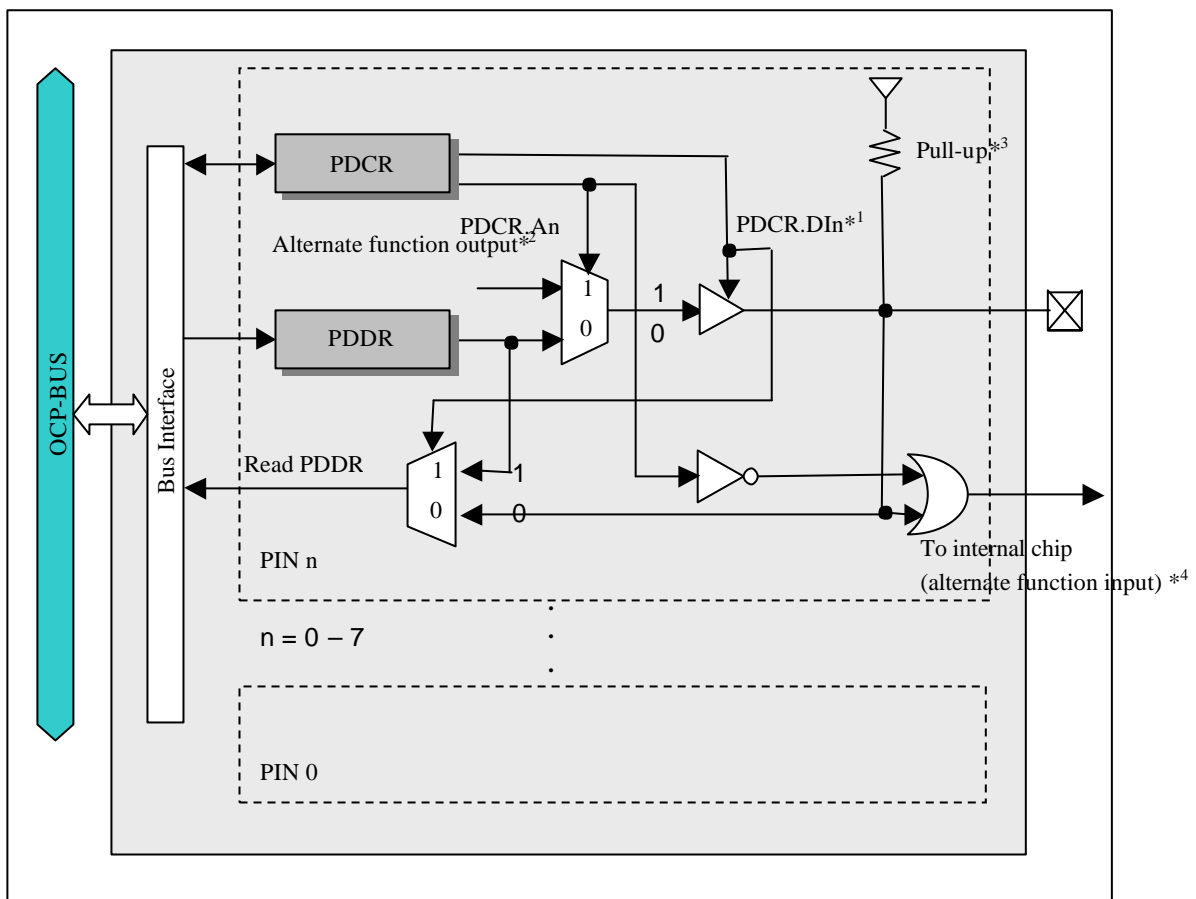


Figure 1-13-5 Port D Block Diagram

Note:

1. $n = 0-7$
2. Bit 7, 6, 5, 4, 2, 0
3. Bit 4, 3
4. Bit 4, 3, 1

13.5.3 Registers Configuration

Table 13-6 Port D Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
PDCR	Control Register	R/W	H'00000000	H'E0000530	32
PDDR	Data Register	R/W	H'00	H'E0000534	8

13.5.4 Port D Data Register (PDDR)

Port D data register (PDDR) is an 8-bit register that always stores the data written into via OCP-bus interface. When a bit is configured as a general purpose input pin, the pin level will be read out instead of the bit value stored in PDDR. Otherwise, the bit value stored in PDDR will be read out. This register can always be written into regardless of its direction and function.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

PDDR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

13.5.5 Port D Control Register (PDCR)

13.5.5.1 Register Description

Port D control register (PDCR) is a 32-bit register that controls the input/output direction, selects pins' function between internal chip function and general-purpose I/O port function.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	A7	A6	A5	A4	A3	A2	A1	A0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Write:								
Reset:	0	0	0	0	0	0	0	0

PDCR is initialized to H'00000000 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

Bit 31~16: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **Alternate Function Control (A):** The control bit specifies the corresponding pin functions as general purpose input/output port or chip function pin.

Bit n: A	Description	
0	Bit n of port D functions as general-purpose I/O port.	(Initial value)
1	Bit n of port D alternates to its internal chip function.	

Note: n = 0~7

- **Direction Control (DI):** The control bit specifies input/output direction for corresponding pin.

Bit n: DI	Description	
0	Bit n of port D is input.	(Initial value)
1	Bit n of port D is output.	

Note: n = 0~7

13.5.5.2 Alternate Function

Table 1-7 Alternate Function When PDCR.An = 1

Bit	7	6	5	4	3	2	1	0
Name	DMA. EOP	DMA. AEN	SCC. CLK	SCC. DATA	UART2. RxD2	UART2. TxD2	UART2. CTS	UART2. RTS
I/O	O	O	O	IO	I	O	I	O

Note: When any bit (bit n) is configured to alternate function (**An** = 1), regardless the value of corresponding **DIn** bit in PDCR, the direction of that pin will be automatically set to input, output or inout according to above table.

13.6 Port E

13.6.1 Features

- Total pin number of port E is 8.
- Pull-up resistors are presented for bit 0, 1, 2.
- Bits 7-3 are multiplexed with ETHC.
- Bits 2-0 are multiplexed with DMA.

13.6.2 Block Diagram

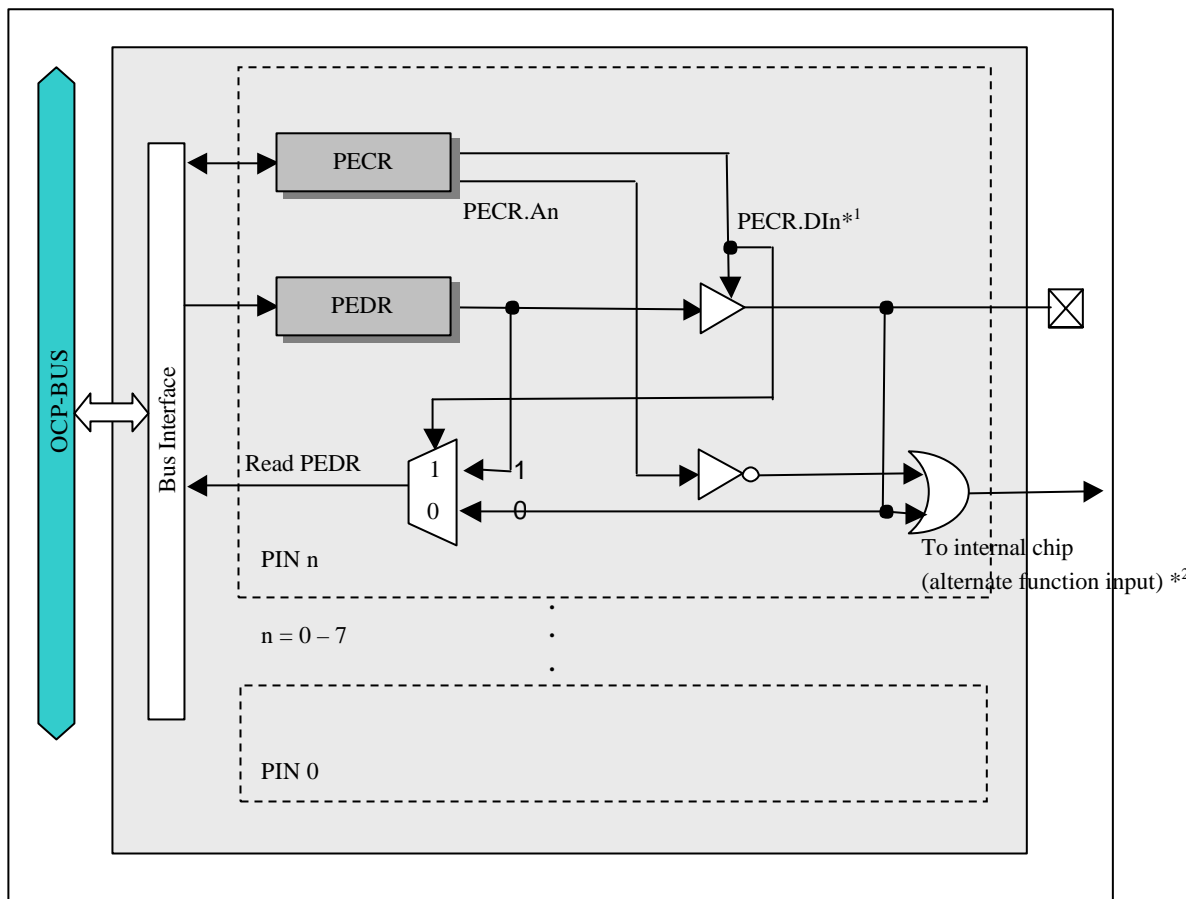


Figure 1-13-6 Port E Block Diagram

Note:

1. $n = 0-7$
2. Bit 0-7

13.6.3 Registers Configuration

Table 13-8 Port E Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
PECR	Control Register	R/W	H'00000000	H'E0000540	32
PEDR	Data Register	R/W	H'00	H'E0000544	8

13.6.4 Port E Data Register (PEDR)

Port E data register (PEDR) is an 8-bit register that always stores the data written into via OCP-bus interface. When a bit is configured as a general purpose input pin, the pin level will be read out instead of the bit value stored in PEDR. Otherwise, the bit value stored in PEDR will be read out. This register can always be written into regardless of its direction and function.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

PEDR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

13.6.5 Port E Control Register (PECR)

13.6.5.1 Register Description

Port E control register (PECR) is a 32-bit register that controls the input/output direction, selects pins' function between internal chip function and general-purpose I/O port function.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	A7	A6	A5	A4	A3	A2	A1	A0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Write:								
Reset:	0	0	0	0	0	0	0	0

PECR is initialized to H'00000000 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

Bit 31~16: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **Alternate Function Control (A):** The control bit specifies the corresponding pin functions as general purpose input/output port or chip function pin.

Bit n: A	Description	
0	Bit n of port E functions as general-purpose I/O port.	(Initial value)
1	Bit n of port E alternates to its internal chip function.	

Note: n = 0~7

- **Direction Control (DI):** The control bit specifies input/output direction for corresponding pin.

Bit n: DI	Description	
0	Bit n of port E is input.	(Initial value)
1	Bit n of port E is output.	

Note: n = 0~7

13.6.5.2 Alternate Function

Table 1-9 Alternate Function When PECR.An = 1

Bit	7	6	5	4	3	2	1	0
Name	ETHC. RXD3	ETHC. RXD2	ETHC. RXD1	ETHC. RXD0	ETHC. RX_ER	DMA. DREQ2_	DMA. DREQ1_	DMA. DREQ0_
I/O	I	I	I	I	I	I	I	I

Note: When any bit (bit n) is configured to alternate function (**An** = 1), regardless the value of corresponding **DIn** bit in PECR, the direction of that pin will be automatically set to input according to above table.

13.7 Port F

13.7.1 Features

- Total pin number of port F is 8.
- Bits 7-0 are multiplexed with ETHC.

13.7.2 Block Diagram

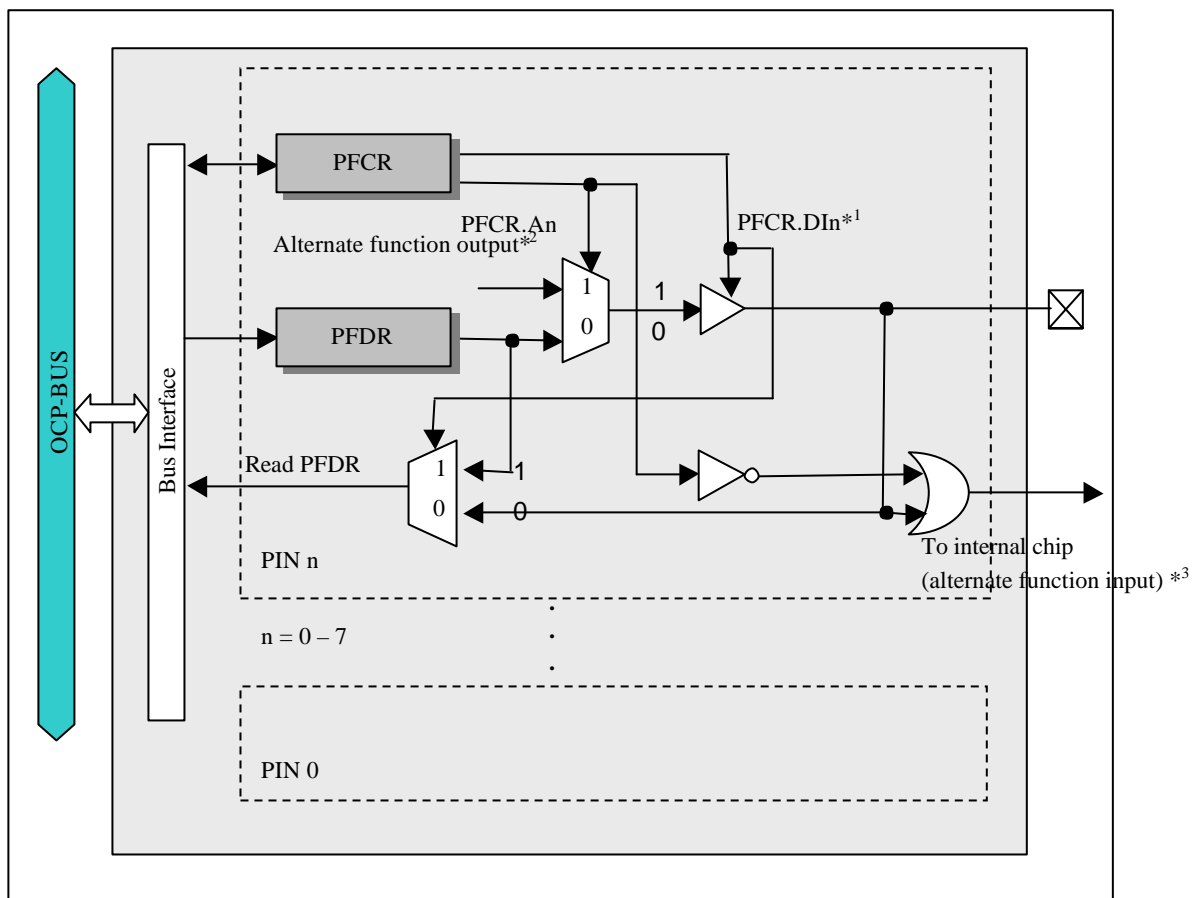


Figure 1-13-7 Port F Block Diagram

Note:

1. *n* = 0-7
2. Bit 5, 4, 3, 2, 1
3. Bit 7, 6, 0

13.7.3 Registers Configuration

Table 13-10 Port F Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
PFCR	Control Register	R/W	H'00000000	H'E0000550	32
PFDR	Data Register	R/W	H'00	H'E0000554	8

13.7.4 Port F Data Register (PFDR)

Port F data register (PFDR) is an 8-bit register that always stores the data written into via OCP-bus interface. When a bit is configured as a general purpose input pin, the pin level will be read out instead of the bit value stored in PFDR. Otherwise, the bit value stored in PFDR will be read out. This register can always be written into regardless of its direction and function.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

PFDR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

13.7.5 Port F Control Register (PFCR)

13.7.5.1 Register Description

Port F control register (PFCR) is a 32-bit register that controls the input/output direction, selects pins' function between internal chip function and general-purpose I/O port function.

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	A7	A6	A5	A4	A3	A2	A1	A0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Write:								
Reset:	0	0	0	0	0	0	0	0

PFCR is initialized to H'00000000 by a power-on reset. It is not initialized by a manual reset. In standby mode, it retains its contents.

Bit 31~16: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **Alternate Function Control (A):** The control bit specifies the corresponding pin functions as general purpose input/output port or chip function pin.

Bit n: A	Description	
0	Bit n of port F functions as general-purpose I/O port.	(Initial value)
1	Bit n of port F alternates to its internal chip function.	

Note: n = 0~7

- **Direction Control (DI):** The control bit specifies input/output direction for corresponding pin.

Bit n: DI	Description	
0	Bit n of port F is input.	(Initial value)
1	Bit n of port F is output.	

Note: n = 0~7

13.7.5.2 Alternate Function

Table 1-11 Alternate Function When PFCR.An = 1

Bit	7	6	5	4	3	2	1	0
Name	ETHC. COL	ETHC. CRS	ETHC. TXD3	ETHC. TXD2	ETHC. TXD1	ETHC. TXD0	ETHC. TX_EN	ETHC. RX_DV
I/O	I	I	O	O	O	O	O	I

Note: When any bit (bit n) is configured to alternate function (**An** = 1), regardless the value of corresponding **DIn** bit in PFCR, the direction of that pin will be automatically set to input or output according to above table.

13.8 Usage Notes

If a pin is programmed as general purpose I/O pin, any associated functionality of the peripheral device with that pin can not be utilized at the same time. For example, in port C, if any bit in bit 0-3 is set to general-purpose I/O function, the other 3 pins can not work as AC97 pins.

14 Universal Asynchronous Receiver/Transmitter

14.1 Overview

Universal asynchronous receiver/transmitter (UART) provides asynchronous serial communication with external devices such as modems and other computers. Sixteen-stage FIFO registers are provided for both transmission and reception, enabling fast, efficient, and continuous communication.

14.1.1 Features

Some of the key features are as the following:

- Full-duplex operation which enables transmission and reception to be performed simultaneously.
- Asynchronous serial communication which is executed using an asynchronous system in which synchronization is achieved character by character. 7 or 8 bit operation with optional even or odd parity and one or two stop bits.
- Generation of break.
- Receive error detection: Parity, frame, overrun, break and timeout errors.
- On-chip baud rate generator allows any bit rate to be selected.
- The DMA controller can be activated to execute a data transfer by issuing a DMA transfer request in the event of a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- Low Power mode by halting the clock supply to UART to reduce power consumption.
- Modem control functions (RTS and CTS) are provided.
- The amount of data in the transmit/receive FIFO registers can be ascertained.
- Four maskable interrupt sources that can issue requests independently: transmit-FIFO-data-empty, receive-FIFO-data-full, receive-error, receive-timeout.

14.1.2 Block Diagram

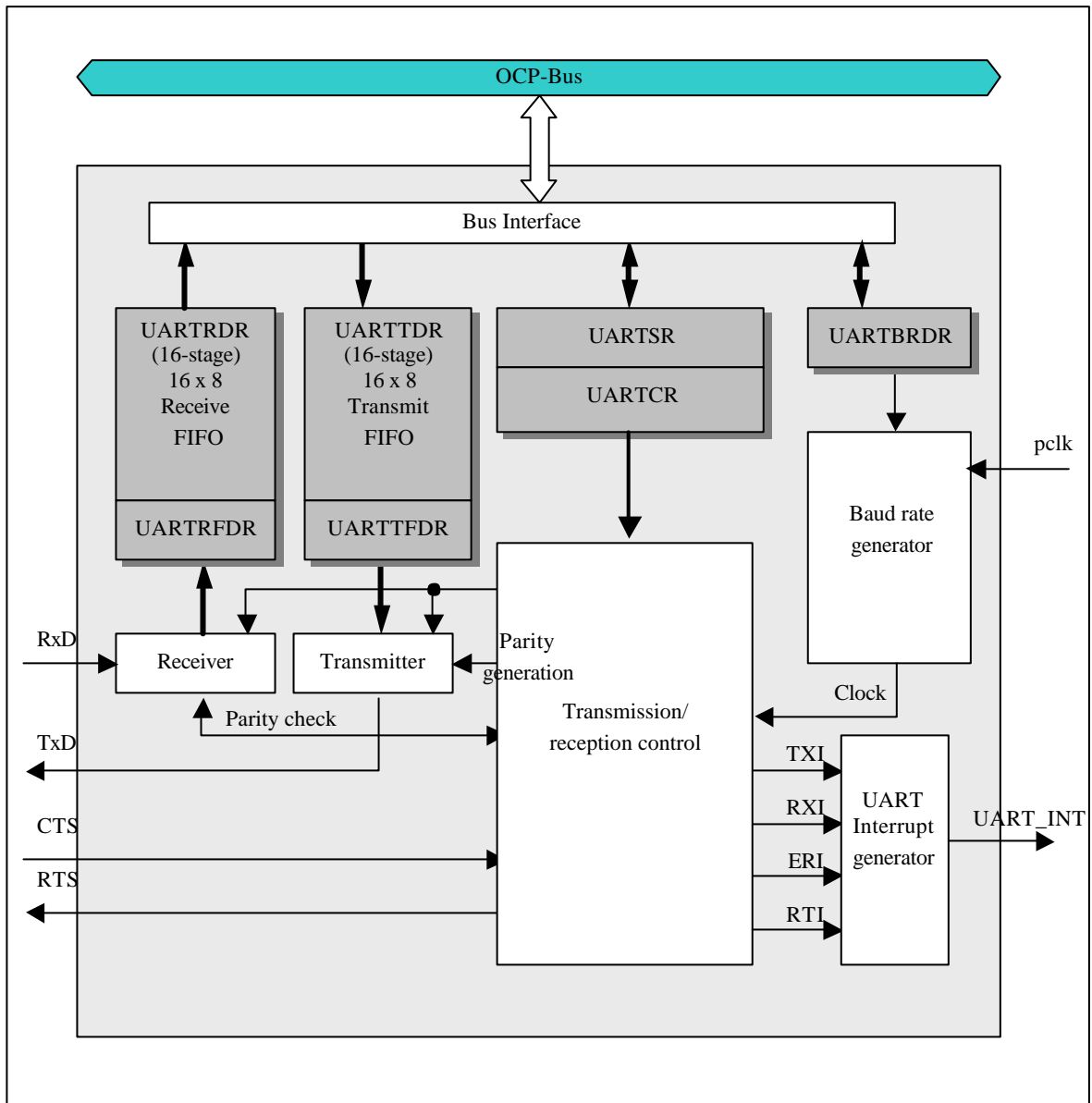


Figure 14-1 UART Block Diagram

14.2 Pin configuration

Table 14-1 UART Pins

Name	Full Name	I/O	Function
RxD	Receive data pin	Input	Receive data input
TxD	Transmit data pin	Output	Transmit data output
CTS	Modem control pin	Input	Transmission enabled (Clear To Send)
RTS	Modem control pin	Output	Transmission request (Request To Send)

Note: When they are disabled for UART function if any of TxD and RxD is used as GPIO; when any of CTS and RTS pins is used as GPIO, modem function is disabled.

14.2.1 RxD Pin

This is the UART Receiver serial input pin. When UART is not in use, it retains high level.

14.2.2 TxD Pin

This is the UART Transmitter serial output pin. When UART is not in use, it retains high level.

14.2.3 CTS Pin — Clear to Send

This is an input pin to UART. External device-like modem requests data from the UART by asserting this pin to low level to enable transmission when modem control is enabled. When CTS goes high during transmission, UART finishes sending the current character and stop transmission.

14.2.4 RTS Pin — Request to Send

This is an output pin from UART. UART signals an external device that it is ready to receive data by asserting this pin to low level when modem control is enabled. When UART detects receive FIFO has 15 or more bytes of data, it negates this pin to high level to disable receiving new data.

14.3 Registers Configuration

Table 14-2 UART Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
UARTBRDR	Bit Rate Divisor Register	R/W	H'0000	H'E0000600	16
UARTCR	UART Control Register	R/W	H'00000000	H'E0000604	32
UARTCRL	UART Control Register L	R/W	H'0000	H'E0000604	16
UARTCRH	UART Control Register H	R/W	H'0000	H'E0000606	16
UARTSR	UART Status Register	R/(W)	H'00000060	H'E0000608	32
UARTRDR	Receive FIFO Data Register	R	Undefined	H'E000060C	8
UARTTDR	Transmit FIFO Data Register	W	Undefined	H'E000060D	8
UARTRFDR	Receive FIFO Data Count Register	R	H'00	H'E000060E	8
UARTTFDR	Transmit FIFO Data Count Register	R	H'00	H'E000060F	8

14.3.1 Receive FIFO Data Register (UARTRDR)

UARTRDR is a 16-stage FIFO register that stores received serial data. When the UART has received one byte of serial data, it transfers the received data from Receiver to UARTRDR and completes the receive operation. Receiver keeps receive operations until the receive FIFO register is full (16 data bytes) and when the receive FIFO is full, subsequent serial data will be lost. UARTRDR is a read-only register, and cannot be written to by the CPU. The contents of UARTRDR are undefined after a power-on reset or manual reset or when there is no receive data in the receive FIFO.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	—	—	—	—	—	—	—	—

14.3.2 Transmit FIFO Data Register (UARTTDR)

UARTTDR is a 16-stage FIFO register that stores data for serial transmission. When transmit is enabled and UARTTDR is not empty, the UART starts to transfer the transmitted data written in UARTTDR to Transmitter. Software should guarantee no more data written to UARTTDR when it is full with 16 bytes of data. UARTTDR is a write-only register, and cannot be read by the CPU. The contents of UARTTDR are undefined after a power-on reset or manual reset.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	—	—	—	—	—	—	—	—

14.3.3 UART Control Register (UARTCR)

The UARTCR register is a 32-bit register which performs enabling or disabling of UART transfer operations, loopback function, break sending, interrupt and setting of UART's serial transfer format, FIFO flushing and trigger data number for the transmit and receive FIFO registers. UARTCR can be read or written to by the CPU at all times. UARTCR is initialized to H'00000000 by a power-on reset or manual reset.

UARTCR is comprised of two 16-bit linked registers (UARTCRH and UARTCRL), so it can be accessed by halfword (UARTCRH and UARTCRL) or word (UARTCR). UARTCRH is mainly used as transfer and FIFO mode register and UARTCRL is used as transfer control and FIFO flush control.

UARTCRH is mapped on bit 31 to 16 of UARTCR, and UARTCRL is mapped on bit 15 to 0 of UARTCR.

Bit:	31	30	29	28	27	26	25	24
Read:	RTRG1	RTRG0	TTRG1	TTRG0				
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:					WLEN	PE	PROE	STOP
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	LOOP			SBRK			TFLUS	RFLUS
Write:							H	H
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	TIE	RIE	TE	RE	REIE	RTIE		MCE
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 27 to 20, 14 to 13, 11 to 10 and 1: Reserved. These bits are always read as 0 and written are ignored.

- **Receive FIFO Data Number Trigger (RTRG1, RTRG0):** These bits are used to set the number of receive data bytes that sets the receive data full (RDF) flag in the serial status register (UARTSR). The RDF flag is set when the number of receive data bytes in UARTRDR is equal to or greater than the trigger set number shown in the following table.

Bit 31: RTRG1	Bit 30: RTRG0	Receive Trigger Number	
0	0	1	(Initial value)
0	1	4	
1	0	8	
1	1	14	

- **Transmit FIFO Data Number Trigger (TTRG1, TTRG0):** These bits are used to set the number of remaining transmit data bytes that sets the transmit FIFO data register empty (TDFE) flag in the serial status register (UARTSR). The TDFE flag is set when the number of transmit data bytes left in UARTTDR is equal to or less than the trigger set number shown in the following table.

Bit 29: TTRG1	Bit 28: TTRG0	Transmit Trigger Number	
0	0	8	(Initial value)
0	1	4	
1	0	2	
1	1	1	

- **Word Length (WLEN):** Selects the number of data bits transmitted or received in a frame.

Bit 19: WLEN	Description	
0	8-bit data.	(Initial value)
1	7-bit data.	

Note: When 7-bit data is selected, the MSB (bit 7) of UARTTDR is not transmitted.

- **Parity Enable (PE):** Selects whether or not parity bit checking and generation is enabled in transmission.

Bit 18: PE	Description	
0	Parity bit addition and checking disabled.	(Initial value)
1	Parity bit addition and checking enabled.	

Note: When the PE bit is set to 1, the parity (even or odd) specified by the PROE bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the PROE bit.

- **Parity Odd/Even Mode (PROE):** Selects either even or odd parity generation and checking during transmission and reception. This bit has no effect when the PE bit is cleared to 0.

Bit 17: PROE	Description	
0	Even parity.	(Initial value)
1	Odd parity.	

Notes:

1. When even parity is set, in transmission, parity bit is added so that the total number of 1-bits in the transmit character plus the parity bit is even. In reception, a check is performed for an even total number of 1-bits in the receive character plus the parity bit.
2. When odd parity is set, in transmission, parity bit is added so that the total number of 1-bits in the transmit character plus the parity bit is odd. In reception, a check is performed for an odd total number of 1-bits in the receive character plus the parity bit.

- **Stop Bit Length Select (STOP):** Selects 1 or 2 bits as the stop bit at the end of a frame data.

Bit 16: STOP	Description	
0	1 stop bit.	(Initial value)
1	2 stop bits.	

Notes:

1. When this bit is cleared to 0, in transmission, only one 1-bit (stop bit) is added to the end of a frame of transmit data for sending.
 2. When this bit is set to 1, in transmission, two 1-bits (stop bits) are added to the end of a frame of transmit data for sending. In reception, only the first stop bit is checked, regardless of the STOP bit setting. The external device is responsible for sending two 1-bits, if the second stop bit is 1, it is treated as a stop bit; otherwise, if it is 0, it is treated as the start bit of the next transmit character.
- **Loopback Test (LOOP):** Internally connects the transmit output pin (TxD) and receive input pin (RxD), and the RTS pin and CTS pin, enabling loopback testing.

Bit 15: LOOP	Description	
0	Loopback disabled.	(Initial value)
1	Loopback enabled.	

- **Send Break (SBRK):** This bit forces sending break through TxD pin. If this bit is set to 1, a low level is continually output on the TxD pin, after completing transmission of the current character. The user is responsible for ensuring this bit is high for a sufficient period of time to generate a valid break, that is, this bit must be asserted for at least one complete frame transmission time in order to generate a valid break. The transmitter samples SBRK after every bit is transmitted. The transmit FIFO contents remain unaffected during a break condition. For normal use, this bit must be cleared to 0.

Following completion of the break transmission, UART transmits two mark bits. The user can continue to fill FIFO, and any remaining data will be transmitted when the break is terminated. This bit can't be set until TE in UARTCR is set 1.

Bit 12: SBRK	Description	
0	Forcing Break output disabled.	(Initial value)
1	Forcing Break output enabled.	

- **Transmit FIFO Flush (TFLUSH):** Flush transmit FIFO to empty state similar to a power on or manual reset.

Bit 9: TFLUSH	Description	
0	Flush is disabled	(Initial value)
1	Flush is enabled	

- **Receive FIFO Flush (RFLUSH):** Flush receive FIFO to empty state similar to a power on or manual reset.

Bit 8: RFLUSH	Description	
0	Flush is disabled	(Initial value)
1	Flush is enabled	

- **Transmit Interrupt Enable (TIE):** Enables or disables transmit-FIFO-data-empty interrupt (TXI) request generation when the number of data bytes in the transmit FIFO register falls to or below the transmit trigger set number, and the TDFE flag in the UART status register (UARTSR) is set to 1.

Bit 7: TIE	Description	
0	Transmit-FIFO-data-empty interrupt (TXI) request disabled.	(Initial value)
1	Transmit-FIFO-data-empty interrupt (TXI) request enabled.	

Note: TXI interrupt request can be cleared by writing transmit data exceeding the transmit trigger set number to UARTTDR, or by clearing the TIE bit to 0.

- **Receive Interrupt Enable (RIE):** Enables or disables generation of a receive-data-full interrupt (RXI) request when the RDF flag is set to 1.

Bit 7: TIE	Description	
0	Transmit-FIFO-data-empty interrupt (TXI) request disabled.	(Initial value)
1	Transmit-FIFO-data-empty interrupt (TXI) request enabled.	

Note: RXI interrupt request can be cleared when receive data below the receive trigger or by clearing the RIE bit to 0.

- **Transmit Enable (TE):** Enables or disables the start of serial transmission by the UART.

Bit 5: TE	Description	
0	Transmit disabled.	(Initial value)
1	Transmit enabled.	

Note: If this bit is set to 1, serial transmission is started when transmit data is written to UARTTDR. So before this bit is set to 1, settings of the transmission format, the transmit FIFO flush, and other information in UART control register (UARTCR) must be made.

- **Receive Enable (RE):** Enables or disables the start of serial reception by the UART.

Bit 4: RE	Description	
0	Reception disabled.	(Initial value)
1	Reception enabled.	

Note:

1. Clearing the RE bit to 0 does not affect the DR, ER, BRK, RDF, FER, PER, and ORER flags, which retain their states.
2. When this bit is set to 1, and when a valid start bit is detected, serial transmission is started. Before this bit is set to 1, settings of the reception format, the receive FIFO flush, and other information in UART control register (UARTCR) must be made.

- **Receive Error Interrupt Enable (REIE):** Enables or disables generation of receive-error interrupt (ERI).

Bit 3: REIE	Description	
0	Receive-error interrupt (ERI) requests disabled	(Initial value)
1	Receive-error interrupt (ERI) requests enabled	

Note: Receive-error interrupt (ERI) requests can be cleared by reading 1 from FER, PER, BRK, or ORER flag, then clearing the flag to 0, or by clearing REIE bit to 0.

- **Receive Timeout Interrupt Enable (RTIE):** Enables or disables generation of receive-timeout interrupt request when the DR flag in UARTSR is set to 1. The receive timeout interrupt is asserted when the receive FIFO is neither empty nor full and no further data is received over a period time of 15 etus. The receive timeout interrupt is cleared when the FIFO becomes empty through reading all the data or by clearing RTIE bit to 0.

Bit 2: RTIE	Description	
0	Receive timeout interrupt (RTI) disabled.	(Initial value)
1	Receive timeout interrupt (RTI) enabled.	

Note: etu: Elementary time unit for transferring one bit.

- **Modem Control Enable (MCE):** Enables the RTS and CTS control signal. When modem control is disabled, RTS and CTS are fixed at active 0 level.

Bit 0: MCE	Description	
0	Modem control is disabled	(Initial value)
1	Modem control is enabled	

14.3.4 UART Status Register (UARTSR)

UARTSR is a 32-bit register. It contains status flags that indicate the operating status of the UART. UARTSR can be read or written to by the CPU at all times. However, only 0 can be written to flags FER, PER, ORER and BRK and only after these flags are read as 1 and then be cleared to 0 by software. UARTSR is initialized to H'00000060 by a power-on reset or manual reset.

Bit:	31	20	19	18	17	16
Read:							
Write:							
Reset:	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:								ORER
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Write:								
Reset:	0	1	1	0	0	0	0	0

Note:

- Only 0 can be written to flags of ORER, FER, PER, BRK to clear those flags.
- When clearing ORER, FER, PER and BRK flags, read UARTSR first and write 0 to those bits which are to be cleared and their value are 1 currently, and write 1 to other flag bits to prevent rest of the flag bits from being cleared unintentionally unless all the flags to be cleared.

Bits 31 to 9: Reserved. These bits are always read as 0 and should only be written with 0.

- Overrun Error (ORER):** Indicates that an overrun error occurred during reception. When the next data reception is completed and the receive FIFO are already full (which means Receiver and the receive FIFO are all full), this bit is set to 1, so CPU must read the data in order to empty the FIFO. It is initialized to 0 by power-on or manual reset or by software after reading it as 1.

Bit 8: ORER	Description	
0	Reception in progress, or reception has ended normally	(Initial value)
1	An overrun error occurred during reception	

Notes:

- The ORER flag is not affected and retains its previous state when the RE bit in UARTCR is cleared to 0.
 - The receive data prior to the overrun error is retained in UARTRDR, and the data received subsequently is lost.
 - Serial reception cannot be continued while the ORER flag is set to 1.
- Receive Error (ER):** Indicates that a receive error (a frame error, a parity error, an overrun or a break error) occurred during reception. When there is a receive error in reception, this

bit is set to 1. It is automatically cleared to 0 when no error bits (FER, PER, BRK or ORER) are set or by power-on reset or manual reset.

Bit 7: ER	Description	
0	No receive error occurred during reception	(Initial value)
1	A receive error occurred during reception	

Notes: The ER flag is not affected and retains its previous state when the RE bit in UARTCR is cleared to 0. When a receive error other than overrun error occurs, the receive data is still transferred to UARTRDR, and reception continues.

- **Transmit End (TEND):** Indicates that there is no valid data in UARTTDR when the last bit of the transmitted character is sent which means the transmit FIFO is empty, and transmission has been ended. Writing 1 or 0 by software doesn't affect its value.

Bit 6: TEND	Description	
0	Transmission is in progress. Clear to 0, when <ul style="list-style-type: none"> · transmit data is written to UARTTDR · data is written to UARTTDR by the DMAC 	
1	Transmission has been ended. Setting 1, when <ul style="list-style-type: none"> · power-on reset or manual reset · the TE bit in UARTCR is 0 · there is no transmit data in UARTTDR on transmission of the last bit of a 1-byte serial transmit character 	(Initial value)

- **Transmit Data FIFO Empty (TDFE):** Indicates that data has been transferred from UARTTDR to Transmitter, the number of data bytes left in UARTTDR has fallen to or below the transmit trigger data number set by bits TTRG1 and TTRG0 in the UART control register (UARTCR), and new transmit data can be written to UARTTDR. Setting to 1 or clearing to 0 by software doesn't affect its value.

Bit 5: TDFE	Description	
0	The number of transmit data bytes exceeding the transmit trigger set number have been written to UARTTDR. Automatically cleared to 0 when <ul style="list-style-type: none"> · writing transmit data exceeding the transmit trigger set number to UARTTDR · transmit data exceeding the transmit trigger set number is written to UARTTDR by the DMAC 	
1	The number of transmit data bytes in UARTTDR does not exceed the transmit trigger set number. Setting 1 when <ul style="list-style-type: none"> · power-on reset or manual reset · the number of UARTTDR transmit data bytes falls to or below the transmit trigger set number 	(Initial value)

Note: As UARTTDR is a 16-byte FIFO register, the maximum number of bytes that can be written is 16 - (transmit trigger set number) when TDFE = 1 or 16 when UARTFDR is 0. The number of data bytes in UARTTDR is indicated by the content of UARTFDR.

- **Break Detect (BRK):** Indicates that a receive data break signal has been detected.

Bit 4: BRK	Description	
0	A break signal has not been received. Clear to 0, when · power-on reset or manual reset · 0 is written to BRK by software after reading BRK = 1	(Initial value)
1	A break signal has been received. Setting to 1, when a null character(contains all zeros including the parity bit) with a framing error is received	

Note:

A frame in which all the bits are logic zero. This includes the stop bit, which is normally a logic one, as well as the data bits. If odd parity is selected, parity error will also be set when this bit is set. This kind of a frame is generally sent to signal the end of a message or the beginning of a new message.

When a break is detected, the receive data (H'00) following detection is not transferred to UARTRDR (only one frame of null data written into receive FIFO and subsequent frames of null data are ignored). When the break ends and the receive signal returns to mark "1", receive data transfer is resumed, that is the next character is only enabled after the receive data input goes to mark "1" and the next valid start bit is received.

At least before the next break occurs, BRK should be cleared to 0 after reading 1, otherwise, the new break error can't be detected and the receiver can't work normally when the new break occurs.

- **Framing Error (FER):** Indicates a framing error in the data read from UARTRDR. Frame error generates when there is no stop bit "1" at the end of a frame data. A frame error is always present on the receiver side, when the transmitter is sending breaks.

