MiniRISC® EZ4021-FC Building Blocks

Technical Manual

April 2000

LSI LOGIC®

Order Number C14070

Document DB14-000108-00, First Edition April 2000
This document describes Revision A of the LSI Logic Corporation MiniRISC® EZ4021-FC EasyMACRO Building Blocks and will remain the official reference source for all revisions/releases of this product until rescinded by an update.

To receive product literature, visit us at http://www.lsilogic.com.

Preface

This book is the primary reference and User's Guide for the MiniRISC® EZ4021-FC Microprocessor EasyMACRO Building Blocks. It contains a complete functional description for each building block.

Audience

This document assumes that you have some familiarity with microprocessors and related support devices. The people who benefit from this book are:

- Engineers and managers who are evaluating the EZ4021-FC building blocks for possible use in a system
- Engineers who are designing the processor and associated building blocks into a system

Organization

This document contains the following chapters and appendixes:

- Chapter 1, **Introduction**, introduces the EZ4021-FC feature set and briefly discusses the available building blocks.
- Chapter 2, **SDRAM Controller**, describes the EZ4021-FC SDRAM Controller and its operation.
- Chapter 3, **Quick Bus**, describes the EZ4021-FC's 64 bit high-performance on-chip bus, the QuickBus.
- Chapter 4, **External Bus Controller**, describes the External Bus Controller, which allows off-chip peripherals to communicate with the EZ4021-FC through the QuickBus.

Related Publications

MiniRISC® EZ4021-FC Microprocessor EasyMACRO Technical Manual, Document No. DB15-000080-01

Conventions Used in This Manual

The first time a word or phrase is defined in this manual, it is *italicized.*

The word *assert* means to drive a signal true or active. The word *deassert* means to drive a signal false or inactive.

Hexadecimal numbers are indicated by the prefix “0x” —for example, 0x32CF. Binary numbers are indicated by the prefix “0b” —for example, 0b0011.0010.1100.1111.

Contents

Customer Feedback

Figures

Tables

Chapter 1
Introduction

This chapter introduces the building blocks available with the LSI Logic MiniRISC EZ4021-FC Microprocessor Core.

The chapter contains the following sections:

1.1 System Overview

The MiniRISC EZ4021-FC Microprocessor EasyMACRO core is a compact, high-performance, 64-bit microprocessor subsystem. The EZ4021-FC uses the LSI Logic CoreWare® system-on-a-chip methodology and executes the MIPS III instruction set. It is ideal for high-performance, cost-sensitive embedded processor applications.

For detailed information about the EZ4021-FC, see the *MiniRISC EZ4021FC Microprocessor Core Technical Manual*.

You can easily design the EZ4021-FC into a wide range of products. It can be combined with industry standard cores and proprietary functional building blocks to create a completely customized embedded system on a chip.

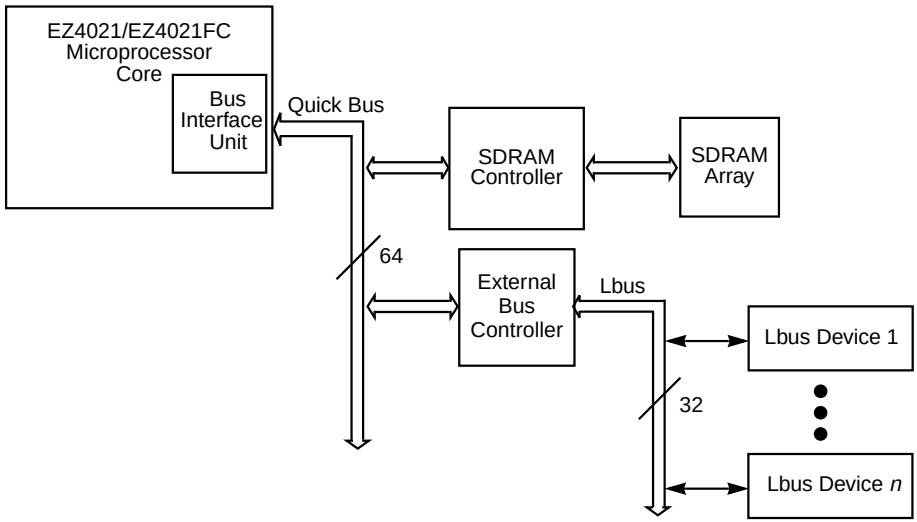
LSI Logic currently provides the following building blocks:

- SDRAM Controller (SDRAMC), described in Chapter 2, “SDRAM Controller.”

- Quick Bus, described in Chapter 3, “Quick Bus.”
- External Bus Controller (EBC), described in Chapter 4, “External Bus Controller.”

The building blocks are briefly described in Section 1.3, “Building Blocks Overview.” System designers can use these building blocks as they are provided, or modify them for specific needs. Figure 1.1 shows how the EZ4021-FC Microprocessor Core interfaces with building blocks in a typical design.

Figure 1.1 EZ4021-FC in a Typical System



1.2 EZ4021-FC Features

This section summarizes the key features of the EZ4021-FC.

- High-performance RISC CPU core for CoreWare ASIC
  - Single issue, five-stage pipeline
  - 250 native MIPS, 250+ Dhrystone MIPS at 250 MHz
  - 250-MHz operation at WC125 (Tj = 125 °C, VDD = 1.71 V, worst-case process)
- MIPS III Instruction Set Architecture (ISA)
  - Thirty-two 64-bit general-purpose registers
  - 32-bit wide MIPS III ISA supporting 64-bit integer operations
  - R4000-style status register and exception processing
  - Wait for Interrupt (WAITI) instruction for power saving
  - Supports SPECIAL2 Multiply-Accumulate extensions
- Both big and little endian support for load and store operations
- Integrated multiplier and divider
  - High-performance eight bit/cycle multiplier
    - ◊ 32-bit unsigned or signed multiplication in five CPU clock cycles
    - ◊ 64-bit unsigned or signed multiplication in nine CPU clock cycles
  - Compact and low performance (1 bit/cycle) divider
    - ◊ 32-bit unsigned or signed division in 34 CPU clock cycles
    - ◊ 64-bit unsigned or signed division in 66 CPU clock cycles
- Integrated instruction and data caches (Harvard architecture)
  - MMU implements 32 dual-entry page translations
  - 16 Kbyte 2-way set-associative instruction cache
    - ◊ Least Recently Used (LRU) algorithm for replacement
    - ◊ Line level lock for instruction cache RAM

  - 16 Kbyte 2-way set-associative data cache
    - ◊ LRU algorithm for replacement
    - ◊ Line level lock for scratchpad memory
    - ◊ Write-through or write-back update policy, programmable on a per-page basis
- Integrated EJTAG debug support features
  - MIPS products standard EJTAG 1.5.3 compliant
    - ◊ Simple instruction and data breakpoints
    - ◊ Program Counter (PC) trace
    - ◊ Processor single step and software debug breakpoints
- System Coprocessor Zero (CP0) Count and Compare registers
- MIPS CPU standard interrupt exceptions (one NMI, one timer, five hardware, two software)
- Serial scan for device testing and EJTAG support for on-chip system debug
- 13 mm^2 core size
- 1.8 V Core VDD
- LSI Logic G12™ CMOS technology (0.18 µ L-drawn, 0.15 µ L-effective)

1.3 Building Blocks Overview

This section gives a brief overview of the available EZ4021-FC building blocks.

1.3.1 SDRAM Controller

The SDRAM Controller allows an EZ4021-FC-based design to interface directly to a 64 Mbit SDRAM array without the need for external logic. The SDRAM Controller generates all commands for the SDRAM array, including Row Address Select (RAS) and Column Address Select (CAS).

The SDRAM Controller supports up to 64 Mbytes of SDRAM.

1.3.2 Quick Bus

The Quick Bus is a demultiplexed 32-bit address, 64-bit data split-transaction on-chip bus. It permits the overlap of command request and data return operations. This feature allows for higher bus utilization and the ability to hide long memory latency times to off-chip devices. The Quick Bus also supports multiple bus masters and data burst transactions.

1.3.3 External Bus Controller

The External Bus Controller (EBC) allows the EZ4021-FC to interface to off-chip devices, such as ethernet controllers, serial I/O devices, and ROMs. The EBC serves as a bridge between the on-chip Quick Bus and the off-chip Local Bus (Lbus). The Lbus is a 32-bit multiplexed address/data bus.

External bus masters on the Lbus can arbitrate for ownership of the Local Bus and thereby make read and write requests on the Quick Bus.

The EBC supports a retry mechanism that can be used if a peripheral device cannot complete a request. The target device can assert retry, which causes the EBC to retry the transaction at a later time.

1.4 MiniRISC Support Tools

The EZ4021-FC has all the tools needed to develop a system on a chip, including:

- The LSI Logic MiniSIM®-20 architectural simulator
- Verilog models
- A system verification environment
- A PROM monitor
- Third-party software support
- A full-featured core evaluation chip (EV4020)
- Bus functional model for Quick Bus

1.5 CoreWare Program

The CoreWare program consists of three main elements:

- A library comprised of a wide range of complex cores based on accepted and emerging industry standards. The library includes high-speed interconnection, digital video, digital signal processing (DSP), and other cores.
- A design development and simulation package. LSI Logic provides a complete framework for device and system development and simulation. The LSI Logic advanced ASIC technologies consistently produce Right-First-Time™ silicon.
- Support for expert applications. The LSI Logic in-house experts provide design support from system architecture definition through chip layout and test vector generation.

Using the CoreWare program, you can combine the EZ4021-FC microprocessor core with other cores on a single chip to create products uniquely suited to specific applications. The program provides unparalleled design flexibility and lets you create high-quality, leading edge products for a wide range of markets.

1.5.1 CoreWare Building Blocks

The CoreWare building blocks include elements based on the LSI Logic high-performance standard products, as well as other industry-standard products. The CoreWare building blocks, which include embedded MiniRISC MIPS processors, bus interface controllers, and a family of floating-point processors, are fully supported library elements for use in the LSI Logic hardware development environment. The building blocks include gate-level simulation models with timing information, so that designers can accurately simulate device performance and trade off various implementation options. In addition to gate-level simulation models, the building blocks also include behavioral simulation models.

1.5.2 Design Environment

ASIC families are supported by the LSI Logic comprehensive system-on-a-chip design methodology. This design methodology uses both internally developed and industry-standard tools integrated with

FlexStream® software and libraries that lets you use third-party software to access LSI Logic technology. You can select from a suite of industry standard simulators, synthesizers, timing analyzers, and test tools that are seamlessly integrated into a common environment for verification and sign-off.

1.5.3 Expert Support

The LSI Logic in-house experts support the CoreWare program with high-level design experience in a wide variety of application areas. These experts provide design support from system architecture definition through chip layout and test-vector generation. They help determine how many functions can be integrated on a single chip to find the most cost-effective solution.

Chapter 2
SDRAM Controller

This chapter explains the operation of the EZ4021-FC's on-chip synchronous DRAM controller, and contains the following sections:

2.1 Overview

The on-chip synchronous DRAM (SDRAM) controller interfaces directly to a 64 Mbit SDRAM memory array, eliminating the need for discrete control logic. This approach saves board space, reduces component count and design complexity, and increases performance by eliminating the inherent delays caused by external discrete components.

2.2 Features

The EZ4021-FC SDRAM Controller operates with industry-standard 64 Mbit devices (4-, 8-, and 16-bit wide) at clock frequencies up to 125 MHz, and offers the following features:

- In-Page Hits

  The SDRAM Controller supports in-page hits (also known as row parking). Once a row address is latched into a bank of the SDRAM, subsequent column addresses in the same row are clocked out according to the programmed column latency. No precharge is required. Each internal bank can store a row address.

- Quick Bus Optimized

  The SDRAM Controller is optimized for use with the Quick Bus. Due to the split transaction nature of the Quick Bus, the requesting device may not immediately acknowledge the output data when it is latched by the SDRAM Controller. The controller output buffer stores output data until the Quick Bus is ready to read. As long as the write buffer is not full, the SDRAM Controller accepts read requests from the Quick Bus. If a burst read request is received while data is in the output buffer, the SDRAM Controller queues the read request and reissues it when the output buffer empties.

- Four Doubleword (Burst) Transaction

  The SDRAM Controller does not support burst mode; all transactions are sequential. However, a four-doubleword request acts as a burst transaction. The SDRAM Controller performs four sequential accesses from a single address. The term *burst* in this document refers to the four-doubleword transaction.

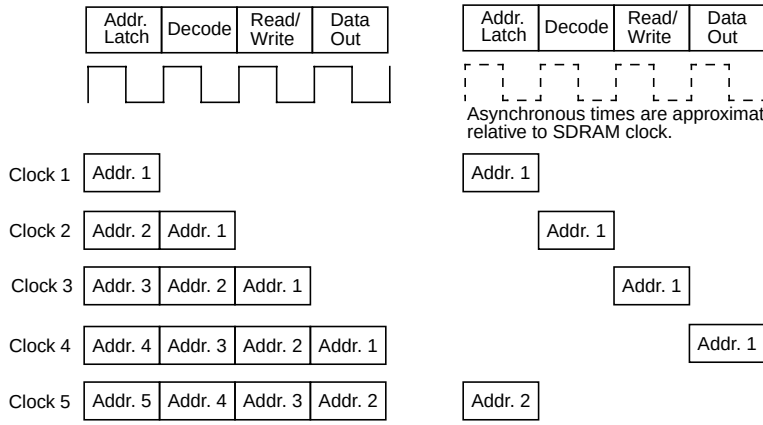
2.3 SDRAM Overview

SDRAMs offer significant performance advantages over standard Fast Page Mode (FPM) or Extended Data Out (EDO) DRAM devices. In standard DRAM architectures, once the processor has requested data, it must wait for a certain number of clocks until data is returned. The amount of delay depends on the design of the memory system and the speed of the DRAMs used.