Bit 3: FER	Description	
0	There is no framing error in the receive data. Clear to 0, when · power-on reset or manual reset · writing 0 to it after reading FER =1 by software	(Initial value)
1	There is a framing error in the receive data. Setting 1, when there is a framing error in receive data	

Note: In 2-stop-bit mode, only the first stop bit is checked for a value of 1, the second stop bit is not checked. So if the first stop bit is 1 and the second stop bit might be 0, no frame error generates.

- **Parity Error (PER):** Indicates a parity error in the receive data. Parity error generates when the number of 1-bits in the receive data plus the parity bit does not match the parity setting specified by PROE bit in UARTCR.

Bit 2: PER	Description	
0	There is no parity error in the receive data Clear to 0, when <ul style="list-style-type: none"> · power-on reset or manual reset · writing 0 to it after reading PER = 1 by software 	(Initial value)
1	There is a parity error in the receive data Setting 1, when there is a parity error in receive data	

- **Receive FIFO Data Full (RDF):** Indicates that the received data has been transferred from Receiver to UARTRDR, and the number of receive data bytes in UARTRDR is equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in the UARTCR.

Bit 1: RDF	Description	
0	The number of receive data bytes in UARTRDR is less than the receive trigger set number Automatically cleared to 0, when <ul style="list-style-type: none"> · power-on reset or manual reset · UARTRDR is read until the number of receive data bytes in UARTRDR falls below the receive trigger set number · UARTRDR is read by the DMAC until the number of receive data bytes in UARTRDR falls below the receive trigger set number 	(Initial value)
1	The number of receive data bytes in UARTRDR is equal to or greater than the receive trigger set number Setting 1, when UARTRDR contains at least the receive trigger set number of receive data bytes	

Note: UARTRDR is a 16-byte FIFO register. When RDF = 1, at least the receive trigger set number of data bytes can be read. If all the data in UARTRDR is read and another read is performed, the data value will be undefined. The number of received data bytes in UARTRDR is indicated by the content of UARTRFDR.

- **Receive Data Ready (DR):** Indicates that there are less than the receive trigger set number of data bytes in UARTRDR (the receive FIFO is not empty, not full), and no further data has arrived for at least 15 etu after the stop bit of the last data received.

Bit 0: DR	Description	
0	Reception is in progress or has ended normally and there is no receive data left in UARTRDR Automatically cleared to 0, when <ul style="list-style-type: none"> · power-on reset or manual reset · all the receive data in UARTRDR has been read · all the receive data in UARTRDR has been read by the DMAC 	(Initial value)
1	No further receive data has arrived Setting to 1, when UARTRDR contains less than the receive trigger set number of receive data bytes, and no further data has arrived for at least 15 etu after the stop bit of the last data received	

Note: Equivalent to 1.5 frames with an 8-bit, 1-stop-bit format.
etu: Elementary time unit (time for transfer of 1 bit).

14.3.5 Bit Rate Divisor Register (UARTBRDR)

UARTBRDR is a 16-bit register that sets the serial transfer bit rate. UARTBRDR can be read or written to by the CPU. UARTBRDR is initialized to H'0000 by a power-on reset or manual reset. A value of zero is illegal, and so no transmission or reception will occur.

Bit:	15	4	3	2	1	0
Read:							
Write:							
Reset:	0	0	0	0	0	0	0

The baud rate divisor is calculated as follows:

$$\text{Baud rate divisor value in UARTBRDR (BAUDDIV)} = (\text{DCLK} / (16 * \text{Baud rate})) - 1$$

Here, DCLK is the peripheral clock frequency. Table 14-3 shows some typical bit rates and their corresponding divisors, given a DCLK of 3.6864MHz as an example.

Table 14-3 Typical baud rates and divisors

Programmed Divisor Value in UARTBRDR	Bit Rate (bps)
0x1	115200
0x2	76800
0x3	57600
0x5	38400
0xb	19200
0xf	14400
0x17	9600
0x5f	2400
0x6e	1200
0x827	110

14.3.6 Receive FIFO Data Count Register (UARTRFDR)

UARTRFDR is an 8-bit register that indicates the number of data bytes stored in UARTRDR. UARTRFDR only can be read by CPU.

Bit:	7	6	5	4	3	2	1	0
Read:				R4	R3	R2	R1	R0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 7 to 5: Reserved. These bits are always read as 0, and should only be written with 0.

- **Bits 4 to 0** show the number of received data bytes in UARTRDR. A value of H'00 indicates that there is no receive data, and a value of H'10 indicates that UARTRDR is full of received data.

14.3.7 Transmit FIFO Data Count Register (UARTTFDR)

UARTTFDR is a 8-bit register that indicates the number of data bytes stored in UARTTDR. UARTTFDR only can be read by CPU.

Bit:	7	6	5	4	3	2	1	0
Read:				T4	T3	T2	T1	T0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bits 7 to 5: Reserved. These bits are always read as 0, and should only be written with 0.

- **Bits 4 to 0** show the number of transmitted data bytes in UARTTDR. A value of H'00 indicates that there is no transmit data, and a value of H'10 indicates that UARTTDR is full of transmitted data.

14.4 Receiver/ Transmitter

14.4.1 Receiver

The receiver accepts a serial data stream and converts it into a parallel character. When enabled, it searches for a start bit, qualifies it, and then samples the succeeding data bits at the bit-center. When one byte of data has been received, it is transferred to the receive FIFO register, UARTRDR, automatically. Receiver cannot be directly read or written to by the CPU.

14.4.2 Transmitter

The transmitter accepts a parallel character from the CPU and transmits it serially. The start, stop, and parity (if enabled) bits are added to the character. To perform serial data transmission, the UART first transfers transmit data from UARTRDR to Transmitter, then sends the data to the TxD pin starting with the start bit, and then LSB (bit 0), etc. When transmission of one byte is completed, the next transmit data is transferred from UARTRDR to Transmitter, so transmission started, automatically. CTS can be used to control the flow of the serial data. If CTS is high, the transmitter finishes sending the character in progress (if any) then stops and waits for CTS to again become asserted (low). A break character (continuous zeros) can be generated by the transmitter as well.

Transmitter cannot be directly read or written to by the CPU.

14.5 UART Operation

14.5.1 Overview

The UART carries out serial communication in asynchronous mode, in which synchronization is achieved character by character.

Sixteen-stage FIFO buffers are provided for both transmission and reception, reducing the CPU overhead and enabling fast, continuous communication to be performed. RTS and CTS signals are also provided as modem control signals.

The transmission format is selected using the serial control register (UARTCR), as shown in Table 14-4.

Table 14-4 UARTCR Settings for Serial Transfer Format Selection

UARTCR Settings (UART Transfer Format)					
Bit 12: WLEN	Bit 11: PE	Bit 10: STOP	Data Length	Parity Bit	Stop Bit Length
0	0	0	8-bit data	No	1 bit
0	0	1	8-bit data	No	2 bits
0	1	0	8-bit data	Yes	1 bit
0	1	1	8-bit data	Yes	2 bits
1	0	0	7-bit data	No	1 bit
1	0	1	7-bit data	No	2 bits
1	1	0	7-bit data	Yes	1 bit
1	1	1	7-bit data	Yes	2 bits

14.5.2 Serial Operation

Table 14-5 shows the data transfer formats that can be used. Any of 8 transfer formats can be selected according to the UARTCR settings.

Table 14-5 Serial Transfer Formats

UARTCR Settings			Serial Transfer Format and Frame Length											
WLEN	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	S	8-bit data								SP		
0	0	1	S	8-bit data								SP	SP	
0	1	0	S	8-bit data								PR	SP	
0	1	1	S	8-bit data								PR	SP	SP
1	0	0	S	7-bit data							SP			
1	0	1	S	7-bit data							SP	SP		
1	1	0	S	7-bit data							PR	SP		
1	1	1	S	7-bit data							PR	SP	SP	

S: Start bit

SP: Stop bit

PR: Parity bit

Data Transfer Operations

UART Initialization: Before transmitting and receiving data or changing transfer format, it is necessary to clear the TE and RE bits in UARTCR to 0 and flush receive and transmit FIFO, then initialize the UART as described below.

The TE bit should be cleared to 0 after all transmit data has been sent and the TEND flag in UARTSR has been set. TE can also be cleared to 0 during transmission, but the data being transmitted will go to the mark state after the clearance. Before setting TE again to start transmission, the TFLUSH bit in UARTCR should first be set to 1 to reset UARTRDR to empty state.

Figure 14-2 shows a sample UART initialization flowchart.

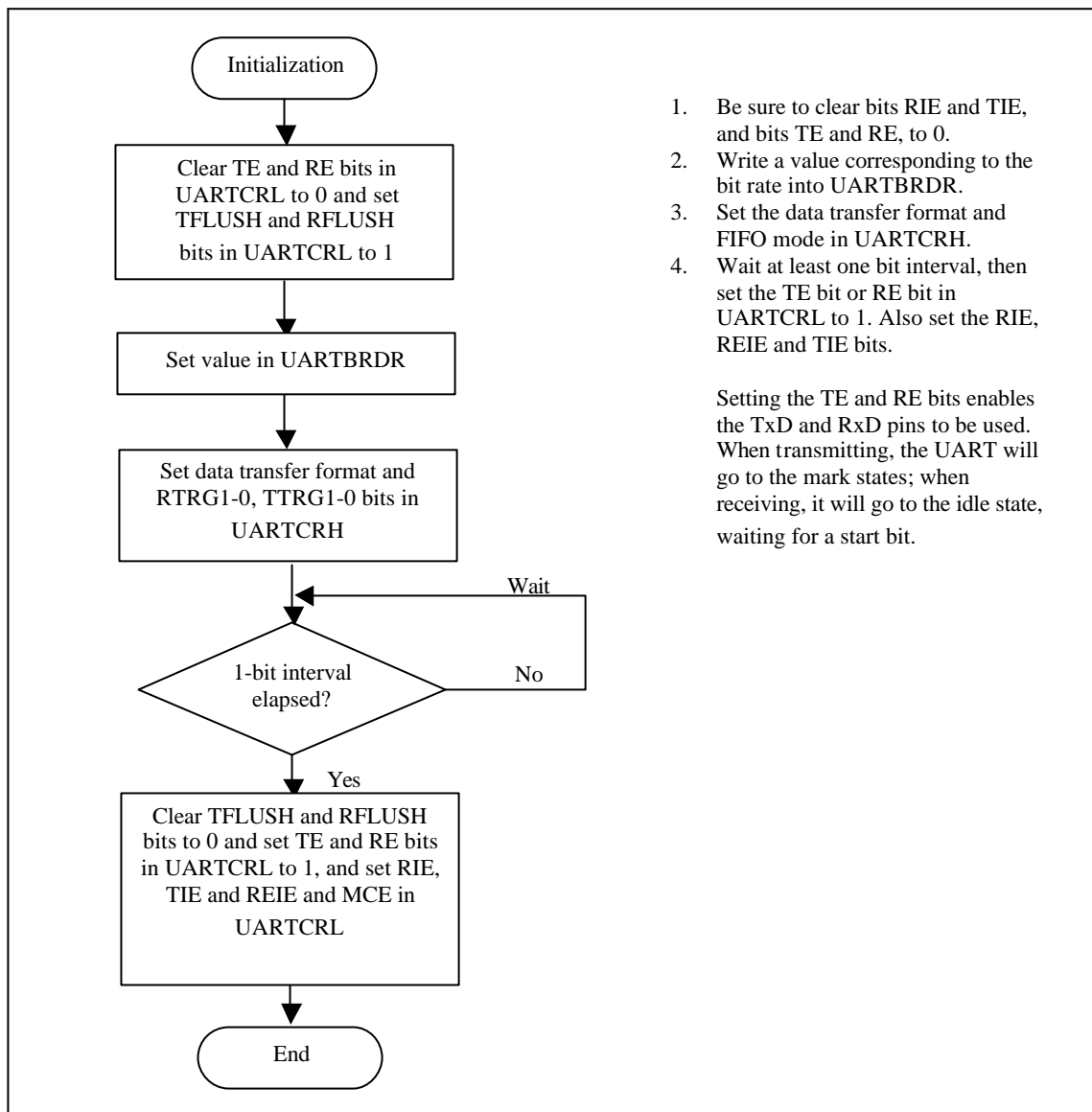


Figure 14-2 Sample of UART Initialization Flowchart

Serial Data Transmissions: Figure 14-3 shows a sample flowchart for UART serial transmission. Use the following procedure for serial data transmission after enabling the UART for transmission.

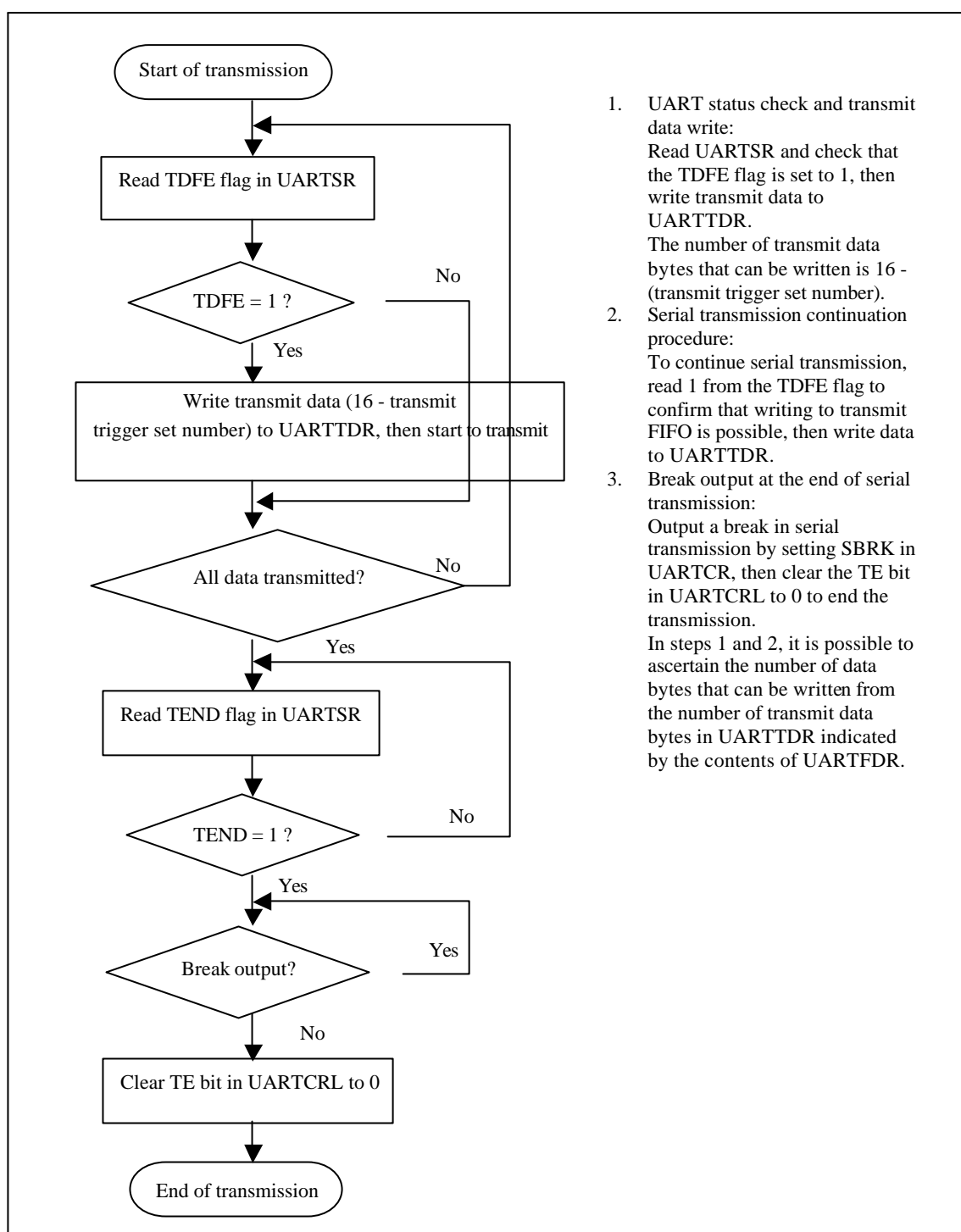


Figure 14-3 Sample UART Serial Transmission Flowchart

In serial transmission, the UART operates as described below.

1. When data is written into UARTTDR, the UART transfers the data from UARTTDR to Transmitter and starts transmitting. Confirm that the TDFE flag in UARTSR is set to 1 before writing transmit data to UARTTDR. The number of data bytes that can be written is at least (16 - transmit trigger setting).
2. When data is transferred from UARTTDR to Transmitter and transmission is started, consecutive transmit operations are performed until there is no transmit data left in UARTTDR. When the number of transmit data bytes in UARTTDR falls to or below the transmit trigger number set in UARTCR, the TDFE flag is set to 1. If the TIE bit in UARTCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated. The serial transmit data is sent from the TxD pin in the following order.
 - a. Start bit: One 0-bit is output.
 - b. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
 - c. Parity bit(if enabled): One parity bit (even or odd parity) is output.
 - d. Stop bit(s): One or two 1-bits (stop bits) are output.
 - e. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.

The UART checks the UARTTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from UARTTDR to Transmitter, the stop bit is sent, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in UARTSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output.

3. Figure 14-4 shows an example of the operation for transmission in asynchronous mode.

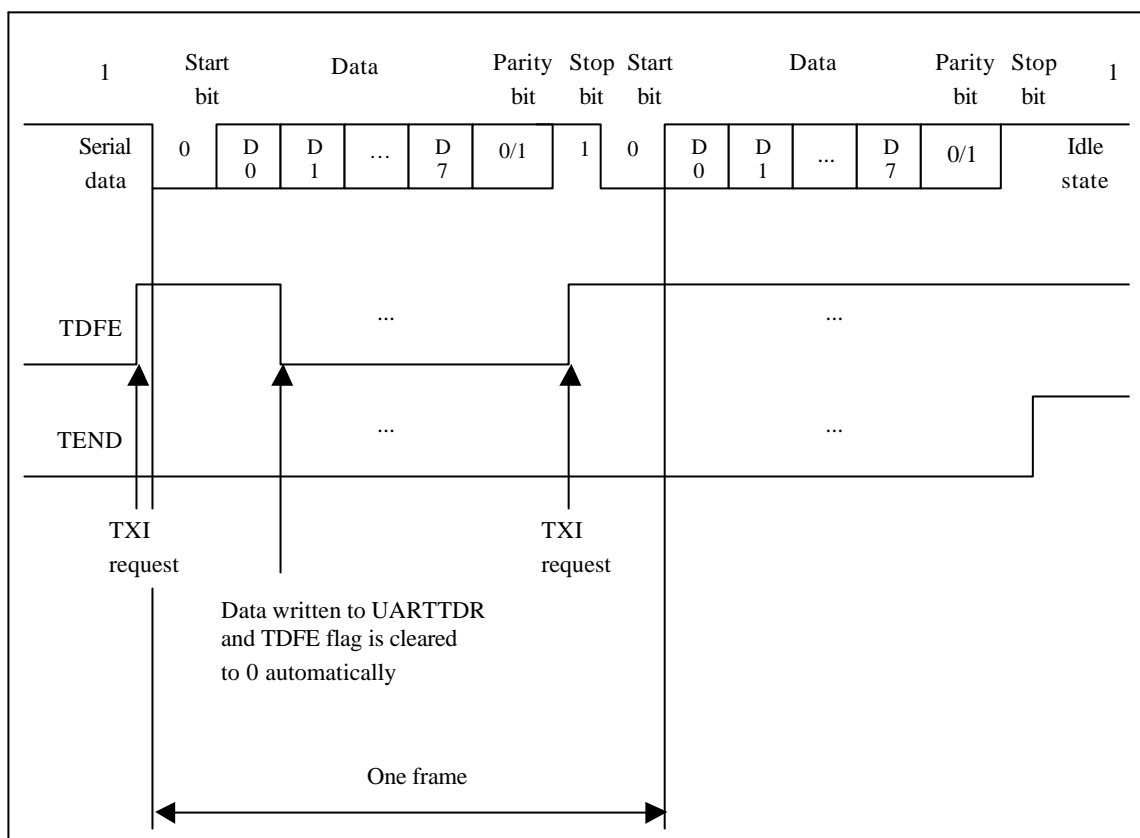


Figure 14-4 Sample of Transmit Operation with 8-bit data, 1 stop bit

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the CTS input value. When CTS is set to 1, if transmission is in progress, the line goes to the mark state after transmission of current frame data. When CTS is set to 0, the next transmit data is output starting from the start bit.

Figure 14-5 shows an example of the operation when modem control is used.

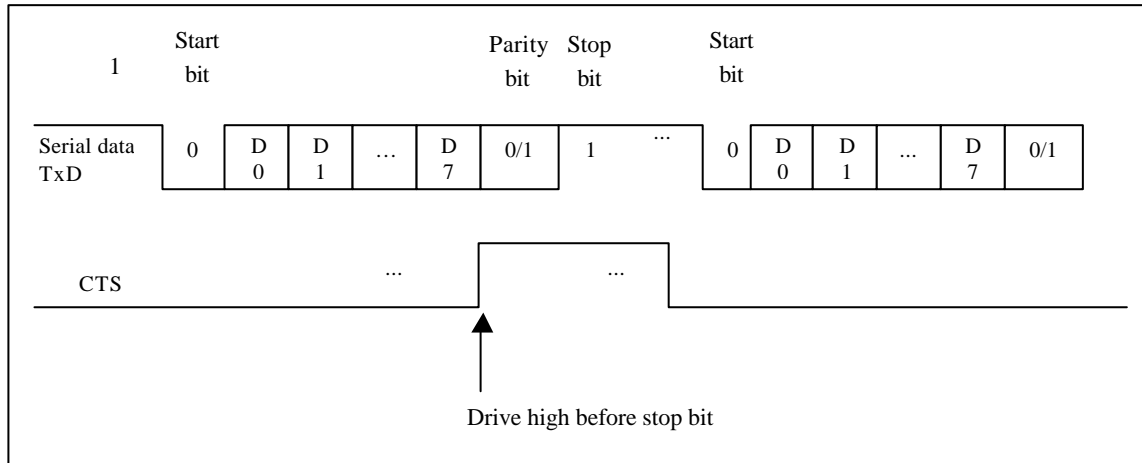


Figure 14-5 Sample of Operation Using Modem Control (CTS)

Serial Data Reception: Figure 14-6 and Figure 14-7 shows a sample flowchart for serial reception. Use the following procedure for serial data reception after enabling the UART for reception.

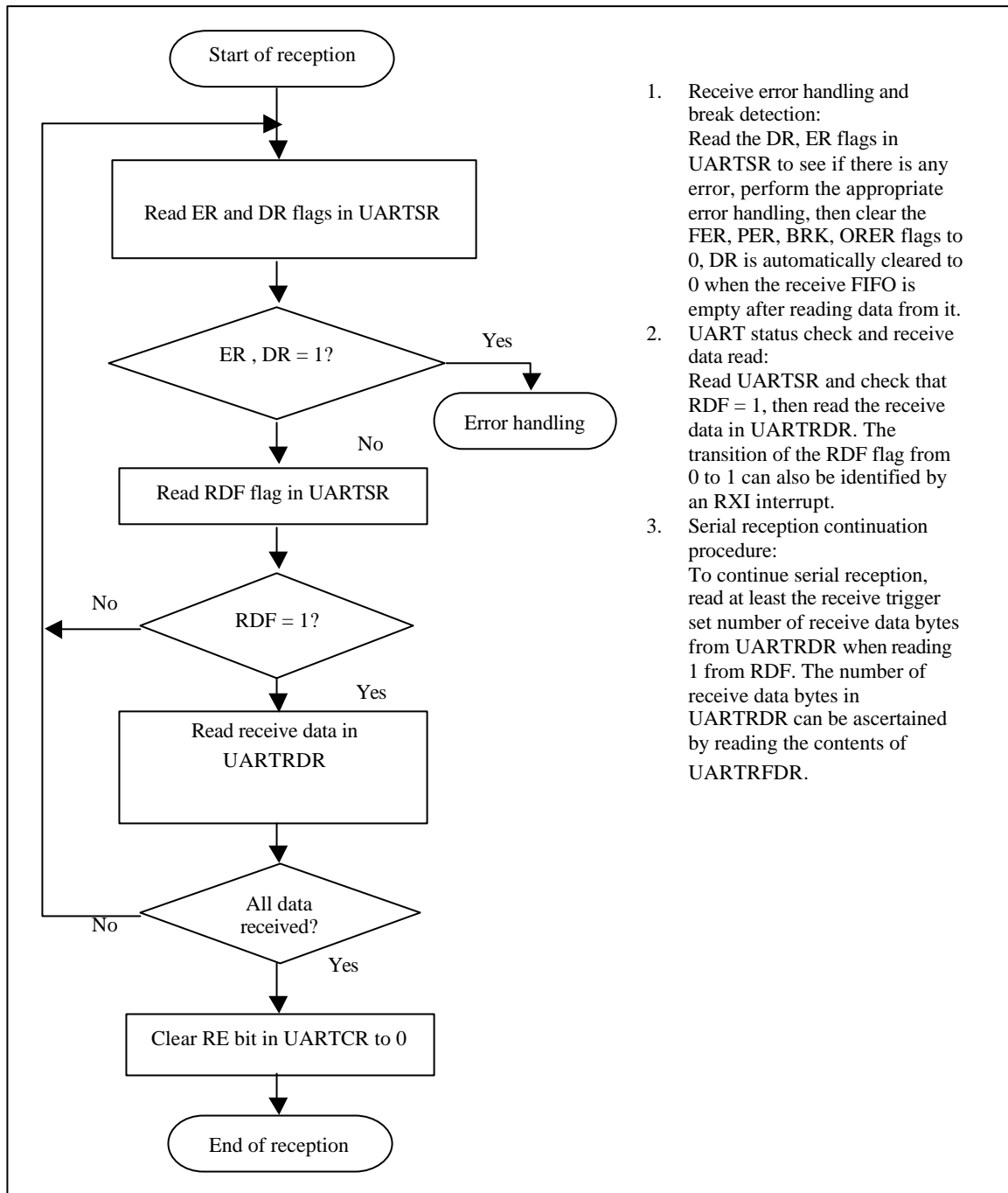


Figure 14-6 Sample of UART Serial Reception Flowchart (1)

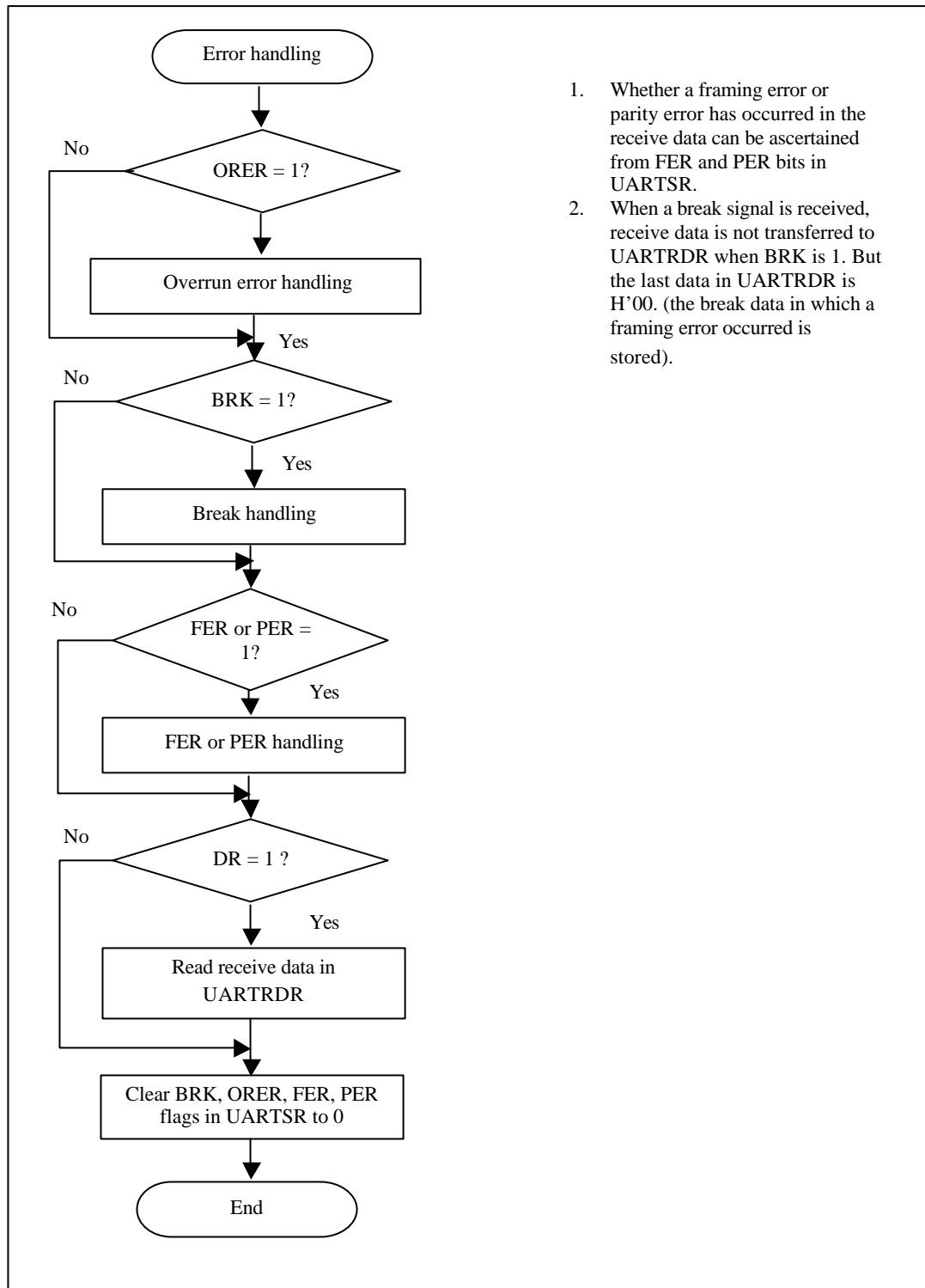


Figure 14-7 Sample of UART Serial Reception Flowchart (2)

In serial reception, the UART operates as described below.

1. The UART monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in Receiver in LSB-to-MSB order.
3. The parity bit and stop bit are received. After receiving these bits, the UART carries out the following checks.
 - a. Stop bit check: The UART checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
 - b. The UART checks whether receive data can be transferred from the Receiver to UARTRDR.
 - c. Overrun error check: The UART checks that the ORER flag is 0, indicating that no overrun error has occurred.
 - d. Break check: The UART checks that the BRK flag is 0, indicating that the break state is not set. If all the above checks are passed, the receive data is stored in UARTRDR.

Note: Reception continues when a receive error other than overrun error occurs.

4. If the RIE bit in UARTCR is set to 1 when the RDF, a receive-FIFO-data- full interrupt (RXI) request is generated. If the REIE bit in UARTCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated, when the DR flag in UARTSR changes to 1 and RTIE bit in UARTCR is set to 1, a receive timeout interrupt request is generated.

Figure 14-8 shows an example of the operation for reception.

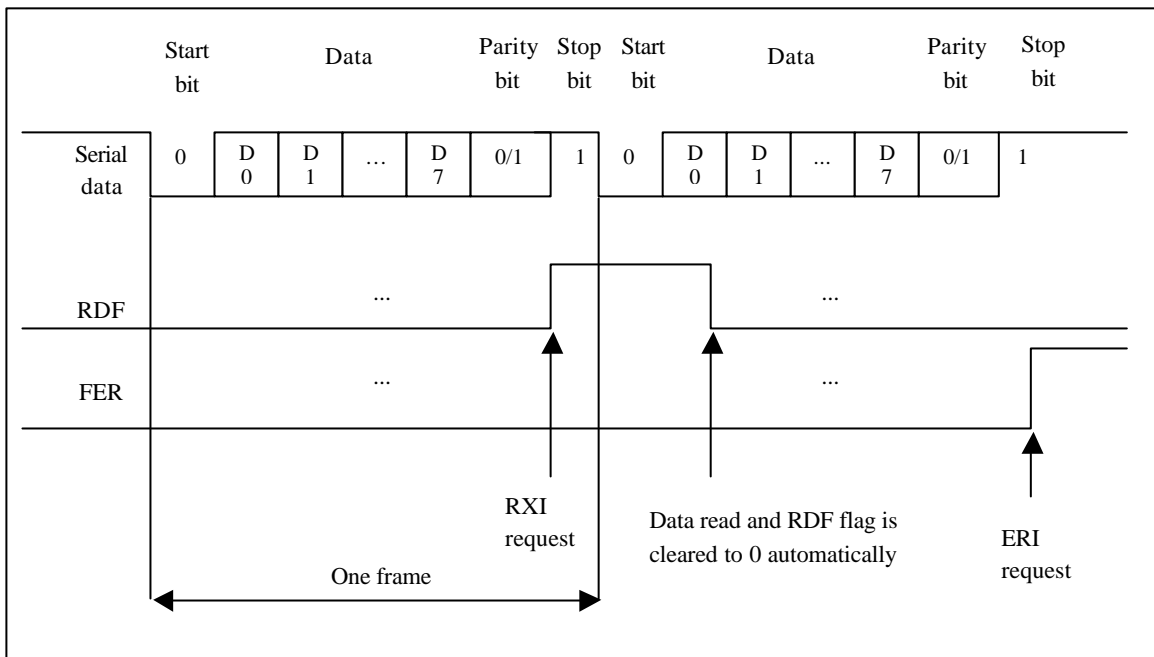


Figure 14-8 Sample of UART Serial Reception Operation with 8-bit data, parity, 1-bit stop

5. When modem control is enabled, the RTS signal is output when UARTRDR is empty. When RTS is 0, reception is possible. When RTS is 1, this indicates that UARTRDR contains 15 or more bytes of data, and there is no free space, reception is not possible.

Figure 14-9 shows an example of the operation when modem control is used.

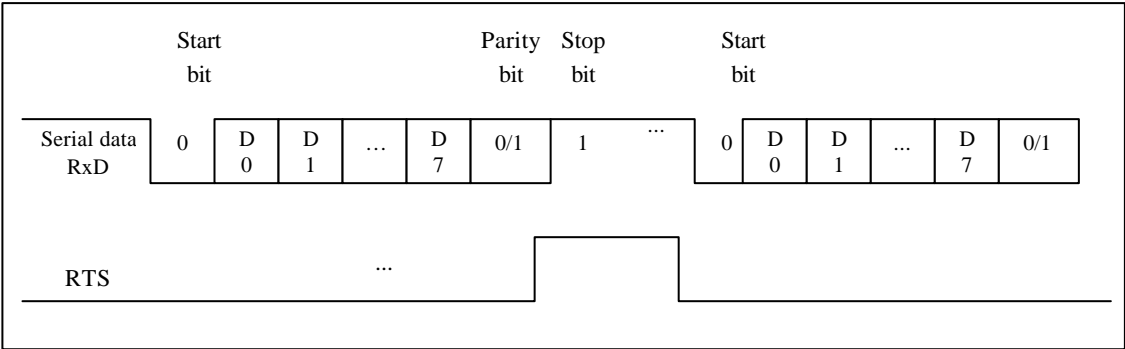


Figure 14-9 Sample of Operation Using Modem Control (RTS)

14.6 UART Interrupt Sources and the DMAC

The UART has four interrupt sources: transmit-FIFO-data-empty interrupt (TXI) request, receive-error interrupt (ERI) request, receive-FIFO-data-full interrupt (RXI) request and receive timeout interrupt (RTI) request.

Table 14-6 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, REIE and RTIE bits in UARTCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When transmission/reception is carried out using the DMAC, output of interrupt requests to the interrupt controller can be inhibited by clearing the RIE and RTIE bit in UARTCR to 0 .

When the TDFE flag in the serial status register (UARTSR) is set to 1, a transmit-FIFO-data-empty request is generated separately from the interrupt request. A transmit-FIFO-data-empty request can activate the DMAC to perform data transfer.

When the RDF flag in UARTSR is set to 1, a receive-FIFO-data-full request is generated separately from the interrupt request. A receive-FIFO-data-full request can activate the DMAC to perform data transfer.

When the DR flag in UARTSR is set to 1, a receive-timeout request is generated separately from the interrupt request. A receive-timeout request can activate the DMAC to perform data transfer.

When using the DMAC for transmission/reception, set and enable the DMAC before making the UART settings.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in UARTRDR.

Table 14-6 UART Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive FIFO data full flag (RDF)	Possible	↓
RTI	Interrupt initiated by receive data ready flag (DR)	Possible	↓
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

14.7 UART Usage Notes

Note the following when using the UART:

UARTTDR Writing and the TDFE Flag: The TDFE flag in UARTSR is set when the number of transmit data bytes written in UARTTDR has fallen to or below the transmit trigger number set by bits TTRG1 and TTRG0 in UARTCR. After TDFE is set, up to (16 - transmit trigger number set) bytes of data can be written to UARTTDR, allowing efficient continuous transmission.

However, if the number of data bytes written in UARTTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again. The number of transmit data bytes in UARTTDR can be found from UARTRFDR.

UARTRDR Reading and the RDF Flag: The RDF flag in UARTSR is set when the number of receive data bytes in UARTRDR has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in UARTCR. After RDF is set, receive data equivalent to the trigger number can be read from UARTRDR, allowing efficient continuous reception.

However, if the number of data bytes in UARTRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again. The number of receive data bytes in UARTRDR can be found from UARTRFDR.

Break Detection and Processing: Break signals can be detected by reading the RxD pin directly with a framing error (FER) also detected. In the break state, the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set when BRK flag is set.

Although the UART stops transferring receive data to UARTRDR after receiving a break, the receive operation continues.

Receive Data Sampling Timing and Receive Margin: The UART operates on a base clock with a frequency of 16 times the bit rate. In reception, the UART synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse.

15 Smart Card Controller (SCC)

15.1 Overview

The SCC (smart card controller) supports smart card interface that conforms to the ISO/IEC standard 7816-3 for identification of cards. The SCC module is designed to work alternatively with the UART. Register settings are switched between UART and SCC.

15.1.1 Features

SCC features:

- A straightforward extension of UART: When SCC is disabled, UART can work as a normal UART.
- Support asynchronous character ($T = 0$) communication modes.
 1. Data length: 8 bits
 2. Parity bit generation
 3. Parity error detection and automatic reporting error and re-transmission data on IO line
 4. Supports both direct convention and inverse convention
- Support asynchronous block ($T = 1$) communication modes.
- SCC clock rate can be selected using on-chip baud rate generator.
- Bit rate can be adjusted according to parameter F/D.
- Support serial clock stop.
- Data communication and error handling are supported by interrupts.

15.1.2 Block Diagram

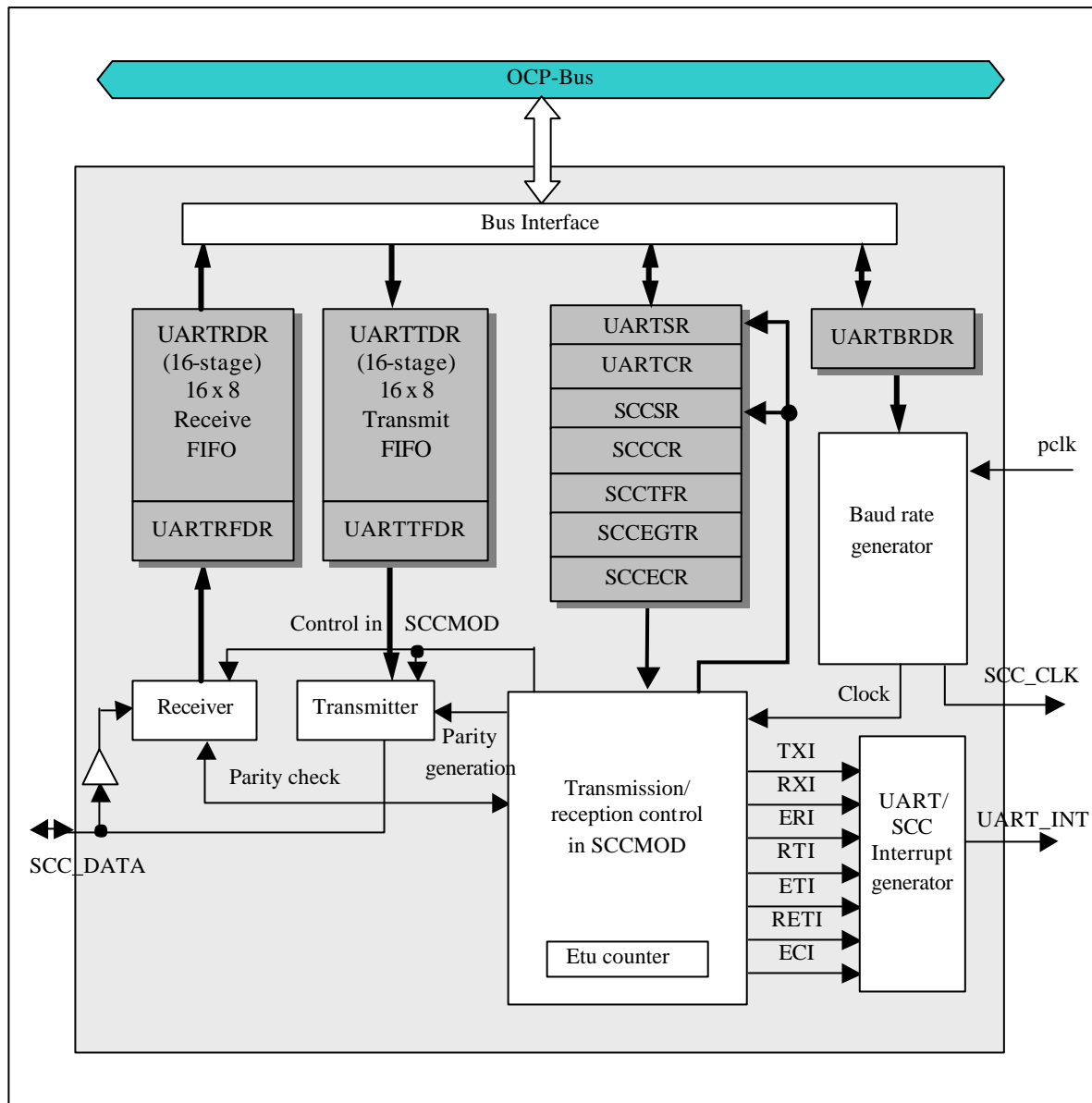


Figure 15-1 SCC Block Diagram (UART functions as SCC)

15.2 Pin configuration

Table 15-1 SCC Pins

Name	Full Name	I/O	Function
DATA	Transmit/Receive data pin	IO	Transmit/Receive data connects SCC and smart card
CLK	Serial clock pin	Output	Serial clock connects SCC and smart card

Note: The above pins are disabled for SCC functions if any of them is used as GPIO.

15.2.1 DATA Pin

This is serial data pin. It is bi-directional. When it is not in use (driven), it retains high level.

15.2.2 CLK Pin

This is serial clock output pin from SCC.

15.3 Registers Configuration

Table 15-2 UART and SCC Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
UARTBRDR	Bit Rate Divisor Register	R/W	H'0000	H'E0000600	16
UARTCR	UART Control Register	R/W	H'00000000	H'E0000604	32
UARTCRL	UART Control Register L	R/W	H'0000	H'E0000604	16
UARTCRH	UART Control Register H	R/W	H'0000	H'E0000606	16
UARTSR	UART Status Register	R/(W)	H'00000060	H'E0000608	32
UARTRDR	Receive FIFO Data Register	R	Undefined	H'E000060C	8
UARTTDR	Transmit FIFO Data Register	W	Undefined	H'E000060D	8
UARTRFDR	Receive FIFO Data Count Register	R	H'00	H'E000060E	8
UARTTFDR	Transmit FIFO Data Count Register	R	H'00	H'E000060F	8
SCCCR	SCC Control Register	R/W	H'0008	H'E0000620	16
SCCSR	SCC Status Register	R/W	H'8000	H'E0000624	16
SCCTFR	SCC Transmission Factor Register	R/W	H'0173	H'E0000628	16
SCCEGTR	SCC Extra Guardtime Register	R/W	H'00	H'E000062C	8
SCCECR	SCC Etu Counter Value Register	R/W	H'00000000	H'E0000630	32

15.3.1 UART Register

(UARTBRDR, UARTCR, UARTCRL, UARTCRH, UARTSR, UARTRDR, UARTTDR, UARTRFDR, UARTTFDR)

Since the SCC module is a straightforward extension of UART, the related UART registers operate in the same way no matter whether the SCC is enabled or not. When SCC is disabled, UART can work independently as a normal UART. Please refer to UART spec for detailed registers' functions.

15.3.2 SCC Control Register (SCCCR)

The SCCCR register is a 16-bit register that is mainly used to control operation mode and transfer format mode. SCCCR is initialized to H'0008 by a power-on reset or manual reset.

Bit:	15	14	13	12	11	10	9	8
Read:	SCCMOD	CONV	TP				ECIE	ETIE
Write:								
Reset:	0	0	0	0	0	0	0	0
Bit:	7	6	5	4	3	2	1	0
Read:	RETIE				TSEND	PX[1]	PX[0]	CLKSTP
Write:								
Reset:	0	0	0	0	1	0	0	0

Bit 12~10, 6~4: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **SCC Mode Select (SCCMOD):** Select whether the SCC module will be used or not.

Bit 15: SCCMOD	Description	
0	Operates as a UART	(Initial value)
1	Operates as a SCC	

Note: When SCCMOD is set to 1, bit 19(WLEN), bit 12 (SBRK) and bit 0(MCE) in UARTCR should be cleared to 0, bit 18(PE) should be set to 1. SCC must operate in half-duplex manner except it performs self-loopback test.

- **SmartCard Data Transfer Convention (CONV):** Selects the serial/parallel conversion format and specifies whether to invert the logic level of the data. This function is used for transmitting and receiving with an inverse convention card. PROE can only be set to 1 when CONV is 1 and cleared to 0 when CONV is 0. CONV doesn't directly invert the logic level of the parity bit, parity bit of both transmitting and receiving is still controlled by PROE bit.

Bit 14: CONV	Description	
0	Transmit data are transferred unchanged and LSB first, receive data is stored unchanged and LSB first.	(Initial value)
1	Transmit data are stored as LSB first but transferred as MSB first and inverted before transfer, receive data is received as MSB first but is inverted and transitioned to LSB first before storage.	

Note: CONV must be cleared to 0 before TS character is received. CONV is set to 1 or maintains its state according to TS character after the character is received.

Smart Card Transmission Protocol (TP): Specifies smart card transmission protocol under asynchronous mode.

Bit 13: TP	Description	
0	Transmission protocol T = 0	(Initial value)
1	Transmission protocol T = 1	

- **SCC ETU Counter Overflow Interrupt Enable (ECIE):** Enables or disables generation of "SCC etu counter overflow" interrupt (ECI). Internal etu counter is designed for counting the character/block wait time and alarming user. The wait time is prepared in SCCECR in unit of 1 etu by software.

Bit 9: ECIE	Description	
0	ECI interrupt is Disabled.	(Initial value)
1	ECI interrupt is Enabled.	

- **Transmit Error Interrupt Enable (ETIE):** Enables or disables generation of transmit-error interrupt (ETI).

Bit 8: ETIE	Description	
0	ETI interrupt is disabled.	(Initial value)
1	ETI interrupt is enabled.	

- **Re-transferring Over 3 Times Interrupt Enable (RETIE):** Enables or disables generation of Re-transferring over-3-times interrupt (RETI).

Bit 7: RETIE	Description	
0	RETI interrupt is disabled.	(Initial value)
1	RETI interrupt is enabled.	

- **Start Character TS End (TSEND):** Specifies whether receiving TS character has ended or not. When TSEND is 0, hardware will not report parity error.

Bit 3: TSEND	Description	
0	Receiving TS character stage has not ended.	
1	Receiving TS character stage has ended.	(Initial value)

Note: Set or clear TSEND flag bit by software. Software should guarantee that TSEND is cleared to 0 before necessary TS character is to be received and that TSEND is promptly set to 1 after the character has been received.

- **Parameter X (PX):** Specifies whether supports “clock stop” mode or not and CLK pin maintains at state high or at state low when supports the mode and enters the mode.

Bit 2-1: PX	Description	
00	SCC does not support clock stop.	(Initial value)
01	Clock line stops at state low.	
10	Clock line stops at state high.	
11	Ignore	

- **Smart Card Clock Stop (CLKSTP):** Software set CLKSTP to start hardware counting 1860 cycles and then stopping SCC_CLK (clearing SCCSR.TRE bit). If SCC_CLK has been stopped, clearing CLKSTP by software will start hardware counting 700 cycles and then setting SCCSR.TRE bit.

Bit 0: CLKSTP	Description	
0	SCC has left or is leaving clock stop mode.	(Initial value)
1	SCC has entered or is entering clock stop mode.	

15.3.3 SCC Status Register (SCCSR)

The SCCSR register is a 16-bit register that is mainly used to show the current states of SCC. SCCSR can be written by software at all times except for TRE bit. SCCSR is initialized to H'8000 by a power-on reset or manual reset.

Bit:	15	14	13	12	11	10	9	8
Read:	TRE						TPER*	
Write:								
Reset:	1	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:				RETR_3*				ECNTO*
Write:								
Reset:	0	0	0	0	0	0	0	0

Note: TPER, RETR_3 and ECNTO can only be written into 0.

Bit 14~10, 8~5, 3~1: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **Transmit and Receive Enable (TRE):** Specifies whether the SCC_CLK is stopped by hardware when executing clock stop function indicated by CLKSTP bit in SCCR. TRE is cleared after 1860 cycles since CLKSTP bit has been set. If software clears CLKSTP when TRE bit has been cleared by hardware, TRE bit will be set to 1 after 700 cycles. SCC can not do data transfer when TRE is cleared.

Bit 15: TRE	Description	
0	In clock stop mode, SCC_CLK is stopped by hardware.	
1	SCC is not in clock stop mode or SCC_CLK has not been stopped by hardware.	(Initial value)

- **Parity Error in Transmission (TPER):** Specifies when SCC in transmission mode (TE = 1), parity error has received from card. Only 0 can be written to this flag. When this bit is 1 and RETR_3 bit is 0, hardware will automatically retransmit the last byte when TP bit is 0.

Bit 9: TPER	Description	
0	Parity error not occurred during transmission.	(Initial value)
1	Parity error occurred during transmission.	

- **Retransfer Over 3 times (RETR_3):** Specifies data have re-transferred over three times with parity error. Only 0 can be written to this flag. During reception or transmission, if RETIE is set to 1, RETI is generated when the bit is set to 1. Hardware re-transmitting will stop when this bit is 1, and will continue when it is cleared by software. When this bit is 1, receiving does not stop.

Bit 4: RETR_3	Description	
0	Data have not re-transferred over 3 times with parity error.	(Initial value)
1	Data have re-transferred for 3 times with parity error.	

Note: Be sure to clear the RETR_3 bit before the next parity bit is sampled.

- **SCC ETU Counter Overflow (ECNTO):** Specifies whether the SCC internal etu counter has overflowed. Overflow here means the counter value is equal to the value in SCCECR. If ECIE is set in SCCR, the etu counter overflow interrupt (ECI) is enabled. Only 0 can be written to this flag.

Bit 0: ECNTO	Description	
0	Overflow not occurred.	(Initial value)
1	Overflow occurred.	

15.3.4 SCC Transmission Factor Register (SCCTFR)

The SCCTFR register is a 16-bit register, which is mainly used to control transmission factor. In the register, bit 15 to 11 are reserved, these bits are always read as 0 and should only be written with 0. The register and UARTBRDR is together used to generate baud rate for SCC. The value of the register (TFR[10:0]) should equal to subtract one from (F/D), that is to say, $SCCTFR = (F/D) - 1$. SCCTFR can be read or written by the CPU at all times. SCCTFR is initialized to H'0173 (D'371) by a power-on reset or manual reset.

The etu (elementary time units: the period for 1 bit to transfer) used on circuit I/O depends upon the actual values of transmission factors F and D (refer to section 6.5.2 in ISO 7816-3 for more information about F and D).

$$1\text{etu} = (F/D) \cdot (1/f)$$

Here:

F: transmission factor;

D: baud rate adjustment factor;

f: clock frequency.

"f" is clock signal that is output on SCC_CLK pin to smart card. The clock is a basic clock with a frequency of (F/D) times the transfer rate, it calculated as follows:

$$f = (F/D) \cdot \text{Baud rate} = \text{PCLK} / [2 * (\text{UARTBRDR} + 1)]$$

Note:

1. UARTBRDR should contain value larger than zero.
2. PCLK is the peripheral clock frequency.

Bit:	15	14	13	12	11	10	9	8
Read:						TFR[10]	TFR[9]	TFR[8]
Write:								
Reset:	0	0	0	0	0	0	0	1

Bit:	7	6	5	4	3	2	1	0
Read:	TFR[7]	TFR[6]	TFR[5]	TFR[4]	TFR[3]	TFR[2]	TFR[1]	TFR[0]
Write:								
Reset:	0	1	1	1	0	0	1	1

15.3.5 SCC Extra Guard Time Register (SCCEGTR)

The SCCEGTR register is an 8-bit register, which is mainly used to control the extra guardtime between two consecutive characters from the SCC to the card. SCC obtains the value of SCCEGTR by parameter N in TC(1) of ATR. If TC(1) is absent, EGTR[7:0] is set to h'00. SCCEGTR can be read or written by the CPU at all times. SCCEGTR is initialized to H'00 by a power-on reset or manual reset.

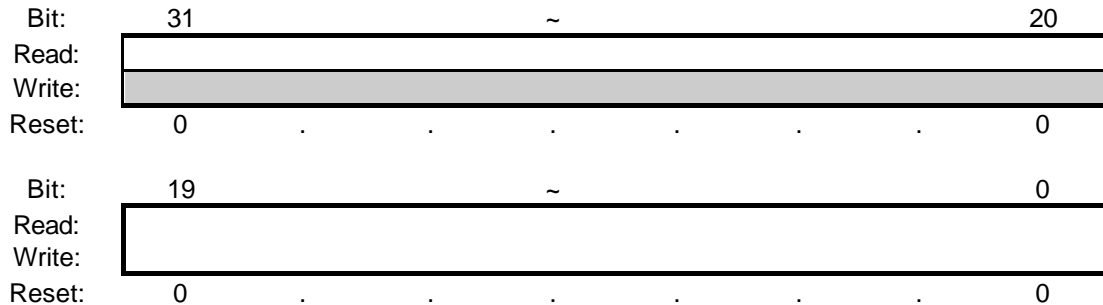
Bit:	7	6	5	4	3	2	1	0
Read:	EGTR	EGTR	EGTR	EGTR	EGTR	EGTR	EGTR	EGTR
Write:	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reset:	0	0	0	0	0	0	0	0

Note: When EGTR is written to h'FF, the minimum delay between the leading edges of two consecutive characters is the same in both directions of transmission. The value is this minimum delay is 12 etu (11 etu) for T = 0 (T = 1).

15.3.6 SCC ETU Counter Value Register (SCCECR)

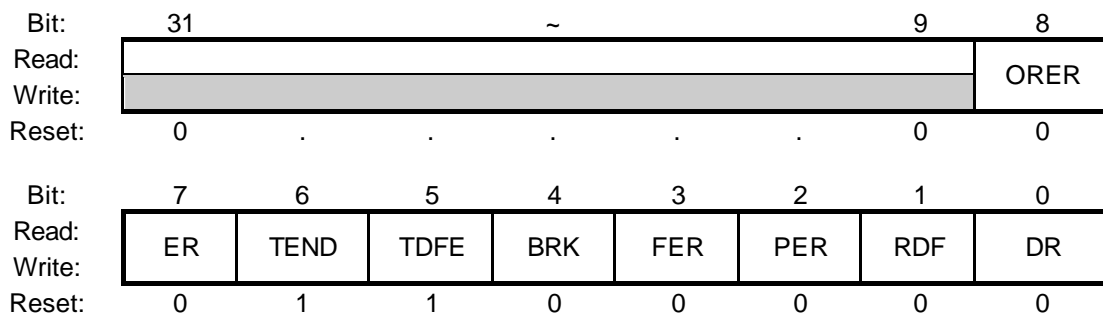
SCCECR is a 32-bit (20 effective bits) register used to set the count time between characters for the internal hardware etu counter. The time is counted in unit of 1 etu.

If the value of SCCECR is not zero, it represents the count target of hardware internal etu counter. Internal etu counter reset itself and starts counting when software writing SCCECR, or clearing SCCSR.ECNT0 bit or when the first edge of start bit of a character (either from card or from SCC) appearing on SCC_DATA pin.



15.3.7 UART Status Register (UARTSR)

In the Smart Card Controller mode, BRK and FER bits have no meaning to SCC. The other bits have the same function as in the ordinary UART. See UART spec, for more information.



15.4 Receiver and Transmitter

15.4.1 Receiver

The receiver accepts serial data stream and converts it into a parallel character. When enabled, it searches for a start bit, qualifies it, and then samples the succeeding data bits at the bit-center. Receiver performs parity check, too. When one valid byte of data has been received, it is transferred to UARTRDR automatically. If a parity error is found, receiver stops writing current data to receive FIFO, drives SCC_DATA pin low from 10.5 etu to 11.5 etu and waits for a repetition of the disputed character when SCC and smart card work under T = 0 mode. When parity error is found, PER bit will be set in both T = 0 and T = 1 mode.

15.4.2 Transmitter

The transmitter accepts a parallel character from the CPU and transmits it serially. The start and parity bits are added to the character. To perform serial data transmission, the SCC first transfers transmit data from UARTRDR to transmitter, then sends the data to the SCC_DATA pin starting with the start bit, and ending with the parity bit. If an error signal is received, SCC sets TPER bit, stops loading new data from transmit FIFO to transmit shift register and retransmit current data in transmitter when SCC and smart card work under T=0 mode.

15.5 Pins Connection

Figure 15-2 shows the pins connection diagram for the Smart Card Controller.

Use on chip's GPIO as the reset signal. The data transfer line is pulled up to the power supply Vcc with a resistor. Apart from these pins, the power and ground pin connections are usually also required.

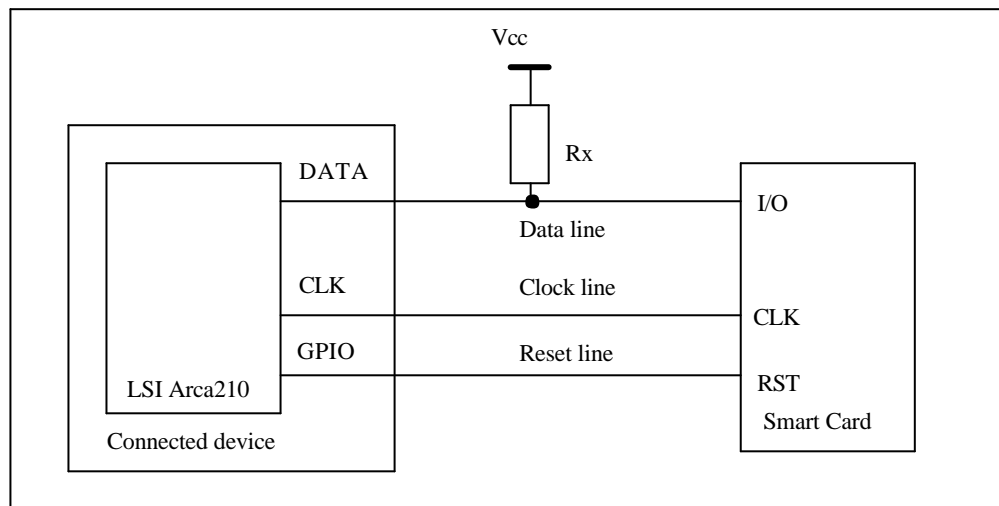


Figure 15-2 Pin Connection Diagram for the Smart Card Controller

15.6 Data Format

The data format for Smart Card Controller are described below:

1. Each frame consists of 8-bit data and 1 parity bit.
2. During transmission, the card leaves a guard time of at least 2 etu (in T=0 mode) or 1 etu (in T=1 mode) from the end of the parity bit to the start of the next frame.
3. During reception, the card outputs an error signal low level for 1 etu after 10.5 etu has elapsed from the start bit if a parity error was detected (in T = 0 mode).
4. During transmission, it automatically transmits the same data after allowing at least 2 etu from the time the error signal is sampled (in T = 0 mode).

Figure 15-3 shows the data format for the Smart Card Controller.

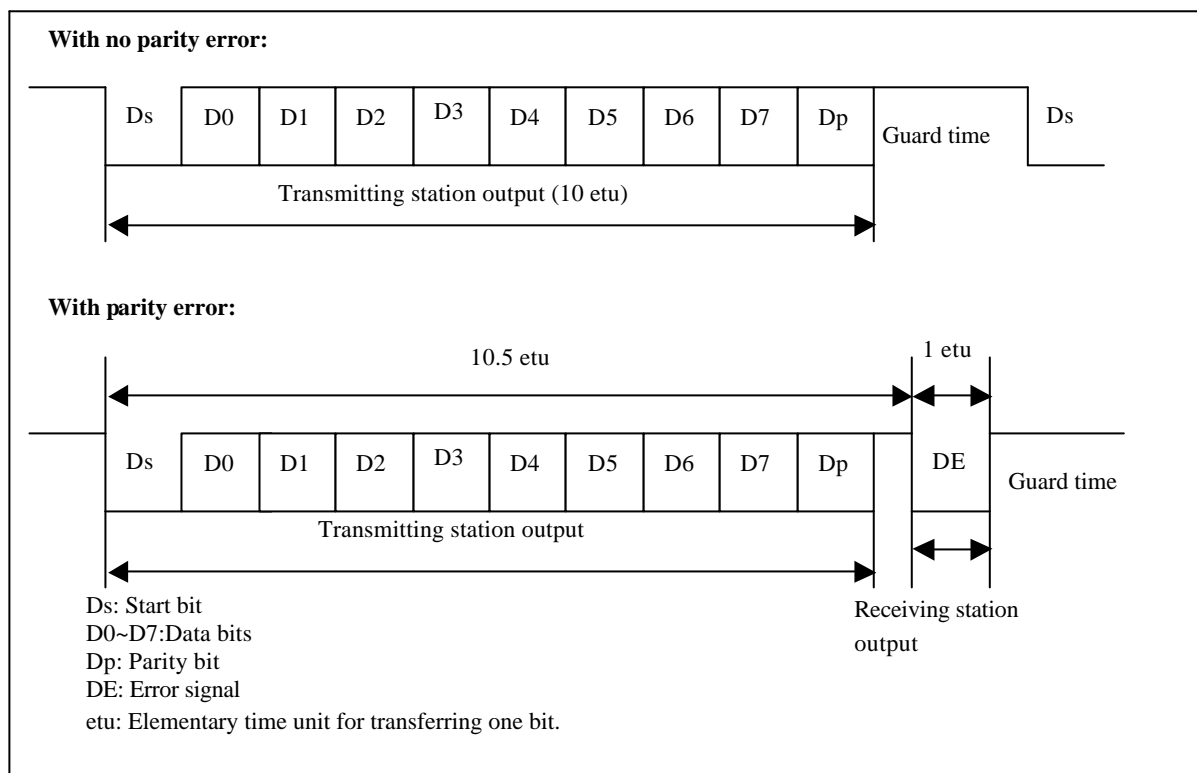


Figure 15-3 Data Format for Smart Card Controller

15.7 Register Related to SCC Setting

Data format for SCC is fixed, so WLEN and PE bits in UARTCR must also be fixed. PROE bit is only set to 0 when CONV is at its initial value (0) and the bit is set according to CONV after TS character has been received. Do not set the TE and RE bits simultaneously unless SCC performing auto-diagnosis. From the start use time of SCC_CLK in card activation to the end use time of SCC_CLK in card deactivation, TE and RE should not be cleared simultaneously because SCC_CLK is enabled by TE or RE.

Figure 15-4 shows sample waveforms for register settings of the two types of smart cards (direct convention and inverse convention) and their start characters. Refer to ISO7816 standards for details.

In the direct convention type, data are not inverted and communication is LSB first. The start character data is H'3B. The parity bit is even (from the smart card standards), and thus 1.

In the inverse convention type, data are inverted and communication is MSB first. The start character data is H'3F. The parity bit is even (from the smart card standards), and thus 0, which corresponds to logical 1 level. Under the condition, only data bits D7~D0 are inverted by the CONV bit. To invert the parity bit, set the PROE bit in UARTCR to odd parity mode. This applies to both transmission and reception.

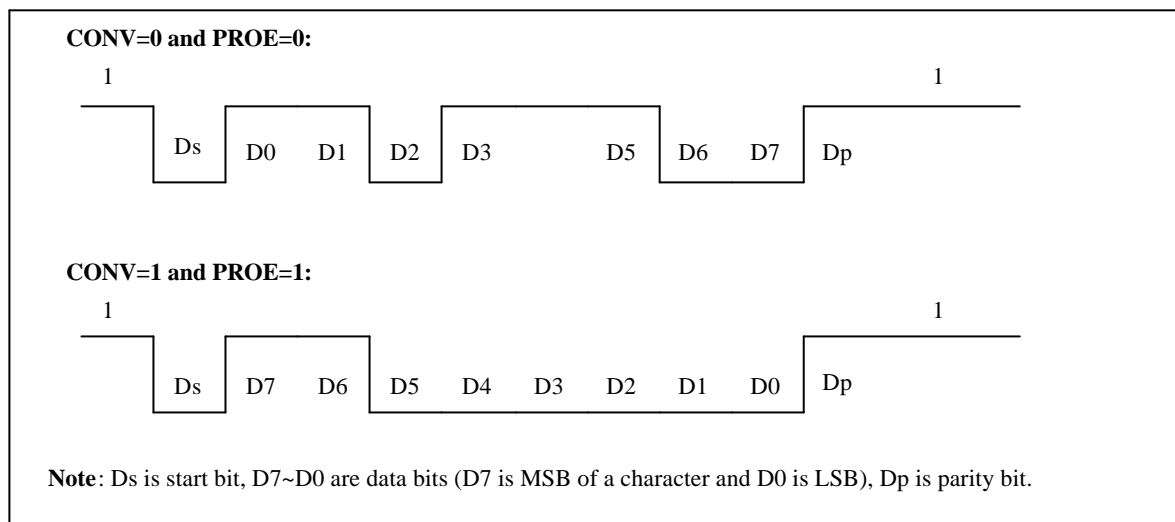


Figure 15-4 Waveform of Start Character

15.8 SCC Operation

15.8.1 Initialization

Initialize the SCC using the following procedure before sending or receiving data. Initialization is also required for switching from transmit mode to receive mode or from receive mode to transmit mode. Figure 15-5 shows a flowchart of the initialization process.

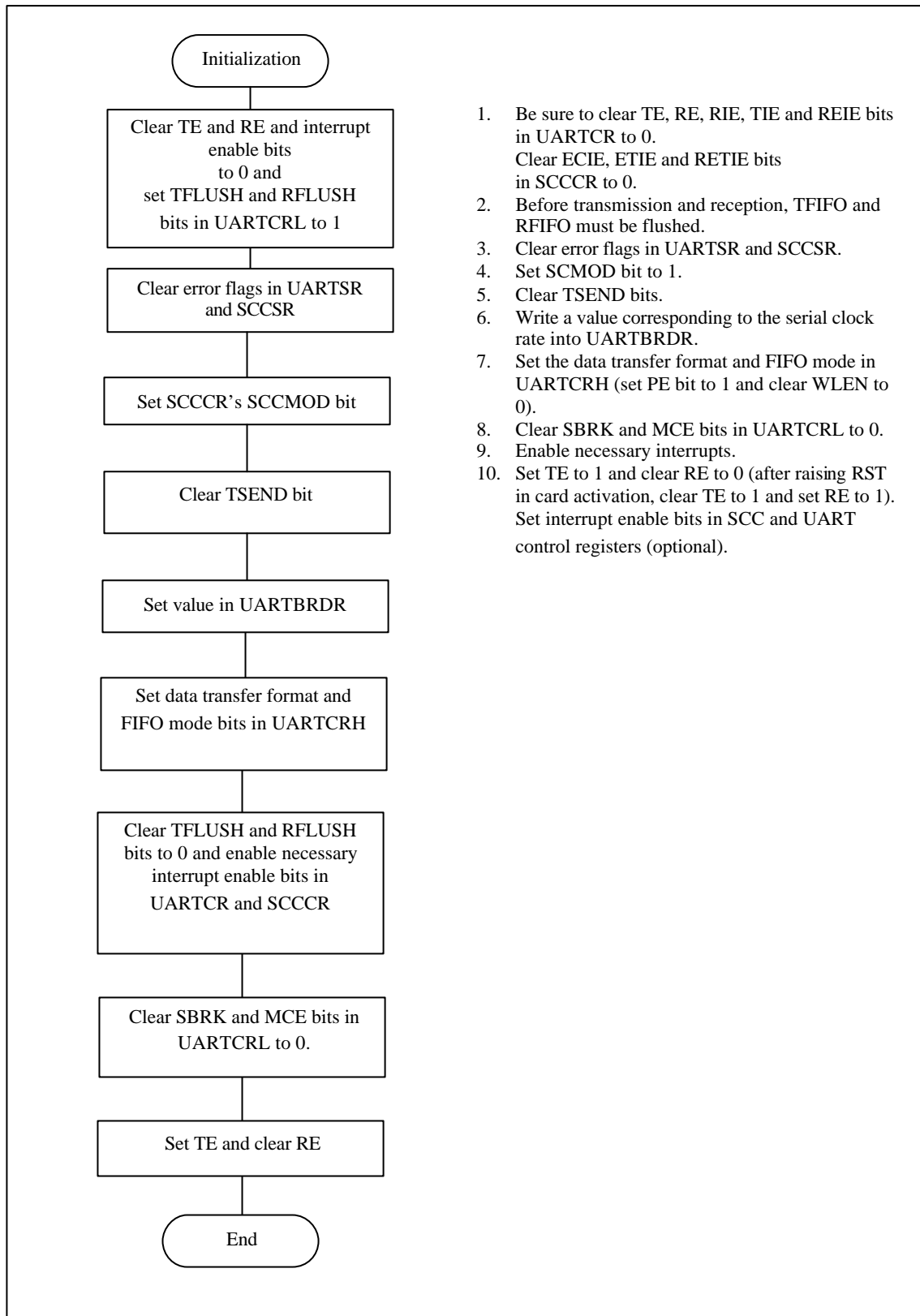


Figure 15-5 Initialization Flowchart

15.8.2 Serial Data Transmission

This transmission process flowchart is shown in Figure 15-6. Assumes that card activation and “answer to reset (ATR)” have finished, TSEND, CONV and PROE have set to proper values.

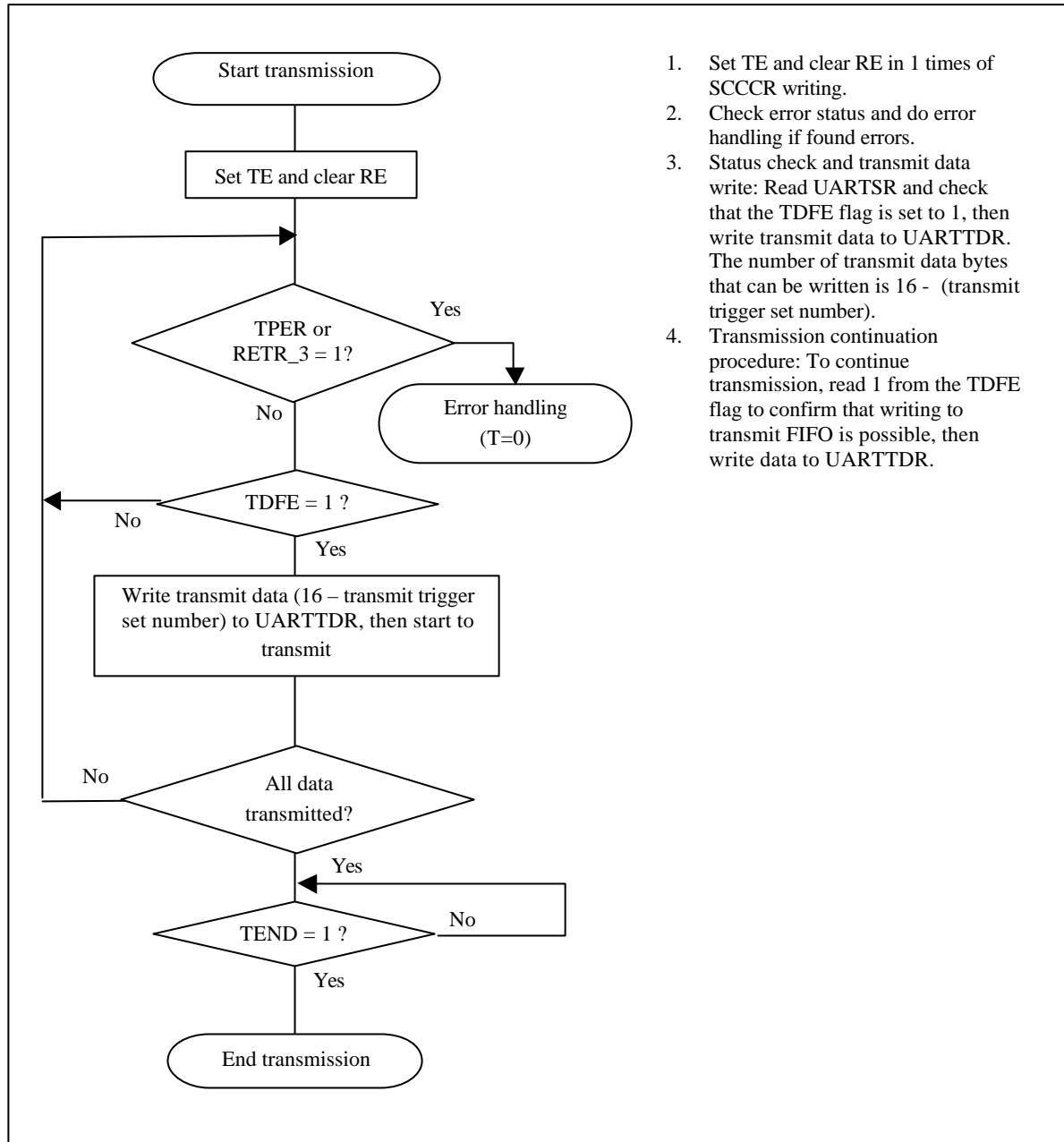


Figure 15-6 Transmission Flowchart

15.8.3 Serial Data Reception

The reception process flowchart is shown in Figure 15-7. Assumes that card activation has finished.

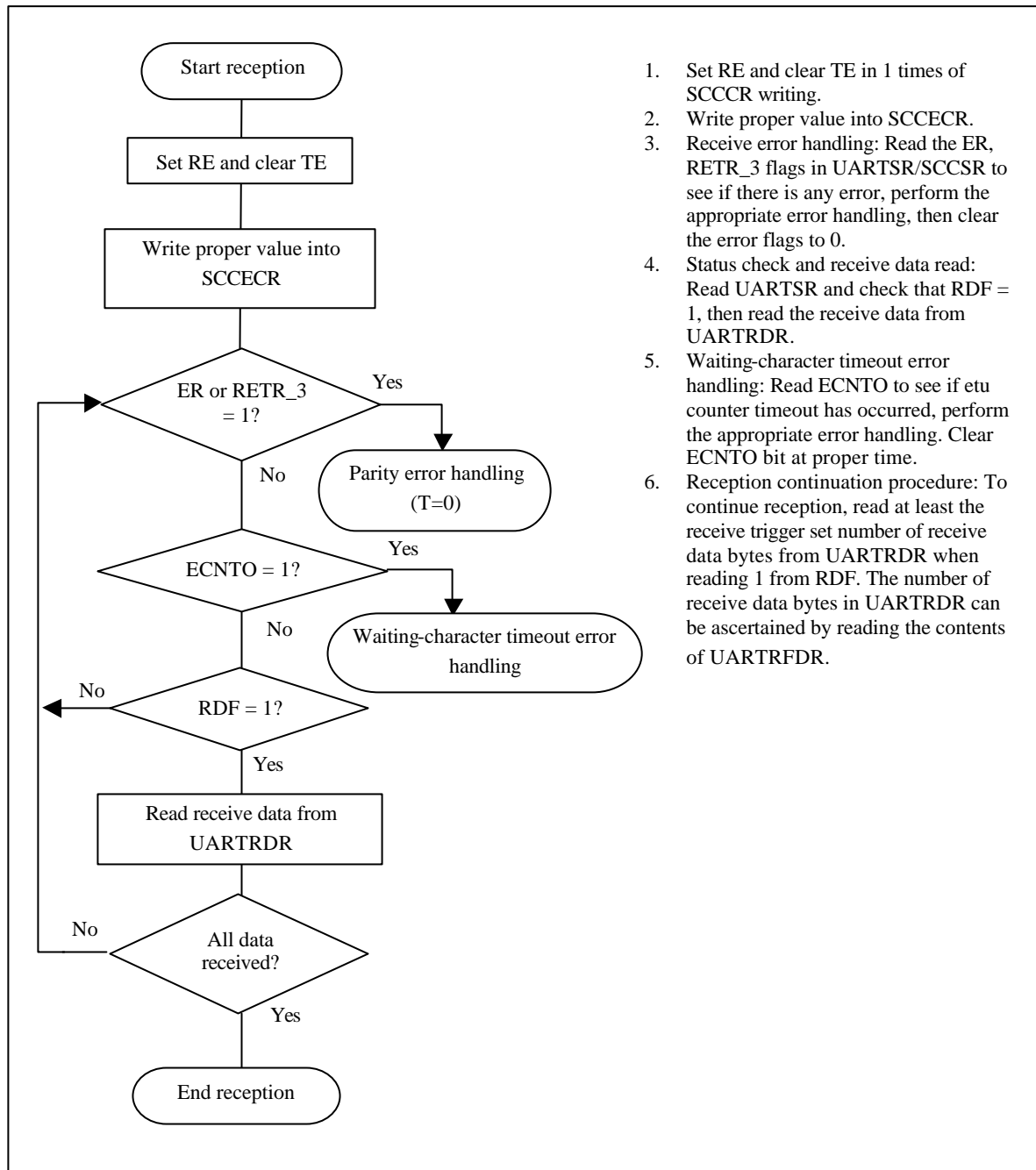


Figure 15-7 Reception Flowchart

15.9 Usage Notes

15.9.1 Interrupt Operation

In the smart card interface mode, in addition to ERI, RXI, RTI and TXI, ETI, RETI and ECI are added (Table 15-3). All these interrupts can be masked by writing 1 into bit 8 of INTC interrupt mask register (IMR).

Table 15-3 Smart Card Mode Operating State and Interrupt Sources

Mode	State	Flag	Register	Mask Bit	Register	Interrupt Source	DMAC Activation
Transmit mode	FIFO empty	TDFE	UARTSR	TIE	UARTCR	TXI	Possible
	Transmit parity error	TPER	SCCSR	ETIE	SCCCR	ETI	Impossible
Receive mode	FIFO full	RDF	UARTSR	RIE	UARTCR	RXI	Possible
	Over time	DR	UARTSR	RTIE	UARTCR	RTI	Possible
	Receive errors	ER, PER, ORER	UARTSR	REIE	UARTCR	ERI	Impossible
Transmit /Receive mode	Re-transfer over 3 times	RETR_3	SCCSR	RETIE	SCCCR	RETI	Impossible
	Counter overflow	ECNTO	SCCSR	ECIE	SCCCR	ECI	Impossible

In the smart card interface mode (SCCMOD is 1), both UARTSR and SCCSR are necessary to be checked by the interrupt handler. The recommended interrupt priority order is ERI > RXI > RTI > ETI > RETI > TXI > ECI.