In standard DRAM, the Row Address Select (RAS) and Column Address Select (CAS) signals must be held active from the start of the cycle until data is retrieved, which means that the DRAM cannot accept new cycle information during this time. Design techniques such as interleaving multiple memory banks helps address this latency problem. With this approach, subsequent accesses go to alternate banks, hiding some of the DRAM latency time. Interleaved memory systems achieve a performance boost over their noninterleaved counterparts, but require large amounts of real estate and increase the controller complexity.

The clocked nature of the SDRAM allows for pipelining. New cycle information is driven to the array on every clock. Once the pipeline is filled, data can be driven out of the SDRAM on each subsequent clock. In addition, each 64 Mbit SDRAM device contains four internal banks, allowing for the implementation of an interleaved memory system without the added component count and controller complexity. Figure 2.1 shows a timing diagram comparing SDRAM and standard DRAM accesses. DRAM devices are asynchronous—access times in Figure 2.1 are relative to an SDRAM device with an equivalent access time.

Figure 2.1 SDRAM Versus Standard DRAM Timing



SDRAMs contain an on-chip mode register that provides a high degree of programmability. On-chip burst counters allow for the support of burst mode without the need for an external discrete counter to increment subsequent addresses of the burst. Column Address Select (CAS)

latency, the number of clocks between the time data is requested and the time it is driven onto the bus, is also programmable. The higher the latency, the longer it takes to retrieve the first data item.

2.4 SDRAM Types and Memory Area

The SDRAM Controller supports 64 Mbit SDRAM devices with 4-, 8-, and 16-bit wide data buses. The 64M bit in the SDRAM Configuration Register must be set. Refer to Section 2.10.1, “SDRAM Controller Configuration Register,” on page 2-20 for more information on the 64M bit.

Table 2.1 shows available SDRAM types, the total number of SDRAM devices required to design a memory system, the total memory size requirements, and the available address ranges.

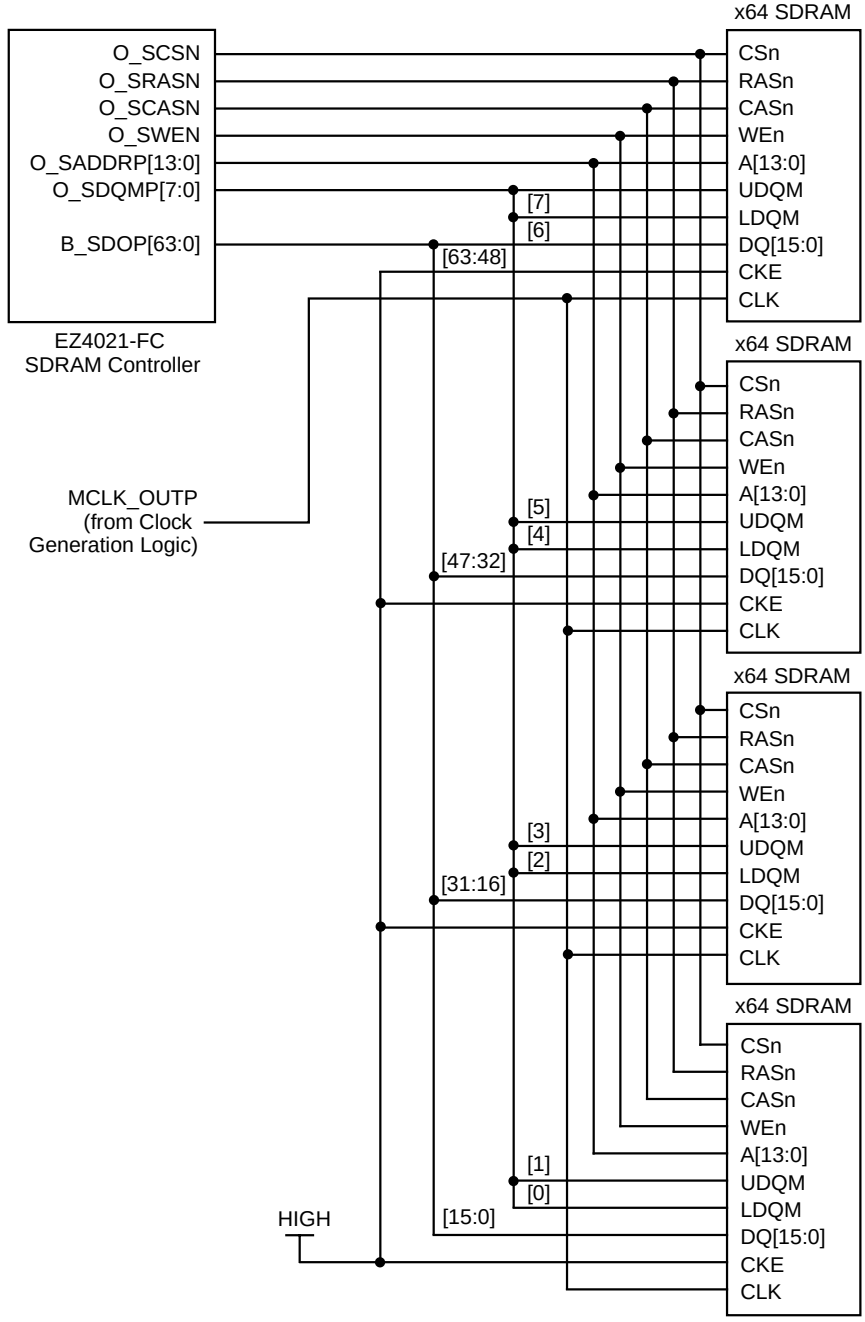
Table 2.1 SDRAM Types and Available Memory Area

SDRAM Type	Devices Needed	Total Memory Size	Address Area
4 Mwords x 16 bits	4	32 Mbytes	0x0000.0000–0x01FF.FFFF
8 Mwords x 8 bits	8	64 Mbytes	0x0000.0000–0x03FF.FFFF
16 Mwords x 4 bits	16	128 Mbytes	0x0000.0000–0x07FF.FFFF

2.5 SDRAM Controller and SDRAM Device Interface

The SDRAM Controller interfaces to the SDRAM array without external glue logic. Figure 2.2 shows a connection diagram for a 1 Mword x 16-bit type SDRAM.

Figure 2.2 SDRAM Controller and SDRAM Array Interface



You must strap the the clock enable (CKE) input HIGH on all SDRAM devices, because the SDRAM Controller does not have a separate CKE signal. Strapping the input HIGH permanently enables the SDRAM devices.

The EZ4021-FC provides the clock for the SDRAM array. Although the EZ4021-FC internally adjusts the phase, the skew in the SDRAM array must be carefully monitored. Placing the devices as close together as possible alleviates some of the skew. Multiple clocks, each derived from the master memory clock, can be used if the array is large.

2.6 SDRAM Addressing

The EZ4021-FC SDRAM Controller passes address information between the Quick Bus and the SDRAM array. The SDRAM Controller receives a 32-bit address from the Quick Bus and communicates with the SDRAM array using the SDRAM 14-bit address bus.

64 Mbit SDRAM devices contain four internal banks. Row address bits O_SADDRP[13:12] select the bank. One row address, specified by row address bits [11:0], is kept in each bank. All four banks may be activated independently.

The controller manages four pages (banks), which are selected by row address bits [13:12]. The SDRAM Controller also holds four sets of address bits [22:11].

2.6.1 Quick Bus to SDRAM Controller

The Quick Bus communicates with the SDRAM Controller using a 32-bit bus, QB_ADDRP[31:0]. The valid bits are encoded in the QB_ADDRP[26:3] signal. Figure 2.3 shows the address bit assignment for the SDRAM Controller.

Figure 2.3 SDRAM Controller Address Bit Assignment (QB_ADDRP[31:0])

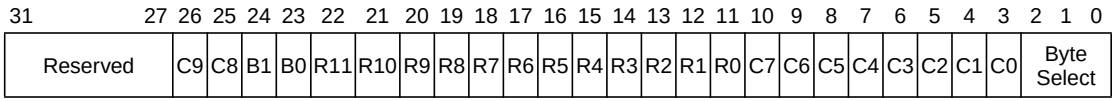


Table 2.2 lists the bit assignments of the address bus shown in Figure 2.3.

Table 2.2 SDRAM Controller Address Bit Assignments (QB_ADDRP[31:0])

Address Bits	Description
31:27	These bits must be zero. The SDRAM Controller does not respond to the transaction if any of these bits are nonzero.
26:25	These bits are assigned to column address bits [9:8]. When 8-bit wide SDRAMs are used, bit 25 is ignored. When 16-bit wide devices are used, both bits are ignored.
24:23	This signal selects one of four internal banks. Note that these banks reside internal to each SDRAM device. A value of 0b00 on these bits selects internal bank 0, 0b01 selects bank 1, 0b10 selects bank 2, and 0b11 selects bank 3. These bits are used as bits 13:12 for both the row and column address.
22:11	SDRAM row address [11:0].
10:3	SDRAM column address [7:0]
2:0	These three bits are ignored during all operation.

2.6.2 SDRAM Controller to SDRAM

The SDRAM Controller communicates with the SDRAM devices using a 14-bit bus, O_SADDRP[13:0]. Row addresses are driven separately from column addresses. Figure 2.4 shows the row and column address bit assignments for 4-, 8-, and 16-bit wide 64 Mbit SDRAM devices. In Figure 2.4, an x indicates a don't care.

Figure 2.4 64 Mbit SDRAM Device Address Bit Assignment (O_SADDRP[13:0]

Row Addressing for all 64 Mbit SDRAM Devices

13	12	11	10	9	8	7	6	5	4	3	2	1	0
B1	B0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0

Column Addressing for 4-bit wide 64 Mbit SDRAM Devices

13	12	11	10	9	8	7	6	5	4	3	2	1	0
B1	B0	x	0	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

Column Addressing for 8-bit wide 64 Mbit SDRAM Devices

13	12	11	10	9	8	7	6	5	4	3	2	1	0
B1	B0	x	0	x	C8	C7	C6	C5	C4	C3	C2	C1	C0

Column Addressing for 16-bit wide 64 Mbit SDRAM Devices

13	12	11	10	9	8	7	6	5	4	3	2	1	0
B1	B0	x	0	x	x	C7	C6	C5	C4	C3	C2	C1	C0

2.7 SDRAM Controller Signals

This section describes the SDRAM Controller signals, which are listed in Table 2.3. Figure 2.5 shows the connections between the Quick Bus controller, the SDRAM Controller, and the SDRAM array. Refer to Figure 2.2 on page 5 for more details on the SDRAM Controller to SDRAM array connections.

Figure 2.5 SDRAM Controller Connection Diagram

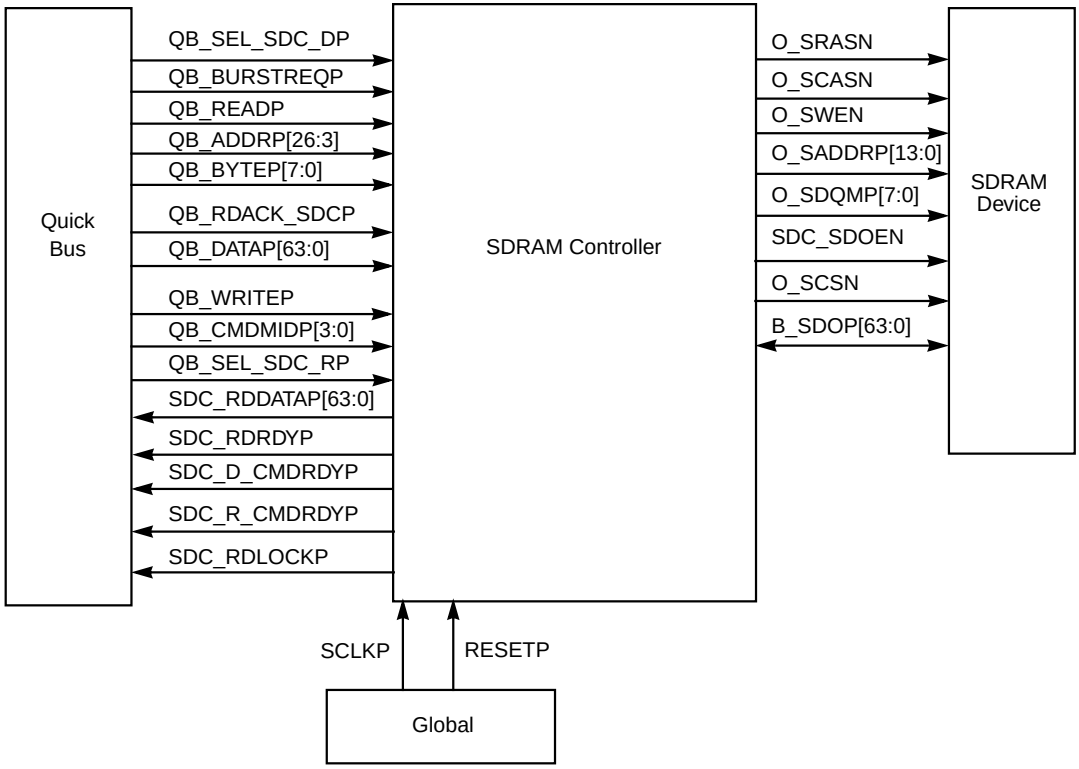


Table 2.3 SDRAM Controller Alphabetical Signal List

Signal Name	Input/Output	Source	Destination	Description
B_SDOP[63:0]	Bidirectional	SDRAM Array SDRAM Cont.	SDRAM Cont. SDRAM Array	Read data from SDRAM Write data to SDRAM
O_SADDRP[13:0]	Output	SDRAM Cont.	SDRAM Array	Address
O_SCASN	Output	SDRAM Cont.	SDRAM Array	Column Address Select (CAS) Command
O_SCSN	Output	SDRAM Cont.	SDRAM Array	Chip Select
O_SDQMP[7:0]	Output	SDRAM Cont.	SDRAM Array	Byte Mask
O_SRASN	Output	SDRAM Cont.	SDRAM Array	Row Address Select (RAS) Command

Table 2.3 SDRAM Controller Alphabetical Signal List (Cont.)

Signal Name	Input/Output	Source	Destination	Description
O_SWEN	Output	SDRAM Cont.	SDRAM Array	Write Enable
QB_ADDRP[26:3]	Input	Quick Bus	SDRAM Cont.	Address
QB_BURSTREQP	Input	Quick Bus	SDRAM Cont.	Four-Doubleword (Burst) Request
QB_BYTEP[7:0]	Input	Quick Bus	SDRAM Cont.	Byte Enable
QB_CMDIDP[3:0]	Input	Quick Bus	SDRAM Cont.	Command ID
QB_WRDATAP[63:0]	Input	Quick Bus	SDRAM Cont.	Write Data
QB_RDACK_SDCP	Input	Quick Bus	SDRAM Cont.	Read Data Acknowledge
QB_READP	Input	Quick Bus	SDRAM Cont.	Read Request
QB_SLSEL_SDC_DP	Input	Quick Bus	SDRAM Cont.	Memory Access Request
QB_SLSEL_SDC_RP	Input	Quick Bus	SDRAM Cont.	Register Access Request
QB_WRITEP	Input	Quick Bus	SDRAM Cont.	Write Request
RESETP	Input	–	SDRAM Cont.	System Reset
SCLKP	Input	Global	SDRAM Cont.	System Clock
SDC_D_CMDRDYP	Output	SDRAM Cont.	Quick Bus	SDRAM Controller Data Command Ready
SDC_R_CMDRDYP	Output	SDRAM Cont.	Quick Bus	SDRAM Controller Register Command Ready
SDC_RDDATAP[63:0]	Output	SDRAM Cont.	Quick Bus	Memory read data
SDC_RDLOCKP	Output	SDRAM Cont.	Quick Bus	Read Return Lock
SDC_RDRDYP	Output	SDRAM Cont.	Quick Bus	Register read ready
SDC_SDOEN	Output	SDRAM Cont.	SDRAM	Data Out Enable

2.7.1 Signal Descriptions

B_SDOP[63:0] Memory Data In/Out **Bidirectional**

This 64-bit bus carries data between the SDRAM Controller and the SDRAM array. The bus operates as an input to the SDRAM Controller during read accesses, and as an output to the SDRAM array during writes. The SDC_SDOEN signal controls the direction of this bus.

O_SADDRP[13:0]

Row/Column Address **Output**

This 14-bit bus outputs the address for a given transaction. During a row address select (RAS) cycle, the SDRAM Controller drives row address information on this bus. During a column address select (CAS) cycle, this bus drives the column address. Refer to Section 2.6, "SDRAM Addressing," on page 2-6 for more information.

O_SCASN Column Address Select (CAS) Command Output

The SDRAM Controller asserts this signal when driving a column address to the SDRAM array. The controller also uses O_SCASN, along with O_SWEN and O_SRASN, as a control signal for signaling commands such as mode register set, active, and precharge to SDRAM devices. Refer to Section 2.9, "SDRAM Commands," on page 2-16, for more information.

O_SCSN Chip Select Output

The SDRAM Controller asserts this signal to enable the SDRAM device.

O_SDQMP[7:0]

Byte Enable **Output**

This 8-bit bus operates as a byte mask for B_SDOP[63:0] on write transactions. The following table shows the correspondence between the byte enable signals and the valid data bits.

Byte Enable	Valid Data Bits	Byte Enable	Valid Data Bits
O_SDQMP[7]	[63:56]	O_SDQMP[3]	[31:24]
O_SDQMP[6]	[55:48]	O_SDQMP[2]	[23:16]
O_SDQMP[5]	[47:40]	O_SDQMP[1]	[15:8]
O_SDQMP[4]	[39:32]	O_SDQMP[0]	[7:0]

O_SRASN **Row Address Select (RAS) Command** **Output**

The SDRAM Controller asserts this signal when driving a row address to the SDRAM array. The controller also uses O_SRASN, along with O_SWEN and O_SCASN, as a control signal for signaling commands such as mode register set, active, and precharge to SDRAM devices. Refer to Section 2.9, “SDRAM Commands,” on page 2-16, for more information.

O_SWEN **Write Enable** **Output**

The SDRAM Controller uses this signal in two ways. The controller asserts this signal to indicate the current transaction is a write transaction. The controller also uses O_SWEN, along with O_SRASN and O_SCASN, as a control signal for signaling commands such as mode register set, active, and precharge to SDRAM devices. Refer to Section 2.9, “SDRAM Commands,” on page 2-16, for more information.

QB_ADDRP[26:3]

Address **Input**

This 24-bit bus carries the read or write address from the Quick Bus to the SDRAM Controller. The SDRAM Controller, in turn, signals address information to the SDRAM device using the O_SADDRP[13:0] signal. Refer to Section 2.6, “SDRAM Addressing,” page 2-6, for more information.

QB_BURSTREQP

Burst Request **Input**

The Quick Bus asserts this signal to indicate a burst (four-doubleword) transaction.

QB_BYTEP[7:0]

Byte EnableInput

The Quick Bus uses this 8-bit bus to indicate which bytes are valid on QB_DATAP[63:0]. The SDRAM Controller passes this data to the SDRAM device using O_SDQMP[7:0]. The following table shows the correspondence between the byte enable signals and the valid data bits.

Byte Enable	Valid Data Bits	Byte Enable	Valid Data Bits
QB_BYTEP[7]	[63:56]	QB_BYTEP[3]	[31:24]
QB_BYTEP[6]	[55:48]	QB_BYTEP[2]	[23:16]
QB_BYTEP[5]	[47:40]	QB_BYTEP[1]	[15:8]
QB_BYTEP[4]	[39:32]	QB_BYTEP[0]	[7:0]

QB_CMDIDP[3:0]
Command ID **Input**
This 4-bit bus carries the command ID for each request from the Quick Bus. On read returns, the SDRAM Controller returns the command ID on the SDC_RDMID_RP[3:0] signal.

QB_DATAP[63:0]
Write Data In **Input**
This 64-bit bus contains write data from the Quick Bus.

QB_RDACK_SDCP
Read Data Acknowledge **Input**
The Quick Bus asserts this signal to acknowledge receiving input data from the SDRAM Controller.

QB_READP **Read Request** **Input**
The Quick Bus asserts this signal to indicate a read request.

QB_SLSEL_SDC_DP
Memory Access Request **Input**
The Quick Bus asserts this signal to request a read or write transaction from the SDRAM Controller.

QB_SLSEL_SDC_RP
Register Access Request **Input**
The Quick Bus asserts this signal to select an internal register in an SDRAM device for read or write access.

QB_WRITEP **Write Request** **Input**
The Quick Bus asserts this signal to indicate a write request to the SDRAM Controller.

RESETP **System Reset** **Input**
Master system reset input. The SDRAM Controller is idle after reset.

SCLKP **System Clock** **Input**

Master system clock input. All transactions occur on the rising edge of the clock.

SDC_D_CMDRDYP

SDRAM Controller Data Command Ready **Output**

The SDRAM Controller asserts this signal when capable of handling data requests. The SDRAM Controller deasserts command ready when the command buffer is full. Typically, command ready is asserted unless a burst request is pending.

SDC_R_CMDRDYP **Output**

SDRAM Controller Register Command Ready

The SDRAM Controller asserts this signal when capable of handling requests to internal registers. The SDRAM Controller deasserts command ready when the command buffer is full. Typically, command ready is asserted unless a burst request is pending.

SDC_RDDATAP[63:0]

Memory Read Data **Output**

This 64-bit bus contains read data from the SDRAM array.

SDC_RDLOCKP

Read Return Lock **Output**

This signal must be tied LOW. The SDRAM Controller never locks the Quick Bus.

SDC_RDRDYP

Memory Data Ready **Output**

The SDRAM Controller asserts this signal to indicate that valid data is on the Quick Bus.

SDC_SDOEN **Data Out Enable** **Output**

This signal determines the direction of the B_SDOP[63:0] bus. The SDRAM Controller asserts SDC_SDOEN to indicate that the B_SDOP[63:0] bus is operating as an output (that is, writing to the SDRAM array). When deasserted, the B_SDOP[63:0] bus operates as an input (that is, reading from the SDRAM array).

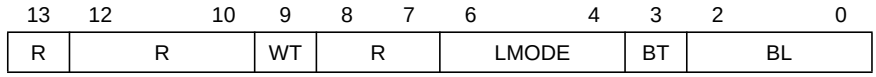
2.8 Programming the SDRAM Mode Register

SDRAM have several programmable parameters, as discussed in Section 2.3. These parameters are set in the mode register of each SDRAM device.

The Mode Register Set (MRS) command causes the SDRAM Controller to send programming information to the SDRAM. The same content is written to all SDRAM devices. The information written depends upon the programming of the SDRAM Controller Configuration Register. Refer to Section 2.10.1, “SDRAM Controller Configuration Register,” page 2-20, for more information.

The SDRAM mode register for 64 Mbit devices contains a 14-bit wide mode register. Figure 2.6 shows the SDRAM mode register.

Figure 2.6 SDRAM Mode Register



R **Reserved** **13**
This bit must be set to zero for 64 Mbit devices.

R **Reserved** **[12:10]**
These bits must be set to zero.

WT **Write-Through Bit** **9**
When cleared, this bit selects write-through mode. The SDRAM Controller only supports write-through mode.

R **Reserved** **[8:7]**
These bits must be set to zero.

LMODE **Column Address Latency (CAS Latency)** **[6:4]**
These bits set the amount of CAS latency on reads. CAS latency is the number of SDRAM clock cycles between the column address being driven to the SDRAM and the SDRAM returning data.

64 Mbit SDRAM devices only support column latencies of 2 or 3 cycles. A latency of one cycle is not supported.

LMODE[2:0]	CAS Latency
000	Reserved
001	Unsupported for 64 MBit Devices
010	2 Cycles
011	3 Cycles
1xx[1]	Reserved

1. x = Don't Care

BT **Burst Type** **3**

This bit must be cleared during initialization. The EZ4021-FC SDRAM Controller does not support burst mode. In lieu of burst mode, the EZ4021-FC SDRAM Controller supports a four-doubleword transaction that performs four sequential accesses to the same row.

BL **Burst Length** **[2:0]**

This field must be set to 0b000, as the EZ4021-FC SDRAM Controller only supports single-word accesses. In lieu of burst mode, the controller supports a four-doubleword transaction that performs four sequential accesses to the same row.

2.9 SDRAM Commands

SDRAM devices support a number of commands. The Quick Bus sends these commands to the SDRAM Controller on the QB_ADDRP[26:3] bus. The SDRAM Controller passes commands to the SDRAM array using the O_SRASN, O_SCASN, O_SWEN, and O_SADDRP[13:0] signals.

Table 2.4 shows a truth table of the supported SDRAM commands. The address bits in Table 2.4 refer to the QB_ADDRP[26:3] bus. The controller maps these bits to the O_SADDRP[13:0] bus.

Table 2.4 SDRAM Command Summary (64 Mbit Mode)[1]

SDRAM Command (Mnemonic)	SDRAM Controller Signals			O_SADDRP[13:0] Bit Encoding[2]				
	O_SRASN	O_SCASN	O_SWEN	[13]	[12]	[11]	[10]	[9:0]
Row Active (ACTV)	LOW	HIGH	HIGH	A[24]	A[23]	A[22]	A[21]	A[20:11]
Mode Register Set (MRS)	LOW	LOW	LOW	A[24]	A[23]	A[22]	A[21]	A[20:11]
No Operation (NOP)	HIGH	HIGH	HIGH	x	x	x	x	x
Precharge All (PALL)	LOW	HIGH	LOW	x	x	x	HIGH	x
Precharge Bank 0 (PRE)	LOW	HIGH	LOW	LOW	LOW	x	LOW	x
Precharge Bank 1 (PRE)	LOW	HIGH	LOW	LOW	HIGH	x	LOW	x
Precharge Bank 2 (PRE)	LOW	HIGH	LOW	HIGH	LOW	x	LOW	x
Precharge Bank 3 (PRE)	LOW	HIGH	LOW	HIGH	HIGH	x	LOW	x
Read (READ)	HIGH	LOW	HIGH	A[24]	A[23]	x	LOW	A[24,23, 10:3]
Refresh (REF)	LOW	LOW	HIGH	x	x	x	x	x
Write (WRIT)	HIGH	LOW	LOW	A[24]	A[23]	x	LOW	A[24,23, 10:3]

1. x = Don't Care.
2. Address bits A refer to the QB_ADDRP[26:3] signal.

2.9.1 Row Active (ACTV) Command

The ACTV command executes prior to any read or write operation. The ACTV command latches and decodes the row address and activates the appropriate row within the SDRAM device. Once the row has been latched, a READ or WRIT command latches the appropriate column address to access the SDRAM.

When the ACTV command completes, read or write operations can occur and a column address is driven.

2.9.2 Mode Register Set (MRS) Command

The MRS command executes on power-up, or whenever the SDRAM Controller modifies the operating parameters of the SDRAM device.

The MRS command initializes the mode register, which contains the operating parameters for each SDRAM device in the memory array. The mode register is programmed during initialization. Refer to Section 2.8, “Programming the SDRAM Mode Register”, on page 2-15, for more information.

2.9.3 No Operation (NOP) Command

The NOP command has no effect on the internal operation of the SDRAM state machines and any cycles in progress are allowed to continue. Some SDRAM vendors require a NOP operation during intialization. To manually issue the NOP command, set the NOP bit in the SDRAM Controller Configuration register.

2.9.4 Precharge All Banks (PALL) Command

The PALL command initiates a precharge operation to all banks of the SDRAM. Precharging a bank clears the previous row address and prepares the bank for subsequent operations. You must issue the precharge all command before issuing a refresh command.

In a 64 Mbit SDRAM, address bit O_SADDRP[10] determines whether all banks are precharged, or only a particular bank. This bit must be set to precharge all banks. The SDRAM ignores address bits O_SADDRP[13:12] during this command.