Fw]

15.9.2 Selection of the Operation Class

Figure 15-8 shows the decision to be made by SCC in selecting the class of operation conditions to be applied to a card. Decisions shown are based upon information implicit in SCC, except where the word “card” is present. In figure, the first operating condition to be applied to the card should be class B. Under class A, a class B card shall not provide an ATR. A delay in the processing is at least 10 ms.

Software should output a signal by a GPIO to power manage unit for providing appropriate voltage.

Some cards conforming to ISO/IEC 7816-3: 1989 could be damaged when operating under class B conditions and should be used only in class A interface devices.

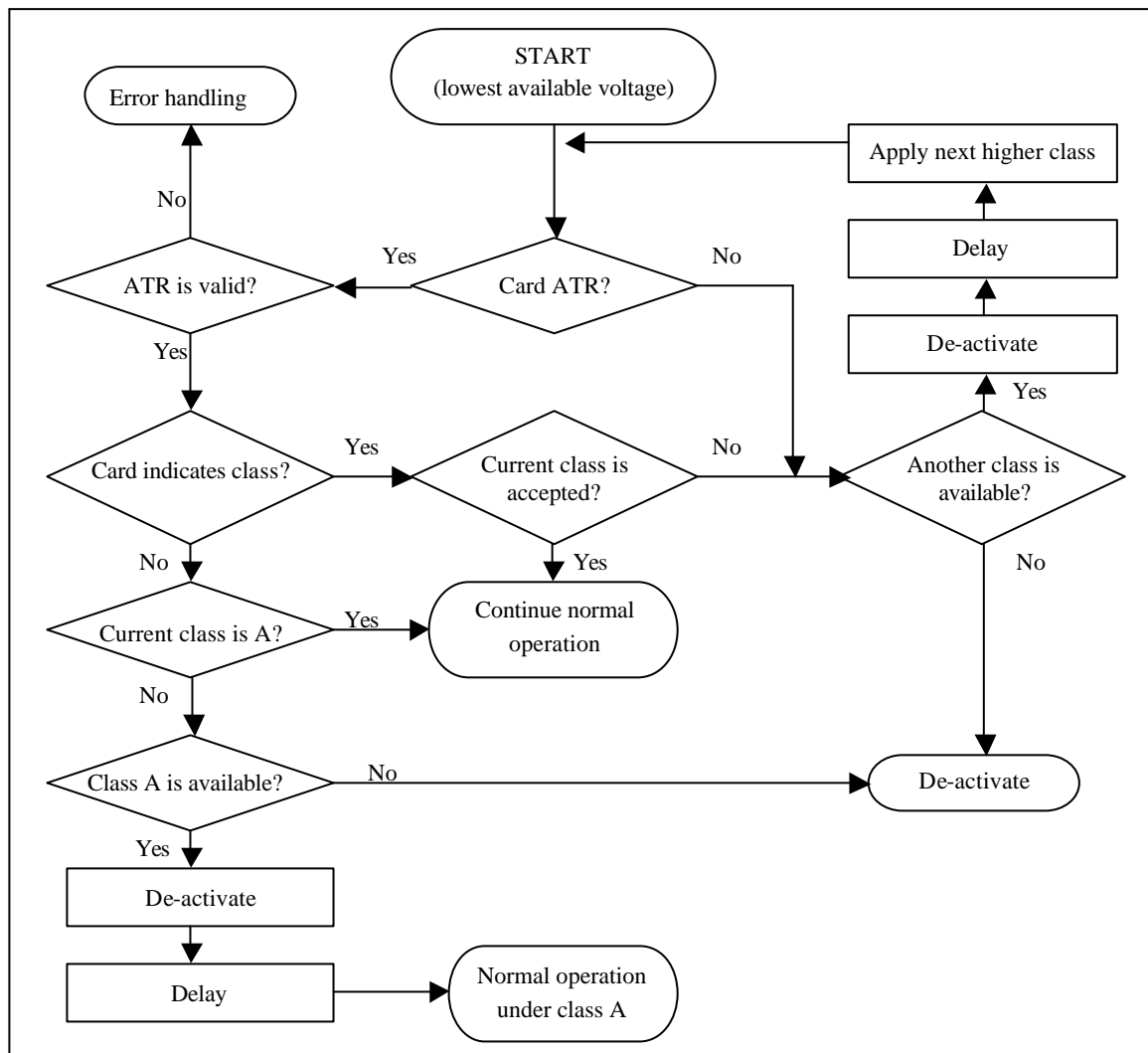


Figure 15-8 Selection of the Class of Operating Conditions by SCC

15.9.3 Reset operation under asynchronous mode

15.9.3.1 Cold Reset

Figure 15-9 shows processing of a cold reset. The clock signal is valid at time T_a . The card should set the I/O line to reception state within t_a being applied to CLK. The card is reset by maintaining RST at logic L level for at least 400 cycles after the clock signal is valid. At time T_b , RST is put to state H. The answer to reset on I/O shall begin between 400 and 40000 cycles after the rising edge of the signal on RST. If the answer is not received within 40000 cycles with RST at state H, the signal on RST should be return to state L and the electrical circuits should be deactivated by SCC.

SCC may initiate a cold reset of the card at its discretion at any time.

Note: TE bit in UARTCR should be set during time “tb” to enable clock output from SCC.

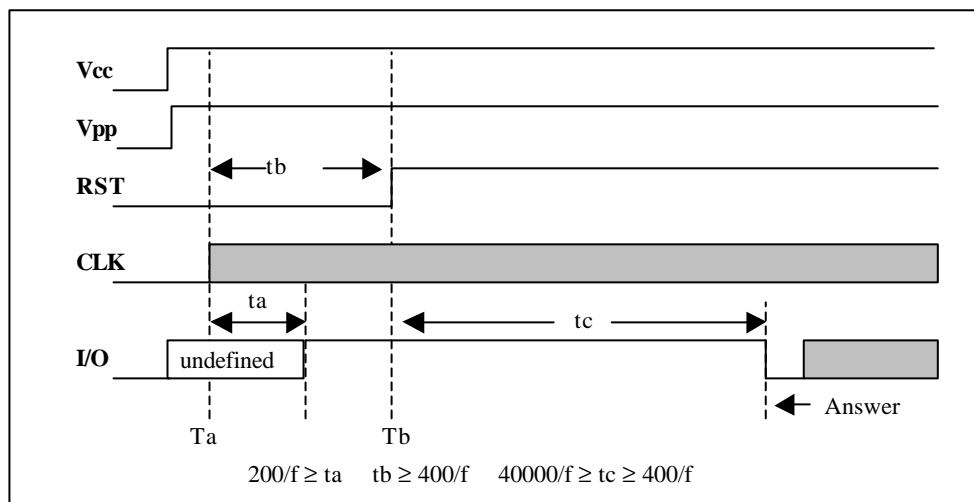


Figure 15-9 Cold Reset

15.9.3.2 Warm Reset

Figure 15-10 shows processing of a warm reset. SCC initiates a warm reset by putting RST to state L for at least 400 cycles when Vcc and CLK remain stable. At time T_d , RST is put to state H. The answer to reset should begin between 400 and 40000 cycles after the rising edge of the signal on RST. If answer is not received within 40000 cycles with RST at state H, the signal on RST should be returned to state L and the electrical circuits shall be deactivated by SCC.

Note: TE bit in UARTCR should be set during time “te” to enable clock output from SCC.

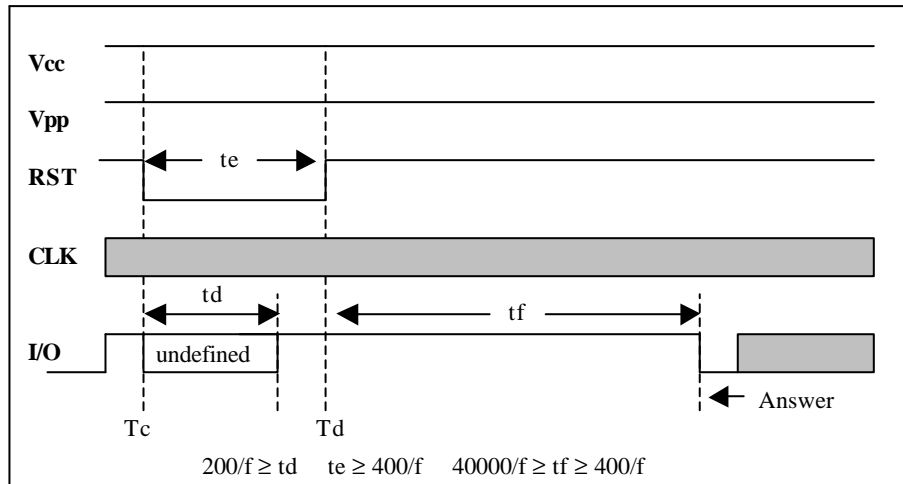


Figure 15-10 Warm Reset

15.9.3.3 Clock Stop

For cards supporting clock stop, when SCC expects no transmission from the card and when I/O has remained at state Z for at least 1860 cycles, then according to Figure 15-11, SCC may stop the clock on CLK. When the clock is stopped, CLK shall be maintained either at state H or L, the state is indicated by PX bit flag in SCCR. At time T_f , SCC restarts the clock and the information exchange on I/O may continue after at least 700 clock cycles.

Note: if both TE and RE bits are cleared to 0, SCC will not output its clock.

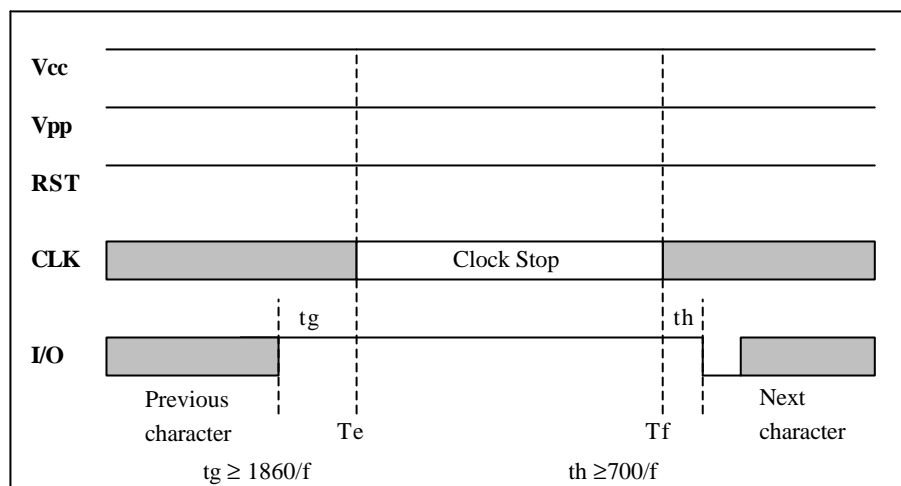


Figure 15-11 Clock Stop

15.9.3.4 Receive Data Timing

During reception, the SCC samples the falling of the start bit using the base clock to achieve internal synchronization. When SCC runs on a basic clock with a frequency of F/D times the transfer rate, receive data is latched internally on the rising edge of the No. $(F/D/2)$ basic clock cycle (Figure 15-12).

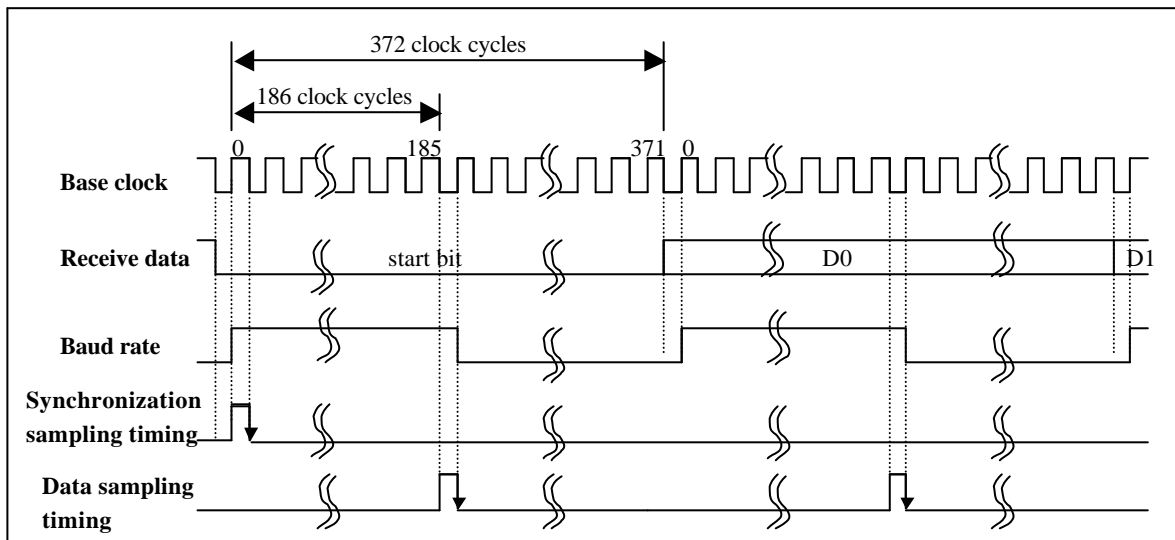


Figure 15-12 Receive Data Sampling Timing under Asynchronous Mode

15.9.3.5 Retransmission (Receive and Transmit Modes under T = 0)

Retransmission by the SCC in Receive Mode: Figure 15-13 shows the retransmission operation in the SCC receive mode.

1. The received parity bit is checked and an error is found, the PER bit in UARTSR is automatically set to 1. At this moment, receive data in receiver can not be written in UARTRDR. Be sure to clear the PER bit before the next parity bit is sampled.
2. When the received parity bit is checked and no error is found, the PER bit in UARTSR is not set, receive character in receiver is written in UARTRDR automatically.

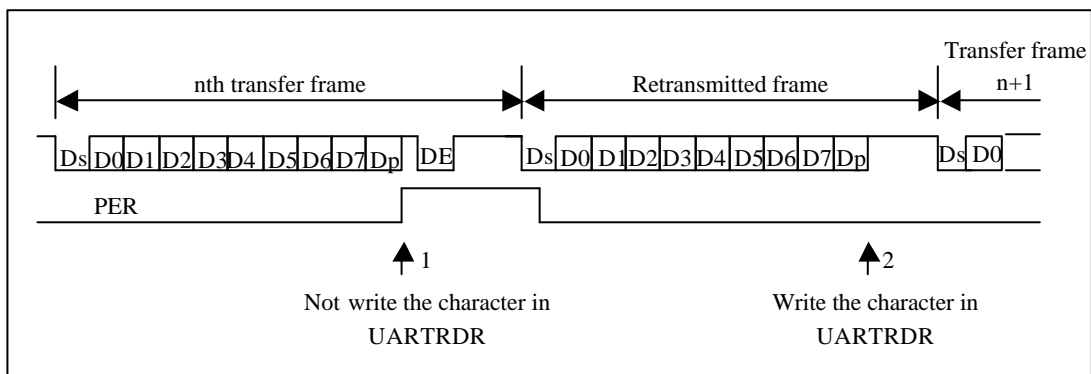


Figure 15-13 Retransmission in SCC Receive Mode

Retransmission by the SCC in Transmit Mode: Figure 15-14 shows the retransmission operation in the SCC transmit mode.

1. After transmission of one frame is completed, the TPER bit in SCCSR is set to 1 when an error signal is returned from the receiving side. At this moment, transmitter can not load data from UARTRDR and retransmit current data in transmitter. Be sure to clear the TPER bit before the next parity bit is sampled.
2. The TPER bit in SCCSR is not set to 1 when no error signal is returned from the receiving side, transmitter loads data from UARTRDR and transmit to SCC_DATA pin, automatically.

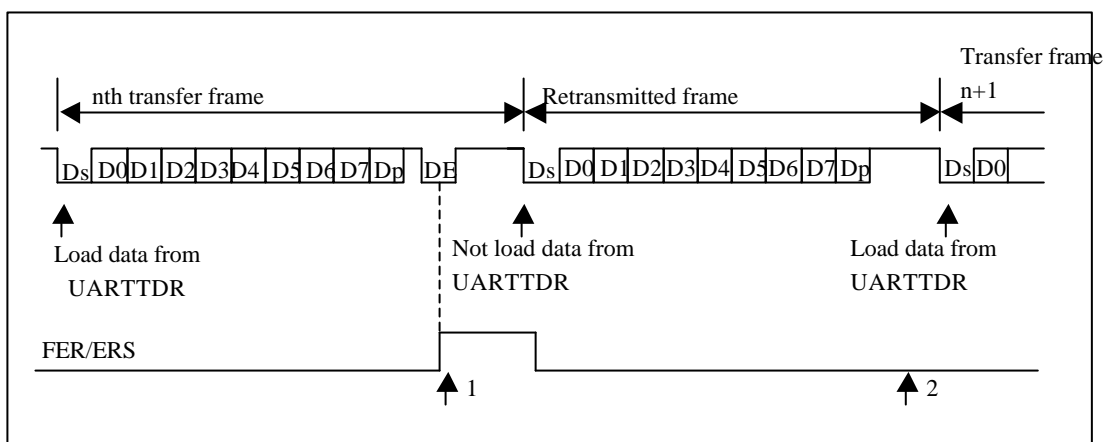


Figure 15-14 Retransmission in SCC Transmit Mode

16 Infrared Data Association (IrDA)

16.1 Overview

This LSI has an on-chip infrared data association (IrDA) interface that is based on the IrDA 1.0 specification and can perform infrared communication, the IrDA module is designed to work with the UART2. The TxD2 and RxD2 pins are multiplex by IrDA and UART2.

The UART2 is the second channel of UART module that performs the same function as UART1 except in UART2 the modem control function and RTS and CTS pins are provided. So the MCE bit in UARTCR2 is added. Another difference between UART1 and UART2 is the LOOP bit in UARTCR2 can set the loopback mode of IrDA if IrDA is enabled.

16.1.1 Features

IrDA features:

- Based on the IrDA 1.0 specification
- Asynchronous infrared serial communication
 - data length: 8 bits
 - stop bit length: 1 bit
 - parity bit: none
- A straightforward extension of UART2: When IrDA is disabled, UART2 can work as a normal UART.
- Polarity of transmitted and received signals selectable
- When transmitting, support normal 3/16 and IrDA low-power mode bit duration
(Here, the “low-power” refers to modulated ‘0’ value bit has a fixed pulse width which is 3/16 of a 115.2Kbps bit duration. This fixed pulse width can lower the power consumption of the optical transceiver out of the chip. So “low-power” in this section has different sense from the “system low power” mode.)
- When receiving, normal 3/16 and low-power mode bit duration signal will both be received as low-power mode bit duration.
- Loopback mode provides a convenient diagnostic method.

16.1.2 Block Diagram

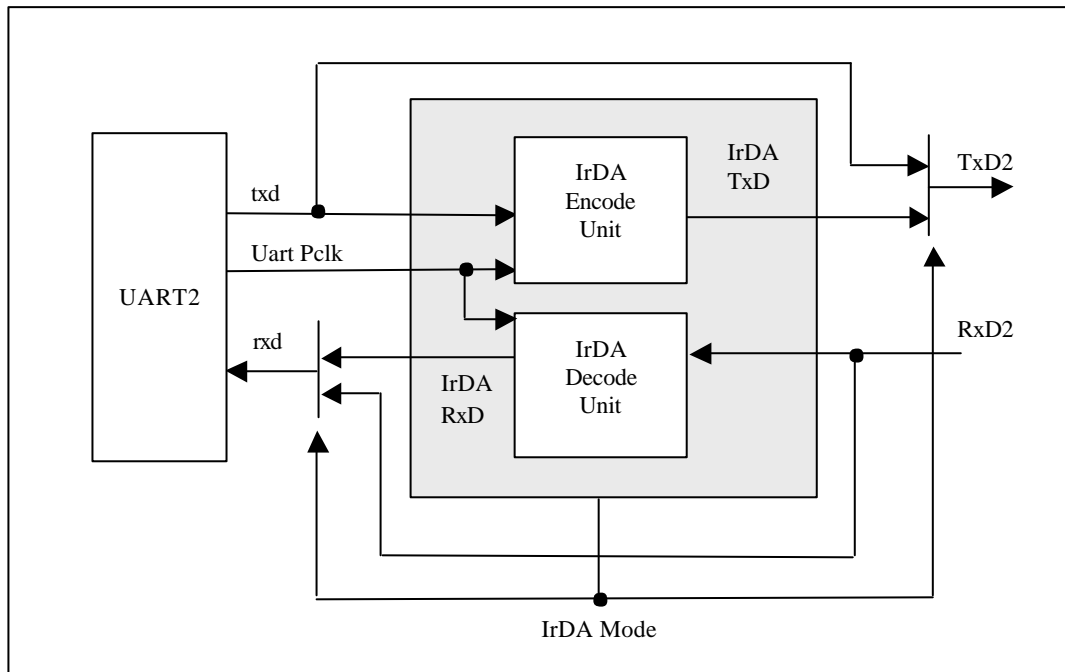


Figure 16-1 IrDA Block Diagram

16.2 Pin configuration

Table 16-1 IRDA Pins

Name	Full Name	I/O	Function
RxD2	Receive data pin	Input	Receive data input
TxD2	Transmit data pin	Output	Transmit data output

Note: The above pins are disabled for IrDA functions if any of them is used as GPIO.

16.2.1 RxD2 Pin

This is serial data input pin to IrDA. When it is not in use (drived), it retains high level.

16.2.2 TxD2 Pin

This is serial data output pin from IrDA. When it is not in use (drived), it retains high level.

16.3 Registers Configuration

Table 16-2 UART2 and IrDA Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
UARTBRDR2	UART2 Bit Rare Divisor Register	R/W	H'0000	H' E0000700	16
UARTCR2	UART2 Control Register	R/W	H'00000000	H' E0000704	32
UARTCRL2	UART2 Control Register L	R/W	H'0000	H' E0000704	16
UARTCRH2	UART2 Control Register H	R/W	H'0000	H' E0000706	16
UARTSR2	UART2 Status Register	R/(W)	H'00000060	H' E0000708	32
UARTRDR2	UART2 Receive FIFO Data Register	R	Undefined	H' E000070C	8
UARTTDR2	UART2 Transmit FIFO Data Register	W	Undefined	H' E000070D	8
UARTRFDR2	UART2 Receive FIFO Data Count Register	R	H'00	H' E000070E	8
UARTTFDR2	UART2 Transmit FIFO Data Count Register	R	H'00	H' E000070F	8
IRDACR	IrDA Control Register	R/W	H'00000000	H' E0000710	32
IRDACRL	IrDA Control Register L	R/W	H'0000	H' E0000710	16
IRDACRH	IrDA Control Register H	R/W	H'0000	H' E0000712	16

16.3.1 UART2 Register

(UARTBRDR2, UARTCR2, UARTCRL2, UARTCRH2, UARTSR2, UARTRDR2, UARTTDR2, UARTRFDR2, UARTTFDR2)

Since the IrDA module is a straightforward extension of UART2, all the UART2 registers are useful to implement the corresponding function. The registers and operation of UART2 are the same as those described in UART section. Please refer to them for details.

The UART2 operates in the same way no matter whether the IrDA is enabled or not and when the IrDA is disabled the UART2 can work independently as a normal UART.

16.3.2 Control Register (IRDACR)

The IRDACR register is a 32-bit register which is comprised of two 16-bit linked registers (IRDACRH and IRDACRL), so it can be accessed by halfword (IRDACRH and IRDACRL) or word (IRDACR). IRDACRH is mapped on bit 31 to 16 of IRDACR, and IRDACRL is mapped on bit 15 to 0 of IRDACR. IRDACR can be read or written by the CPU at all times. IRDACR is initialized to H'00000000 by a power-on reset or manual reset.

IRDACRH is mainly used as operation mode and control register and IRDACRL is used as a divisor to generate a fixed clock of 1.8432MHz (16 times of 115.2Kbps frequency).

Bit:	31	30	29	28	27	26	25	24
Read:	IRMOD	TXP	RXP	PSEL				
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	FCK15	FCK14	FCK13	FCK12	FCK11	FCK10	FCK9	FCK8
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	FCK7	FCK6	FCK5	FCK4	FCK3	FCK2	FCK1	FCK0
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit 27~16: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **IrDA mode register (IRMOD):** Select whether the IrDA module will be used or not.
Note: When IRMOD is set to 1, bits 19~16 in UARTCR2 should be cleared to 0, UART2 must operate in half-duplex manner unless LOOP bit in UARTCR2 is set to 1.

Bit31: IRMOD	Description	
0	IrDA module does not work.	(Initial value)
1	IrDA module works.	

- **IrDA transmitted signal polarity (TXP):** To determine the polarity of transmitted signal.

This bit should be set to match the signal polarity of optical transceiver's requirement. For details, please refer to the manufacturer's user guide.

Bit30: TXP	Description	
0	An active high pulse is decoded as '0'.	(Initial value)
1	An active low pulse is decoded as '0'.	

Note: When TXP=0, data output from the UART is inverted first before being output to the TxD2 pin. When TXP=1, data output from the UART2 to the TxD2 pin is noninverted.

- **IrDA received signal polarity (RXP):** To determine the polarity of received signal.

This bit should be set to match the signal polarity of optical transceiver's requirement. For details, please refer to the manufacturer's user guide.

Bit29: RXP	Description	
0	An active low pulse is decoded as '0'.	(Initial value)
1	An active high pulse is decoded as '0'.	

Note: When RXP=1, data input from the RxD2 pin is first inverted before being sent to UART2. When RXP=0, data input from the RxD2 pin is not inverted before being sent to UART2.

- **Output pulse width select (PSEL):** This bit selects the output '0' pulse width of IrDA that is 3/16 of bit length for 115.2Kbps or 3/16 of a bit length for selected baud rate. In other words, this bit set the low-power mode for transmitting.

Bit28: PSEL	Description	
0	Transmitted '0' pulse width is 3/16 of a bit length.	(Initial value)
1	Transmitted '0' pulse width is 3/16 of 115.2kbps bit length	

- **IrDA Fixed Clock Divisor Register (FCK15~FCK0):** This register is set to generate a fixed clock that is 1.8432MHz when IRMOD bit is set to 1. The 1.8432M is 16 times of 115.2K. This clock is used for capacitating the IrDA to receive '0' value data with 3/16 of a bit duration at any bit rate (including 115.2Kbps low-power mode), and it is also used when IrDA module is transmitting in 115.2Kbps. In receive procedure, IrDA regards a valid modulated '0' pulse only when it sampled 3/16 of 8.68 μ s '0' pulse. Thus, the fixed clock divisor register should be set proper value in receiving at any rate.

Note: Here the "low-power" refers to a fixed pulse width which is 3/16 of a 115.2Kbps bit duration.

The value N determined by FCK15~FCK0 is calculated as follows:

$$N = \text{PCLK} / 1.8432\text{MHz} - 1$$

Here, PCLK is the same that is described in UART section. For example, if the PCLK is 3.6864MHz the value of N is 1. When the value is 0 no clock is generated, and the IrDA will not work.

Bit15~0: FCK15~FCK0	Description	
H'0000	No fixed clock is generated for transmission or reception.	(Initial value)
$N = \text{PCLK} / 1.8432\text{MHz} - 1$	A fixed clock of 1.8432MHz is generated for transmission or reception.	

16.4 IrDA Operation

16.4.1 Overview

According to the IrDA 1.0, data rate is limited to a maximum value of 115.2Kbps. The modulation/demodulation scheme is 3/16 RZI (Return-to-Zero-Inverted). That is a "0" is represented by an optical light pulse and the pulse duration is nominally 3/16 of a bit duration (or 3/16 of a 115.2 kb/s bit duration in IrDA low-power mode).

The IrDA module performs the infrared communication conforming to IrDA 1.0 specification by connecting the UART2 and an external optical transceiver unit. In this module, the UART waveform is modulated to IrDA waveform when transmitting and the IrDA waveform is demodulated to UART waveform when receiving. After finished setting the mode, format, bit rate and frequency divisor, enable the TE or RE in UARTCR2 can start the IrDA to work.

When IrDA is enabled, All the other features of UART2 are the same as UART1 except these (refer to the UART section 1.1):

- Full-duplex operation which enables transmission and reception to be performed simultaneously.
- Selectable 7or 8 bit-length, even or odd parity check and 1 or 2 stop bits.
- Modem control functions (MCE bit controls the RTS and CTS).

The IrDA module can operate only in half-duplex manner except loopback mode because if it operates full-duplex the received signal will become a mixture of signals transmitted by both nodes because the light interfere each other. The mixed signal will make the computer reach a wrong judgment. So the appropriate register in UART2 should be set properly by software, these two bits are TE and RE in UARTCR2, they must not be both 1 at the same time, if it occurs the IrDA will not work.

The IrDA 1.0 specifies a minimum 10ms latency after an optical node ceases transmitting before its receiver recovers its receiving function and this delay must be implemented by software.

In the IrDA 1.0 specification, communication must start up at the rate of 9600bps, but then allows the link to negotiate higher (or lower) data rates if supported by both ends. However, the communication rate will not automatically change. Change, if necessary, is performed by software.

The polarity of transmit/receive signal is selectable to satisfy different external optical transceiver. The default polarity is active high when transmitting and active low when receiving (refer to register description).

16.4.2 Transmit

As for the serial output signal (UART frame) from the UART2 module, its waveforms is modulated to the IrDA waveform by the IrDA module, as shown in Figure 16-2.

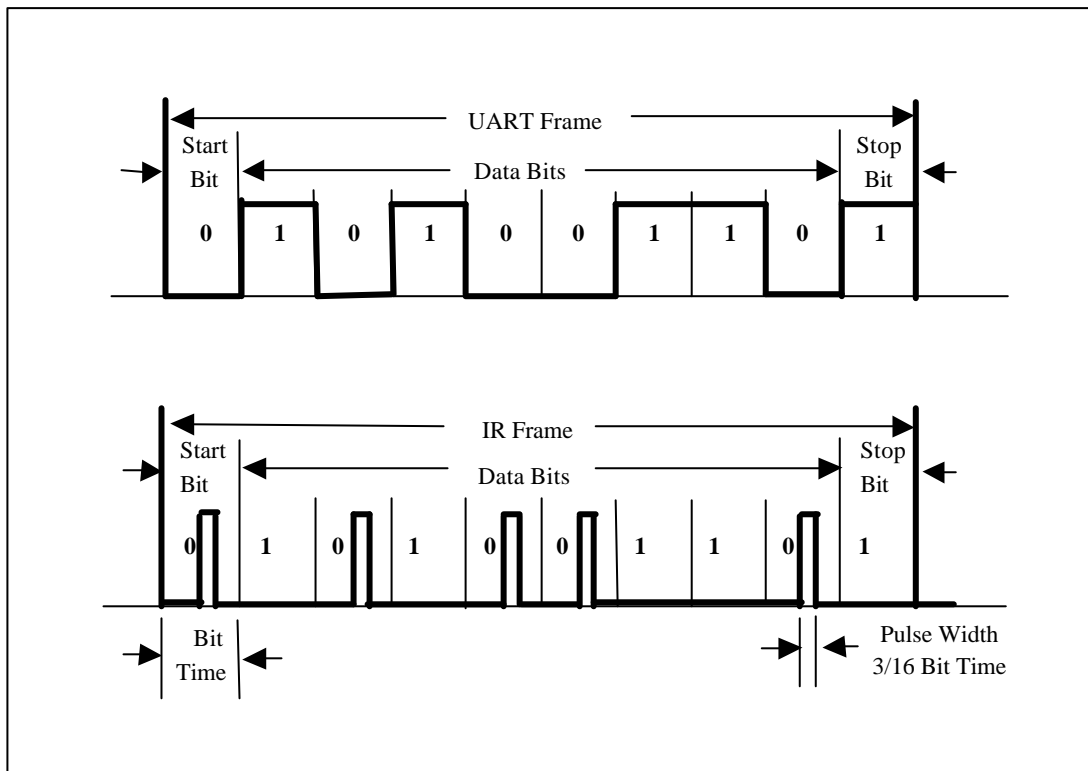


Figure 16-2 UART and IrDA Signal Waveform

When serial data is 0 and transmit signal polarity is active high, the 3/16-bit width pulse is generated and output. When serial data is 1, no pulse is output. When the polarity is active low the waveform is the inverse of active high.

The transmitted pulse width is selectable by PSEL bit in IRDACR and it can be 3/16 of a bit length or 3/16 of 115.2Kbps bit length (nominal 1.62us).

16.4.3 Receive

The received signal of the IR frame is demodulated to the UART frame, as shown in Figure 16-2. The pulse width may be 3/16 of a bit length or 3/16 of 115.2Kbps bit length that is called low-power mode. Regardless the received pulse width is in normal or low-power mode, a start bit is deemed valid if the pulse level is still low, one cycle of 1.8432Mbps (1/16 of 115.2Kbps bit length) after the low was first detected. This allows a normal-mode IrDA to receive data of low-power mode pulse.

Since the received pulse is asynchronous, the pulse edge is the only detectable factor when demodulating. Therefore, the pulse was transmitted at what position of a bit duration is not cared when it is received.

An active high or low pulse is demodulation to 0, and no pulse is demodulation to 1. The polarity of received pulse is selectable by register to match the optical transceiver's specification. The default is active low.

16.4.4 Loopback Mode

LOOP bit in UARTCR2 register sets the loopback mode. When IrDA operates in loopback mode the TxD2 pin is internally connected to the RxD2 pin. This provides a convenient diagnostic method for this module. When IrDA operates in loopback mode, the transmitted and received signal polarities should be the same (both are active high or active low).

When IrDA module is enabled, the loopback test will pass IrDA and UART2, when IrDA module is disabled, the loopback test will pass UART2 only.

16.5 Usage Notes

The bit error ratio caused by inaccurate programmed baud rate derived from pclk must meet the tolerance in the IrDA standard specification. For this reason, some baud rates are not supported by particular frequencies of pclk.

17 I²C-Bus Interface (I2CI)

17.1 Overview

The Inter-Integrated Circuit (I²C) Interface (I²CI) is used for transmit data between Ccore and other I²C slave devices, such as EEPROMs. I²C-bus is a bi-direction, 2-wire bus. It uses two signals — serial data (SDA) and serial clock (SCL) — to transmit data between devices connected to it. This document describes the I²C-bus protocol and the I²C-bus interface.

17.1.1 Features

I2CI features:

- The I2CI supports only single master mode.
- Support of I²C standard-mode and F/S-mode up to 400 kHz.
- I2CI receiver and transmitter are double-buffered.
- Support burst reading or writing of data.
- Support random writing access of data.
- Support general call address and START byte format after START condition.
- Independent, programmable serial clock generator.
- Support slave coping with fast master during data transfers by holding the SCL line on a bit level.
- The number of devices that you can connect to the same I2C-bus is limited only by the maximum bus capacitance of 400pF.

17.1.2 Block Diagram

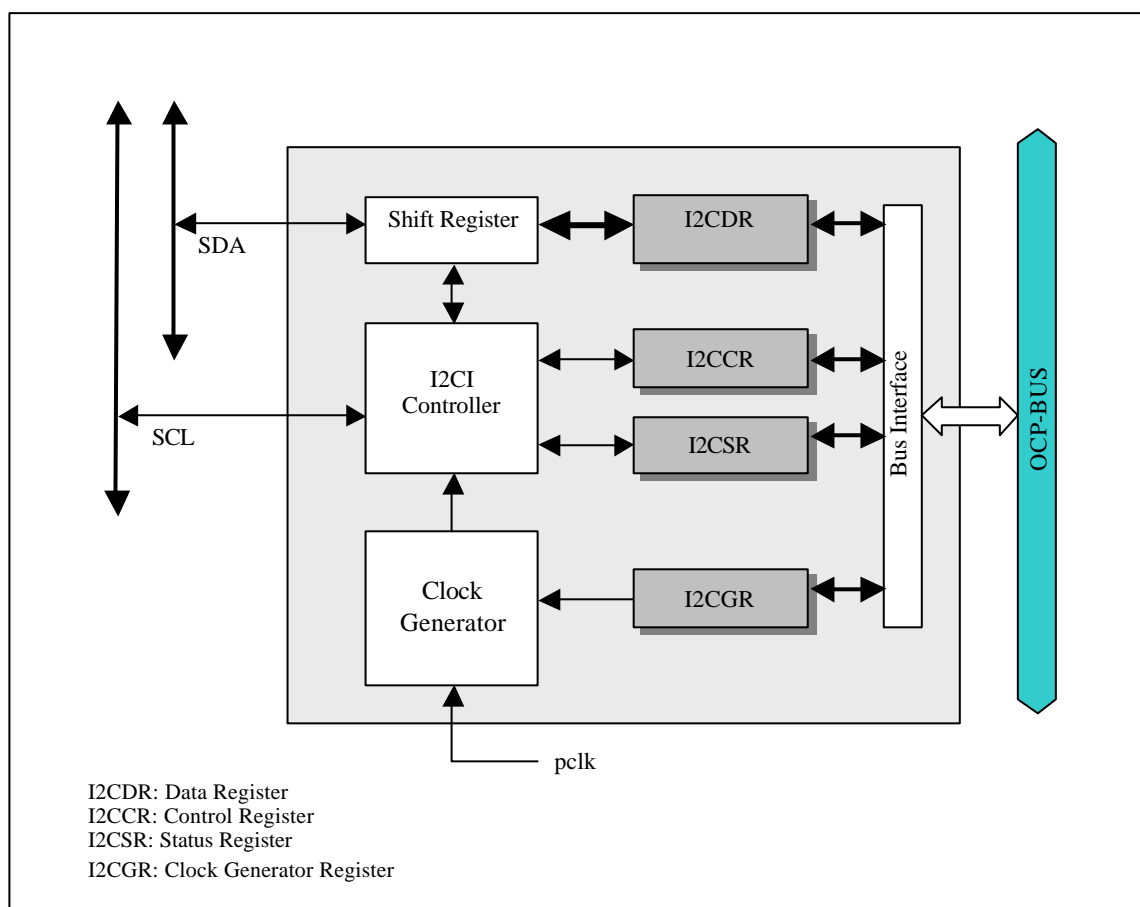


Figure 17-1 I2CI Block Diagram

17.2 Pin configuration

Table 17-1 I2CI Pins

Name	Full Name	I/O	Function
SDA	Serial data pin	I/O	Serial data between I2CI and slave
SCL	Serial clock pin	I/O	Serial clock between I2CI and slave

17.2.1 SDA Pin

This pin is connected with outer I²C data bus. When data bus is idle, it retains high level.

17.2.2 SCL Pin

This pin is connected with outer I²C clock bus. When clock bus is idle, it retains high level.

17.3 Registers Configuration

Table 17-2 I2CI Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
I2CDR	Data Register	R/W	H'00	H' E0000800	8
I2CCR	Control Register	R/W	H'00	H' E0000804	8
I2CSR	Status Register	R/W	H'04	H' E0000808	8
I2CGR	Clock Generator Register	R/W	H'0000	H' E000080C	16

17.3.1 Data Register (I2CDR)

I2CDR is an 8-bit read/write register that is used as a data buffer in transmit/receive operation. In transmit (master write) operation, data in I2CDR is copied to an internal shift register and then transmitted to SDA line bit-by-bit. In receive (master read) operation, data is copied from the internal shift register to I2CDR after one byte is received. This register is not initialized by any reset.

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	—	—	—	—	—	—	—	—

17.3.2 Control Register (I2CCR)

I2CCR is an 8-bit read/write register which enables the I2CI and I2CI interrupt, controls the START/STOP condition and the voltage level of the acknowledge bit. It is initialized to H'00 by power-on reset or manual reset.

Bit:	7	6	5	4	3	2	1	0
Read:				IEN	STA*	STO*	AC	I2CIE
Write:								
Reset:	0	0	0	0	0	0	0	0

Note: STA and STO can only be written with 1.

Bit 7~5: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **Interrupt Enable (IEN):** Enable I2CI interrupt. When this bit is set, I2CI interrupt will occur when register I2CSR bit DRF = 0 in transmission or when DRF = 1 in reception. A transmission is stopped when BUSY is 0 but a reception will not terminate in I2CI until DRF and BUSY are both cleared. During the time just after a START condition has been sent out, if DRF is 0, I2CI interrupt will also be generated. After sending the address byte, if the hardware finds that current transaction is a master read, the I2CI interrupt due to address byte transmission will be stopped and new interrupt will be issued if DRF is 1 in reception.

Bit 4: IEN	Description	
0	Disable I2CI interrupt.	(Initial value)
1	Enable I2CI interrupt.	

- **START (STA):** Writing 1 to this bit can initiate hardware sending a new START condition. Hardware will clear this bit to 0 automatically after 1 just has been written into. This bit is always read as 0. Write 0 to this bit has no effect.

If a STA bit is written into 1 during the transferring of an address byte or a data byte, hardware will guarantee that the STOP condition will be sent out after that byte and the following acknowledge pulse complete.

Writing a 1 into STA bit can break any type of holding SCL line in LOW level by master.

Bit 3: STA	Description	
0	START condition will not be sent to I ² C bus.	(Initial value)
1	START condition will be sent to I ² C bus.	

- **STOP (STO):** Writing 1 to this bit can initiate hardware sending a new STOP condition. Hardware will clear this bit to 0 automatically after 1 just has been written into. This bit is always read as 0. Write 0 to this bit has no effect.

Only when I2C bus is in busy status (BUSY bit in I2CSR register is 1), can the STO bit be written into I2CCR. If a STO bit is written into 1 during the transferring of an address byte or a data byte, hardware will guarantee that the STOP condition will be sent out after that byte and the following acknowledge pulse complete.

When user set STA and STO to 1 simultaneously, neither bit will be written into I2CCR.

Writing a 1 into STO bit can break any type of holding SCL line in LOW level by master.

Bit 2: STO	Description	
0	STOP condition won't be sent to I ² C bus.	(Initial value)
1	STOP condition will be sent to I ² C bus.	

Note: I2CI has a 2-bit internal queue buffer recording the command of sending START/STOP condition. Whenever the buffer is full, the bit at the queue tail will be set to overlapped manner, writing 1 into STA/STO bit will update that bit into the START/STOP condition. If the queue contains START/STOP conditions, they will be sent out just at the proper time.

- **Acknowledge Control Bit (AC):** This bit is used in read operation when I2CI is acting as a master-receiver. The value of AC bit will be sent to I²C bus as acknowledge signal directly.

Bit 1: AC	Description	
0	0 will be sent to I ² C bus as LOW level acknowledge signal.	(Initial value)
1	1 will be sent to I ² C bus as HIGH level acknowledge signal.	

- **Enable of I²C interface (I2CIE):** When this bit is cleared to 0, I2CI will be disabled and goes into a low power status. To prevent unpredictable results, software should guarantee that this bit is cleared at I2C bus free state.

Bit 0: I2CIE	Description	
0	I2CI module is disabled	(Initial value)
1	I2CI module is enabled	

17.3.3 Status Register (I2CSR)

I2CSR is an 8-bit read/write register that describes the status of I2CI. It is initialized to H'04 by power-on reset or manual reset.

Bit:	7	6	5	4	3	2	1	0
Read:				STX	BUSY	TEND	DRF	ACKF
Write:								
Reset:	0	0	0	0	0	1	0	0

Bit 7~5: Reserved. These bits always read as 0. Write data to these bits are ignored.

- **STA/STO Command is On (STX):** This bit representing whether there are pending STA/STO bits in the STA/STO FIFO in I2CI. Clearing I2CIE in I2CCR will empty the STA/STO FIFO, consequently clear this status bit.

Bit 4: STX	Description	
0	STA/STO FIFO buffer is empty.	(Initial value)
1	STA/STO FIFO buffer is not empty.	

- **I2C Bus Busy (BUSY):** The I2C bus is considered busy after the START condition. The bus is considered free a certain time after the STOP condition. The bus stays busy if a repeated START (Sr) is generated instead of a STOP condition.

Bit 3: BUSY	Description	
0	I2C bus is free.	(Initial value)
1	I2C bus is busy.	

- **Transmission End Flag (TEND):** 0 of this bits indicates an transmission idle state. When I2CI is transmitting, TEND is 1 means the byte of data in I2CI internal shift register has been sent out and the reception of acknowledge bit for that byte is over. It also indicates that I2CDR is regarded as empty now since DRF is 0 at this moment. When I2CI is receiving (determined by the least bit in the address byte), this bit is always 1. This bit is read-only for software.

Bit 2: TEND	Description	
0	The I2CI is proceeding a transmission or a reception of an acknowledgement.	
1	The I2CI is not proceeding a transmission and a reception of an acknowledgement.	(Initial value)

- **Data Register Valid Flag (DRF):** This bit indicates that data in I2CDR is valid.

In write operation, I2CDR register and internal shift register construct a two-byte buffer. The depth of the valid data in this buffer is indicated by DRF and TEND. TEND = 0 and DRF = 1 means the buffer for transmission is full. When the buffer is empty, user can continuously twice writing data into I2CDR and setting DRF to 1, because DRF = 1 can immediately let hardware load the data from I2CDR into shift register and clear DRF to 0 when the shift register is idle or the byte in shift register is sent out and the acknowledge bit for that byte is received into ACKF. Software should follow the step that set DRF to 1 after it has put data in I2CDR each time.

At the beginning of a repeated-START condition in the process of I2CI writing, this transfer two-byte buffer will be reset (invalidated). The DRF will be cleared and the TEND will be set.

In read operation, After a byte of data has been received into shift register and before acknowledge bit has been sent, hardware copies received data from internal shifter to I2CDR and sets DRF to 1. Software reads the data and clears DRF to 0.

In these cases, I2CI will hold the SCL in its LOW level state and wait.

1. After START condition, DRF is 0.
2. When I2CI shifts out a complete byte of data and the acknowledge bit pulse for this byte is over, DRF is 0 this time (before a underrun).
3. When I2CI shifts in a complete byte of data before loading it to I2CDR, DRF is 1 this time (before a overrun).
4. When a HIGH-level acknowledge bit is sent or received after the data byte or address byte. (This type of master hold can only be broken by writing 1 into STA or STO in I2CCR).

Bit 1: DRF	Description	
0	Data in I2CDR is invalid.	(Initial value)
1	Data in I2CDR is valid.	

Acknowledge Level Flag (ACKF): In write operation, this bit indicates the voltage level of received acknowledge signal. “0” indicates an acknowledge and “1” indicates a not-acknowledge. Software can only read the ACKF to get acknowledge level for the last byte which has just been sent out. When ACKF is 1, the I2CI will go into a master-hold state. In read operation, this bit has no use. This bit is read only to software. A START/repeated-START or STOP condition will clear the ACKF bit to 0.

Bit 0: ACKF	Description	
0	The acknowledge signal from I ² C-bus is “0”	(Initial value)
1	The acknowledge signal from I ² C-bus is “1”	

17.3.4 Clock Generator Register (I2CGR)

I2CGR is 16-bit read/write register that sets the frequency of serial clock. The serial clocks frequency is calculated as follows:

$$[\text{Value of I2CGR (SCL clock frequency)}] = [\text{Frequency of PCLK}] / 16 - 1$$

I2CGR is initialized to H'0 as well as a power-on reset or manual reset. Except I2CI is sending a START byte, I2CGR should not be written a new value when I2CI is busy. During the START byte is in process, if user writes I2CGR, the new value will be saved by hardware and will not be used until the START byte is totally sent out.

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

17.4 I²C-BUS PROTOCOL

17.4.1 Bit Transfer

Due to the variety of different technology devices (CMOS, NMOS, bipolar) which can be connected to the I²C-bus, the levels of the logical '0' (LOW) and '1' (HIGH) are not fixed and depend on the associated level of VDD. One clock pulse is generated for each data bit transferred.

17.4.2 Data Validity

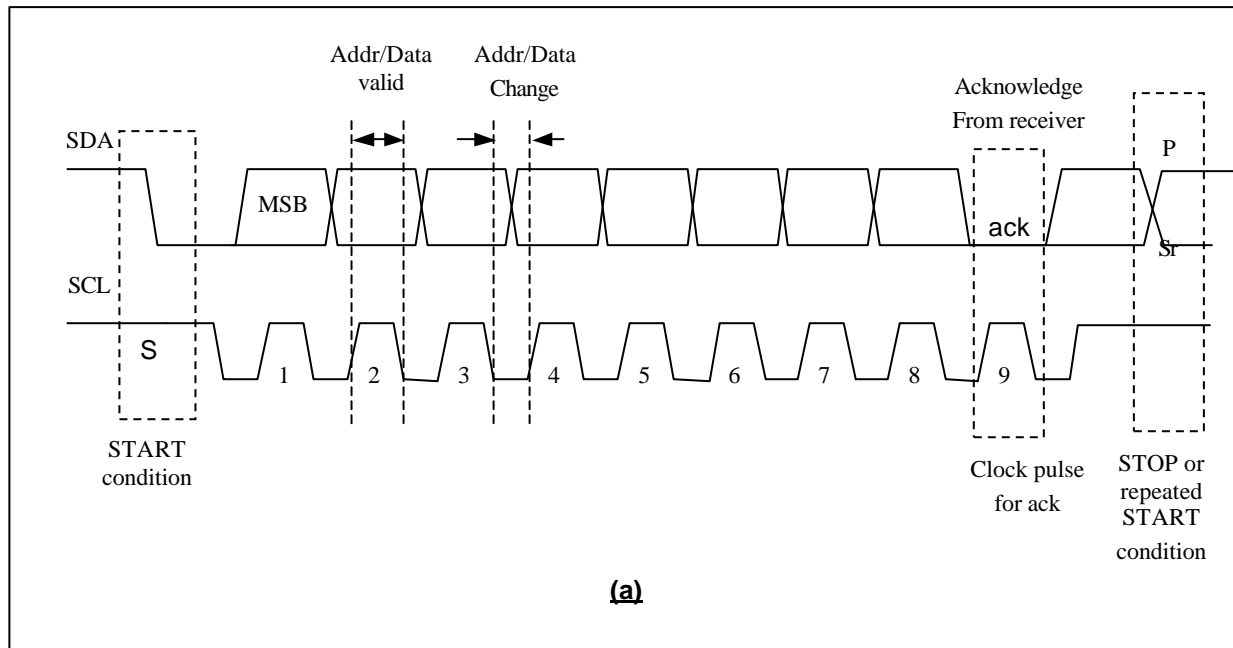
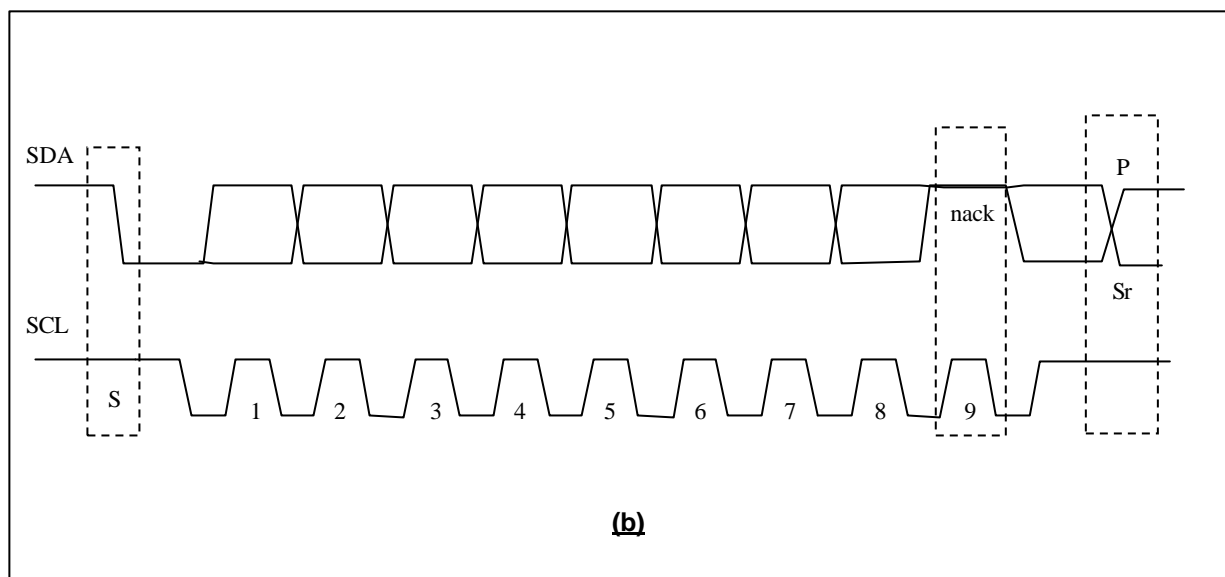
The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW states of the data line can only change when the clock signal on the SCL line is LOW.

17.4.3 START and STOP Conditions

A HIGH to LOW transition on the SDA line while SCL is HIGH indicates a START condition. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP condition.

17.4.4 Byte Format

1. Every byte put on the SDA line must be 8-bits long.
2. The number of bytes that can be transmitted/received per transfer is unrestricted.
3. Each byte has to be followed by an acknowledge (ack/nack) bit.
4. Data is transferred with the most significant bit (MSB) first.
5. Data transfer with an acknowledge signal (acknowledge or not-acknowledge) is obligatory.
6. The acknowledge_ related clock pulse is generated by the master.
7. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse.
8. Slave can hold the SCL line LOW during the SCL in LOW level at any bit to force the master to proceed a lower speed of transfer.

Figure 17-2 I²C-bus ProtocolFigure 17-3 I²C-bus Protocol (cont.)**Notes:**

1. Sr means repeated START condition. P means STOP condition.
2. In Fig (a), if the master does not generate Sr or P, the next data byte follows the ack.
3. In Fig (b), nack is received, the master generates Sr or P and the transfer terminates.

17.4.5 Data Transfer Format

17.4.5.1 First Byte

The first byte is a term indicates the address byte after START condition.

17.4.5.1.1 Normal 7-bit Address

After the START condition, the addressing procedure for the I²C-bus is such that the first byte usually determines which slave will be selected by the master.

The first seven bits of the first byte make up the slave address (see Figure 17-4). The eighth bit is the LSB (least significant bit). It determines the direction of the message. A 'zero' in the least significant position of the first byte means that the master will write information to a selected slave. A 'one' in this position means that the master will read information from the slave.

When an address is sent, each device in a system compares the first seven bits after the START condition with its address. If they match, the device considers itself addressed by the master as a slave-receiver or slave-transmitter, depending on the R/W bit.

A slave address can be made-up of a fixed and a programmable part. Since it's likely that there will be several identical devices in a system, the programmable part of the slave address enables the maximum possible number of such devices to be connected to the I²C-bus. The number of programmable address bits of a device depends on the number of pins available. For example, if a device has 4 fixed and 3 programmable address bits, a total of 8 identical devices can be connected to the same bus.

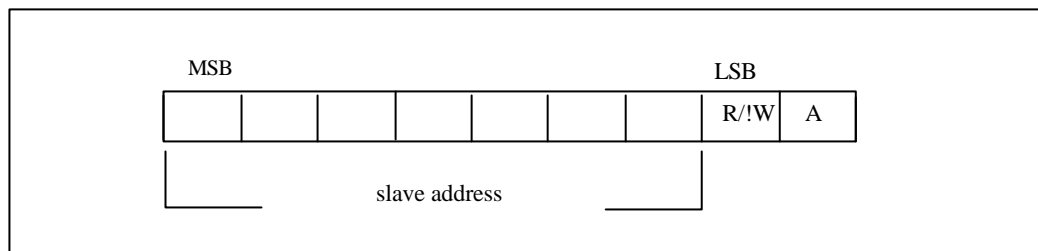


Figure 17-4 Normal 7 Bit Address after START Condition

17.4.5.1.2 General Call Address

Address byte with all bits are "0" is defined as "general call address". When this address is used, all devices should, in theory, respond with an acknowledge. However, if a device doesn't need any of the data supplied within the general call structure, it can ignore this address by not issuing an acknowledgment. If a device does require data from a general call address, it will acknowledge this address and behave as a slave-receiver. The second and following bytes will be acknowledged by every slave-receiver capable of handling this data. A slave that cannot process one of these bytes must ignore it by not-acknowledging.

The second byte of the general call address then defines the action to be taken.

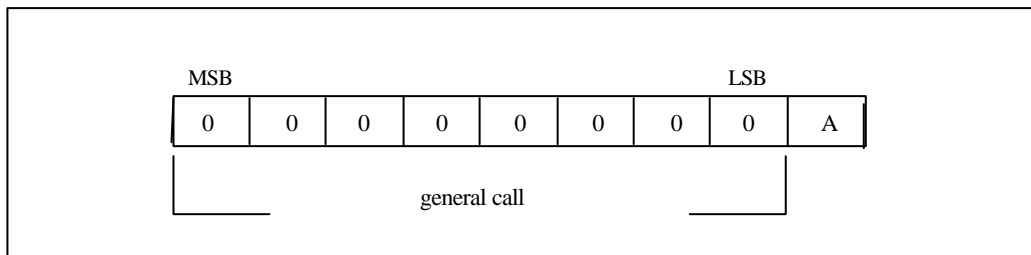


Figure 17-5 General Call Address after START Condition

17.4.5.1.3 START Byte Address

START Byte:

After the START condition S has been transmitted by the master, data transfer can be preceded by a start procedure which is much longer than normal. The start procedure consists of:

- **A START condition (S)**
- **A START byte (00000001)**
- **An acknowledge clock pulse (ACK)***
- **A repeated START condition (Sr)**

Note: An acknowledge-related clock pulse is generated after the START byte. This is present only to conform to the byte handling format used on the bus. No device is allowed to acknowledge the START byte.

When the START byte (00000001) is transmitted, another microcontroller (the slave) can therefore sample the SDA line at a low sampling rate (also determined by the I2CGR) until one of the seven zeros in the START byte is detected. After detection of this LOW level on the SDA line, the microcontroller can switch to a higher sampling rate to find the repeated START condition Sr which is then used for synchronization.

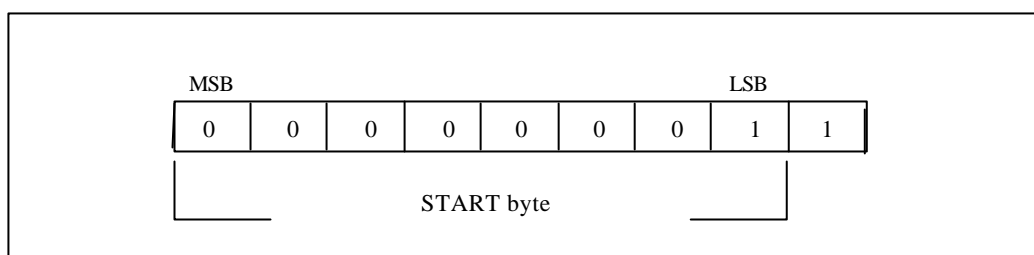


Figure 17-6 START Byte after START Condition

17.4.5.2 Transfer Format

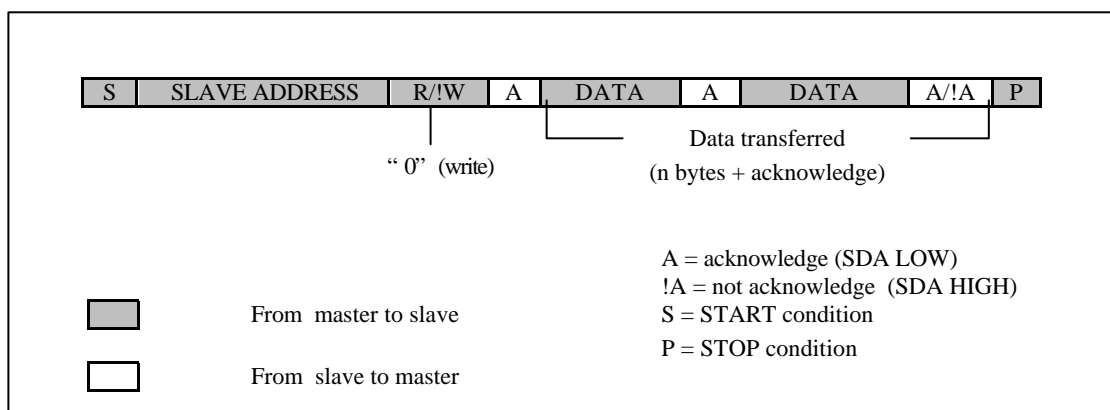
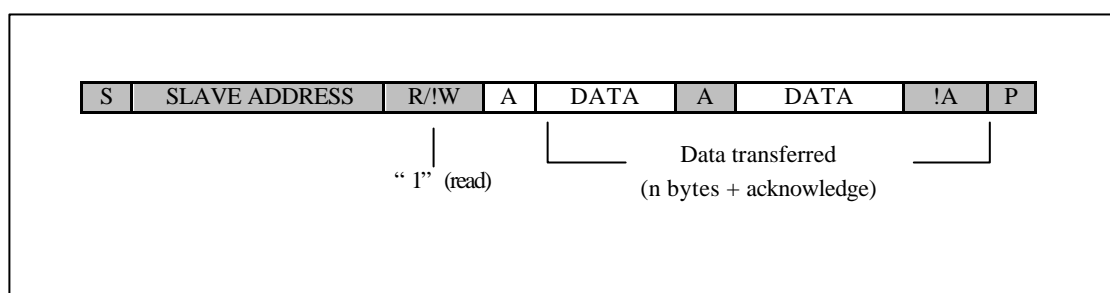
A data transfer is always terminated by a STOP condition (P) generated by the master. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition (Sr) and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.

Possible data transfer formats are:

- Master-transmitter transmits to slave-receiver. The transfer direction is not changed.
- Master reads slave immediately after first byte. At the moment of the first acknowledge, the master-transmitter becomes a master-receiver and the slave-receiver becomes a slave-transmitter.
- This first acknowledge is still generated by the slave. The STOP condition is generated by the master, which has previously sent a not-acknowledge.

Notes:

1. Combined formats can be used, for example, to control a serial memory. During the first data byte, the internal memory location has to be written. After the START condition and slave address is repeated, data can be transferred.
2. All decisions on auto-increment or decrement of previously accessed memory locations etc. are taken by the designer of the device.
3. Each byte is followed by an acknowledgment bit as indicated by the 'A' or '!A' blocks in the sequence.

**Figure 17-7 A Master-Transmitter Addresses a Slave Receiver with a 7-Bit Address****Figure 17-8 A Master Reads the Slave Immediately after the First Byte (Master-Receiver)**

17.5 I²Ci Operation

17.5.1 I²Ci Initialize

Before transmitting and receiving data, set the I2CIE bit in I2CCR to 1 to enable I2Ci operation and set I2CGR for proper serial clock frequency. Set the I2CIE bit to 0 after transmitting or receiving data for low power dissipation.

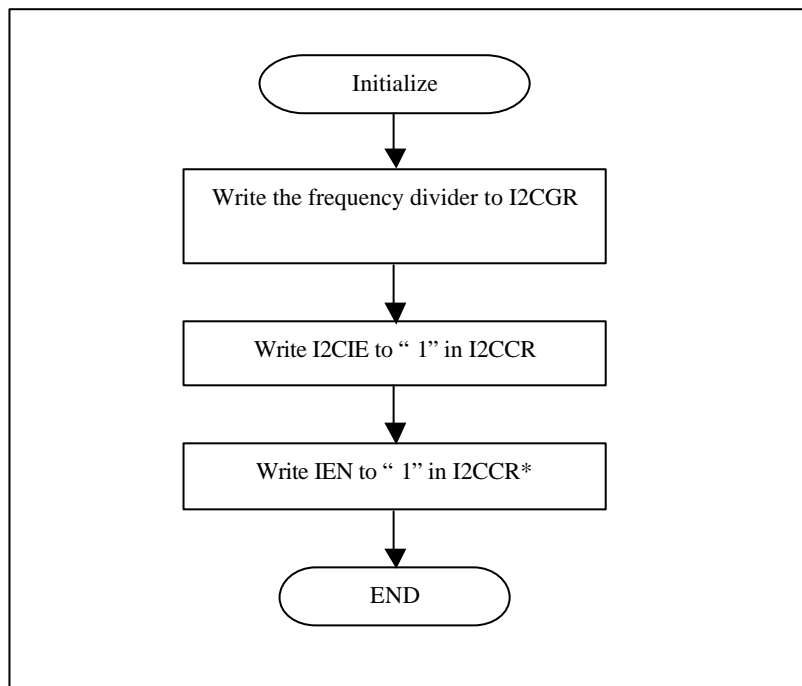


Figure 17-9 I²Ci Initialization

Note: This step is selectable.

17.5.2 Write Operation

Following figure illustrates the flow of a write operation.

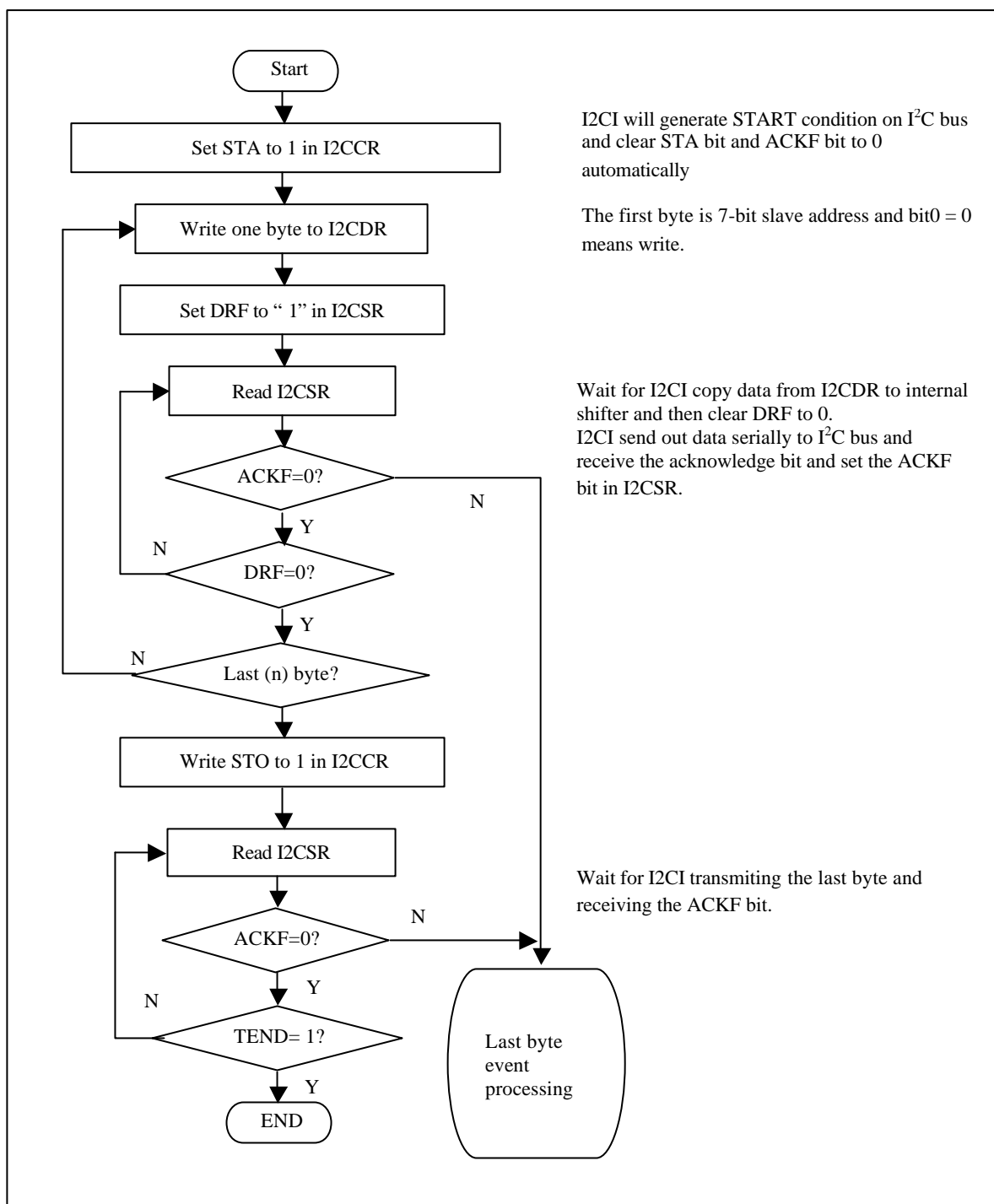


Figure 17-10 I2CI Write Operation Flowchart

17.5.3 Read Operation

Following figure illustrates the flow of read operation.

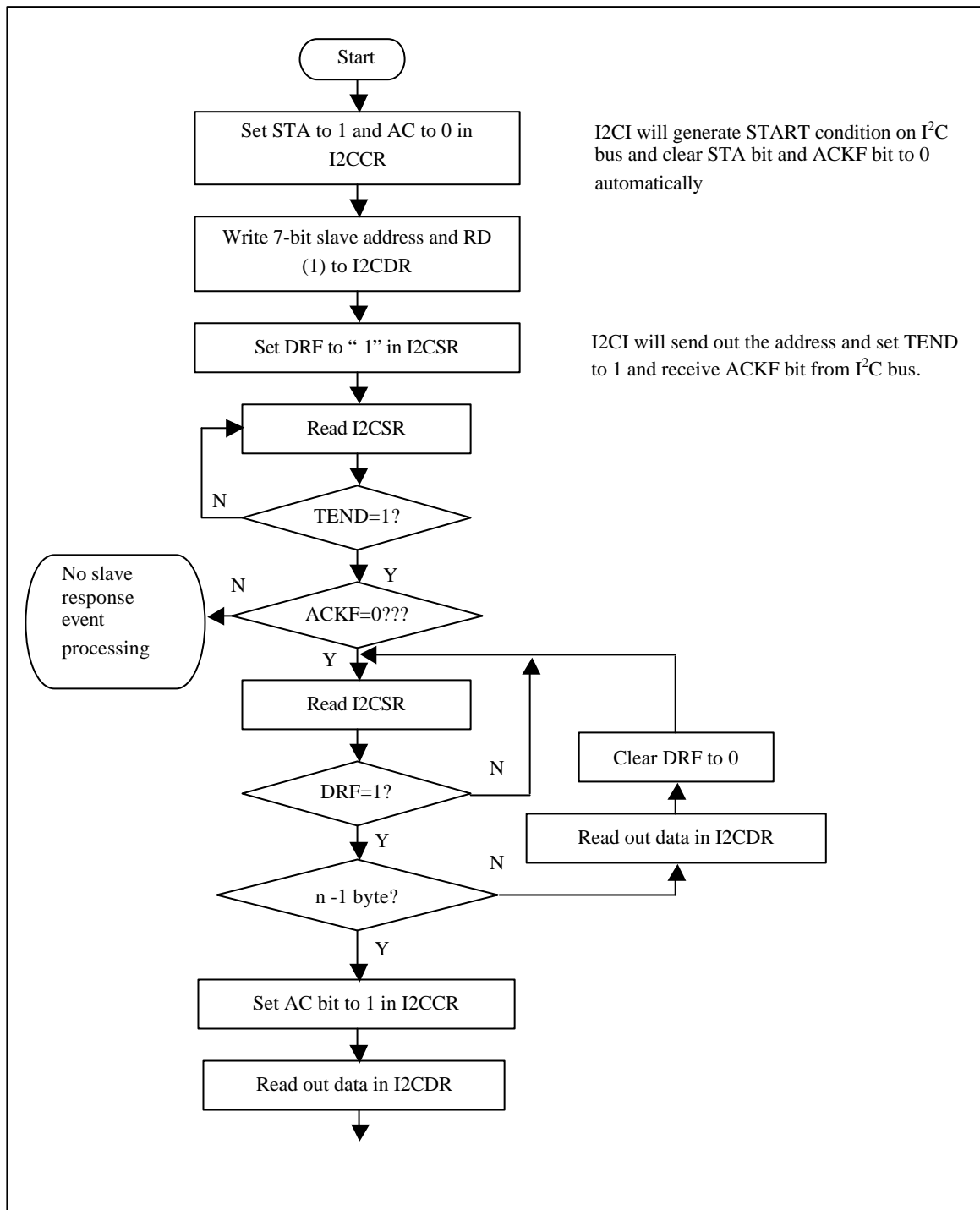


Figure 17-11 I2CI Read Operation Flowchart

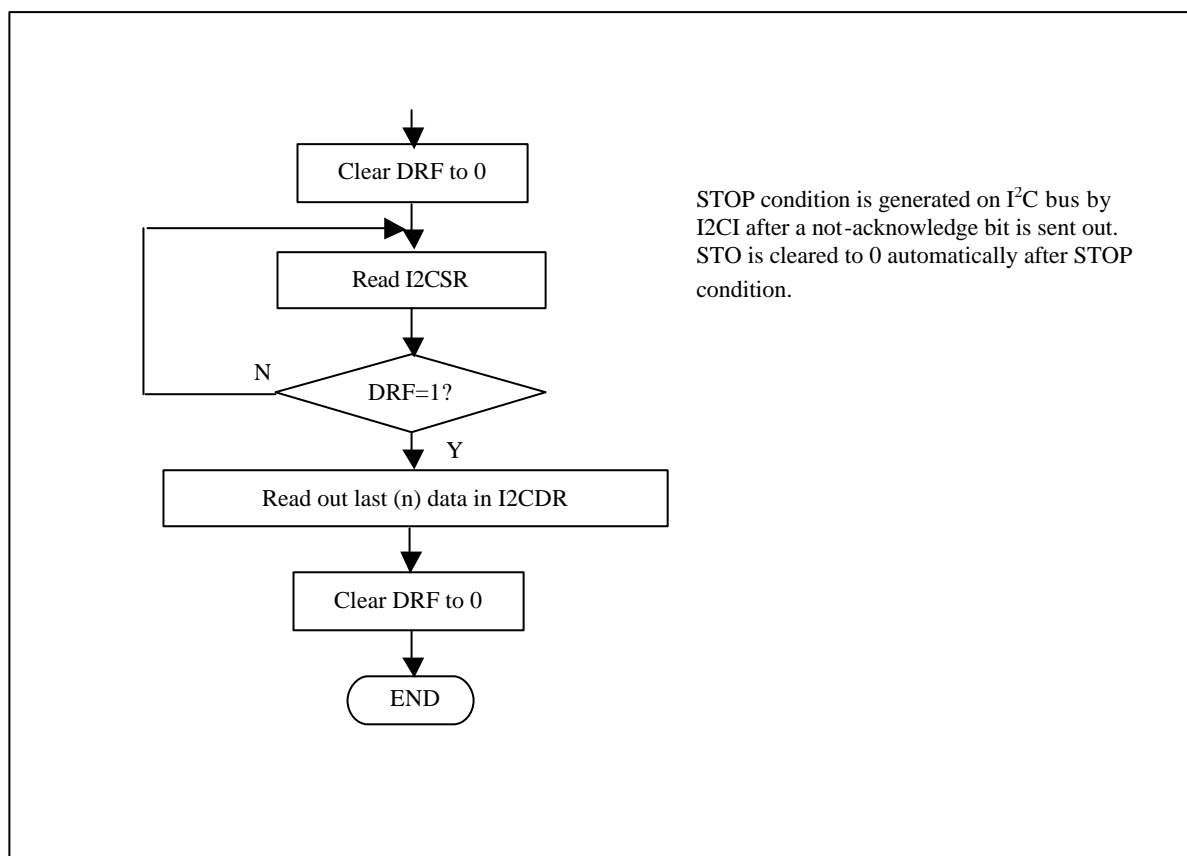


Figure 17-12 Read Operation Flowchart (cont.)

17.6 Appendix: Terms Referred from I2C Standard Specification

1. START condition
2. repeated START condition
3. STOP condition
4. START byte
5. general call
6. I2C bus busy
7. acknowledge
8. not-acknowledge

18 AC97 Controller

18.1 Overview

The full-duplex serial port for audio transfers serialized audio data to and from external devices using AC-Link (a 5 bit digital serial interface). Many popular audio D-to-A converters and codecs use this format.

The AC97 controller fully supports AC-Link functions that are compliant with AC'97 Component Specification 2.2. For AC97 controller, digitized audio can pass in both directions simultaneously using sample sizes up to 20 bits.

The AC97 controller can use DMA to transfer data between the external codec and system memory without processor intervention. If DMA is not enabled then interrupt transfers can be used. Interrupts to the host processor are minimized by inclusion of dual FIFOs that store up or free down to 16 samples in the corresponding direction.

If the AC97 Controller function is disabled, its pins may be used for GPIO functions.

18.1.1 Feature

Some of the main features are:

- Multiple Sample Size (16, 18 and 20 bit) Support
- Programmable Output channels and Input channels Support
- Power Down Mode Support
- Two Wake-Up mode Support
- DMA transfer mode Support
- Programmable Interrupt function Support

18.1.2 Block Diagram

AC97 Controller Block Diagram is described in Figure 18-1 AC97 Controller Block Diagram as shown in following.

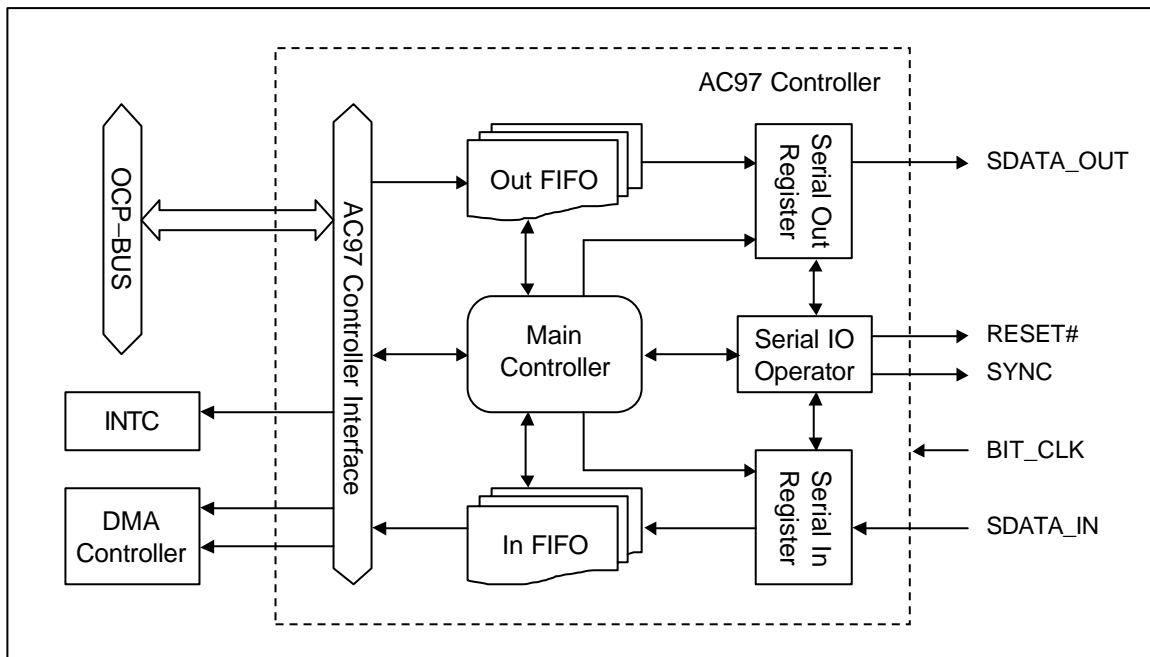


Figure 18-1 AC97 Controller Block Diagram

18.1.3 Pin configuration

Table 18-3 AC97 Controller Pins

Name	Full Name	I/O	Description
SDATA_OUT	Serial data out pin	Output	Serial audio data out, to Codec
BIT_CLK	Serial Clock pin	Input	12.288 MHz serial data clock
SDATA_IN	Serial data in pin	Input	Serial audio data in, from Codec
SYNC	Frame Synchronization pin	Output	48 kHz fixed rate sample sync
RESET#	Reset pin	Output	AC'97 Master Hardware Reset

Note: When any of these pins are used as GPIO, the AC97 function is disabled.