Other cycles cannot execute while the PALL command is in progress. At the completion of the PALL command, the controller enters the idle state.

2.9.5 Precharge Selected Bank (PRE) Command

The PRE command precharges a single bank of the SDRAM device. Precharging a bank clears the previous row address and prepares the

bank for subsequent operations. The SDRAM Controller does not precharge the bank if subsequent operations are to the same row.

In a 64 Mbit SDRAM, address bits O_SADDRP[13:12] select between the four banks. Address bit O_SADDRP[10] must be deasserted to use the PRE command.

O_SADDRP[13:12]	O_SADDRP[10]	Selected Bank
0b00	0b0	Bank 0
0b01	0b0	Bank 1
0b10	0b0	Bank 2
0b11	0b0	Bank 3
0bxx	0b1	All Banks

The state of the control signals for a PRE command is identical to the PALL command with the exception of address bit O_SADDRP[10]. If O_SADDRP[10] is LOW, a PRE command executes; if O_SADDRP[10] is HIGH, a PALL command executes.

2.9.6 Read (READ) Command

The READ command signals a read operation to the SDRAM. During execution of a READ command, a column address is latched into the appropriate SDRAM device. The row address for the read operation is latched by the ACTV command.

When a READ command executes, data is available either two or three cycles later, according to the value programmed in the CAS latency field of the SDRAM mode register. At the completion of the READ command, the SDRAM output buffers are placed into the high-impedance state.

2.9.7 Refresh (REF) Command

The REF command refreshes the voltage in the SDRAM device to prevent data loss (the REF command is equivalent to a CAS-before-RAS refresh in a standard DRAM). You must precharge all banks prior to the REF command.

The SDRAM Controller must be idle to execute the REF command; all cycles in progress complete prior to command execution. The refresh command halts operations in progress (except for burst reads). Burst reads finish prior to REF command execution.

An on-chip refresh counter provides refresh address and bank select bits; no external address counter is required. Each bank is automatically precharged after a REF operation—the PRE or PALL commands are not required following a refresh operation.

The REF command cannot be interrupted. The SDRAM Controller returns to the idle state after completing the refresh operation.

2.9.8 Write (WRIT) Command

The WRIT command signals a write operation to the SDRAM. During execution of a WRIT command, a column address is latched into the appropriate SDRAM device. The row address for the write operation is latched by the ACTV command.

2.10 SDRAM Controller Registers

There are two SDRAM controller registers located at the following virtual addresses.

- SDRAM Controller Configuration register (0xBEFF.FFD0)
- SDRAM Controller Refresh register (0xBEFF.FFD4)

These registers are defined in the following subsections.

2.10.1 SDRAM Controller Configuration Register

The SDRAM Controller Configuration register contains the general operating parameters for the SDRAM controller. This register is located at physical address 0x1EFF.FFD0.

Figure 2.7 SDRAM Controller Configuration Register (R/W)

31	30	29	28	27 25	24	23 22	21 20	19	18 16	15 12	11 8	7 3	2 1	0
64M	PC	MRS	REF	R	NOP	R	CL	R	RCD	RC	RAS	Reserved	RP	DPL

64M **64 Mbit SDRAM** **31**

This bit selects the address format for the memory array. This bit must be set to use 64 Mbit SDRAMs. This bit is cleared on reset.

PC **Manual Precharge** **30**

When set, this bit causes the SDRAM Controller to issue a manual precharge command. The bit is cleared on reset and automatically cleared after the precharge command is issued.

During SDRAM initialization, the PC, MRS, and REF bits must be set during the initial write to the configuration register. Setting these bits causes the following sequence of operations: precharge, mode register set, refresh.

MRS **Mode Register Set** **29**

When set, this bit causes the SDRAM Controller to issue a mode register set command. The mode register set command programs the individual SDRAM devices. The MRS bit is cleared automatically when the command is issued.

During SDRAM initialization, the PC, MRS, and REF bits must be set during the initial write to the configuration register. Setting these bits causes the following sequence of operations: precharge, mode register set, refresh.

REF **Manual Refresh Request** **28**

When set, this bit causes the SDRAM Controller to issue a manual refresh request for all banks of the SDRAM device. The bit is cleared at reset and automatically after the refresh operation is executed.

During SDRAM initialization, the PC, MRS, and REF bits must be set during the initial write to the configuration register. Setting these bits causes the following sequence of operations: precharge, mode register set, refresh.

R **Reserved** **[27:25]**

These bits are read as zero.

NOP **Manual NOP Command** **24**

When set, this bit causes the SDRAM Controller to generate a NOP (No Operation) command to the SDRAM array. This bit is cleared on reset and automatically cleared after the NOP command is executed.

R **Reserved** **[23:22]**

These bits are read as zero.

CL **CAS Latency** **[21:20]**

This field defines the latency, in MCLK_OUTP cycles, between the time when the READ command is issued by the SDRAM Controller and when data is driven onto the bus by the SDRAM. CAS latency applies to all read transactions.

The value in this field is programmed into the LMODE field of each individual SDRAM device. The column latency parameter applies to all read transactions (single and burst).

64 Mbit SDRAM devices only support column latencies of two or three cycles. A latency of one cycle is not supported. The encoding for this field is shown below.

CL[2:0]	CAS Latency
0b000	Reserved
0b001	Unsupported for 64 MBit Devices
0b010	2 Cycles
0b011	3 Cycles
0b1xx[1]	Reserved

1. x = Don't Care

R **Reserved** **19**

This bit is read as zero.

RCD **Active RAS to Read/Write Command Period** **[18:16]**

This field defines the number of clock cycles between the row active (ACTV) command and a READ or WRIT command. This parameter is also known as t_{RCD}.

The encoding for this field is shown below.

RCD[2:0]	ACTV to READ/WRIT Latency t_{RCD} (MCLK_OUTP Cycles)
0b000	Undefined
0b001	1 Cycle
0b010	2 Cycles
0b011	3 Cycles
0b100	4 Cycles
0b101	5 Cycles
0b110	6 Cycles
0b111	7 Cycles

RC **Refresh-to-Refresh/Active Command Period [15:12]**
This field defines the minimum number of clock cycles between a refresh command (REF) and the next refresh or row active (ACTV) command. Select a value between 2 and 15 cycles.

This parameter is also known as t_{RC}.

The encoding for this field is shown below.

RC [15:12]	REF to REF/ACTV Command Period t_{RC} (MCLK_OUTP Cycles)
0b0000	Undefined
0b0001	Undefined
0b0010	2 Cycles
0b0011	3 Cycles
0b0100	4 Cycles
0b0101	5 Cycles
0b0110	6 Cycles
0b0111	7 Cycles
0b1000	8 Cycles
0b1001	9 Cycles

RC [15:12]	REF to REF/ACTV Command Period t_{RC} (MCLK_OUTP Cycles) (Cont.)
0b1010	10 Cycles
0b1011	11 Cycles
0b1100	12 Cycles
0b1101	13 Cycles
0b1110	14 Cycles
0b1111	15 Cycles

RAS **Active to Precharge Command Period [11:8]**

This field defines the minimum number of clock cycles between a row active (ACTV) command and a precharge (PRE or PALL) command.

Select a value between three and seven clock cycles.

This parameter is also known as t_{RAS}.

The encoding for this field is shown below.

RAS[2:0]	ACTV to PRE/PALL Command Period t_{RAS} (MCLK_OUTP Cycles)
0b000–0b010	Undefined
0b011	3 Cycles
0b100	4 Cycles
0b101	5 Cycles
0b110	6 Cycles
0b111	7 Cycles

R **Reserved [7:3]**

These bits are read as zero.

RP **Precharge/MRS to Active Command Period [2:1]**

This field defines the minimum number of clock cycles from a precharge (PRE/PALL) command to the next row active (ACTV) command and the minimum number of cycles from a mode register set (MRS) command to the next ACTV command.

This parameter is also known as t_{RP}.

Select a value between two and five clock cycles. The programmed value sets both parameters. The encoding for this field is shown below.

RP[1:0]	PRE/PALL to ACTV and MRS to ACTV Command Period t_{RP} (MCLK_OUTP Cycles)
0b00	2 Cycles
0b01	3 Cycles
0b10	4 Cycles
0b11	5 Cycles

DPL **Data In to Precharge Command Period** **0**

This bit defines the minimum number of clock cycles from a write (WRIT) command (data in) to a precharge (PRE/PALL) command. Setting this bit indicates a delay of two clock cycles. Clearing this bit indicates a delay of one clock cycle.

This parameter is also known as t_{DPL}.

2.10.1.1 Configuration Register Example

The values programmed into the SDRAM Controller Configuration register depend upon the frequency and type of SDRAM devices used. Refer to the SDRAM vendor's documentation for proper refrseh values.

Table 2.5 shows example values for NEC 64 Mbit SDRAM uPD4564441 and uPD4564841-80, -10, -12 parts including the minimum clock cycles (in ns) for each parameter.

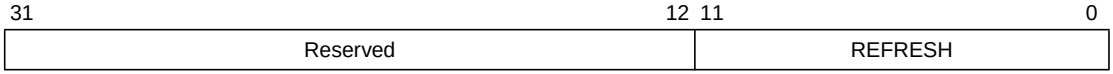
Table 2.5 Configuration Register Programming Example

SDRAM Clock Type/Frequency			Minimum Clock Cycles					
-80 version	**-10 version**	**-12 version**	**CL**	**RCD**	**RC**	**RAS**	**RP**	**DPL**
125 MHz	100 MHz	83 MHz	3	3	10	6	3	1
83 MHz	66 MHz	55 MHz	2	2	7	4	2	1
80 MHz	–	–	2	2	7	4	2	1

2.10.2 SDRAM Controller Refresh Register

All SDRAM devices lose charge over time. To prevent data loss, the voltage in the SDRAM array must be refreshed periodically. The SDRAM Controller Refresh Register stores the refresh interval parameter (the time between refresh operations), which depends upon the SDRAM frequency. This register is located at physical address 0x1EFF.FFD4. Figure 2.8 shows the refresh register.

Figure 2.8 SDRAM Controller Refresh Register (R/W)



Reserved **Reserved** **[31:12]**
This field is read as zero.

REFRESH **Refresh Interval Time** **[11:0]**
This field defines the SDRAM refresh cycle interval time. Table 2.6 shows recommended refresh intervals based on SDRAM clock cycle time.

Table 2.6 SDRAM Refresh Register Programming Value

SDRAM Clock Frequency	SDRAM Clock Cycle Time	Recommended CAS Latency	Recommended Refresh Cycle	
			Decimal	Hex
125 MHz	8 ns	3	1942	0x796
100 MHz	10 ns	3	1551	0x61A
80 MHz	12.5 ns	3	1238	0x4D6
66 MHz	15 ns	2	1031	0x407

2.11 SDRAM Initialization

The SDRAM Controller initializes the memory array at system power-up. During initialization, the controller:

- Assures that V_{DD} is stable
- Precharges all SDRAM banks
- Performs a minimum of eight refresh operations
- Drives the Mode Register Set command

Refer to Section 2.12, “Mbus Timing Waveforms,” page 2-27, for the waveforms associated with SDRAM initialization.

SDRAM requires a minimum of 200 µs between the time when V_{CC} is stable and the PALL command is executed. The time t_{RP} is the amount of time required for precharge of a bank. This time must be a minimum of three cycles.

A minimum of eight consecutive refresh commands are issued by the SDRAM Controller. The time t_{RC} is the row address select cycle time and must be a minimum of eight clocks. The time t_{MRD} is the delay between the MRS command and the initial bank activation (ACTV) command and must be a minimum of three clocks.

2.12 Mbus Timing Waveforms

The SDRAM Controller transfers data to and from memory based on the parameters programmed into the on-chip SDRAM Controller Configuration register. This section shows signal waveforms associated with the following SDRAM transactions:

- single read
- single write
- burst read
- burst write
- initialization

This section shows signal names for bidirectional signals (such as B_SDOP[63:0]) explicitly as input or output signals. For example, B_SDOP[63:0] is shown as I_SDOP[63:0] when it operates as an input signal and O_SDOP[63:0] when it operates as an output signal.

The waveforms also show the internal state of the SDRAM controller for clarity, using the STATE signal. The STATE signal is not externally visible to the user. The acronyms used to indicate state are given in Table 2.7.

Table 2.7 STATE Acronyms

STATE	Description
CAS	Column Address
CBR	Refresh
IDLE	Idle State (inactive)
MRW	Mode Register Set (write)
PCA	Precharge
RA	Row Address
RCDx	x-Cycle Wait State (Active to Read/Write Command)

2.12.1 Single Transactions

This section shows example timing waveforms for EZ4021-FC SDRAM Controller single read and write transactions. On a read transaction, the CASn latency parameter in the SDRAM Mode register determines the number of clocks between the SDRAM device latching the column address and the SDRAM Controller driving data out. The column latency parameter applies to all read transactions (single and burst).

Figure 2.9 shows the waveforms for a single read transaction with a column latency value of two cycles.

Figure 2.9 Single Read Transaction (CL = 2)

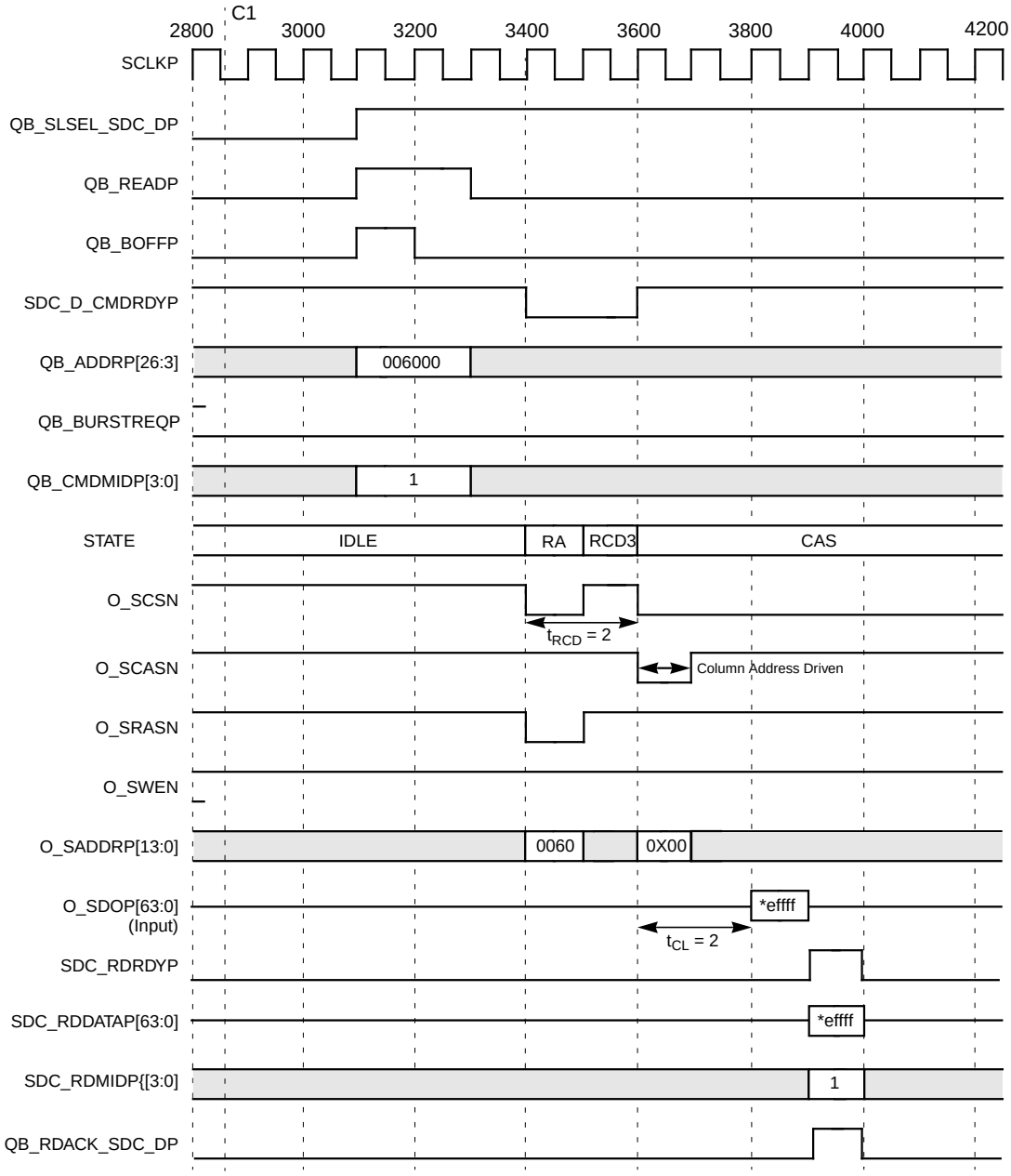


Figure 2.10 shows the timing waveforms for a single read with a CAS latency of three cycles.

Figure 2.10 Single Read Transaction (CL = 3)

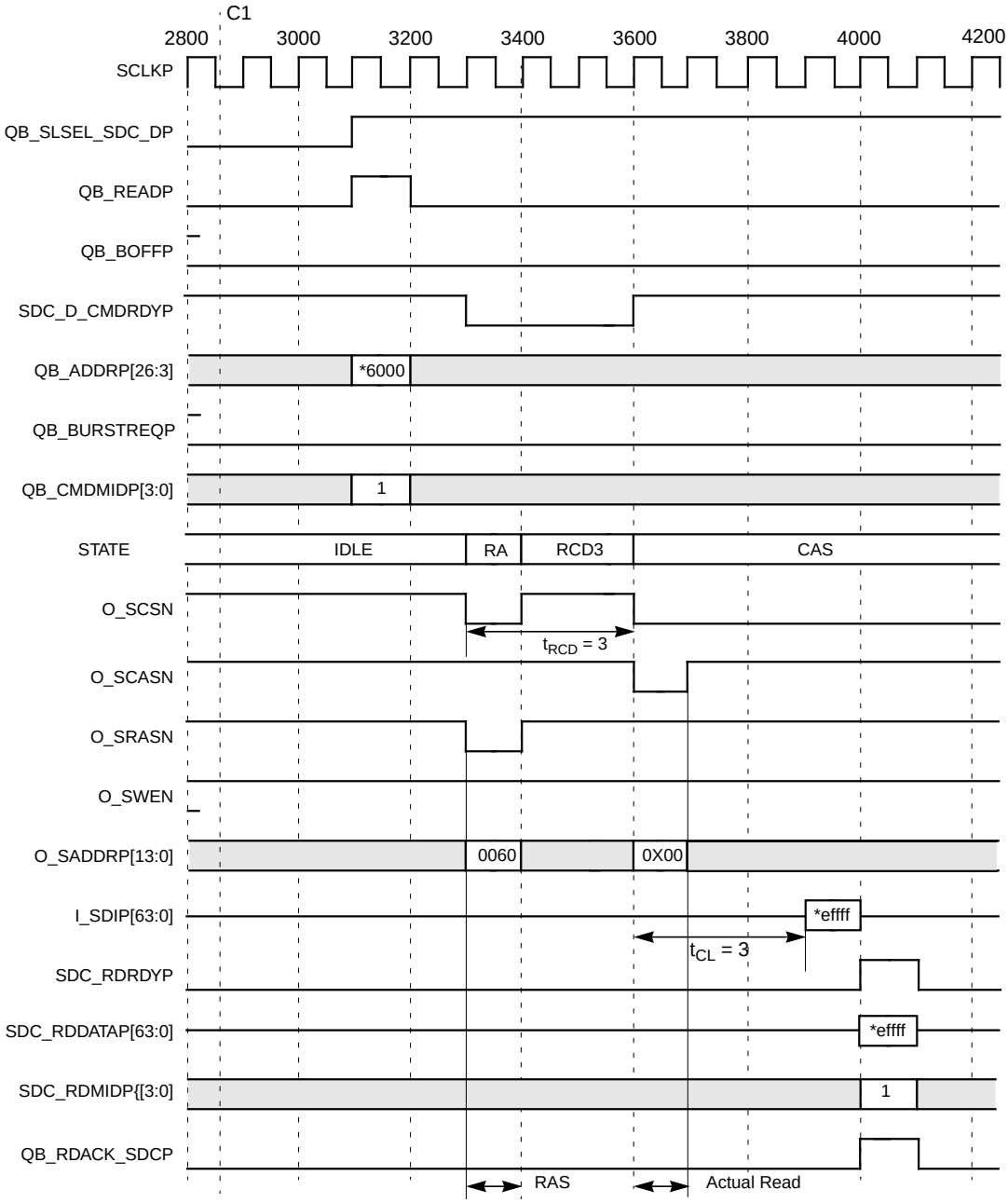
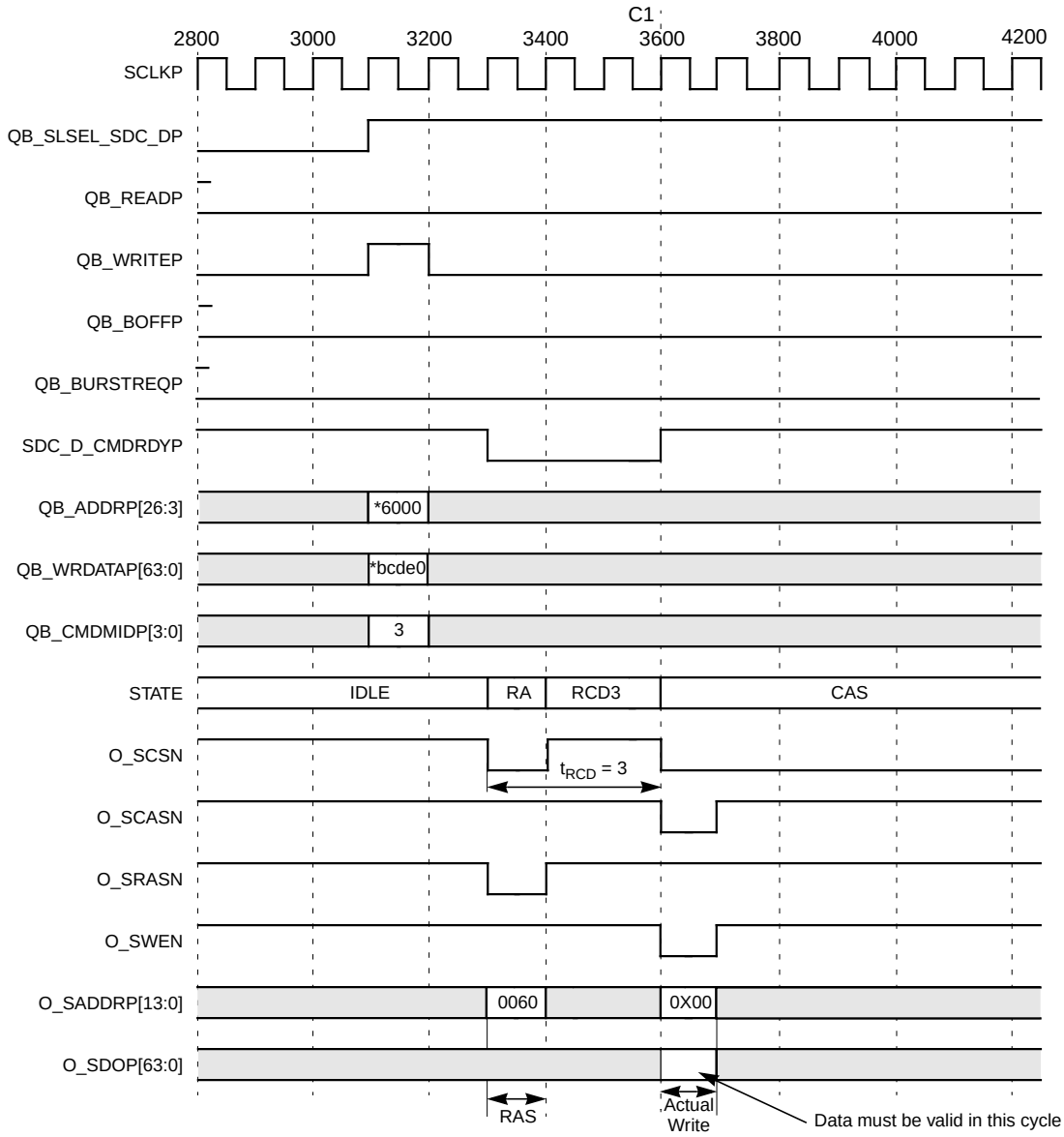


Figure 2.11 shows a single write transaction. The SDRAM Controller asserts O_SWEN to signal a write transaction. The SDRAM Controller drives write data and the column address simultaneously.

Figure 2.11 Single Write Transaction



2.12.2 Burst Transactions

This section shows example timing waveforms for EZ4021-FC SDRAM Controller burst read and write transactions. Asserting QB_BURSTREQP signals a burst transaction.

On a read transaction, the CASn latency parameter in the SDRAM mode register determines the number of clocks between the SDRAM device latching the column address and the SDRAM Controller driving data out.

Figure 2.12 shows the timing waveforms for a burst read transaction with CAS latency set to two cycles.

Figure 2.12 Burst Read Transaction (CL = 2)