18.1.3.1 SDATA_OUT Pin

This is Serial audio data pin, which outputs to codec. If AC97 Controller is disabled, it retains the low.

18.1.3.2 BIT_CLK Pin

This is 12.288 MHz serial data clock, which inputs from the primary codec. If AC97 Controller is disabled, its state is undefined.

18.1.3.3 SDATA_IN Pin

This is Serial audio data pin, which inputs from codec. If AC97 Controller is disabled, its state is undefined.

18.1.3.4 SYNC Pin

This is 48 kHz fixed rate sample sync, which outputs to codec. If AC97 Controller is disabled, it retains the low.

18.1.3.5 RESET# Pin

This is AC'97 Master Hardware Reset, which outputs to codec. If AC97 Controller is disabled, it retains the high.

18.2 Register Configuration

There are 9 registers and 2 FIFO data ports in the AC97 Controller block: one configuration, two controls, one status, one interrupt control, four AC-link control registers, and two FIFO data port registers.

- Configuration register is used to program common control.
- Two control registers contains to configure transfer slots and to control reset and synchronization signal of AC-link.
- The Status Registers signal the state of the FIFO buffers and the status of the interface, which is selected by the common control register.
- The Interrupt Enable Register controls the corresponding flag to generate an interrupt.
- The AC-link Control Registers are Command Address, Command Data, Status Address, and Status Data for AC-link interface.
- Two FIFO data port registers are mapped as 32-bit word address, which physically points to two 32-bit registers. One register is for WRITES, and transfers data to the Transmit FIFO; the other is for READS, and takes data from the Receive FIFO.

Table 18-4 AC97 Controller Registers

Name	Full Name	R/W	Initial value	Address	Access Size
ACFR	AC97 Controller Configuration Register	R/W	H'00007700	H'E000 0900	32
ACCR1	AC97 Controller Control Register 1	R/W	H'0???0000 ¹	H'E000 0904	32
ACCR2	AC97 Controller Control Register 2	R/W	H'00000000	H'E000 0908	32
ACSR	AC97 Controller Status Register	R/W	H'???0???? ²	H'E000 090C	32
ACIER	AC97 Controller Interrupt Enable Register	R/W	H'00000000	H'E000 0910	32
ACCAR	AC97 Controller Command Address Register	R/W	H'00000000	H'E000 0914	32
ACCDR	AC97 Controller Command Data Register	R/W	H'00000000	H'E000 0918	32
ACSAR	AC97 Controller Status Address Register	R	H'000????? ¹	H'E000 091C	32
ACSDR	AC97 Controller Status Data Register	R	H'000????? ¹	H'E000 0920	32
ACODR	AC97 Controller Out FIFO Data Port Register	W	Undefined	H'E000 0924	32
ACIDR	AC97 Controller In FIFO Data Port Register	R	Undefined	H'E000 0928	32

1. If BIT_CLK pin inputs a clock signal, this value is H'0000-0000; otherwise, undefined.

2. If BIT_CLK pin inputs a clock signal, this value is H'0000-0008; otherwise, undefined.

18.2.1 Register Descriptions

18.2.1.1 AC97 Controller Configuration Register (ACFR)

This register is used to program common control.

Note: The power-up/reset default value of this register is H'0000-7700.

H'E000 0900

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	RFTH				TFTH			
Write:								
Reset:	0	1	1	1	0	1	1	1

Bit:	7	6	5	4	3	2	1	0
Read:					RST			ENB
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit 31 ~ 16, 7 ~ 4, 2, 1: Reserved bits, ignored in write operation, always 0 in read operation.

- **RFTH (Bit 15 ~ 12):** Receive FIFO interrupt or DMA threshold; set to value 0 – 15. This value should be set to the desired threshold value minus one. When RFL (the valid PCM data level value of receive FIFO in ACSR) is great or equal than the setting of the threshold about receive FIFO, the RFS in ACSR is being set. Larger values provide a longer latency than smaller values. Using threshold values that are too large can result in an overflow condition of receive FIFO. The optimum value, which is system dependent, results in the FIFO being half full.

Bit 15 ~ 12: RFTH	Description	
H'0	When RFL >= H'1, RFS is set.	
H'1	When RFL >= H'2, RFS is set.	
H'2	When RFL >= H'3, RFS is set.	
H'3	When RFL >= H'4, RFS is set.	
H'4	When RFL >= H'5, RFS is set.	
H'5	When RFL >= H'6, RFS is set.	
H'6	When RFL >= H'7, RFS is set.	
H'7	When RFL >= H'8, RFS is set.	(Initial value)
H'8	When RFL >= H'9, RFS is set.	
H'9	When RFL >= H'A, RFS is set.	

H'A	When RFL >= H'B, RFS is set.	
H'B	When RFL >= H'C, RFS is set.	
H'C	When RFL >= H'D, RFS is set.	
H'D	When RFL >= H'E, RFS is set.	
H'E	When RFL >= H'F, RFS is set.	
H'F	When RFL >= H'10, RFS is set.	

- **TFTH (Bit 11 ~ 7):** Transmit FIFO interrupt or DMA threshold; set to value 0 – 15. This value should be set to the desired threshold value minus one. When TFL (the valid PCM data level value of transmit FIFO in ACSR) is little or equal than the setting of the threshold about transmit FIFO, the TFS in ACSR is being set. Smaller values provide a longer latency than larger values. Using threshold values that are too small can result in an underflow condition of transmit FIFO.

Bit 11 ~ 8: TFTH	Description	
H'0	When TFL <= H'1, TFS is set.	
H'1	When TFL <= H'2, TFS is set.	
H'2	When TFL <= H'3, TFS is set.	
H'3	When TFL <= H'4, TFS is set.	
H'4	When TFL <= H'5, TFS is set.	
H'5	When TFL <= H'6, TFS is set.	
H'6	When TFL <= H'7, TFS is set.	
H'7	When TFL <= H'8, TFS is set.	(Initial value)
H'8	When TFL <= H'9, TFS is set.	
H'9	When TFL <= H'A, TFS is set.	
H'A	When TFL <= H'B, TFS is set.	
H'B	When TFL <= H'C, TFS is set.	
H'C	When TFL <= H'D, TFS is set.	
H'D	When TFL <= H'E, TFS is set.	
H'E	When TFL <= H'F, TFS is set.	
H'F	When TFL <= H'10, TFS is set.	

- **RST (Bit 3):** Reset the AC97 Controller Control and FIFOs except this register, **it is always read 0.**

Bit 3: RST	Description	
0	Not reset	(Initial value)
1	Reset is Active to Other AC97 Controller Registers and FIFOs.	

- **ENB (Bit 0):** Enable AC97 Controller function, this bit is used to enable or disable the AC97 function.

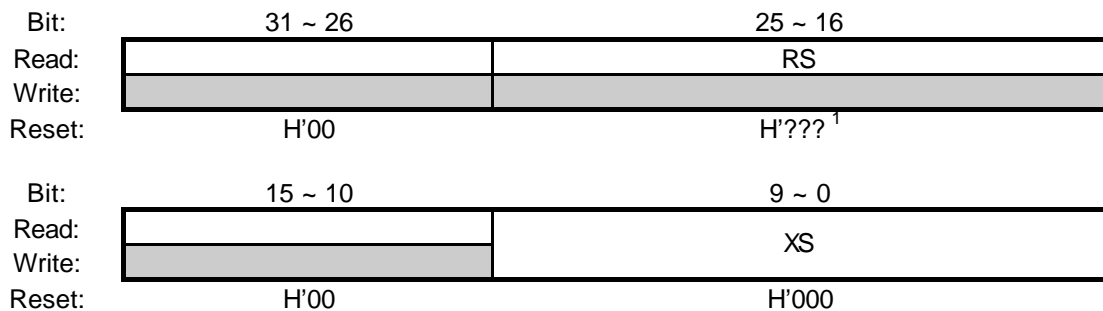
Bit 0: ENB	Description	
0	Disable AC97 Controller	(Initial value)
1	Enable AC97 Controller	

18.2.1.2 AC97 Controller Control Register 1 (ACCR1)

AC97 Controller Control Register 1 contains bits necessary to be configured to determine the transfer slots.

Note: If BIT_CLK pin inputs a clock signal, the power-up/reset default value of this register is H'0000-0000; otherwise, it is H'0???-0000.

H'E000 0904



1. If BIT_CLK inputs a clock signal, it is 0; otherwise, undefined.

Bit 31 ~ 26, 15 ~ 10: Reserved bits, ignored in write operation, always 0 in read operation.

- **RS (Bit 25 ~ 16):** Receive Slots, the read only bits making up slots are taken from the valid bits in the AC'97 tag (slot 0 of SDATA_IN) and indicate which incoming slots have valid PCM data. Slot 3 is mapped to bit 16, slot 4 to 17 and so on. If the corresponding bit is set it indicates that valid PCM data will be in the respective slot.

Bit 25 ~ 16: RS		Description	
Bit 25: RS[9]	0	Slot 12 is invalid.	(Initial value)
	1	Slot 12 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	
Bit 24: RS[8]	0	Slot 11 is invalid.	(Initial value)
	1	Slot 11 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	
Bit 23: RS[7]	0	Slot 10 is invalid.	(Initial value)
	1	Slot 10 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	

Bit 25 ~ 16: RS		Description	
Bit 22: RS[6]	0	Slot 9 is invalid.	(Initial value)
	1	Slot 9 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	
Bit 21: RS[5]	0	Slot 8 is invalid.	(Initial value)
	1	Slot 8 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	
Bit 20: RS[4]	0	Slot 7 is invalid.	(Initial value)
	1	Slot 7 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	
Bit 19: RS[3]	0	Slot 6 is invalid.	(Initial value)
	1	Slot 6 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	
Bit 18: RS[2]	0	Slot 5 is invalid.	(Initial value)
	1	Slot 5 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	
Bit 17: RS[1]	0	Slot 4 is invalid.	(Initial value)
	1	Slot 4 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	
Bit 16: RS[0]	0	Slot 3 is invalid.	(Initial value)
	1	Slot 3 has valid PCM data.	
	X	When BIK_CLK has not a clock, this bit is undefined.	

- **XS (Bit 9 ~ 0):** Transmit Slots, the bits making up slots map to the valid bits in the AC'97 tag (slot 0 on SDATA_OUT) and indicate which outgoing slots have valid PCM data. Bit 0 maps to slot 3, bit 1 to slot 4 and so on. Setting the corresponding bit indicates to the CODEC that valid PCM data will be in the respective slot. The number of valid bits will designate how many words will be pulled out of the FIFO per audio frame.

Bit 9 ~ 0: XS		Description	
Bit 9: XS[9]	0	Slot 12 is invalid.	(Initial value)
	1	Slot 12 has valid PCM data.	
Bit 8: XS[8]	0	Slot 11 is invalid.	(Initial value)
	1	Slot 11 has valid PCM data.	
Bit 7: XS[7]	0	Slot 10 is invalid.	(Initial value)
	1	Slot 10 has valid PCM data.	
Bit 6: XS[6]	0	Slot 9 is invalid.	(Initial value)
	1	Slot 9 has valid PCM data.	
Bit 5: XS[5]	0	Slot 8 is invalid.	(Initial value)
	1	Slot 8 has valid PCM data.	
Bit 4: XS[3]	0	Slot 7 is invalid.	(Initial value)
	1	Slot 7 has valid PCM data.	

Bit 9 ~ 0: XS		Description	
Bit 3:	0	Slot 6 is invalid.	(Initial value)
XS[3]	1	Slot 6 has valid PCM data.	
Bit 2:	0	Slot 5 is invalid.	(Initial value)
XS[2]	1	Slot 5 has valid PCM data.	
Bit 1:	0	Slot 4 is invalid.	(Initial value)
XS[1]	1	Slot 4 has valid PCM data.	
Bit 0:	0	Slot 3 is invalid.	(Initial value)
XS[0]	1	Slot 3 has valid PCM data.	

18.2.1.3 AC97 Controller Control Register 2 (ACCR2)

AC97 Controller Control Register 2 contains bits to program common control, and reset and synchronization signal of AC-link.

Note: The power-up/reset default value of this register is H'0000-0000.

H'E000 0908

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:	RDMS	TDMS					OASS	
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:	IASS		ENLBF	ERPL	EREC	SR	SS	SA
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit 31 ~ 16, 13 ~ 10: Reserved bits, ignored in write operation, always 0 in read operation.

- **RDMS (Bit 15):** Receiving DMA Mode Switch, this bit is used to enable or disable the DMA mode during receiving audio data.

Bit 15: RDMS	Description	
0	Disable DMA when receiving	(Initial value)
1	Enable DMA when receiving	

- **TDMS (Bit 14):** Transmitting DMA Mode Switch, this bit is used to enable or disable the DMA mode during transmitting audio data.

Bit 14: TDMS	Description	
0	Disable DMA when transmitting	(Initial value)
1	Enable DMA when transmitting	

- **OASS (Bit 9 ~ 8):** Output Audio Sample Size.

Bit 9 ~ 8: OASS	Description	
H'0	Audio sample is 20 bit.	(Initial value)
H'1	Audio sample is 18 bit.	
H'2	Audio sample is 16 bit.	
H'3	Reserved	

Note: The OASS determines the valid bits, when the software accesses ACODR (transmitting FIFO data port, see 18.2.1.10).

- **IASS (Bit 7 ~ 6):** Input Audio Sample Size.

Bit 7 ~ 6: IASS	Description	
H'0	Audio sample is 20 bit.	(Initial value)
H'1	Audio sample is 18 bit.	
H'2	Audio sample is 16 bit.	
H'3	Reserved	

Note: The IASS determines the valid bits, when the software accesses ACIDR (receiving FIFO data port, see 18.2.1.10).

- **ENLBF (Bit 5):** Enable AC97 Loop Back Function (for test), this bit is used to enable or disable the internal loop back function of AC97 Controller.

Bit 5: ENLBF	Description	
0	AC97 Loop Back Function is Disabled	(Initial value)
1	AC97 Loop Back Function is Enabled	

- **ENRPL (Bit 4):** Enable Playing Back function of AC97, this bit is used to disable or enable the audio sample data transmitting.

Bit 4: ENRPL	Description	
0	AC97 Playing Back Function is Disabled	(Initial value)
1	AC97 Playing Back Function is Enabled	

- **ENREC (Bit 3):** Enable Recording Function of AC97, this bit is used to disable or enable the audio sample data receiving.

Bit 3: ENREC	Description	
0	AC97 Recording Function is Disabled	(Initial value)
1	AC97 Recording Function is Enabled	

- **SR (Bit 2):** Specify RESET# signal, this bit is used to drive the RESET# signal of AC97 Controller in AC-link Interface.

Bit 2: SR	Description	
0	RESET# signal is driven to high	(Initial value)
1	RESET# signal is active low	

Note: The operation of this bit must be compliant with AC'97 Component Specification 2.2. Otherwise, the result may be undesired.

- **SS (Bit 1):** When this bit is read, it **describes the actual value of SYNC**; When this is written, this bit controls the value of the SYNC signal when SA is set to 1. See SA description in following.
- **SA (Bit 0):** SYNC Alternation, this bit is used to determine the driven signal of SYNC. When SA is 0, SYNC being driven by serial clock; otherwise, SYNC being controlled by the SS in ACCR2. The truth value table is described in following.

Bit 0: SA	Bit 1: SS	Description	
0	0	When read, indicated SYNC is 0	(Initial value)
		When write, not effect	
	1	When read, indicated SYNC is 1	
		When write, not effect	
1	0	When read, indicated SYNC is 0	
		When write, SYNC is driven to 0	
	1	When read, indicated SYNC is 1	
		When write, SYNC is driven to 1	

18.2.1.4 AC97 Controller Status Registers (ACSR)

AC97 Controller Status Register (ACSR) is used for the AC-link Interface and FIFO Status. This status register is a read-only register. Among them, some bit can be written a 0 to clear their interrupt.

Note: If BIT_CLK pin inputs a clock signal, the power-up/reset default value of this register is H'0000-0008; otherwise, it is H'???0-????.

H'E000 090C

Bit:	31	30	29	28	27	26	25	24
Read:						RFL		
Write:								
Reset:	0	0	0	? ¹	? ¹	? ¹	? ¹	? ¹

Bit:	23	22	21	20	19	18	17	16
Read:				CRDY	CLPM			
Write:						RSTO	SADR	CADT
Reset:	0	0	0	? ¹	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:						TFL		
Write:								
Reset:	0	0	0	? ¹	? ¹	? ¹	? ¹	? ¹

Bit:	7	6	5	4	3	2	1	0
Read:								
Write:		ROR	TUR	RFS	TFS			
Reset:	0	0	0	? ¹	? ²	0	0	0

1. If BIT_CLK inputs a clock signal, it is 0; otherwise, undefined.
2. If BIT_CLK inputs a clock signal, it is 1; otherwise, undefined.

Bit 31 ~ 29, 23 ~ 21, 15 ~ 13, 7, 2~0: Reserved bits, ignored in write operation, always 0 in read operation.

- **RFL (Bit 28 ~ 24):** A 5-bit Receive FIFO Level, the bits indicate the amount of valid PCM data in Receive FIFO.

Bit 28 ~ 24: RFL	Description	
H'00	No valid PCM data in Receive FIFO	(Initial value)
H'01	1 valid PCM data in Receive FIFO	
H'02	2 valid PCM data in Receive FIFO	
H'03	3 valid PCM data in Receive FIFO	
H'04	4 valid PCM data in Receive FIFO	
H'05	5 valid PCM data in Receive FIFO	
H'06	6 valid PCM data in Receive FIFO	
H'07	7 valid PCM data in Receive FIFO	
H'08	8 valid PCM data in Receive FIFO	
H'09	9 valid PCM data in Receive FIFO	
H'0A	10 valid PCM data in Receive FIFO	

H'0B	11 valid PCM data in Receive FIFO	
H'0C	12 valid PCM data in Receive FIFO	
H'0D	13 valid PCM data in Receive FIFO	
H'0E	14 valid PCM data in Receive FIFO	
H'0F	15 valid PCM data in Receive FIFO	
H'10	16 valid PCM data in Receive FIFO	
H'11 ~ H'1F	Reserved.	

- **CRDY (Bit 20):** External Codec Ready, this bit derives from the Codec Ready bit of Slot 0 in SDATA_IN, and indicates the external AC97 Codec is or not ready.

Bit 20: CRDY	Description	
0	When read, indicated Codec no ready	(Initial value)
	When write, not effect	
1	When read, indicated Codec ready	
	When write, not effect	

- **CLPM (Bit 19):** External Codec Low Power Mode, this bit indicates the external Codec is switched to low power mode or BIT_CLK is active from Codec after wake up.

Bit 19: CLPM	Description	
0	When read, indicated BIT_CLK is active	(Initial value)
	When write, not effect	
1	When read, indicated Codec is switched to low power mode	
	When write, not effect	

- **RSTO (Bit 18):** External Codec Registers Read Status Time Out, this bit indicates that the read status time out is or not detected, on condition that 4 Frames pass after a command to read the external Codec status registers. When this bit is written a 0 is clear itself and 1 is no effect.

Bit 18: RSTO	Description	
0	When read, indicated time out not occurred	(Initial value)
	When write, clear itself	
1	When read, indicated read status time out	
	When write, not effect	

- **SADR (Bit 17):** External Codec Registers Status Address and Data Received, this bit indicates that an address and data of external AC97 Codec registers has or not been received.

Bit 17: SADR	Description	
0	When read, indicated reception not occurred	(Initial value)
	When write, clear itself	
1	When read, indicated a reception done	
	When write, not effect	

- **CADT (Bit 16):** Command Address and Data Transmitted, this bit indicates that a command transmission of read/write external AC97 Codec registers is or not done.

Bit 16: CADT	Description	
0	When read, indicated command not transmitted	(Initial value)
	When write, clear itself	
1	When read, indicated a command done	
	When write, not effect	

- **TFL (Bit 12 ~ 8):** A 5-bit Transmit FIFO Level, the bits indicate the amount of valid PCM data in Transmit FIFO.

Bit 12 ~ 8: TFL	Description	
H'00	No valid PCM data in Transmit FIFO	(Initial value)
H'01	1 valid PCM data in Transmit FIFO	
H'02	2 valid PCM data in Transmit FIFO	
H'03	3 valid PCM data in Transmit FIFO	
H'04	4 valid PCM data in Transmit FIFO	
H'05	5 valid PCM data in Transmit FIFO	
H'06	6 valid PCM data in Transmit FIFO	
H'07	7 valid PCM data in Transmit FIFO	
H'08	8 valid PCM data in Transmit FIFO	
H'09	9 valid PCM data in Transmit FIFO	
H'0A	10 valid PCM data in Transmit FIFO	
H'0B	11 valid PCM data in Transmit FIFO	
H'0C	12 valid PCM data in Transmit FIFO	
H'0D	13 valid PCM data in Transmit FIFO	
H'0E	14 valid PCM data in Transmit FIFO	
H'0F	15 valid PCM data in Transmit FIFO	
H'10	16 valid PCM data in Transmit FIFO	
H'11 ~ H'1F	Reserved.	

- **ROR (Bit 6):** Receive FIFO Over Run, this bit indicates that Receive FIFO has or not experienced an overrun.

Bit 6: ROR	Description	
0	When read, indicated no overrun	(Initial value)
	When write, clear itself	
1	When read, indicated that attempted data write to full Receive FIFO	
	When write, not effects	

- **TUR (Bit 5):** Transmit FIFO Under Run, this bit indicates that Transmit FIFO has or not experienced an under-run.

Bit 5: TUR	Description	
0	When read, indicated no under-run	(Initial value)
	When write, clear itself	
1	When read, indicated that attempted data read from empty Transmit FIFO	
	When write, not effect	

- **RFS (Bit 4):** Receive FIFO Service Request, this bit indicates that Receive FIFO level is or not below RFL threshold (RFTH, the setting in ACFR, see 18.2.1.1).

Bit 4: RFS	Description	
0	When read, indicated that Receive FIFO level below RFL threshold	(Initial value)
	When write, not effect	
1	When read, indicated Receive FIFO level is at or above RFL threshold	
	When write, not effect	

- **TFS (Bit 3):** Transmit FIFO Service Request, this bit indicates that Transmit FIFO level exceeds TFL threshold (TFTH, the setting in ACFR, see 18.2.1.1).

Bit 3: TFS	Description	
0	When read, indicated Transmit FIFO level exceeds TFL threshold	(Initial value)
	When write, not effect	
1	When read, indicated that Transmit FIFO level is at or below TFL threshold	
	When write, not effect	

18.2.1.5 AC97 Controller Interrupt Enable Registers (ACIER)

AC97 Controller Interrupt Enable Register (ACIER) is used to enable or disable the interrupt request for the relating function in ACSR.

Note: The power-up/reset default value of this register is H'0000-0000.

H'E000 0910

Bit:	31	30	29	28	27	26	25	24
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	23	22	21	20	19	18	17	16
Read:						ERSTO	ESADR	ECADT
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit:	7	6	5	4	3	2	1	0
Read:		EROR	ETUR	ERFS	ETFS			
Write:								
Reset:	0	0	0	0	0	0	0	0

Bit 31 ~ 19, 15 ~ 7, 2 ~ 0: Reserved bits, ignored in write operation, always 0 in read operation.

- **ERSTO (Bit 18):** Enable RSTO Interrupt, this bit is used to control the RSTO interrupt enable or disable.

Bit 18: ERSTO	Description	
0	Disable RSTO interrupt function	(Initial value)
1	Enable RSTO interrupt function	

- **ESADR (Bit 17):** Enable SADR Interrupt, this bit is used to control the SADR interrupt enable or disable.

Bit 17: ESADR	Description	
0	Disable SADR interrupt function	(Initial value)
1	Enable SADR interrupt function	

- **ECADT (Bit 16):** Enable CADT Interrupt, this bit is used to control the CADT interrupt enable or disable.

Bit 16: ECADT	Description	
0	Disable CADT interrupt function	(Initial value)
1	Enable CADT interrupt function	

- **EROR (Bit 6):** Enable ROR Interrupt, this bit is used to control the ROR interrupt enable or disable.

Bit 6: EROR	Description	
0	Disable ROR interrupt function	(Initial value)
1	Enable ROR interrupt function	

- **ETUR (Bit 5):** Enable TUR Interrupt, this bit is used to control the TUR interrupt enable or disable.

Bit 5: ETUR	Description	
0	Disable TUR interrupt function	(Initial value)
1	Enable TUR interrupt function	

- **ERFS (Bit 4):** Enable RFS Interrupt, this bit is used to control the RFS interrupt enable or disable.

Bit 4: ERFS	Description	
0	Disable RFS interrupt function	(Initial value)
1	Enable RFS interrupt function	

- **ETFS (Bit 3):** Enable TFS Interrupt, this bit is used to control the TFS interrupt enable or disable.

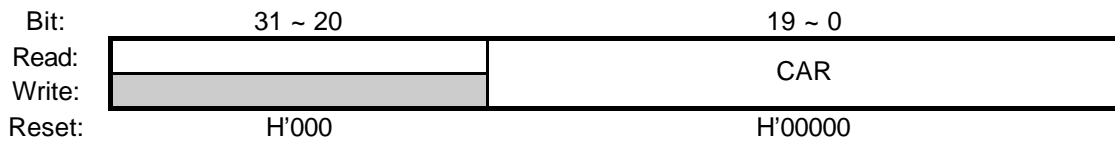
Bit 3: ETFS	Description	
0	Disable TFS interrupt function	(Initial value)
1	Enable TFS interrupt function	

18.2.1.6 AC97 Controller Command Address Register (ACCAR)

AC97 Controller Command Address Register (ACCAR) is used to transmit the read/write flag and Codec registers address in Slot 1 of out frame. AC97 Controller is automatically executing the external Codec register access command transfer after a read/write command is issued by the software. A read command is a data writing to ACCAR and a write command is two data writing to ACCAR and ACCDR. The data format want writing to ACCAR and ACCDR is compliant with AC'97 Component Specification 2.2. When a read/write command transfer is done CADT bit of ACSR is set. When a read/write command is issued, **the software must make certain that CADT is 0**. Otherwise, this command is invalid. (See 18.3.4).

Note: The power-up/reset default value of this register is H'0000-0000.

H'E000 0914



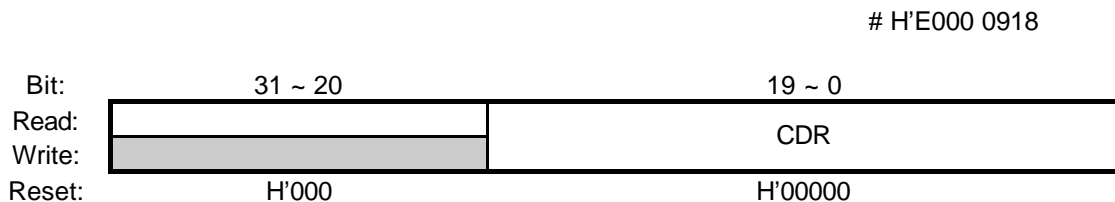
Bit 31 ~ 20: Reserved bits, ignored in write operation, always 0 in read operation.

- **CAR (Bit 19 ~ 0):** Command Address Register, a 20-bit data for AC97 Controller Command Address Port (Slot 1 in SDATA_OUT).

18.2.1.7 AC97 Controller Command Data Register (ACCDR)

AC97 Controller Command Data Register (ACCDR) is used to transmit the data want to write into Codec registers in Slot 2 of out frame. The data format want writing to ACCAR and ACCDR is compliant with AC'97 Component Specification 2.2. (See 18.3.4).

Note: The power-up/reset default value of this register is 0000-0000h.



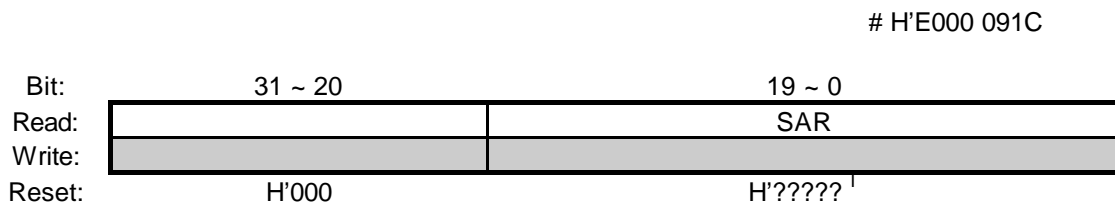
Bit 31 ~ 20: Reserved bits, ignored in write operation, always 0 in read operation.

- **CDR (Bit 19 ~ 0):** Command Data Register, a 20-bit data for AC97 Controller Command Data Port (Slot 2 in SDATA_OUT).

18.2.1.8 AC97 Controller Status Address Register (ACSAR)

When AC97 Controller has received the external Codec registers address and data from Slot 1 and Slot 2 in SDATA_IN, SADR bit of ACSR is set and the address and data are held in ACSAR and ACSDR. (See 18.3.4).

Note: If BIT_CLK pin inputs a clock signal, the power-up/reset default value of this register is H'0000-0000; otherwise, it is H'000?-????.



Bit 31 ~ 20: Reserved bits, ignored in write operation, always 0 in read operation.

- **SAR (Bit 19 ~ 0):** Status Address Register, a 20-bit read only register, 20-bit data from AC97 Controller Status Address Port (Slot 1 of Input Frame). Echo of register index for which data is being returned. The write operation is ignored.

18.2.1.9 AC97 Controller Status Data Register (ACSDR)

When AC97 Controller has received the external Codec registers address and data from Slot 1 and Slot 2 in SDATA_IN, SADR bit of ACSR is set and the address and data are held in ACSAR and ACSDR. (See 18.3.4).

Note: If BIT_CLK pin inputs a clock signal, the power-up/reset default value of this register is H'0000-0000; otherwise, it is H'000?-????.

	# H'E000 0920	
Bit:	31 ~ 20	19 ~ 0
Read:		SDR
Write:		
Reset:	H'000	H'?????

Bit 31 ~ 20: Reserved bits, ignored in write operation, always 0 in read operation.

- **SDR (Bit 19 ~ 0):** Status Data Register, a 20-bit read only register, 20-bit data from AC97 Controller Status Data Port (Slot 2 of Input Frame). The Register Data of external Codec is returned. The write operation is ignored.

18.2.1.10 AC97 Controller FIFO Data Port Registers (ACODR & ACIDR)

The AC97 Controller Out FIFO Data Port Register (ACODR) is the data input port of transmit FIFO and it is write-only. The AC97 Controller In FIFO Data Port Register (ACIDR) is the data output port from the receive FIFO and it is read-only. Each FIFO is 16 Samples deep. Care should be taken to monitor the status register to insure that there is room for data in the FIFO for a read or write transaction. This will be taken care of automatically by using DMA.

	# H'E000 0924 ACODR	
Bit:	31 ~ N ^N	N-1 ^N ~ 0
Read:		
Write:		DATA
Reset:	Undefined	

	# H'E000 0928 ACIDR	
Bit:	31 ~ N ^N	N-1 ^N ~ 0
Read:		DATA
Write:		
Reset:	H'000	Undefined

Note: The N indicates the size of audio sample. The OASS and IASS in ACCR2 specify the sample (as known PCM data) size. The default of sample size is 20 bit. (See 18.2.1.3).

18.3 Operation

AC97 Controller supports a full-duplex serial audio interface between an external codec and itself. AC97 controller is used to attach to one external AC'97-compliant codecs. The external codec generates a 12.288 MHz clock and sends it as BIT_CLK to the AC97 Controller. In the AC97 Controller, the clock is divided by 256 to yield a 48 KHz audio-sample clock. This is used internally, and sent back to the codec as SYNC. This signal defines the start of each "frame". There are 13 slots within a frame and each slot (except the first) contains 20 bits. The timing of BIT_CLK and SYNC are applied by controllers at each end to sample the two serial data streams - one inbound (SDATA_IN), the other outbound (SDATA_OUT) - and capture serial audio bits going in both directions. The AC97 audio output/input frame is described in following figure.

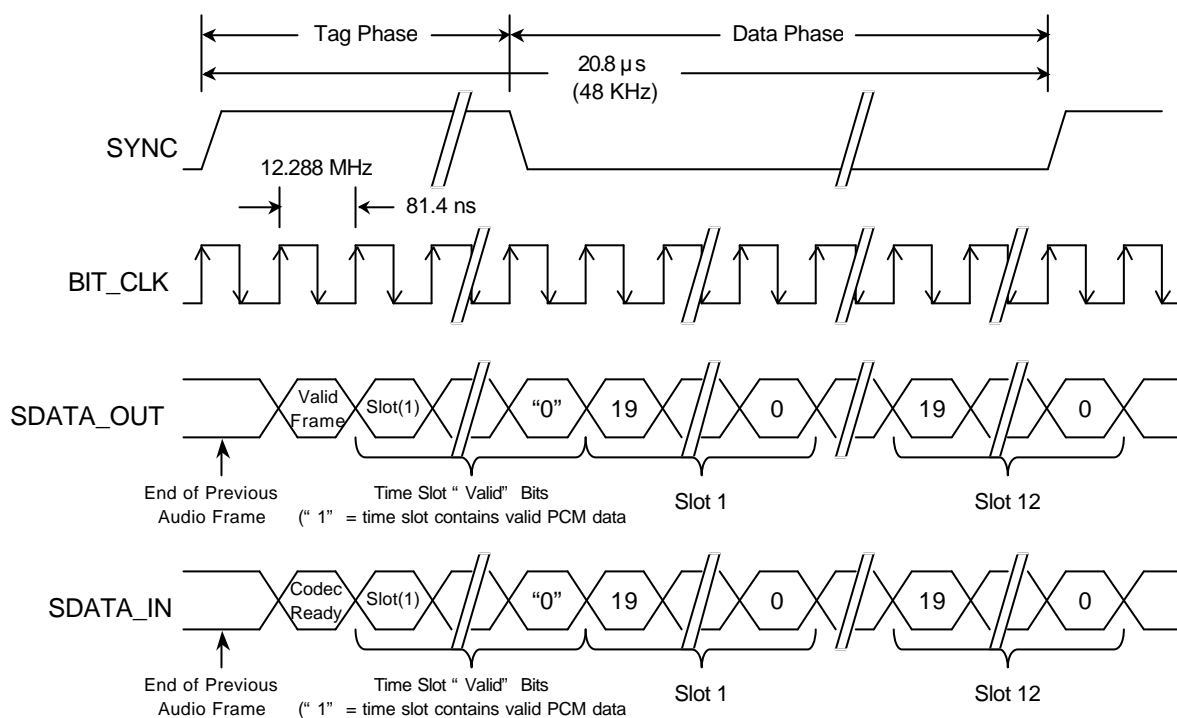


Figure 18-2 AC-link Audio Output/Input Frame

18.3.1 Initialization and Reset

Audio data is transferred between the AC97 Controller and an external codec through the AC-link interface. The system accesses the AC97 Controller through OCP Bus. The system must be initiate the AC'97 codec and AC97 Controller after system power-on or reset. And, If the error has been happened in data transferring, or because of other reasons, the system must be to resettlement the AC97 controller (and AC'97 codec). The initial/reset sequence is described in following flow chart.

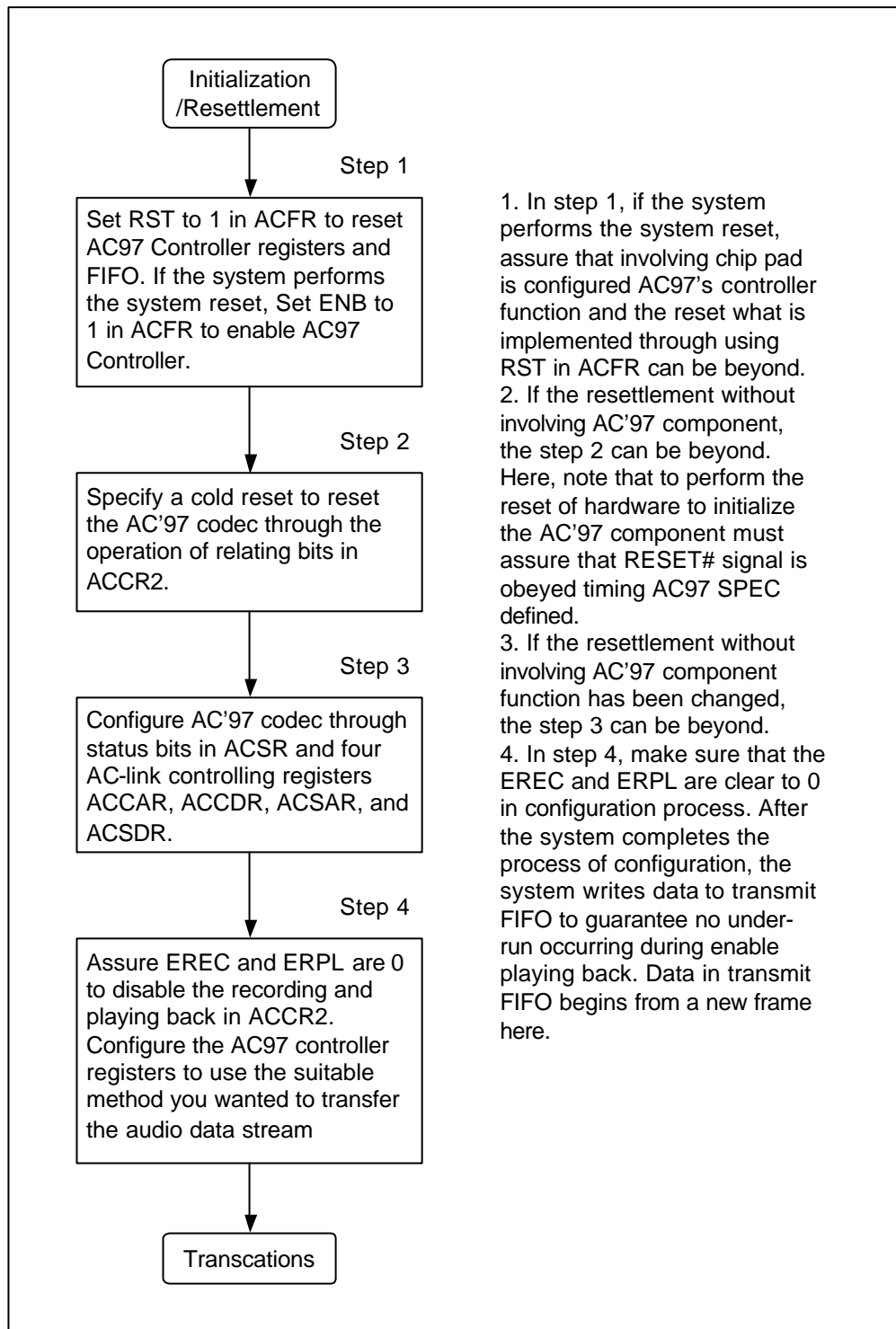


Figure 18-3 AC97 Controller Initialization/Resettlement Flowchart

18.3.2 Data Transfer

FIFO status is available to the system processor in the AC97 Controller Status Registers (ACSR). The system can be set the enable bits in ACCR2, ACIER to support the various transfer methods or their combination. Note, the system can use different transfer method between the transmission and the reception.

Outgoing data (from system to codec) is written through the OCP Bus to the AC97 Controller's Transmit FIFO. The AC97 controller then takes the data from the FIFO, serializes it, and sends it over the serial wire (SDATA_OUT) to the codec.