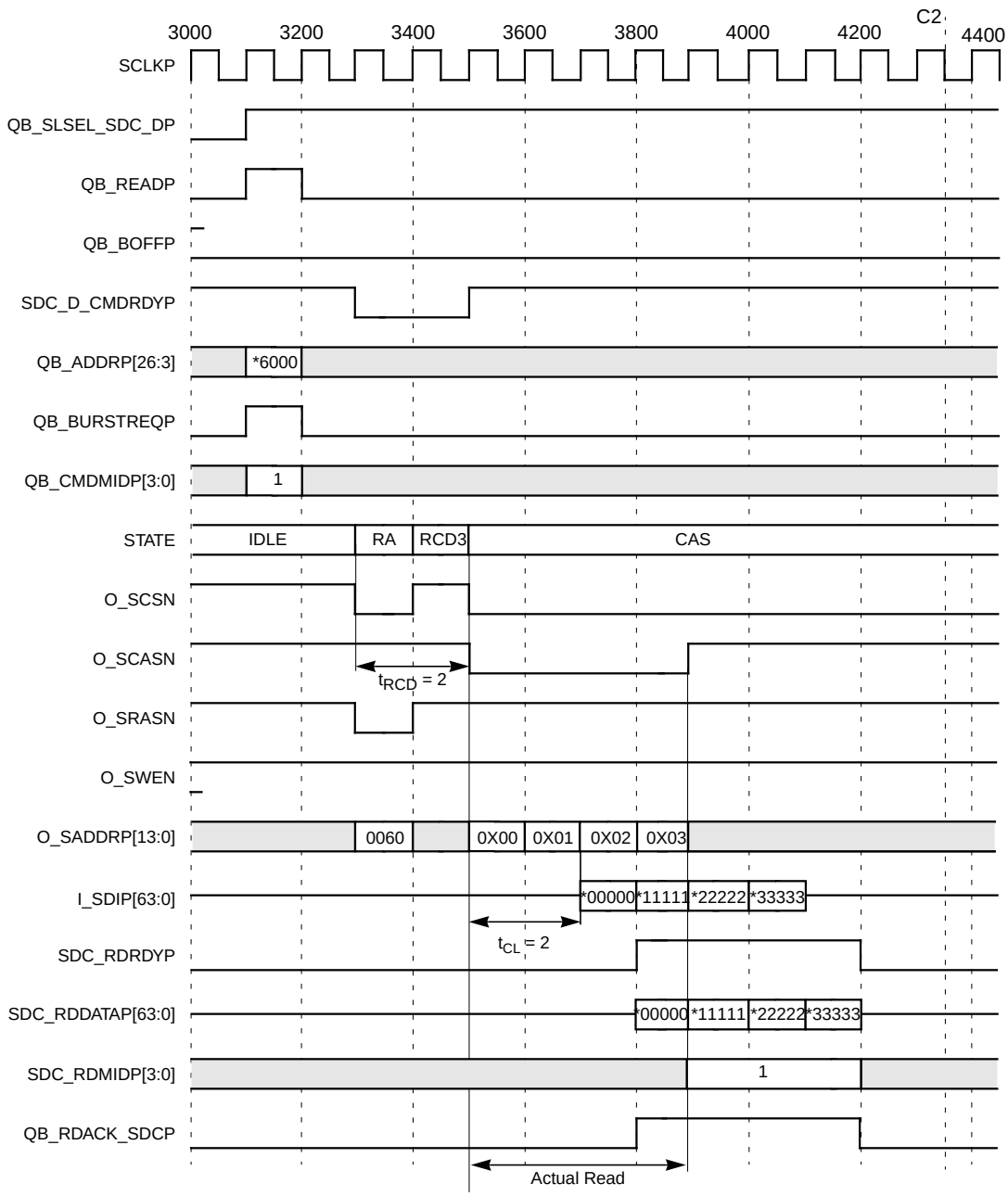
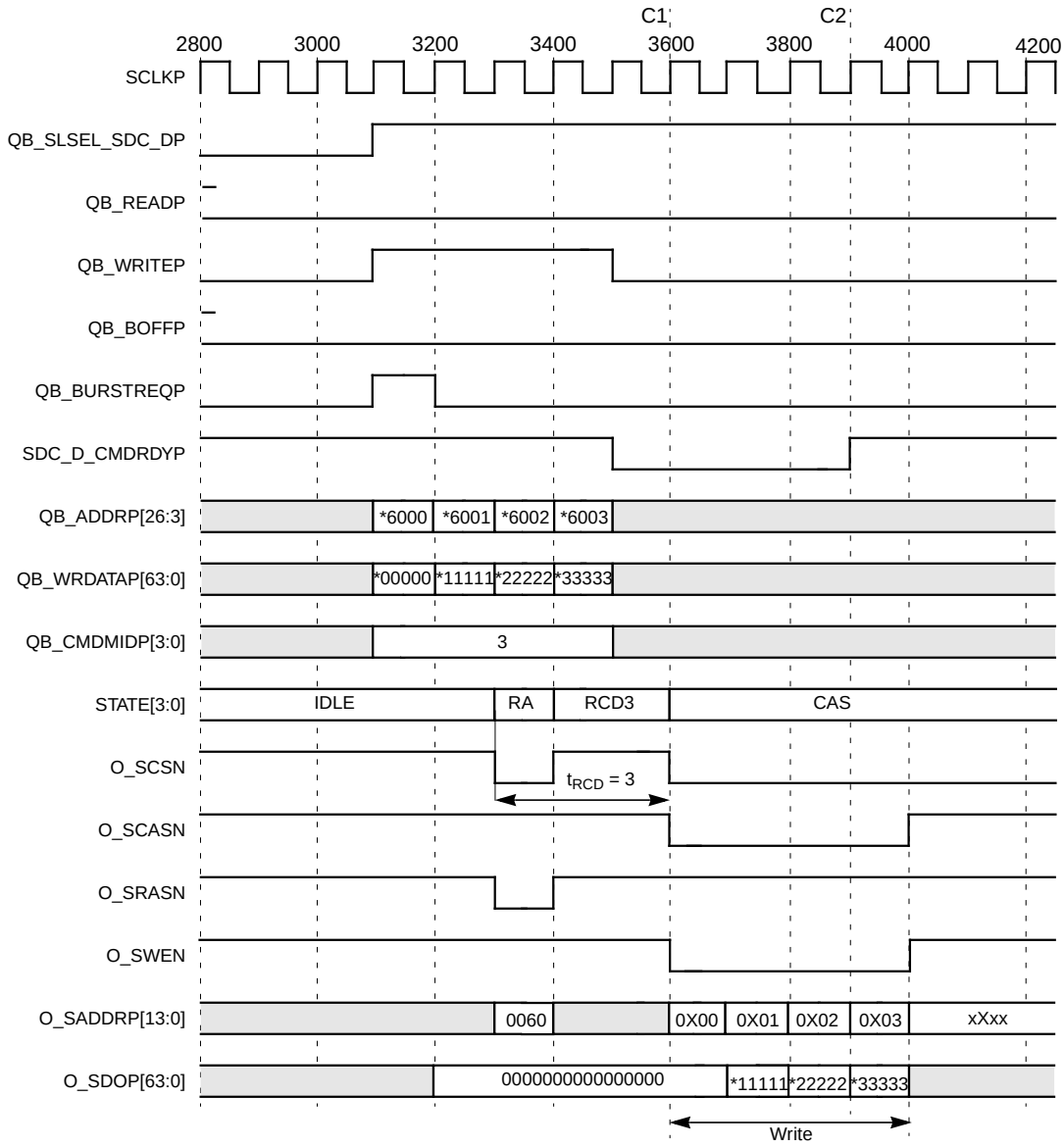


Figure 2.13 shows a burst write transaction. The SDRAM Controller asserts QB_WRITEP to signal a write transaction.

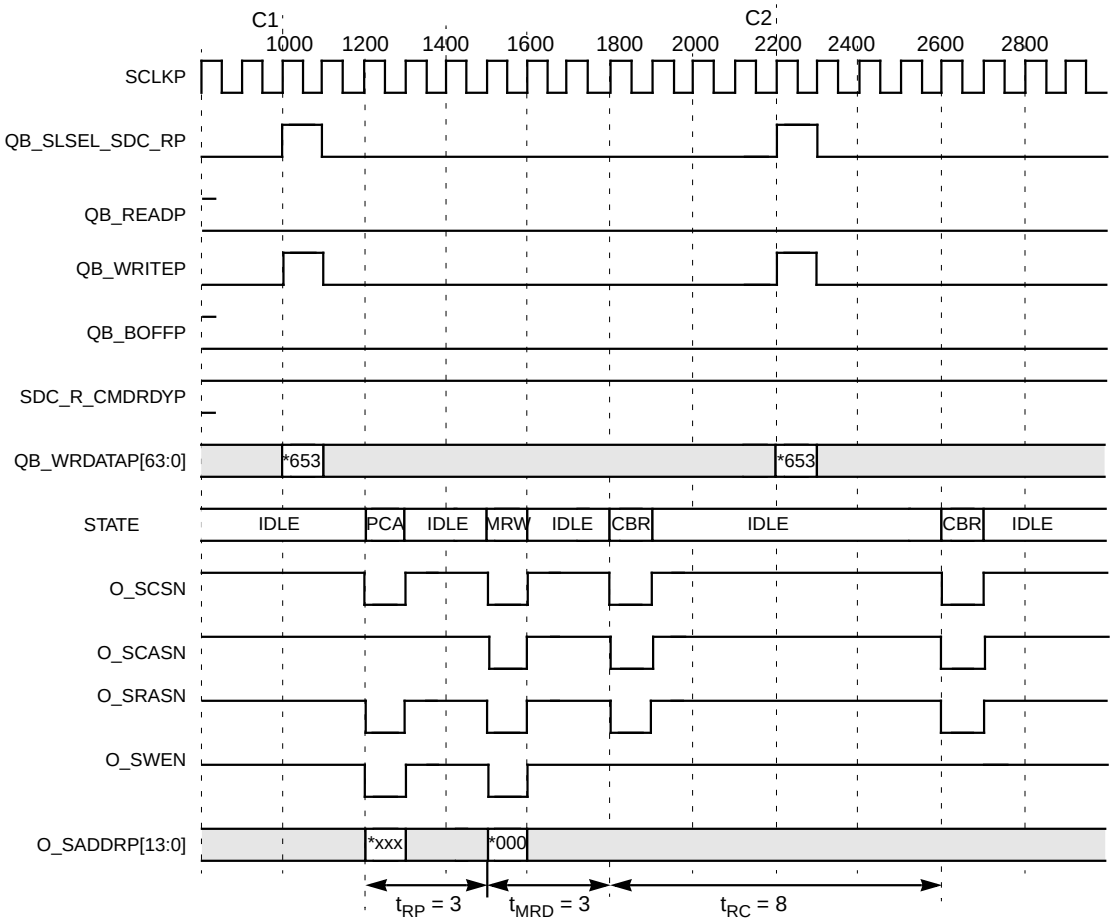
Figure 2.13 Burst Write Transaction



2.12.3 Initialization

The SDRAM array must be initialized on power-up. Figure 2.14 shows the waveforms associated with initialization. The timing parameters for a particular SDRAM vendor may differ from Figure 2.14.

Figure 2.14 Initialization Sequence



Chapter 3
Quick Bus

This chapter explains the operation of the EZ4021-FC's on-chip bus, the Quick Bus, and contains the following sections:

3.1 Overview and Features

The Quick Bus is a 64-bit, high-performance, on-chip bus that allows rapid data transfer between high-bandwidth, high-latency modules that include synchronous DRAMs, pipelined SRAMs, and pipelined Flash EPROMs. It is also suitable for use with low-bandwidth, high-latency devices, such as boot PROMs.

The Quick Bus uses a split-transaction protocol, which makes requests from a master and the response from a slave completely independent. All transactions require one command cycle to send a command from a master to a slave. Read transactions require an additional read return cycle. After a bus master issues a data request, the bus remains unlocked while the master waits for the read return.

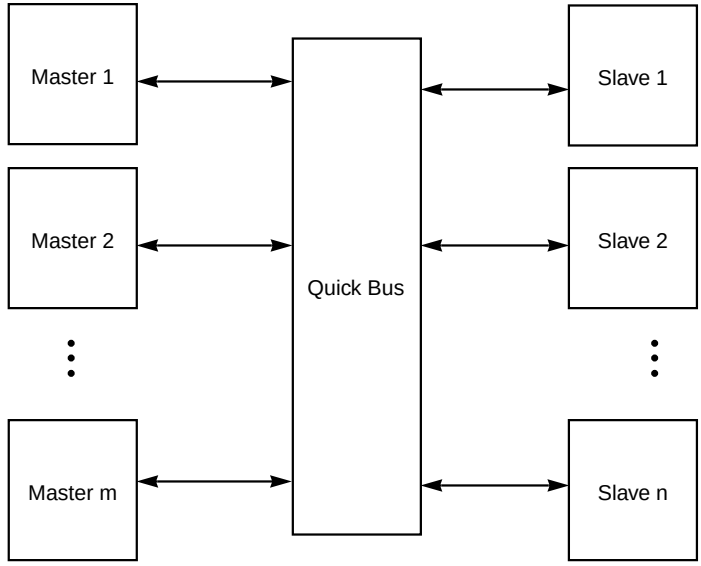
The Quick Bus includes the following features:

- Uses split-transaction protocol
- Handles high-latency devices

- Allows application-dependent arbitration
- Supports these bus operations:
  - burst mode - a standard burst is reading or writing four 64-bit words
  - non-pipelined transactions - only one read or write transaction outstanding at any given time
  - pipelined transactions - multiple read or write transactions outstanding at any given time
- Supports multiple bus masters:
  - Multiple CPUs
    - ◊ I-Cache (each CPU)
    - ◊ D-Cache (each CPU)
  - Off-chip DMA
  - On-chip DMA

Figure 3.1 shows a block diagram of the Quick Bus and attached devices.

Figure 3.1 Quick Bus Block Diagram



3.2 Quick Bus Transactions

The Quick Bus uses a general protocol that works for any application. A Quick Bus master initiates a read or write transaction with a single command cycle. For write transactions, the single command cycle is all that is required. For read transactions, an additional cycle is required to return the requested data from the slave to the requesting master.

3.3 Device Attachment Criteria

Deciding what devices to attach to the Quick Bus is a decision the system designer makes based on the performance and latency of the individual device. The criteria listed here are guidelines with examples. For a more complete list see the next section.

- High-latency devices. Normally, all off-chip devices are high latency.
  - controllers for off-chip memory devices, such as: DRAM, SDRAM, and EPROMs
  - controllers for external buses, such as: PCI, ISA, PC/AT, and PCMCIA
- High bandwidth devices (such as Fibre Channel)
- Masters, such as the following devices that are tightly coupled to memory:
  - CPUs
  - DMA - for example, when serving as a data pump handling I/O “fly-by” transfers to memory

3.4 Supported Devices

Below is a list of devices that should work well on the Quick Bus. Selection of these devices is based on the criteria described in Section 3.3, “Device Attachment Criteria,” page 3-3.

- Bus Masters
- Display Controller

- DMA Controllers
- Ethernet-110
- GigaBlaze® SeriaLink® core
- GigaBlaze Transceiver
- IEEE 1394 Link
- Memory Controllers for:
  - SDRAMs
  - SRAMs
  - Flash EPROMS
  - PROMs
- Merlin® DL Fibre Channel Controller
- PC/ATA Interface
- PC Card (PCMCIA)
- PCI-66 FlexCore® Controller
- RDRAM Controllers
- SCSI Transceiver
- Video DAC
- 2x SmartCard

3.5 Signal Descriptions

This section describes the Quick Bus signals. The signals are divided among the following functional groups:

- Master Command
- Master Read Return
- Slave Command
- Slave Read Return

Signal names in this document are in upper case, whereas in the RTL source code the same signals appear in lower case. The notation <Mm> or <Sn> in a signal name indicates the signal is associated with m number of master devices or n number of slave devices, respectively.

Figure 3.2 shows the Quick Bus signals grouped according to function. Table 3.1 summarizes the Quick Bus signals.