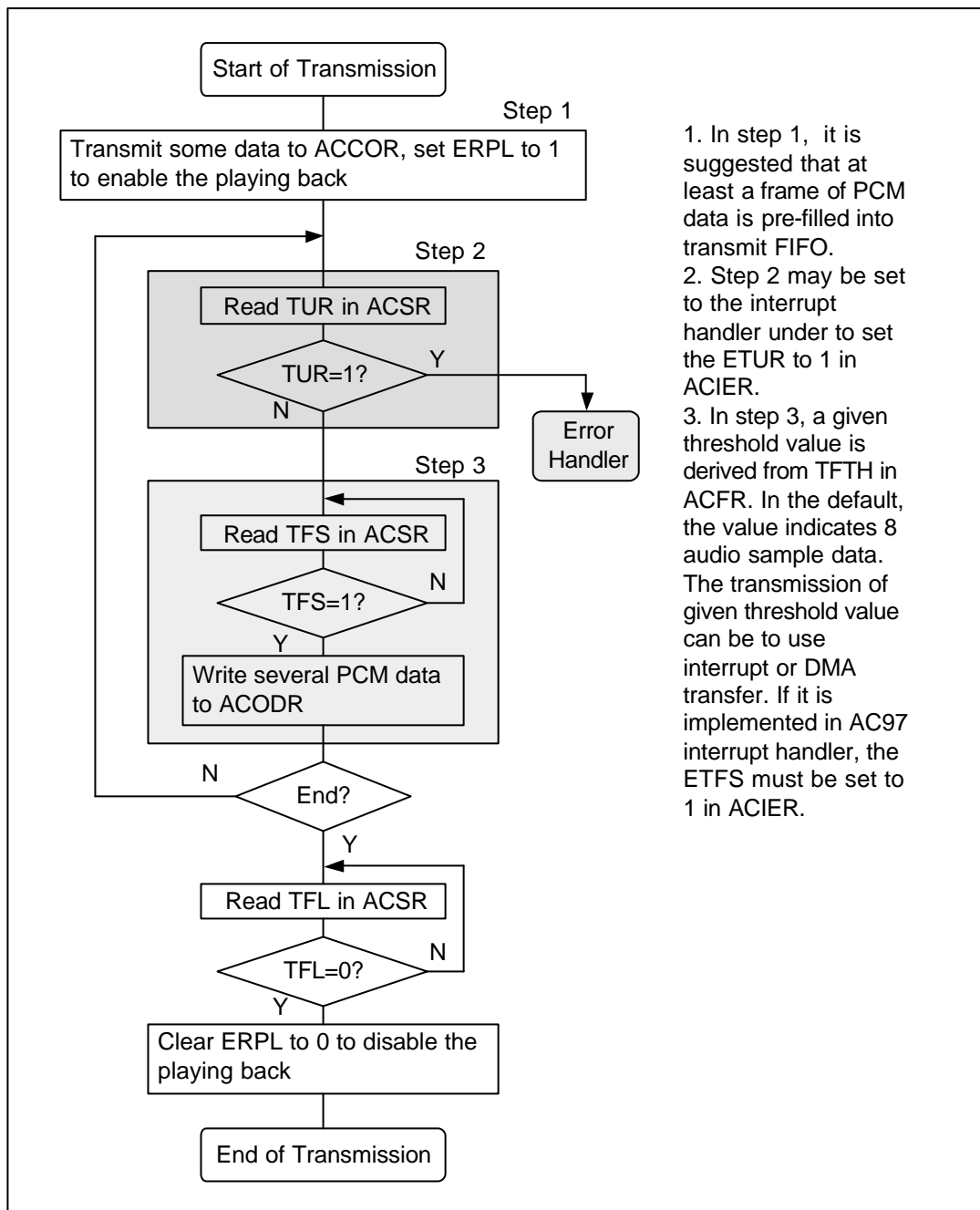


Figure 18-4 AC97 Controller Transmission Flowchart

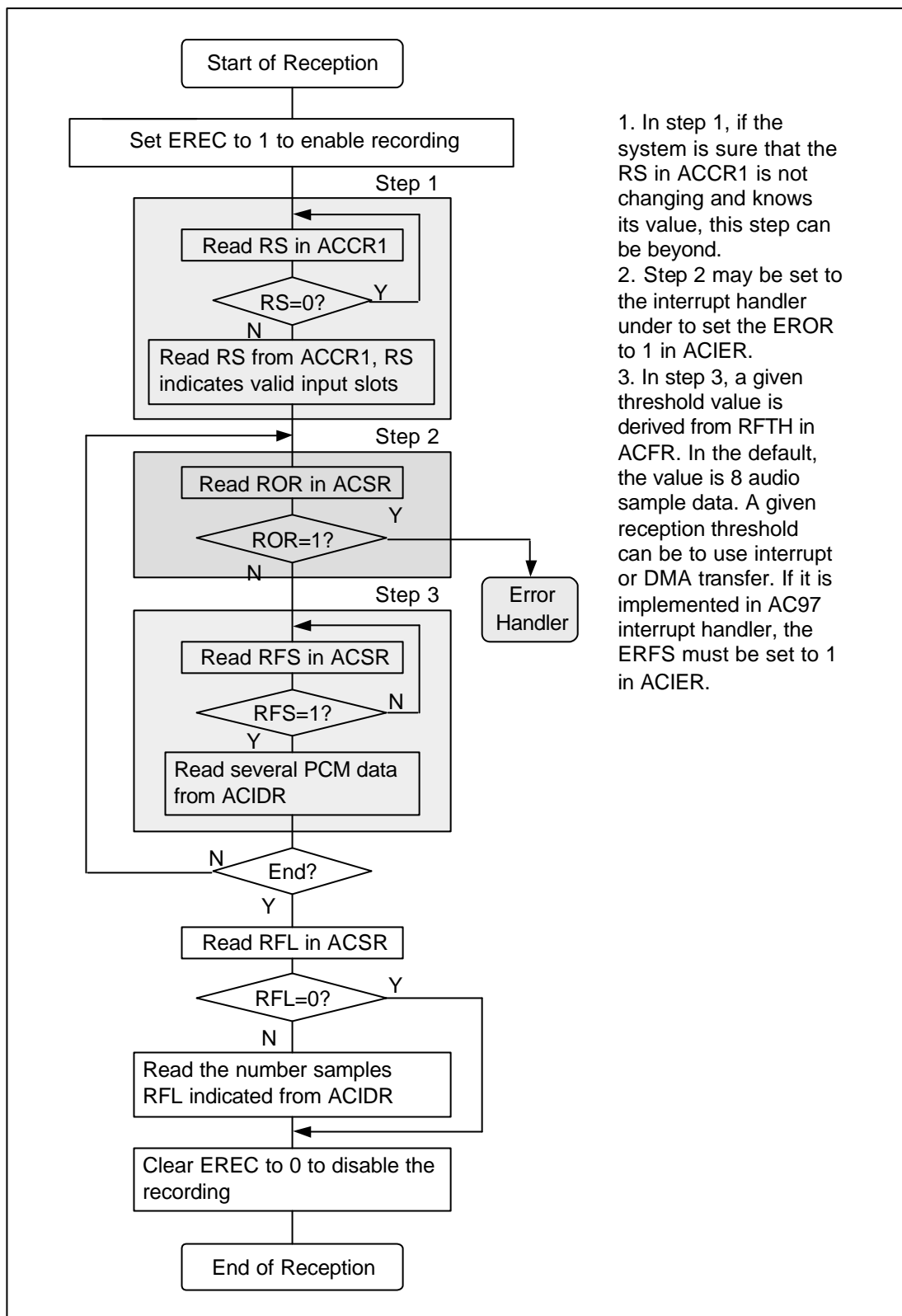


Figure 18-5 AC97 Controller Reception Flowchart

Incoming data from the external codec (on SDATA_IN) is converted to parallel words and stored in the Receive FIFO buffer. If ERFS is 1, a programmable threshold may be triggering an interrupt to the Interrupt Controller. The interrupt service routine responds by identifying the source of the interrupt and then doing a read from the inbound FIFO buffer.

The error handler is to perform the corresponding handle activity with various error conditions. If AC97 controller and AC97 codec are reset by the hardware reset, the system will perform an initial sequence. If only a register reset or FIFO flush (implementation through RST in ACFR) is demanded, the system will perform a reset sequence. In the meantime, the writing or reading data to FIFO is starting from the first valid slot of a frame.

18.3.3 DMA Audio Data Transfers

The AC97 Controller can transfer data directly to and from system memory if DMA operation is enabled (TDMS and RDMS are set to 1). In the meantime, the interrupt ought to be disabled (ETFS and ERFS are set to 0). The TTFH and RFTH in ACFR determine the number of transfer data every DMA request. The DMA configures that the destination address is pointed to ACODR in transmission and the source address is pointed to ACIDR in reception.

When data is to be transferred (generally in response to a Transmit or Receive FIFO condition), the AC97 Controller requests the service using ac97_rece_dreq_n or ac97_xmit_dreq_n.

The DMA Controller drives address onto the OCS bus through PBB selecting AC97 Controller (ACODR or ACIDR) and then AC97 Controller drives data onto the OCP bus through OCS bus (converted by PBB) transferring to or from system memory.

18.3.4 External Codec Registers Access Operation

The ACCAR, ACCDR, ACSAR, ACSDR and CADT, SADR of ACSR are used to access the registers that locate in the external AC'97 codec [see AC'97 Component Specification, Revision 2.2, Section 5.7]. The function that to access external codec registers is independent of audio data transferring operation. The ACCAR and ACCDR are used to send an operation command to external AC'97 codec [see AC'97 Component Specification, Revision 2.2, Section 4.3]. The ACSAR and ACSDR are used to receive a status register contain from external AC'97 codec [see AC'97 Component Specification, Revision 2.2, Section 4.4]. Because of the codec registers read/write command sending and the status registers contain returning is asynchronous, the operation flow is separated and respectively shown in following.

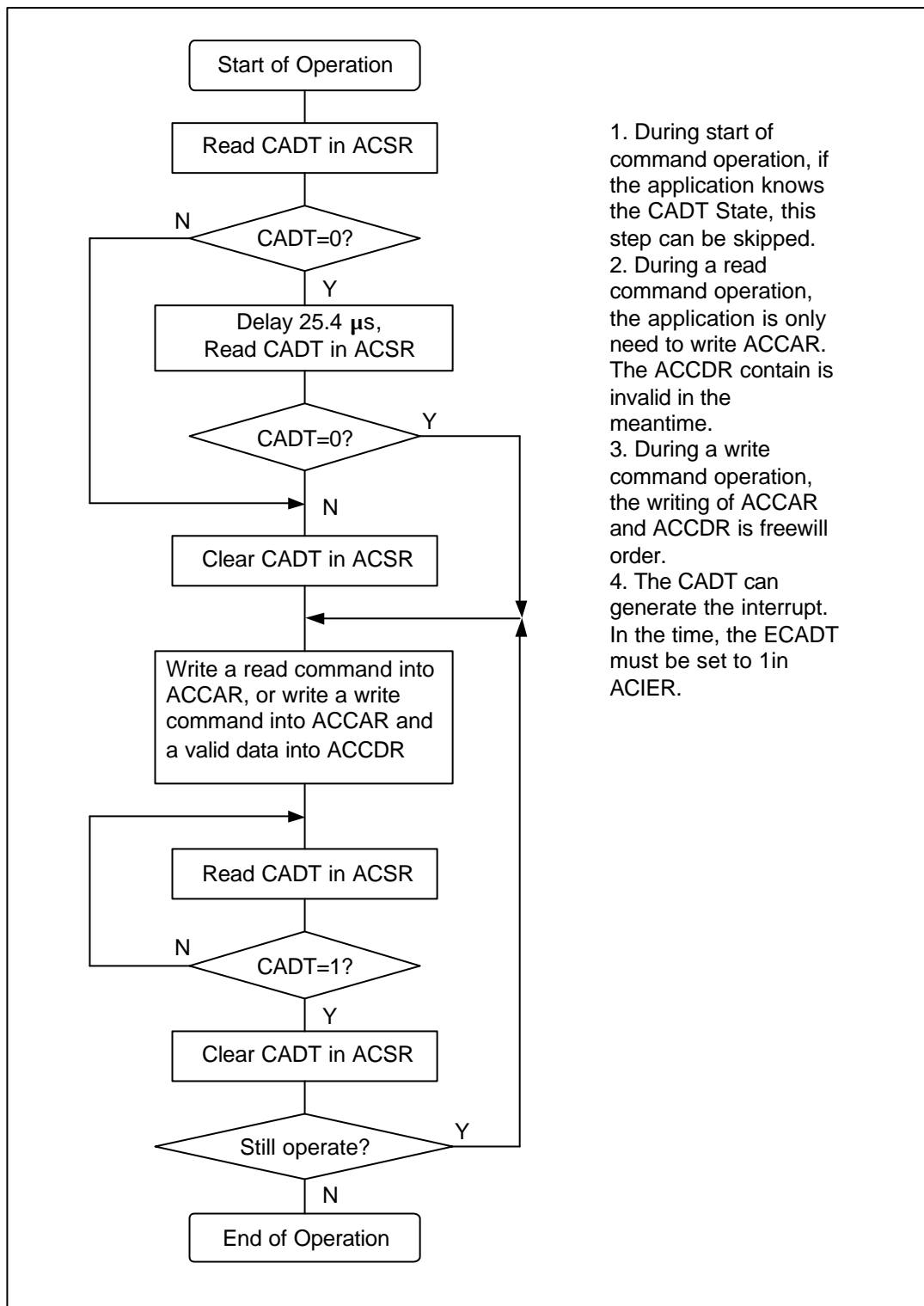


Figure 18-6 AC97 Codec Register Access Command Send Flow Chart

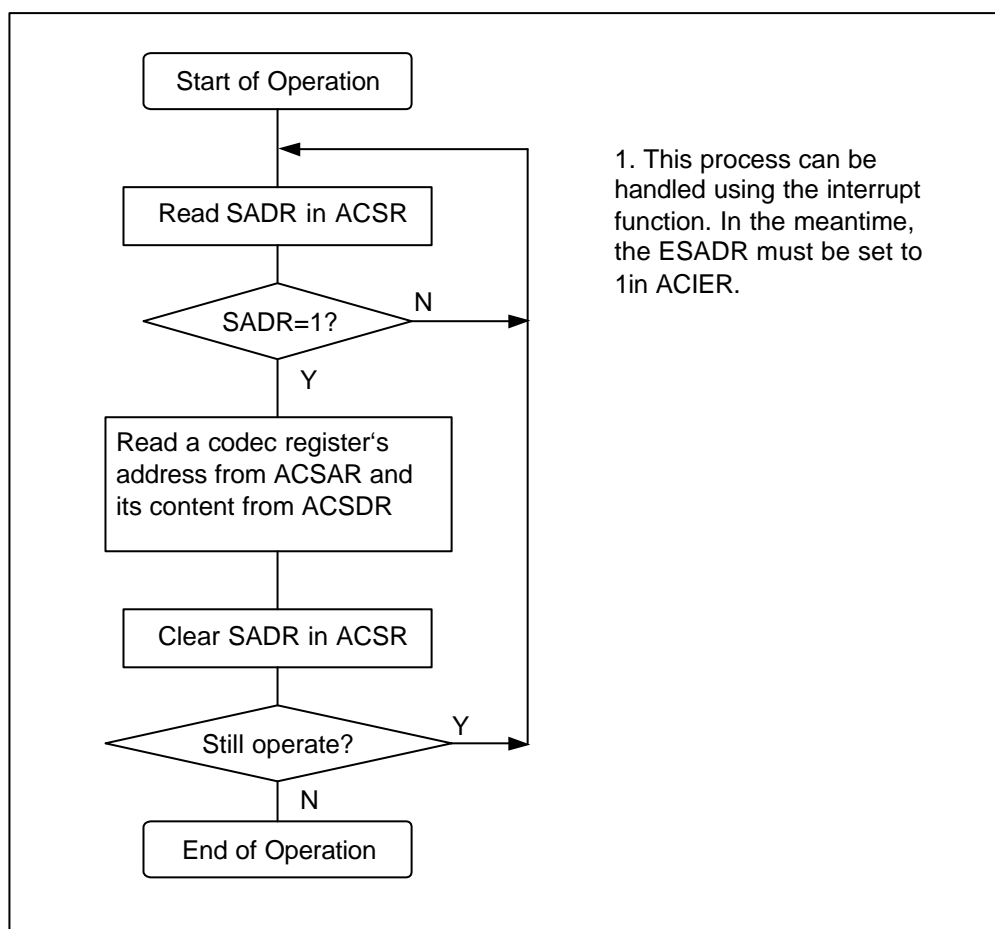


Figure 18-7 Bring AC97 Codec Register Back Reception Flow Chart

18.3.5 Power Down Mode

The AC-link signals can be placed in a low power mode. When AC'97's Power-down Register (26h), is programmed to the appropriate value, both BIT_CLK and SDATA_IN will be brought to and held at a logic low voltage level.

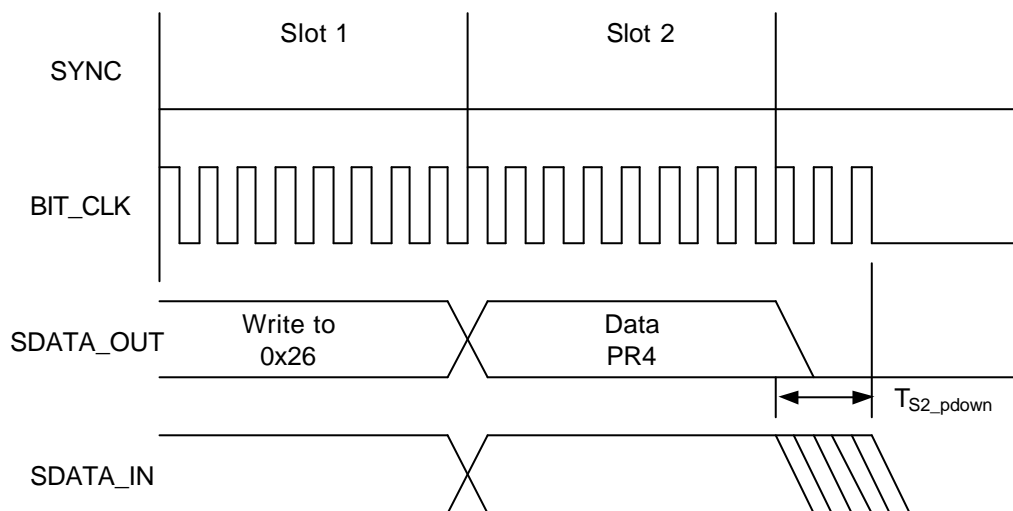


Figure 18-8 AC-Link Low Power Mode Timing

Table 18-5 AC-Link Low Power Mode Timing Parameters

Parameter	Symbol	Min	Type	Max	Units
End of Slot 2 to BIT_CLK, SDATA_IN low	T_{S2_pdown}	-	-	1.0	us

There are two methods for bringing the AC-link out of a low power, halted mode. Regardless of the method, it is the AC97 Controller that performs the wake-up task.

18.3.6 AC'97 Reset

AC-link protocol provides for a "Cold AC'97 Reset", and a "Warm AC'97 Reset". The current power down state would ultimately dictate which form of AC '97 reset is appropriate. Unless a "cold" or "register" reset (a write to the Reset register) is performed, wherein the AC '97 registers are initialized to their default values, registers are required to keep state during all power down modes.

Once powered down, re-activation of the AC-link via re-assertion of the SYNC signal must not occur for a minimum of four audio frame times following the frame in which the power down was triggered. When AC-link powers up it indicates readiness via the Codec Ready bit (input slot 0, bit 15).

18.3.6.1 Cold AC'97 Reset

A cold reset is achieved by asserting RESET# for the minimum specified time. By driving RESET# low, BIT_CLK, and SDATA_OUT will be activated, or re-activated as the case may be, and all AC'97 control registers will be initialized to their default power on reset values. RESET# is an asynchronous AC'97 input.

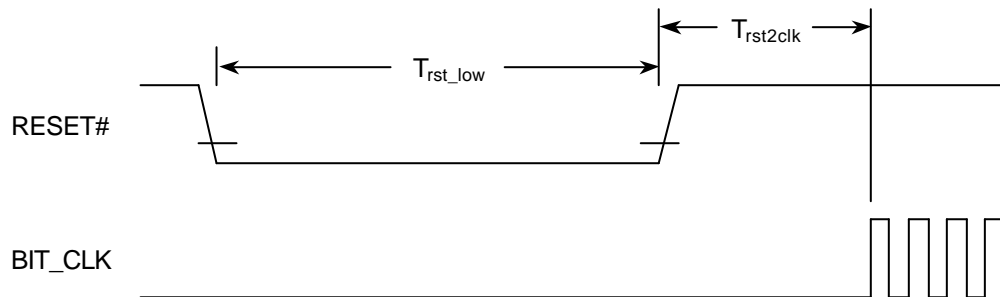


Figure 18-9 Cold Reset Timing

Table 18-6 Cold Reset Timing parameters

Parameter	Symbol	Min	Type	Max	Units
RESET# active low pulse width	T_{rst_low}	1.0	-	-	us
RESET# inactive to BIT_CLK startup delay	$T_{rst2clk}$	162.8	-	-	Ns

18.3.6.2 Warm AC '97 Reset

A warm AC'97 reset will re-activate the AC-link without altering the current AC'97 register values. Driving SYNC high for a minimum of 1us in the absence of BIT_CLK signals a warm reset.

Within normal audio frames SYNC is a synchronous AC'97 input. However, in the absence of BIT_CLK, SYNC is treated as an asynchronous input used in the generation of a warm reset to AC'97.

AC'97 MUST NOT respond with the activation of BIT_CLK until AC '97 has sampled SYNC low again. This will preclude the false detection of a new audio frame.

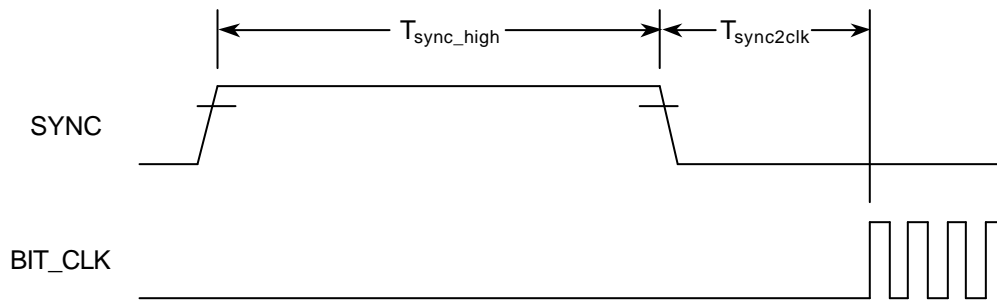


Figure 18-10 Warm Reset Timing

Table 18-7 Warm Reset Timing Parameters

Parameter	Symbol	Min	Type	Max	Units
SYNC active high pulse width	T_{sync_high}	1.0	-	-	us
SYNC inactive to BIT_CLK startup delay	$T_{sync2clk}$	162.8	-	-	Ns

18.3.7 ATE Test Mode

AC'97 has two test modes. One is for ATE in circuit test and the other is for vendor-specific tests. AC'97 enters the ATE in circuit test mode if SDATA_OUT is sampled high at the trailing edge of RESET#. AC'97 enters the vendor-specific test mode when coming out of RESET if SYNC is high. These cases will never occur during standard operating conditions.

Regardless of the test mode, the AC '97 Controller must issue a cold reset to resume normal operation of the AC '97 Codec.

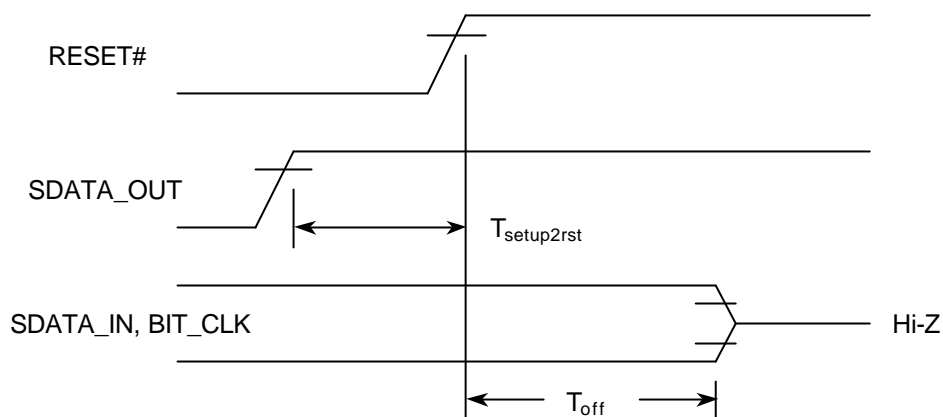


Figure 18-11 ATE Test Mode Timing

Note: The ATE Test Mode can be entered and escaped through that the AC97 pins used as GPIO.

Table 18-8 ATE Test Mode Timing Parameters

Parameter	Symbol	Min	Type	Max	Units
Setup to trailing edge of RESET# (also applies to SYNC)	T _{setup2rst}	15.0	-	-	ns
Rising edge of RESET# to Hi-Z delay	T _{off}	-	-	25.0	ns

Notes:

1. All AC-link signals are normally low through the trailing edge of RESET#. Bringing SDATA_OUT high for the trailing edge of RESET# causes AC '97's AC-link outputs to go high impedance which is suitable for ATE in circuit testing.
2. Bringing SYNC high for the trailing edge of RESET# can enter a vendor-specific internal test mode. This mode has no effect on AC '97 AC-link output signal levels.
3. Once either of the two test modes has been entered, AC '97 must be issued another RESET# with all AC-link signals low to return to the normal operating mode.

19 Boot Time Configuration

There are some function modes that must be set during the system reset phase by setting the state of the function mode pins, and cannot be changed after the reset. These function mode pins are multiple usage. When the system runs normally based on the mode that is set when reset, these pins are used for different function of the system.

The pins can be found in the pin list table of the Arca210. The detailed information about their function is described in relevant chapter of the manual.

19.1 Bank 0 Memory Size – MD_mem

The memory data bus width can be set for bank 0 at power-on reset. The correspondence between the external pin (MD_mem[1:0]) and memory size is listed in the table below.

MD_mem[1]	MD_mem[0]	Bank 0 Memory Size
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	Reserved

19.2 Clock Mode – MD_clk

Following table shows PLL status and iclk, sclk, and pclk division ratio, CFCR initial value, input and output clock frequency range in different clock mode.

Clock mode	MD_clk [2:0]	PLL	CFCR initial value	CPU clock (icl) divider	System clock (sclk) divider	Device clock (pclk) divider
0	000	ON	H'xxxx 0800	x 1	x 1	x 1
1	001	ON	H'xxxx 1848	x 1	x 1/2	x 1/2
2	010	ON	H'xxxx 2890	x 1	x 1/3	x 1/3
3	011	ON	H'xxxx 38D8	x 1	x 1/4	x 1/4
4	100	OFF	H'xxxx 4000	x 1	x 1	x 1
5	101	ON	H'xxxx 5920	x 1	x 1/6	x 1/6
6	110	ON	H'xxxx 6968	x 1	x 1/8	x 1/8

Note: In clock mode 4, extal is used directly as internal clocks (icl/sclk/pclk). The dividers are bypassed. Any low power function and frequency change will not affect the frequency of iclk, sclk and pclk, their frequency are always equal to that of external crystal.

19.3 PLL Feedback Division Ratio Bit 5-1 – MD_pll_f

Pins MD_pll_f[5:1] are the bit 5-1 of PLL Feedback Division Ratio. PLL Feedback Division Ratio (PLLFD): Controls the PLL feedback divider. The MD_PLLFD[5:1] pins are registered into PLLFD5~1 during power-on. PLLFD8~6 and PLLFD0 are initialized to 0 by a power-on reset

19.4 PLL Output Division Ration Bit 0 – MD_pll_od

PLL Output Division Ratio (PLLOD): Controls the PLL output divider. The MD_PLLOD[1:0] pin is registered into PLLOD0-1 during power-on reset.

Bit 17: PLLOD1	Bit 16: PLLOD0	Divided by
0	0	1
0	1	2
1	0	2
1	1	4

19.5 PCI Host Mode – MD_pci

The operation mode is controlled by pin MD_pci when system reset.

MD_pci	Description
0	Satellite mode. A device in a hosted PCI system. Configured by the host, for example, a PCI card in a PC.
1	Hosts mode. The main processor in a PCI system. Responsible for configuration, and interrupt handling.

19.6 PCI Host Arbiter – MD_pciarb

Arca210 has an internal PCI-bus arbiter that can be used when Arca210 is in host mode.

MD_pciarb	Description
0	Disable the Arca210 internal PCI arbiter when in PCI host mode.
1	Enable the Arca210 internal PCI arbiter when in PCI host mode.

19.7 TAP Select – MD_tap_sel

Arca210 has an internal core TAP and boundary TAP.

MD_tap_sel	Description
0	Select and enable boundary TAP
1	Select and enable core TAP

19.8 OCS-Bus Data Access Timeout

The data access on OCS-bus, may be no reply by any slave device of OCS-bus. The reason may be an invalid address set by programmer, or the slave has errors. It will block up the system processing.

In the arbiter of OCS-bus – SARB, time-out register is used to set a time-out value which is compared with an internal counter. During the data access phase of OCS-bus that is not acknowledged, if the internal counter increase to equal this time-out register, it means a timeout has occurred. The arbiter will save the address of the failed access to register SAADDR, and acknowledge the access to not block the system.

Time-out register is initialized to H'0000FFFF by a power-on reset.

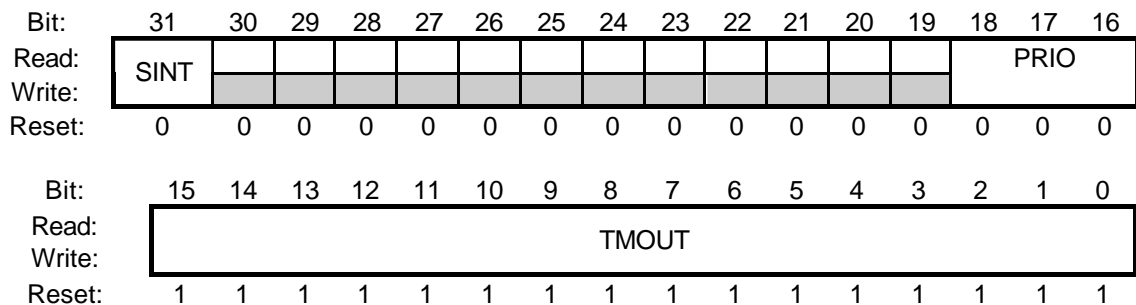
When time-out occur, SARB would send out an interrupt to INTC to indicate that current access is error because no any device answer this access, data read or write by master is invalid. The master must read out this time-out address and store it before clear SINT bit to zero, otherwise, time-out address would be cover by new address when SINT bit is cleared.

Table 19-1 SARB Registers

Name	Full Name	R/W	Initial Value	Address	Access Size
SATMOUT	SARB time-out register	R/W	H'0000FFFF	H'E1010000	32
SAADDR	SARB address record register	R/W	H'xxxxxxxx	H'E1010004	32

19.8.1 Timeout Register

Timeout register is used to set the time-out value, arbitration priority and record a time-out interrupt event.



– **Bits 30 ~ 19:** Reserved bits. These bits always read 0. The write value are ignored.

– **Bit 31 — (SINT):** This bit is use for marking a time-out interrupt.

Bit 31 (SINT)	Meaning	
0	No time-out occur	(Initial value)
1	Time-out occur	

- **Bit 18 ~ 16 — (PRIO):** These bits are use to set the priority for each master.

Bit 18 ~ 16 (PRIO)	Priority (High -> Low)	Application
000	DMA > MAC > PCI > USB > CBB (Initial value)	DMA supports AC97, floppy etc, and there are four channels in DMA
001	DMA > PCI > MAC > USB > CBB	
010	PCI > DMA > MAC > USB > CBB	PCI supports ALI South Bridge, and so many PCI master devices
011	PCI > MAC > DMA > USB > CBB	
100	USB > DMA > MAC > PCI > CBB	USB supports storage, video etc devices
101	USB > PCI > DMA > MAC > CBB	
110	CBB > DMA > MAC > PCI > USB	If ARCA cache miss too long to response INTC request
111	CBB > PCI > DMA > MAC > USB	

- **Bit 15 ~ 0 — (TIMEOUT):** These bits are use to set a time-out cycles.

Bit 15~0 (TMOUT)	Meaning	
'h0000~'hFFFF	Time-out cycles	(65535 cycles is Initial value)

19.8.2 Address Record register

SARB address record register is used for store a time-out address. When no time-out occurred, it equal to each access address, when time-out occur, this register be freezed until the SINT bit in time-out register is cleared. CPU can read this register to known the time-out address before it clear the SINT bit.

Bit:	31	12	11	10	9	8	7	6	5	4	3	2	1	0
Read:															
Write:															
Reset:															

20 Pin Description

Arca210 has 319 pins, 226 of which are function pins. The following figures and tables list all the function pins.

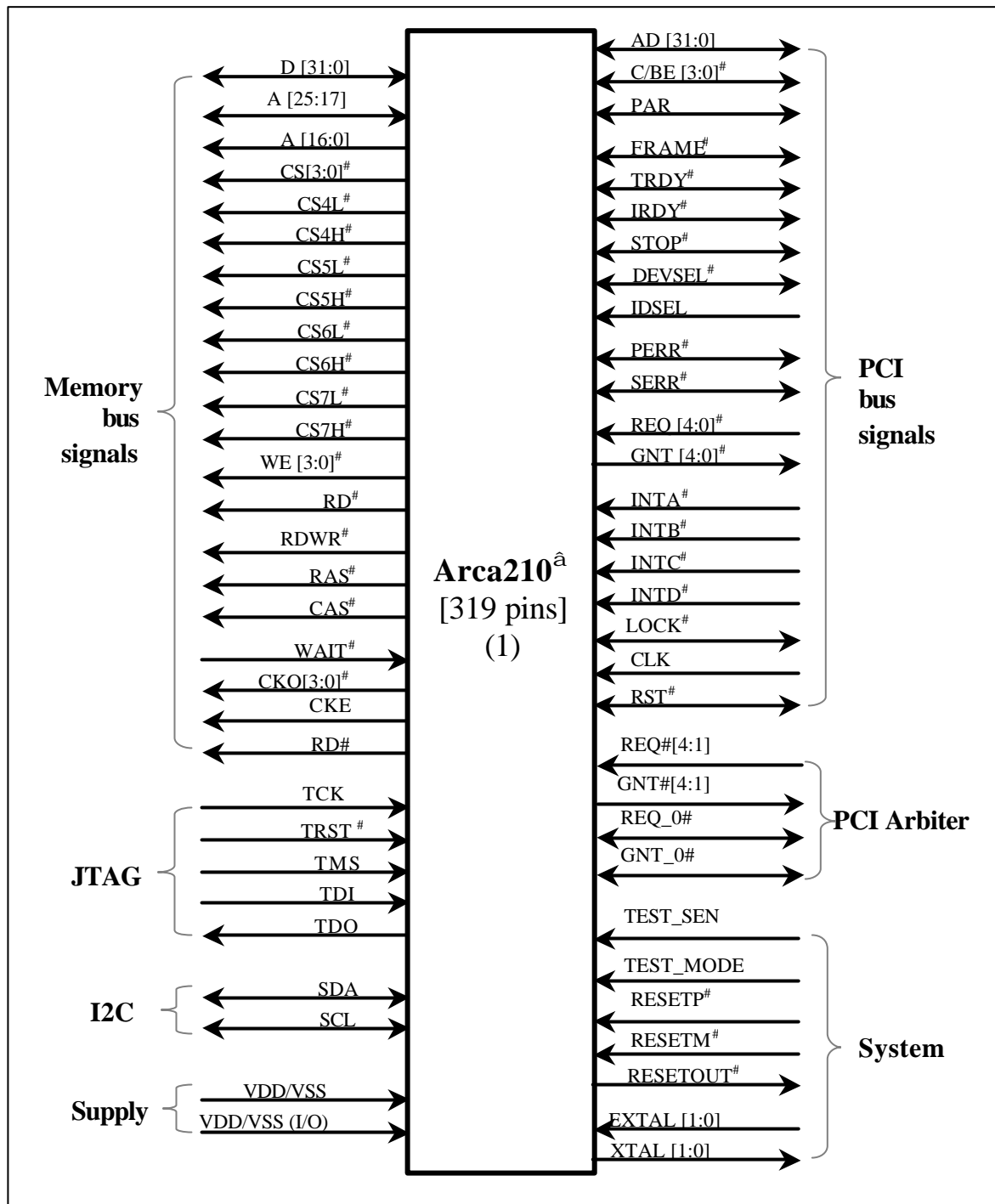


Figure 20-1 Arca210 Pin Diagram (1)

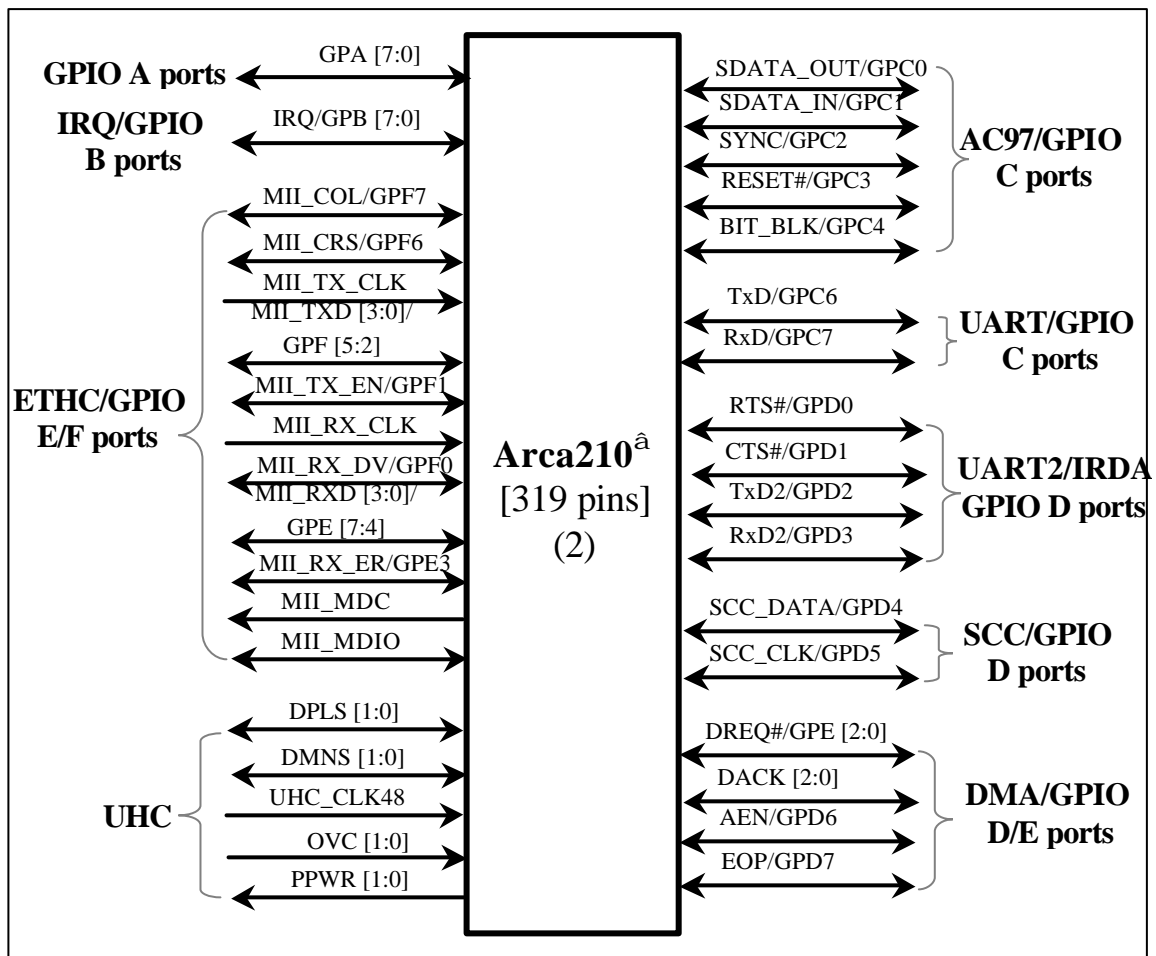


Figure 20-2 Arca210 Pin Diagram (2)