Figure 3.2 Quick Bus Signals

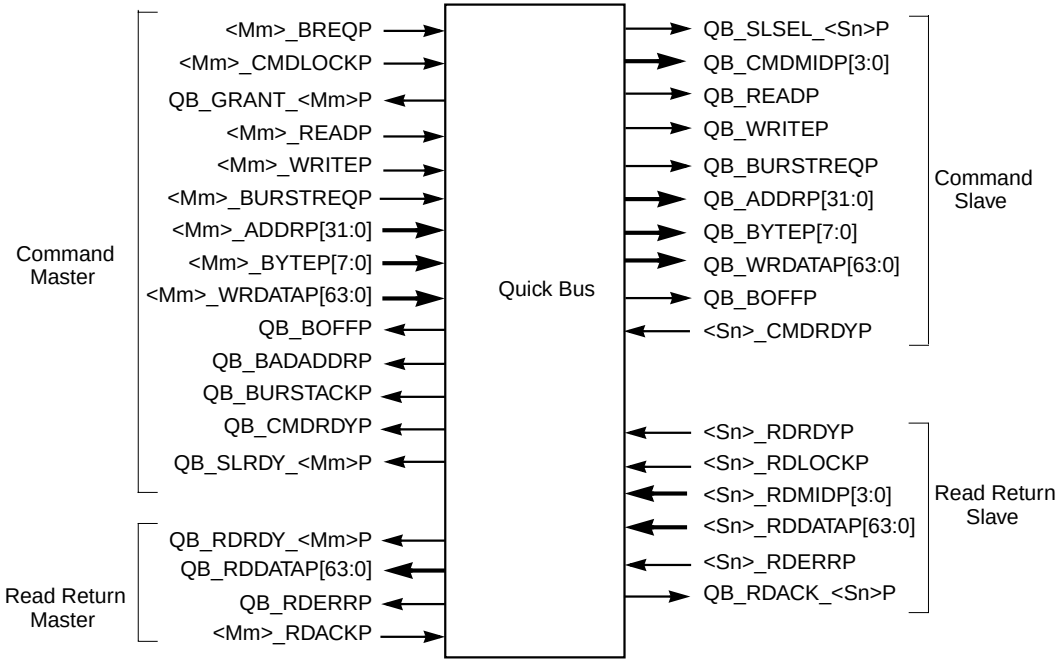


Table 3.1 Quick Bus Signal Summary

Signal Name	Input/ Output	Source	Destination	Description
<Mm>_ADDRP[31:0]	I	Master	Quick Bus	Address
<Mm>_BREQP	I	Master	Quick Bus	Bus Request
<Mm>_BURSTREQP	I	Master	Quick Bus	Burst Request (optional)
<Mm>_BYTEP[7:0]	I	Master	Quick Bus	Byte Enable
<Mm>_CMDLOCKP	I	Master	Quick Bus	Command Lock (optional)
<Mm>_RDACKP	I	Master	Quick Bus	Read Data Acknowledge
<Mm>_READP	I	Master	Quick Bus	Read Request
(Sheet 1 of 3)				

Table 3.1 Quick Bus Signal Summary (Cont.)

Signal Name	Input/ Output	Source	Destination	Description
<Mm>_WRDATAP[63:0]	I	Master	Quick Bus	Write Data
<Mm>_WRITEP	I	Master	Quick Bus	Write Request
<Sn>_CMDRDYP	I	Slave	Quick Bus	Command Ready
<Sn>_RDDATAP[63:0]	I	Slave	Quick Bus	Read Data
<Sn>_RDERRP	I	Slave	Quick Bus	Read Error
<Sn>_RDLOCKP	I	Slave	Quick Bus	Read Lock (optional)
<Sn>_RDMIDP[3:0]	I	Slave	Quick Bus	Read Master ID
<Sn>_RDRDYP	I	Slave	Quick Bus	Read Ready
QB_ADDRP[31:0]	O	Quick Bus	Slave	Address
QB_BADADDRP	O	Quick Bus	Master	Bad Address
QB_BOFFP	O	Quick Bus	Master/Slave	Back Off
QB_BURSTACKP	O	Quick Bus	Master	Burst Acknowledge (optional)
QB_BURSTREQP	O	Quick Bus	Slave	Burst Request
QB_BYTEP[7:0]	O	Quick Bus	Slave	Byte Enables
QB_CMDMIDP[3:0]	O	Quick Bus	Slave	Command Master ID
QB_CMDRDYP	O	Quick Bus	Slave	Command Ready
QB_GRANT_<Mm>P	O	Quick Bus	Master	Bus Grant
QB_RDACK_<Sn>P	O	Quick Bus	Slave	Read Acknowledge
QB_RDDATAP[63:0]	O	Quick Bus	Slave	Read Data
QB_RDERRP	O	Quick Bus	Slave	Read Data Error
QB_RDRDY_<Mm>P	O	Quick Bus	Master	Read Ready
QB_READP	O	Quick Bus	Slave	Read Request
QB_SLRDY_<Mm>P	O	Quick Bus	Master	Slave Ready

(Sheet 2 of 3)

Table 3.1 Quick Bus Signal Summary (Cont.)

Signal Name	Input/ Output	Source	Destination	Description
QB_SLSEL_<Sn>P	O	Quick Bus	Slave	Slave Select
QB_WRDATAP[63:0]	O	Quick Bus	Slave	Write Data
QB_WRITEP	O	Quick Bus	Slave	Write Request
(Sheet 3 of 3)				

3.5.1 Master-Command Signals

This interface provides a path between bus masters and the Quick Bus. Bus masters use this interface to send read and write commands to the Quick Bus.

A Quick Bus master initiates a read or write transaction with a single command cycle. For write transactions, the single command cycle is all that is required. For read transactions, an additional cycle is required to return the requested data from the slave to the requesting master.

Each bus command triggers one of three handshake responses:

- bad address
- command ready
- command not ready

A *bad address* response means the master must abort the command, because no slave device exists at the address specified. A *command ready* response indicates that the addressed slave can accept the command. A *command not ready* means the addressed slave is busy, and that the master must retry the command later.

Read returns can get out of sequence when a master initiates reads with more than one slave at a time. You can avoid this risk by prohibiting a master from having outstanding read requests with multiple slaves. Since slaves typically handle read requests on a first-in-first-out basis, a single slave can handle multiple read requests without risk of getting the return transactions out of sequence.

To manage multiple outstanding requests, a device can interface with the Quick Bus as more than one master device. For example, the EZ4021-FC CPU uses three types of requests: instructions, data, and EJTAG debug. For each request type, the EZ4021-FC can function as a separate Quick Bus master. Although most of the bus signals are shared among the bus masters, there are several signals that must be replicated for each master. The signals include: <Mm>_BREQP, QB_GRANT_<Mm>P, and QB_SLRDY_<Mm>P. The EZ4021-FC CPU provides three sets of these signals, each identified with a different letter (I, D, or E).

If there are multiple CPUs on a chip, each instance of a CPU is also identified by a prefix letter. For example, the <Mm>_BREQ signal might be prefixed by <X>_I, <X>_D, or <X>_E, where <X> identifies a particular CPU instance.

<Mm>_ADDRP[31:0]

Address **M->QB**

This is the read or write address sent from a bus master to the Quick Bus.

<Mm>_BREQP

Bus Request **M->QB**

A bus master asserts this signal to request access to the Quick Bus.

<Mm>_BURSTREQP

Burst Request (optional) **M->QB**

A bus master asserts this signal to request a burst transfer.

<Mm>_BYTEP[7:0]

Byte Enables **M->QB**

A bus master asserts the byte enable signal(s) to indicate which read or write data bytes on the bus are valid.

Byte Enable	Enables Data Bits
<Mm>_BYTEP[0]	[07:00]
<Mm>_BYTEP[1]	[15:08]
<Mm>_BYTEP[2]	[23:16]
<Mm>_BYTEP[3]	[31:24]
<Mm>_BYTEP[4]	[39:32[

Byte Enable	Enables Data Bits
<Mm>_BYTEP[5]	[47:40]
<Mm>_BYTEP[6]	[55:48]
<Mm>_BYTEP[7]	[63:56]

<Mm>_CMDLOCKP

Command Lock (optional) **M->QB**

When a bus master asserts this signal, it overrides the bus arbitration logic and forces a bus grant. Do not use this signal unless the bus was previously granted through normal arbitration. Refer to Section 3.6.5, “Bus Locking,” on page 3-24.

<Mm>_READP

Read Request **M->QB**

A bus master asserts this signal to indicate that the current bus request is a read transaction.

<Mm>_WRDATAP[63:0]

Write Data **M->QB**

A bus master drives these 64 bits of write data along with a write request and the appropriate byte enable signals during a write transaction.

<Mm>_WRITEP

Write Request **M->QB**

A bus master asserts this signal to indicate that the current bus request is a write transaction.

QB_BADADDRP

Bad Address **QB->M**

If the Quick Bus receives a read or write request for a nonexistent device address, the Quick Bus returns a QB_BADADDRP signal to the requesting master device. This signal tells the master device to abort the command because no slave exists at the address specified.

QB_BOFFP

Back Off **QB->M**

This signal causes both the master and the slave to ignore the current command.

QB_BURSTACKP

Burst Acknowledge (optional) **QB->M**

The Quick Bus drives this signal to indicate that the selected slave is capable of supporting burst operations. Typically, the address decode logic generates the Burst Acknowledge signal by ORing together the select signals of all the burst capable slave devices.

QB_CMDRDYP

Command Ready **QB->M**

The Quick Bus forwards this signal to the bus master to indicate whether a slave device can accept a read or write command. QB_CMDRDYP HIGH indicates a slave can accept a read or write command. When the signal is LOW, the slave is busy.

QB_GRANT_<Mm>P

Bus Grant **QB->M**

The Quick Bus asserts this signal to grant access to a bus master.

QB_SLRDY_<Mm>P

Slave Ready **QB->M**

A master can monitor this signal to determine when a Slave is ready. The Quick Bus monitors the <Sn>_CMDRDYP signals and asserts QB_SLRDY_<Mm>P to the master when the slave accessed last is ready to accept a command.

3.5.2 Master-Read Return Signals

This interface provides a path between the Quick Bus and the bus master. The Quick Bus uses this interface to forward read data from the slave to the master.

QB_RDDATAP[63:0]

Read Data **QB->M**

These 64 bits of data are sent to a bus master in response to an earlier read request.

<Mm>_RDACKP

Read Data Acknowledge **M->QB**

The master asserts this signal when it is ready to receive read data. For optimum bus performance, bus masters

normally should be ready to receive data, so this signal is seldom deasserted.

QB_RDERRP **Read Data Error** **QB->M**

The Quick Bus asserts this signal to tell a bus master that the data on the bus is invalid due to an error. Typically, this signal is used to report a nonexistent address or other read error on a bus that is attached to the Quick Bus by a bus bridge.

QB_RDRDY_<Mm>P

Read Ready **QB->M**

The Quick Bus asserts this signal to tell a bus master that read data is on the bus.

3.5.3 Slave-Command

This interface provides a path between the Quick Bus and the slave device. The Quick Bus uses this interface to forward bus commands and related signals from a bus master to the slave.

An individual slave device can have multiple Slave Select (address decodes) and Command Ready signals. For example, a DRAM controller with memory mapped configuration and control registers will have different slave addresses for the DRAM and the memory mapped registers.

QB_ADDRP[31:0]

Address **QB->S**

This is the read or write address the Quick Bus forwards to a slave device from a bus master.

QB_BOFFP **Back Off** **QB->S**

This signal causes both the master and the slave to ignore the current command.

QB_BURSTREQP

Burst Request **QB->S**

The Quick Bus forwards the Burst Request signal to a slave device from a bus master.

QB_BYTEP[7:0]

Byte Enables **QB->S**

These are the read or write data byte enables the Quick Bus forwards to a slave device from a bus master.

Byte Enable	Enables Data Bits
QB_BYTEP[0]	[07:00]
QB_BYTEP[1]	[15:08]
QB_BYTEP[2]	[23:16]
QB_BYTEP[3]	[31:24]
QB_BYTEP[4]	[39:32[
QB_BYTEP[5]	[47:40]
QB_BYTEP[6]	[55:48]
QB_BYTEP[7]	[63:56]

QB_CMDMIDP[3:0]

Command Master ID **QB->S**

Encoded identification of the master that is sending a read or write request command.

QB_READP

Read Request **QB->S**

The Quick Bus sends this signal to a slave to indicate that a bus master is making a data read request.

QB_SLSEL_<Sn>P

Slave Select **QB->S**

The Quick Bus asserts this signal to indicate which slave device is selected to receive the current read/write request. The Quick Bus generates this signal by decoding the read or write data address.

QB_WRDATAP[63:0]

Write Data **QB->S**

This signal contains 64-bit bus of write data that accompanies a write request. The Quick Bus forwards this data from a bus master to the slave device.

QB_WRITEP

Write Request **QB->S**

The Quick Bus asserts this signal to indicate to a slave that a bus master is making a data write request.

<Sn>_CMDRDYP

Command Ready **S->QB**

This signal is always active, even when a slave is not selected. The command ready signal indicates whether a slave can accept a read or write command from a bus master. When <Sn>_CMDRDYP is HIGH, the slave can accept a read or write command. When LOW, the slave is busy.

After a master addresses a slave, the Quick Bus Slave Ready logic forwards the <Sn>_CMDRDYP signal to the master by asserting QB_SLRDY_<Mm>P.

3.5.4 Slave-Read Return

This interface provides a path from the slave to the Quick Bus for sending read data.

QB_RDACK_<Sn>P

Read Acknowledge **QB->S**

This signal tells a slave that it is granted the Read Return bus and the master acknowledges the read return.

A slave must continue to assert a read return until acknowledged. A read return is defined as <Sn>_RDRDYP, <Sn>_RDDATAP[63:0], and <Sn>_RDERRP.

<Sn>_RDDATAP[63:0]

Read Data **S->QB**

These 64 bits of data are being sent from a slave device in response to an earlier read request from a master device.

<Sn>_RDERRP

Read Error **S->QB**

The slave asserts this signal to indicate that the read data currently on the bus is invalid due to an error. Typically, this signal is used to report a nonexistent address or other read error on a bus that is attached to the Quick Bus by a bus bridge.