Table 20-1 EMI pins (total 85)

Pin Name	Size	I/O	Pin Description
DAT[31:0]	32	IO	memory data bus
ADD[16:2]	15	O	SDRAM memory address
ADD[17]	1	O	non-SDRAM memory addr
ADD [25:18]	8	IO	non-SDRAM memory addr / MD_pll_f, MD_clk
ADD[1:0]	2	IO	non-SDRAM memory address / MD_pll_od
CS0_	1	O	Chip select 0
CS1_	1	O	Chip select 1
CS2_	1	O	Chip select 2
CS3_	1	O	Chip select 3
CS4L_	1	O	Chip select 4 low
CS4H_	1	O	Chip select 4 high
CS5L_	1	O	Chip select 5 low
CS5H_	1	O	Chip select 5 high
CS6L_	1	O	Chip select 6 low
CS6H_	1	O	Chip select 6 high
CS7L_	1	O	Chip select 7 low
CS7H_	1	O	Chip select 7 high
RAS_	1	O	Row address strobe
CAS_	1	O	Column address strobe
WE0_	1	O	Byte 0 write enable
WE1_	1	O	Byte 1 write enable
WE2_	1	O	Byte 2 write enable
WE3_	1	O	Byte 3 write enable
RDWR_	1	O	1 – read, 0 – write
RD_	1	O	SRAM read strobe
WE_	1	O	Byte Control SRAM write strobe
CKE	1	O	Clock enable for SDRAM
CKO0	1	O	Clock for SDRAM
CKO1	1	O	Clock for SDRAM
CKO2	1	O	Clock for SDRAM
CKO3	1	O	Clock for SDRAM
WAIT_	1	I	Wait for slow memory

Table 20-2 GPIO A pins (total 8)

Pin Name	Size	I/O	Pin Description
GPA0	1	IO	GPIO A Inout0
GPA1	1	IO	GPIO A Inout1
GPA2	1	IO	GPIO A Inout2
GPA3	1	IO	GPIO A Inout3
GPA4	1	IO	GPIO A Inout4
GPA5	1	IO	GPIO A Inout5
GPA6	1	IO	GPIO A Inout6
GPA7	1	IO	GPIO A Inout7

Table 20-3 IRQ/GPIO B pins (total 8)

Pin Name	Size	I/O	Pin Description
IRQ0/GPB0	1	IO	GPIO B Inout0/ IRQ0 Input
IRQ1/GPB1	1	IO	GPIO B Inout1/ IRQ1 Input
IRQ2/GPB2	1	IO	GPIO B Inout2/ IRQ2 Input
IRQ3/GPB3	1	IO	GPIO B Inout3/ IRQ3 Input
IRQ4/GPB4	1	IO	GPIO B Inout4/ IRQ4 Input
IRQ5/GPB5	1	IO	GPIO B Inout5/ IRQ5 Input
IRQ6/GPB6	1	IO	GPIO B Inout6/ IRQ6 Input
IRQ7/GPB7	1	IO	GPIO B Inout7/ IRQ7 Input

Table 20-4 AC97/GPIO C pins (total 5)

Pin Name	Size	I/O	Pin Description
SDATA_OUT/GPC0	1	IO	AC97 SDATA_OUT/GPIO C Inout0
SDATA_IN/GPC1	1	IO	AC97 SDATA_IN/GPIO C Inout1
SYNC/GPC2	1	IO	AC97 SYNC/GPIO C Inout2
RESET_/GPC3	1	IO	AC97 RESET_/GPIO C Inout3
BIT_CLK/GPC4	1	IO	AC97 BIT_CLK/GPIO C Inout4

Table 20-5 UART/GPIO C pins (total 2)

Pin Name	Size	I/O	Pin Description
TxD/GPC6	1	IO	UART TxD/GPIO C Inout6
RxD/GPC7	1	IO	UART RxD/GPIO C Inout7

Table 20-6 UART2/IRDA GPIO D pins (total 4)

Pin Name	Size	I/O	Pin Description
RTS_/GPD0	1	IO	UART2 RTS_/GPIO D Inout0
CTS_/GPD1	1	IO	UART2 CTS_/GPIO D Inout1
TxD2/GPD2	1	IO	UART2/IRDA TxD /GPIO D Inout2
RxD2/GPD3	1	IO	UART2/IRDA RxD /GPIO D Inout3

Table 20-7 SCC/GPIO D pins (total 2)

Pin Name	Size	I/O	Pin Description
SCC_DATA/GPD4	1	IO	SCC_DATA/GPIO D Inout4
SCC_CLK/GPD5	1	IO	SCC_CLK/GPIO D Inout5

Table 20-8 DMA/GPIO D/E pins (total 8)

Pin Name	Size	I/O	Pin Description
DREQ0_/GPE0	1	IO	DMA External Request 0/GPIO E Inout0
DREQ1_/GPE1	1	IO	DMA External Request 1/GPIO E Inout1
DREQ2_/GPE2	1	IO	DMA External Request 2/GPIO E Inout2
DACK0	1	IO	DMA External Acknowledge 0/MD_tap_sel
DACK1	1	IO	DMA External Acknowledge 1/MD_mem[0]
DACK2	1	IO	DMA External Acknowledge 2/MD_mem[1]
AEN/GPD6	1	IO	DMA Transaction Enable/GPIO D Inout6
EOP/GPD7	1	IO	DMA Transaction Complete/GPIO D Inout7

Table 20-9 I2C pins (total 2)

Pin Name	Size	I/O	Pin Description
SDA	1	IO	I2C Serial Data Input/Output
SCL	1	IO	I2C Serial Clock Input/Output

Table 20-10 SYSTEM pins (total 9)

Pin Name	Size	I/O	Pin Description
EXTAL	1	I	System Crystal Input + Crystal Output
XTAL	1	O	
EXTAL2	1	I	RTC Crystal Input + Crystal Output
XTAL2	1	O	
RESETP_	1	I	System Power on reset input
RESETM_	1	I	System Manual reset input
RESETOUT_	1	O	Chip reset output
TEST_SEN	1	I	scan enable for scan-reg
TEST_MODE	1	I	Chip test mode

Table 20-11 JTAG pins (total 5)

Pin Name	Size	I/O	Pin Description
TRSTN_	1	I	JTAG Reset
TMS	1	I	JTAG Mode Select
TDI	1	I	JTAG Serial Data Input
TCK	1	I	JTAG Clock
TDO	1	O	JTAG Serial Data Output

Table 20-12 PCI pins (total 52)

Pin Name	Size	I/O	Pin Description
AD	32	IO	PCI: Address/Data bus
C_	4	IO	PCI: Command/Byte enable
PAR	1	IO	PCI: Even parity across AD[31:0] and C_[3:0].
CLK	1	I	PCI: PCI clock
RST_	1	IO	PCI: PCI reset
FRAME_	1	IO	PCI: PCI frame
TRDY_	1	IO	PCI: target ready
IRDY_	1	IO	PCI: initiator ready
STOP_	1	IO	PCI: current target request master to stop current transaction
DEVSEL	1	IO	PCI: device select
LOCK_	1	I	PCI: lock access to memory /scan chain in 6
IDSEL_	1	I	PCI: initialization device select/scan chain in 7
PERR_	1	IO	PCI: parity error
SERR_	1	O	PCI: system error
INTA_	1	IO	PCI: interrupt A
INTB_	1	I	PCI: interrupt B / scan chain in 8
INTC_	1	I	PCI: interrupt C / scan chain in 9
INTD_	1	I	PCI: interrupt D / scan chain in 10

Table 20-13 PCI Arbiter pins (total 10)

Pin Name	Size	I/O	Pin Description
REQ_0_	1	IO	PCIART: bus request 0, used by Arca210
REQ_1_	1	I	PCIART: bus request 1 / scan chain in 1
REQ_2_	1	I	PCIART: bus request 2 / scan chain in 2
REQ_3_	1	I	PCIART: bus request 3 / scan chain in 3
REQ_4_	1	I	PCIART: bus request 4 / scan chain in 4
GNT_0_	1	IO	PCIART: bus grant 0, used by Arca210
GNT_1_	1	O	PCIART: bus grant 1 / scan chain out 1
GNT_2_	1	O	PCIART: bus grant 2 / scan chain out 2
GNT_3_	1	O	PCIART: bus grant 3 / scan chain out 3
GNT_4_	1	O	PCIART: bus grant 4 / scan chain out 4

Table 20-14 ETHC/GPIO E/F pins (total 17)

Pin Name	Size	I/O	Pin Description
MII_COL/GPF7	1	IO	Ethernet Collision /GPIO F Inout0
MII_CRS/GPF6	1	IO	Ethernet Carrier Sense /GPIO F Inout1
MII_TX_CLK	1	I	Ethernet Transmit Clock
MII_TXD[3]/GPF5	1	IO	Ethernet Transmit Data/GPIO F Inout5
MII_TXD[2]/GPF4	1	IO	Ethernet Transmit Data/GPIO F Inout4
MII_TXD[1]/GPF3	1	IO	Ethernet Transmit Data/GPIO F Inout3
MII_TXD[0]/GPF2	1	IO	Ethernet Transmit Data/GPIO F Inout2
MII_TX_EN/GPF1	1	IO	Ethernet Transmit Enable/GPIO F Inout1
MII_RX_CLK	1	I	Ethernet Receive Clock
MII_RX_DV/GPF0	1	IO	Ethernet Receive Data Valid/GPIO F Inout0
MII_RXD[3]/GPE7	1	IO	Ethernet Receive Data/GPIO E Inout7
MII_RXD[2]/GPE6	1	IO	Ethernet Receive Data/GPIO E Inout6
MII_RXD[1]/GPE5	1	IO	Ethernet Receive Data/GPIO E Inout5
MII_RXD[0]/GPE4	1	IO	Ethernet Receive Data/GPIO E Inout4
MII_RX_ER/GPE3	1	IO	Ethernet Receive Error/GPIO E Inout3
MII_MDC	1	O	Ethernet Management Clock
MII_MDIO	1	IO	Ethernet Management Data Inout

Table 20-15 UHC pins (total 9)

Pin Name	Size	I/O	Pin Description
UHC_CLK48	1	I	USB Clock Input
DPLS0	1	IO	USB Port 0 Data Plus
DMNS0	1	IO	USB Port 0 Data Minus
OVC0	1	I	USB Port 0 Over-current Input
PPWR0	1	O	USB Port 0 Power Switch
DPLS1	1	IO	USB Port 1 Data Plus
DMNS1	1	IO	USB Port 1 Data Minus
OVC1	1	I	USB Port 1 Over-current Input
PPWR1	1	O	USB Port 1 Power Switch

Sum of all above pin = 226

Table 20-16 Test port / MD pins (5)

Pin Name	Size	I/O	Pin Description
TEST_PORT[4:0]	5	IO	test port / MD_pci, MD_pciarb

21 System Memory Map

Space Allocation: In the Arca210 architecture, both logical spaces and physical spaces have 32-bit address spaces. The 4Gbyte physical space is divided into several partitions for static memory, SDRAM, PCI and internal I/O. Following table shows the physical space map.

Table 21-1 Physical Address Space Map

Physical Address	Connectable Memory	Capacity
H'0000 0000 to H'03FF FFFF	Static memory bank 0	64 Mbytes
H'0400 0000 to H'07FF FFFF	Static memory bank 1	64 Mbytes
H'0800 0000 to H'0BFF FFFF	Static memory bank 2	64 Mbytes
H'0C00 0000 to H'0FFF FFFF	Static memory bank 3	64 Mbytes
H'1000 0000 to H'1FFF FFFF	SDRAM bank 4	256 Mbytes
H'2000 0000 to H'2FFF FFFF	SDRAM bank 5	256 Mbytes
H'3000 0000 to H'3FFF FFFF	SDRAM bank 6	256 Mbytes
H'4000 0000 to H'4FFF FFFF	SDRAM bank 7	256 Mbytes
H'7000 0000 to H'CFFF FFFF	Reserved	1792 Mbytes
H'E000 0000 to H'E3FF FFFF	PCI I/O space	64 Mbytes
H'E400 0000 to H'E7FF FFFF	PCI memory space 0	64 Mbytes
H'E800 0000 to H'EBFF FFFF	PCI memory space 1	64 Mbytes
H'EC00 0000 to H'FFFF FFFF	PCI memory space 2	64 Mbytes
H'F000 0000 to H'FFFF FFFF	Internal I/O	256 Mbytes

21.1 Internal Register Map Address and Definition

In Arca210 system, all of the internal registers of CPU-core can be accessed through core module interface CLD/CST instruction.

And the other registers of on chip devices can be accessed by means of memory-mapped load/store. Their memory-mapped addresses are allocated in space PS4: H'F0000000 – H'FFFFFFFF. CPU can access these registers by load/store instructions in privileged mode.

21.2 Address Space Allocation

Table 21-2 shows the address space allocated for 3 groups of Arca210 components.

Table 21-2 Address space allocation for different component groups

Component Group	Address Range	Address Space Allocated
CPU-Core Modules	ID: 0~8; CR: 0~1024	
OCS-Bus Devices	F0000000~FFFFFFFF	240MB, each device allocates 16MB
OCP-Bus Devices	FF000000~FFFFFFFF	16MB

Table 21-3 lists the base address of the memory-mapped registers for each module.

Table 21-3 Base address memory-mapped register for each module

Group	Module	Address	Note
CPU-Core	MMU	ID= 0; CR= 0~4	
	Cache	ID= 0; CR= 5	
OCS-Bus	Arbiter	F0000000	The arbiter of OCS-Bus
	EMI	FE000000	
	PCI	FD000000	
	DMA	FC000000	
OCP-Bus	UART	FFFFFFF00	
	INTC	FFFFFFE00	
	RTC	FFFFFFD00	
	TMU	FFFFFFC00	
	WDT	FFFFFFB00	
	CGU/PMC	FFFFFFA00	
	I2CI	FFFFFF800	
	GPIO	FFFFFF700	

21.3 Register Address

21.3.1 MMU

Table 21-4 MMU memory-mapped register

Name	Full Name	Address	Width
MCR	MMU Control Register	ID= 0; CR= 000	32
ASI	Address Space Identifier Register	ID= 0; CR= 001	32
CED	Configure Exchange Data Register	ID= 0; CR= 100	32
MEA	MMU Exception Address Register	ID= 0; CR= 010	32
TTB	Translation Table Base Register	ID= 0; CR= 011	32

21.3.2 Cache

Table 21-5 Cache memory-mapped register

Name	Full Name	Address	Width
CCR	Cache Control Register	ID= 0; CR= 101	32

21.3.3 OCS-Bus arbiter

Table 21-6 OCS-Bus arbiter memory-mapped register

Name	Full Name	Address	Width
SATMOUT	SARB Timeout register	F0000000	32
SAADDR	SARB Address record register	F0000004	32

21.3.4 EMI

Table 21-7 EMI memory-mapped register

Name	Full Name	Address	Width
BCR	Bus Control Register	FE000000	32
SMCR0	Static Memory Control Register 0	FE000004	32
SMCR1	Static Memory Control Register 1	FE000008	32
SMCR2	Static Memory Control Register 2	FE00000C	32
SMCR3	Static Memory Control Register 4	FE000010	32
DMCR	DRAM Control Register	FE000014	32
RTCSR	Refresh Time Control/Status Register	FE000018	16
RTCNT	Refresh Timer Counter	FE00001C	16
RTCOR	Refresh Time Constant Register	FE000020	16
SDMR	SDRAM mode register, bank 4	FEFFA000~FEFFAFFF	8/16/32
	SDRAM mode register, bank 5	FEFFB000~FEFFBFFF	
	SDRAM mode register, bank 6	FEFFC000~FEFFCFFF	
	SDRAM mode register, bank 7	FEFFD000~FEFFDFFF	

21.3.5 PCI

Table 21-8 PCI memory-mapped register

Name	Full Name	Address	Width
PTA_MAILRR0	PCI-to_Arca Mailbox Receive Register 0	FD000000	32
PTA_MAILRR1	PCI-to_Arca Mailbox Receive Register 1	FD000004	32
PTA_MAILRR2	PCI-to_Arca Mailbox Receive Register 2	FD000008	32
PTA_MAILRR3	PCI-to_Arca Mailbox Receive Register 3	FD00000C	32
ATP_MAILTR0	Arca-to-PCI Mailbox Transmit Register 0	FD000010	32
ATP_MAILTR1	Arca-to-PCI Mailbox Transmit Register 1	FD000014	32
ATP_MAILTR2	Arca-to-PCI Mailbox Transmit Register 2	FD000018	32
ATP_MAILTR3	Arca-to-PCI Mailbox Transmit Register 3	FD00001C	32
PCIC_MAILSR	PCIC Mailbox Status Register	FD000020	32
PTA_DB	PCI-to-Arca Doorbell Register	FD000024	32
ATP_DB	Arca-to-PCI Doorbell Register	FD000028	32
PCIC_CFGAR	PCIC Configuration Address Register	FD00002C	32
PCIC_CFGDR	PCIC Configuration Data Register	FD000030	32
PCIC_SR	PCIC Status Register	FD000034	32
PCIC_IER	PCIC Interrupt Enable Register	FD000038	32
PTA_MEMATR	PCI-to-Arca Memory Address Translation Register	FD000040	32
ATP_MTR0	Arca-to-PCI Memory Translation Register 0	FD000044	32
ATP_MTR1	Arca-to-PCI Memory Translation Register 1	FD000048	32
ATP_MTR2	Arca-to-PCI Memory Translation Register 2	FD00004C	32
ATP_IOTR	Arca-to-PCI IO Translation Register	FD000050	32
VENDOR_ID	PCI vendor ID	FD000100	16
DEVICE_ID	PCI device ID	FD000102	16
COMMAND	PCI command	FD000104	16
STATUS	PCI status	FD000106	16
REVISION_ID	PCI revision ID	FD000108	8
CLASS_CODE	PCI class code	FD000109	24
CACHELINE_SIZE	PCI cacheline size	FD00010C	8
LATENCY_TIMER	PCI latency timer	FD00010D	8
HEADER_TYPE	PCI header type	FD00010E	8
BIST	PCI bist	FD00010F	8
PCI_MBR0	PCI memory base address0	FD000110	32
PCI_MBR1	PCI memory base address1	FD000114	32
SS_VENDOR_ID	PCI subsystem vendor ID	FD00012C	16
SS_ID	PCI subsystem ID	FD00012E	16
INTR_LINE	PCI interrupt line	FD00013C	8
INTR_PIN	PCI interrupt pin	FD00013D	8
MIN_GNT	PCI minimum grant	FD00013E	8
MAX_LAT	PCI latency timer	FD00013F	8
TRDY_TIMER	PCI trdy timer	FD000140	8
RETRY_TIMER	PCI retry timer	FD000141	8

21.3.6 UART

Table 21-9 UART memory-mapped register

Name	Full Name	Address	Width
UARTBRDR	Bit rate divisor register	FFFFFFF00	16
UARTCR	UART control register	FFFFFFF04	32
UARTCRL	UART control register L	FFFFFFF04	16
UARTCRH	UART control register H	FFFFFFF06	16
UARTSR	UART status register	FFFFFFF08	32
UARTRDR	Receive FIFO data register	FFFFFFF0C	8
UARTTDR	Transmit FIFO data register	FFFFFFF0D	8
UARTRFDR	Receive FIFO data count register	FFFFFFF0E	8
UARTTFDR	Transmit FIFO data count register	FFFFFFF0F	8

21.3.7 INTC

Table 21-10 INTC memory-mapped register

Name	Full Name	Address	Width
ISR	Interrupt Source Register	FFFFFFE00	32
IMR	Interrupt Mask Register	FFFFFFE04	32
IMSR	Interrupt Mask Set Register	FFFFFFE08	32
IMCR	Interrupt Mask Clear Register	FFFFFFE0C	32
IPR	Interrupt Pending Register	FFFFFFE10	32

21.3.8 RTC

Table 21-11 RTC memory-mapped register

Name	Full Name	Address	Width
RCR	RTC control register	FFFFFD00	8
RSR	RTC second register	FFFFFD04	32
RFR	RTC fraction register	FFFFFD08	32
RSAR	RTC second alarm register	FFFFFD0C	32
RFAR	RTC fraction alarm register	FFFFFD10	32

21.3.9 TMU

Table 21-12 TMU memory-mapped register

Abbreviation	Full Name	Address	Width
TER	Timer enable register	FFFFFFC00	8
TRDR0	Timer reload data register 0	FFFFFFC04	32
TCNT0	Timer counter 0	FFFFFFC08	32
TCSR0	Timer control/status register 0	FFFFFFC0C	16
TRDR1	Timer reload data register 1	FFFFFFC10	32
TCNT1	Timer counter 1	FFFFFFC14	32
TCSR1	Timer control/status register 1	FFFFFFC18	16
TRDR2	Timer reload data register 2	FFFFFFC1C	32
TCNT2	Timer counter 2	FFFFFFC20	32
TCSR2	Timer control/status register 2	FFFFFFC24	16

21.3.10 WDT

Table 21-13 WDT memory-mapped register

Name	Full Name	Address	Width
WTCSR	Watchdog timer Control/Status Register	FFFFFB00	R8/W16
WTCNT	Watchdog Timer Counter	FFFFFB04	R8/W16

21.3.11 CGU

Table 21-14 CGU memory-mapped register

Name	Full Name	Address	Width
CFCR	Clock Frequency Control Register	FFFFFA00	32
LPCR	Low Power Control Register	FFFFFA04	16
RSTR	Reset Status Register	FFFFFA08	8
PSTR	PLL Stabilization Time Register	FFFFFA0C	8

21.3.12 PMC

Table 21-15 PMC memory-mapped register

Name	Full Name	Address	Width
LPCR	Low Power Control Register	FFFFFA04	16
RSTR	Reset Status Register	FFFFFA08	8
PSTR	PLL Stabilization Time Register	FFFFFA0C	8

21.3.13 I2CI

Table 21-16 I2CI memory-mapped register

Name	Full Name	Address	Width
I2CDR	Data Register	FFFFFF800	8
I2CCR	Control Register	FFFFFF804	8
I2CSR	Status Register	FFFFFF808	8
I2CGR	Clock generator Register	FFFFFF80C	16

21.3.14 GPIO

Table 21-17 GPIO memory-mapped register

Name	Full Name	Address	Width
GPCRA	Port A Control Register	FFFFFF700	16
GPDRA	Port A Data Register	FFFFFF804	8
GPTRA	Port A Interrupt Detect Manner Register	FFFFFF808	16
GPFRA	Port A Edge Flag Register	FFFFFF80C	8

List of Figures

Figure 1-1 Block Diagram	2
Figure 3-1 EMI Block Diagram	8
Figure 3-2 Physical Address Space Map	12
Figure 3-3 Synchronous DRAM Mode Register Configuration	26
Figure 3-4 Example of 32-Bit Data Width Flash Connection	35
Figure 3-5 Example of 16-Bit Data Width Flash Connection	35
Figure 3-6 Example of 16-Bit Data Width Flash Connection	36
Figure 3-7 Example of 8-Bit Data Width Flash Connection	36
Figure 3-8 Basic Timing of Normal Memory Read	37
Figure 3-9 Basic Timing of Normal Memory Write	38
Figure 3-10 Normal Memory Read Timing With Wait (Software Wait Only)	39
Figure 3-11 Normal Memory Write Timing With Wait (Software Wait Only)	40
Figure 3-12 Normal Memory Read Timing With Wait (Wait Cycle Insertion by WAIT# pin)	41
Figure 3-13 Example of 32-Bit Data Width Byte Control SRAM Connection	42
Figure 3-14 Byte Control SRAM Read Timing	43
Figure 3-15 Byte Control SRAM Write Timing	44
Figure 3-16 Burst ROM Read Timing (Software Wait Only)	45
Figure 3-17 Example of Synchronous DRAM Chip Connection (1)	48
Figure 3-18 Example of Synchronous DRAM Chip Connection (2)	49
Figure 3-19 Example of Synchronous DRAM 168-pin DIMMs Connection	50
Figure 3-20 Example of Synchronous DRAM 100-pin DIMMs Connection	51
Figure 3-21 Example of Synchronous DRAM 144-pin DIMMs Connection	52
Figure 3-22 Synchronous DRAM 16-Byte Burst Read Timing (Non-Page Mode)	55
Figure 3-23 Synchronous DRAM 32-Byte Burst Read Timing (Non-Page Mode)	56
Figure 3-24 Synchronous DRAM Single Read Timing (Non-Page Mode)	57
Figure 3-25 Synchronous DRAM 16-Byte Burst Write Timing (Non-Page Mode)	58
Figure 3-26 Synchronous DRAM 32-Byte Burst Write Timing (Non-Page Mode)	59
Figure 3-27 Synchronous DRAM Single Write Timing (Non-Page Mode)	60
Figure 3-28 Synchronous DRAM 16-byte Burst Read Timing (Page Mode, Different Row)	62
Figure 3-29 Synchronous DRAM 16-byte Burst Read Timing (Page mode, Same Row)	63
Figure 3-30 Synchronous DRAM 16-byte Burst Write Timing (Page Mode, Different Row)	64
Figure 3-31 Synchronous DRAM 16-byte Burst Write Timing (Page Mode, Same Row)	65
Figure 3-32 Synchronous DRAM 32-byte Burst Read Timing (Page Mode, Different Row)	66
Figure 3-33 Synchronous DRAM 32-byte Burst Read Timing (Page Mode, Same Row)	67
Figure 3-34 Synchronous DRAM 32-byte Burst Write Timing (Page Mode, Same Row)	68
Figure 3-35 Synchronous DRAM 32-byte Burst Write Timing (Page Mode, Different Row)	69
Figure 3-36 Synchronous DRAM Single Read Timing (Page Mode, Different Row)	70
Figure 3-37 Synchronous DRAM Single Read Timing (Page mode, Same Row)	71
Figure 3-38 Synchronous DRAM Single Write Timing (Page Mode, Different Row)	72
Figure 3-39 Synchronous DRAM Single Write Timing (Page Mode, Same Row)	73
Figure 3-40 SDRAM Power-Down Mode Timing (CKO Stopped)	74
Figure 3-41 SDRAM Power-Down Mode Timing (Clock Supplied)	74
Figure 3-42 Synchronous DRAM Auto-Refresh Operation	75
Figure 3-43 Synchronous DRAM Auto-Refresh Timing (Non-page mode)	76
Figure 3-44 Synchronous DRAM Auto-Refresh Timing (Page Mode)	76
Figure 3-45 Synchronous DRAM Self-Refresh Timing	78
Figure 3-46 Synchronous DRAM Self-Refresh Timing (Page Mode)	78
Figure 3-47 SDRAM Mode Register Write Timing 1 (Pre-charge All Banks)	80
Figure 3-48 SDRAM Mode Register Write Timing 2 (Mode Register Set)	81
Figure 4-1 DMA Block Diagram	83
Figure 4-2 Flowchart of DMA Transfer	95
Figure 4-3 Two-address Transfer Example	98

Figure 4-4 Timing in Two-address DMA Transfer	99
Figure 4-5 Example of DMA Transfer in Single Mode	100
Figure 4-6 Example of DMA Transfer in Block Mode.....	101
Figure 4-7 Bus Handling with Two DMA Channels Operating.....	103
Figure 4-8 Example of Single Mode, DREQ Low Level Detection	104
Figure 4-9 Example of Single Mode, DREQ Falling Edge Detection.....	105
Figure 4-10 Example of Block Mode, DREQ Low Level Detection	106
Figure 4-11 Example of Block Mode, DREQ Edge Detection.....	107
Figure 5-1 PCIC Block Diagram	111
Figure 5-2 PCIC Master Address Mapping	113
Figure 5-3 PCI Address Map to 8M-byte Arca210 Address Space	114
Figure 5-4 PCIC Target Address Mapping	115
Figure 5-5 Mailbox and Doorbell Registers Access.....	120
Figure 5-6 PCI Arbiter Diagram	143
Figure 5-7 Arbitration Scheme	144
Figure 5-8 Bus Parking	145
Figure 5-9 Arbitration in Bus Idle	147
Figure 5-10 Master of Group B Preempted	148
Figure 6-1 ETHC Block Diagram	150
Figure 6-2 Descriptor Ring and Chain Structure Examples	178
Figure 6-3 Receive Descriptor Format	179
Figure 6-4 Transmit Descriptor Format	183
Figure 6-5 Transmit Data Flow.....	187
Figure 6-6 Receive Data Flow	187
Figure 6-7 Ethernet Controller Operation	188
Figure 6-8 Block Diagram of MAC Block.....	195
Figure 6-9 VLAN Topology	200
Figure 6-10 VLAN Frame.....	201
Figure 7-1 UHC Block Diagram	204
Figure 8-1 CGU Block Diagram.....	209
Figure 2 Block Diagram of PLL	217
Figure 9-1 Doze Mode Operation	228
Figure 9-2 Sleep Mode Cancelled By Interrupt Sequence	229
Figure 9-3 Sleep Mode Cancelled By Manual Reset	230
Figure 9-4 Sleep Cancelled By Power-on Reset.....	230
Figure 9-5 Standby Mode Cancelled by Interrupt (Pllen=1)	232
Figure 9-6 Standby Mode Cancelled by Power-on Reset (clock mode 0, 1, 2 and 3, Pllen=1).....	232
Figure 9-7 Standby Mode Cancelled by Manual Reset (clock mode 0, 1, 2 and 3, Pllen=1)	233
Figure 9-8 Power-On Reset via RESETP Pin (clock mode 0, 1, 2, 3)	236
Figure 9-9 Power-On Reset via RESETP Pin (clock mode 4)	237
Figure 9-10 Manual Reset via RESETM Pin.....	237
Figure 9-11 WDT Manual Reset.....	238
Figure 10-1 INTC Block Diagram.....	240
Figure 10-2 Flowchart	245
Figure 11-1 TMU Block Diagram	249
Figure 11-2 TMU Operation	253
Figure 11-3 TMU Channel 0 Decrease and Reload Timing	254
Figure 12-1 WDT Block Diagram.....	256
Figure 13-1-1 Port A Block Diagram	261
Figure 1-13-2 Port B Block Diagram	264
Figure 1-13-3 Port B Block Diagram	264
Figure 1-13-4 Port C Block Diagram.....	268
Figure 1-13-5 Port D Block Diagram.....	271
Figure 1-13-6 Port E Block Diagram	274
Figure 1-13-7 Port F Block Diagram.....	277
Figure 14-1 UART Block Diagram	282
Figure 14-2 Sample of UART Initialization Flowchart	298

Figure 14-3 Sample UART Serial Transmission Flowchart	299
Figure 14-4 Sample of Transmit Operation with 8-bit data, 1 stop bit	300
Figure 14-5 Sample of Operation Using Modem Control (CTS)	301
Figure 14-6 Sample of UART Serial Reception Flowchart (1)	302
Figure 14-7 Sample of UART Serial Reception Flowchart (2)	303
Figure 14-8 Sample of UART Serial Reception Operation with 8-bit data, parity, 1-bit stop	304
Figure 14-9 Sample of Operation Using Modem Control (RTS)	305
Figure 15-1 SCC Block Diagram (UART functions as SCC)	309
Figure 15-2 Pin Connection Diagram for the Smart Card Controller	317
Figure 15-3 Data Format for Smart Card Controller	318
Figure 15-4 Waveform of Start Character	319
Figure 15-5 Initialization Flowchart	321
Figure 15-6 Transmission Flowchart	322
Figure 15-7 Reception Flowchart	323
Figure 15-8 Selection of the Class of Operating Conditions by SCC	325
Figure 15-9 Cold Reset	326
Figure 15-10 Warm Reset	327
Figure 15-11 Clock Stop	327
Figure 15-12 Receive Data Sampling Timing under Asynchronous Mode	328
Figure 15-13 Retransmission in SCC Receive Mode	329
Figure 15-14 Retransmission in SCC Transmit Mode	329
Figure 16-1 IrDA Block Diagram	331
Figure 16-2 UART and IrDA Signal Waveform	337
Figure 17-1 I2CI Block Diagram	341
Figure 17-2 I2C-bus Protocol	349
Figure 17-3 I ² C-bus Protocol (cont.)	349
Figure 17-4 Normal 7 Bit Address after START Condition	350
Figure 17-5 General Call Address after START Condition	351
Figure 17-6 START Byte after START Condition	351
Figure 17-7 A Master-Transmitter Addresses a Slave Receiver with a 7-Bit Address	352
Figure 17-8 A Master Reads the Slave Immediately after the First Byte (Master-Receiver)	352
Figure 17-9 I ² CI Initialization	353
Figure 17-10 I2CI Write Operation Flowchart	354
Figure 17-11 I2CI Read Operation Flowchart	355
Figure 17-12 Read Operation Flowchart (cont.)	356
Figure 18-1 AC97 Controller Block Diagram	358
Figure 18-2 AC-link Audio Output/Input Frame	376
Figure 18-3 AC97 Controller Initialization/Resettlement Flowchart	377
Figure 18-4 AC97 Controller Transmission Flowchart	379
Figure 18-5 AC97 Controller Reception Flowchart	380
Figure 18-6 AC97 Codec Register Access Command Send Flow Chart	382
Figure 18-7 Bring AC97 Codec Register Back Reception Flow Chart	383
Figure 18-8 AC-Link Low Power Mode Timing	384
Figure 18-9 Cold Reset Timing	385
Figure 18-10 Warm Reset Timing	386
Figure 18-11 ATE Test Mode Timing	386
Figure 20-1 Arca210 Pin Diagram (1)	392
Figure 20-2 Arca210 Pin Diagram (2)	393

List of Tables

Table 1-1 Arca210 Features	3
Table 1-2 Characteristics.....	5
Table 3-1 EMI Pin Configuration	9
Table 3-2 Physical Address Space Map	11
Table 3-3 EMI Register Configuration	13
Table 3-4 32-Bit External Device/Big Endian Access and Data Alignment	30
Table 3-5 16-Bit External Device/Big Endian Access and Data Alignment	31
Table 3-6 8-Bit External Device/Big Endian Access and Data Alignment	31
Table 3-7 32-Bit External Device/Little Endian Access and Data Alignment	32
Table 3-8 16-Bit External Device/Little Endian Access and Data Alignment	32
Table 3-9 8-Bit External Device/Little Endian Access and Data Alignment.....	33
Table 3-10 SDRAM Command Encoding (Notes: 1)	47
Table 3-11 Relationship between CA, BA and Address Multiplex Output* ⁶	53
Table 4-1 DMA Pins	84
Table 4-2 DMA Registers.....	85
Table 4-3 RS4-0 Settings for External Transfer Request.....	96
Table 4-4 Relationship between DMA Transfer Source, Request Mode and Transfer Mode	101
Table 5-1 PCI Mode Configuration Pin.....	112
Table 5-2 PCI Arbiter Mode Configuration Pin	112
Table 5-3 PCI Commands Support	117
Table 5-4 PCIC Control Registers	119
Table 5-5 PCIC_MAILSR Bit Meaning	123
Table 5-6 PCI_MAILSR Bit Meaning	124
Table 5-7 Memory Size As PCI Slave	132
Table 5-8 PCI Configuration Registers.....	134
Table 6-1 ETHC Pin Configuration	151
Table 6-2 ETHC Register Configuration	154
Table 6-3 RDES0 Receive Descriptor 0 Description	180
Table 6-4 RDES1 Receive Descriptor 1 Description	182
Table 6-5 RDES2 Receive Descriptor 2 Description	182
Table 6-6 RDES3 Receive Descriptor 3 Description	183
Table 6-7 TDES0 Receive Descriptor 0 Description.....	184
Table 6-8 TDES1 Receive Descriptor 1 Description.....	185
Table 6-9 TDES2 Transmit Descriptor 2 Description.....	186
Table 6-10 TDES3 Transmit Descriptor 3 Description	186
Table 6-11 Destination Address Bit 0.....	197
Table 6-12 Address Filtering Mode.....	198
Table 7-1 UHC Pin Configuration	205
Table 7-2 UHC Registers	206
Table 8-1 Arca210 Clock Source.....	208
Table 8-2 CGU Pin Configuration.....	210
Table 8-3 CGU Clock Modes	211
Table 8-4 CFCR Setting and Clock Frequency.....	212
Table 8-5 CGU Registers Configuration.....	214
Table 9-1 Low-Power Modes Status	222
Table 9-2 Power/Reset Management Controller Registers Configuration	223
Table 10-1 INTC Pins	241
Table 10-2 INTC Register	242
Table 10-3 Interrupt Sources	243
Table 11-1 TMU Registers	250
Table 12-1 WDT Registers	257
Table 13-1 Port A Registers	262

Table 13-2 Port B Registers	265
Table 1-3 Alternate Function When PBCR.An = 1	266
Table 13-4 Port C Registers.....	269
Table 1-5 Alternate Function When PCCR.An = 1	270
Table 13-6 Port D Registers.....	272
Table 1-7 Alternate Function When PDCR.An = 1	273
Table 13-8 Port E Registers	275
Table 1-9 Alternate Function When PECR.An = 1	276
Table 13-10 Port F Registers	278
Table 1-11 Alternate Function When PFCR.An = 1.....	279
Table 14-1 UART Pins.....	283
Table 14-2 UART Registers	284
Table 14-3 Typical baud rates and divisors	294
Table 14-4 UARTCR Settings for Serial Transfer Format Selection	297
Table 14-5 Serial Transfer Formats	297
Table 14-6 UART Interrupt Sources	306
Table 15-1 SCC Pins.....	310
Table 15-2 UART and SCC Registers	311
Table 15-3 Smart Card Mode Operating State and Interrupt Sources	324
Table 16-1 IRDA Pins	332
Table 16-2 UART2 and IRDA Registers	333
Table 17-1 I2CI Pins	342
Table 17-2 I2CI Registers	343
Table 18-3 AC97 Controller Pins	358
Table 18-4 AC97 Controller Registers	360
Table 18-5 AC-Link Low Power Mode Timing Parameters	384
Table 18-6 Cold Reset Timing parameters	385
Table 18-7 Warm Reset Timing Parameters	386
Table 18-8 ATE Test Mode Timing Parameters	387
Table 19-1 SARB Registers	390
Table 20-1 EMI pins (total 85).....	394
Table 20-2 GPIO A pins (total 8)	394
Table 20-3 IRQ/GPIO B pins (total 8)	394
Table 20-4 AC97/GPIO C pins (total 5)	395
Table 20-5 UART/GPIO C pins (total 2)	395
Table 20-6 UART2/IRDA GPIO D pins (total 4)	395
Table 20-7 SCC/GPIO D pins (total 2)	395
Table 20-8 DMA/GPIO D/E pins (total 8).....	396
Table 20-9 I2C pins (total 2).....	396
Table 20-10 SYSTEM pins (total 9)	396
Table 20-11 JTAG pins (total 5)	396
Table 20-12 PCI pins (total 52)	397
Table 20-13 PCI Arbiter pins (total 10).....	397
Table 20-14 ETHC/GPIO E/F pins (total 17).....	398
Table 20-15 UHC pins (total 9).....	398
Table 20-16 Test port / MD pins (5).....	398
Table 21-1 Physical Address Space Map.....	399
Table 21-2 Address space allocation for different component groups	399
Table 21-3 Base address memory-mapped register for each module	400
Table 21-4 MMU memory-mapped register	400
Table 21-5 Cache memory-mapped register	400
Table 21-6 OCS-Bus arbiter memory-mapped register	400
Table 21-7 EMI memory-mapped register	401
Table 21-8 PCI memory-mapped register	402
Table 21-9 UART memory-mapped register	403
Table 21-10 INTC memory-mapped register	403
Table 21-11 RTC memory-mapped register	403

Table 21-12 TMU memory-mapped register	404
Table 21-13 WDT memory-mapped register	404
Table 21-14 CGU memory-mapped register	404
Table 21-15 PMC memory-mapped register	404
Table 21-16 I2CI memory-mapped register	405
Table 21-17 GPIO memory-mapped register	405