<Sn>_RDLOCKP

Read Lock (optional) **S->QB**

When the slave asserts this signal, it overrides the read return bus arbitration logic and forces a bus grant. Do not

use this signal unless the bus was previously granted through normal arbitration. Typically, read return bus locking is not used. It can be used if a slave needs to prevent rearbitration in the middle of a burst transaction. Refer to Section 3.6.5, “Bus Locking,” on page 3-24.

<Sn>_RDMIDP[3:0]

Read Master ID **S->QB**

Encoded identification of the master that is to receive the returning read data.

<Sn>_RDRDYP

Read Ready **S->QB**

This signal is the read return bus request. A slave device asserts this signal when it is ready to send data that a bus master previously requested.

3.6 Functional Description

This section describes the functional operation of the Quick Bus.

3.6.1 Command Cycle

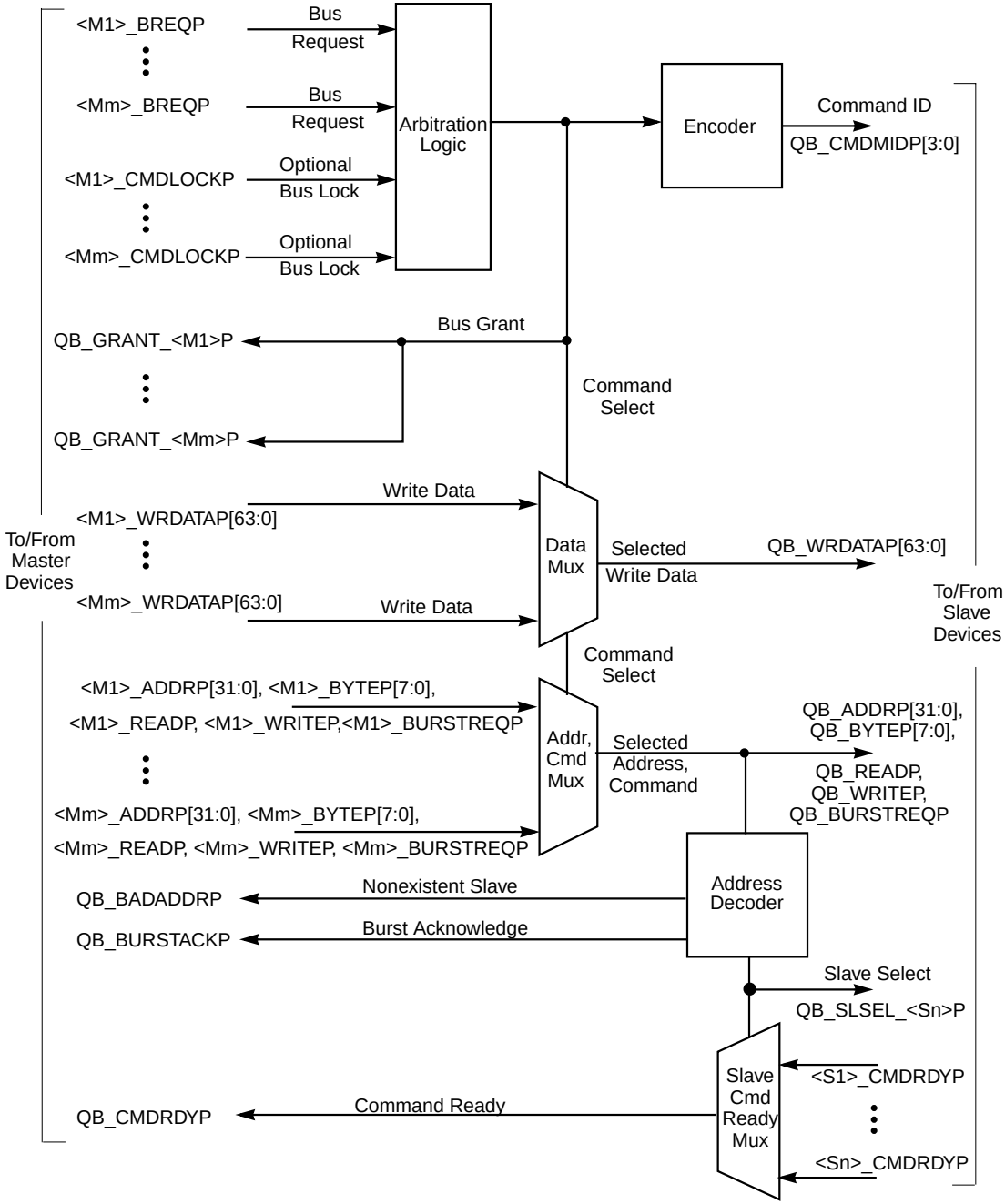
During a command cycle, a master bus device makes a read or write request and the slave acknowledges the request.

There are four steps in a command cycle:

1. Arbitrating for the Command Bus
2. Selecting the Command Source
3. Decoding the Address
4. Selecting the Command Ready from a Slave

Figure 3.3 shows typical Quick Bus Command logic. The text that follows Figure 3.3 explains what the command logic does during a command cycle. Figure 3.4 provides a diagram of the Command timing.

Figure 3.3 Typical Quick Bus Command Logic



3.6.1.1 Arbitrating for the Command Bus

During command arbitration, the Quick Bus arbitrates one or more requests from bus master devices. A master that wins the arbitration receives a bus grant signal from the Quick Bus arbitration logic. If a bus master asserts the optional command lock signal (<Mm>_CMDLOCKP), the lock signal overrides the arbitration logic and forces a bus grant to that master. Bus masters are not permitted to use the command lock unless they won the bus grant on the previous bus cycle. Once a master wins the bus grant, it can use the command lock with each subsequent bus cycle indefinitely.

3.6.1.2 Selecting a Command Source

Each bus master has a unique bus grant signal associated with it. When a particular bus master (Bus Master m) wins a bus grant, its bus grant signal is also used as a mux select signal for a Quick Bus multiplexer. The mux select signal enables the command information from the winning bus master to pass through the Quick Bus multiplexer. The bus grant multiplexing function steers the following type of information through the mux:

- address
- burst request
- byte enables
- read request
- write request
- write data

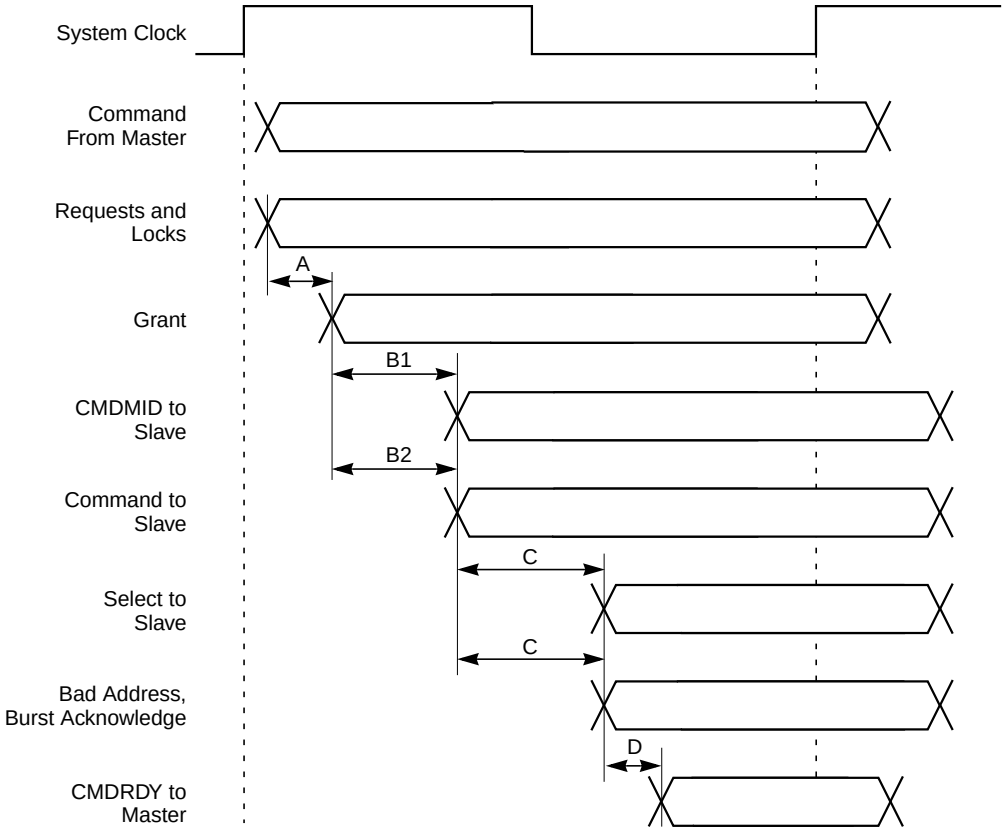
3.6.1.3 Decoding the Address

The Quick Bus address decode logic receives the data address from the winning bus master, decodes the address, and generates a select signal for the appropriate slave device. If the address is for a nonexistent slave device, the Quick Bus decode logic issues a Bad Address (QB_BADADDRP) signal and sends it to the bus master. The address decode logic also generates the Burst Acknowledge (QB_BURSTACKP) for burst-capable slaves.

3.6.1.4 Selecting Command Ready from Slave

The slave select signals (generated by the Quick Bus address decoder) also control the Slave Command Ready Mux. The mux selects the command ready (<Sn>_CMDRDYP) inputs from one of the slave devices. A slave select signal steers the command ready signal from the selected slave device through the mux. The mux outputs the selected Command Ready (QB_CMDRDYP) signal to the bus master. If QB_CMDRDYP is HIGH, it means the slave device accepts the command. If the QB_CMDRDYP is LOW, it indicates that the slave device is busy. If the slave is busy, the command request is ignored.

Figure 3.4 Quick Bus Command Timing



A = Arbitration Delay

B1 = Encoder Delay

B2 = Mux Delay

C = Address Decode Delay

D = Mux Delay

Command = address, burst request, byte, read, write, write data

3.6.2 Read Return Cycle

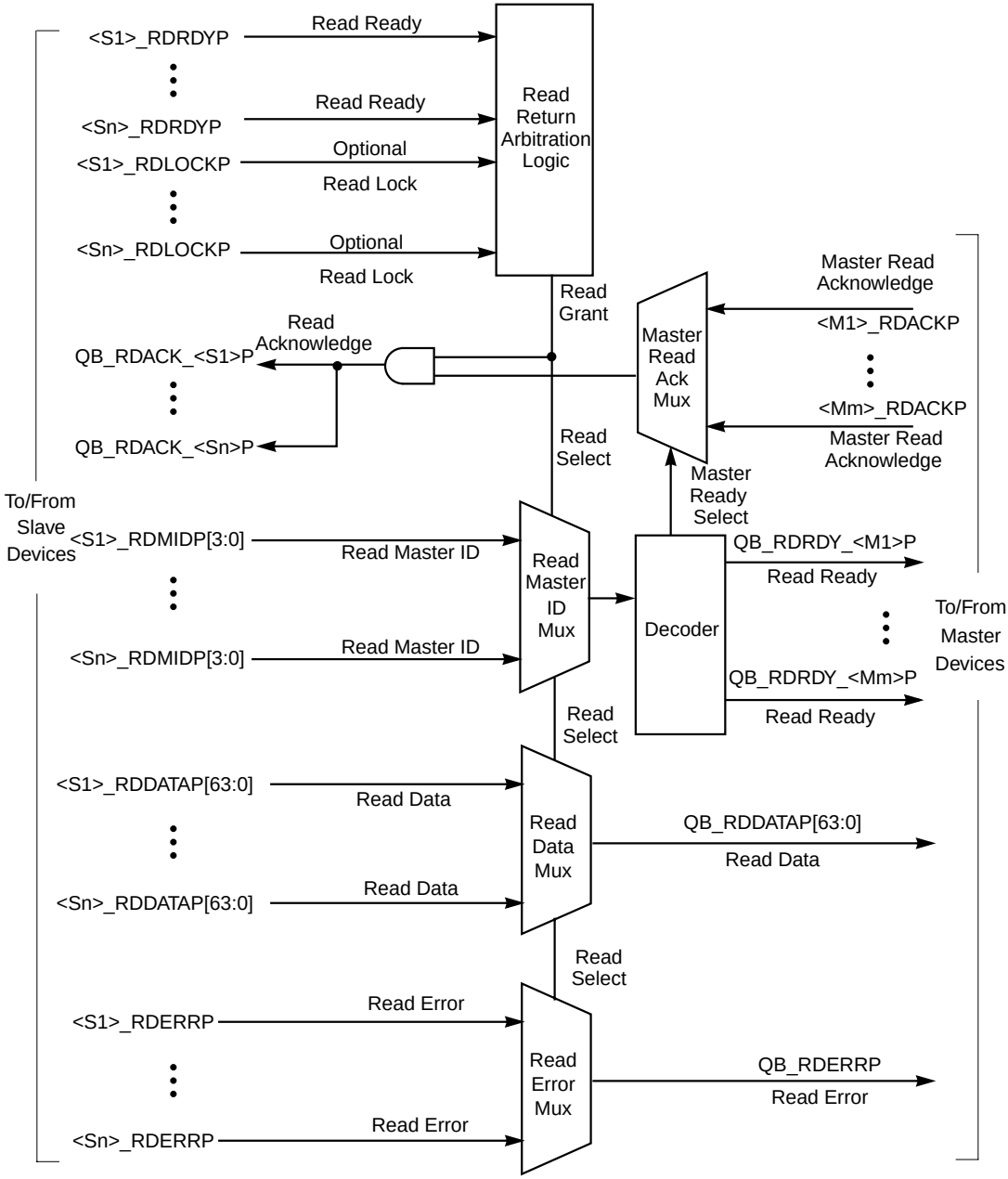
After a slave device obtains the requested data and is ready to send it to the master device, the slave initiates a Read Return Cycle. To do this, the slave sends a read ready signal to the Quick Bus along with the 64 data bits, the master ID, and the read error signal (<Sn>_RDRDYP, <Sn>_RDDATAP[63:0], <Sn>_RDMIDP[3:0], and <Sn>_RDERRP).

There are three steps in a Read Return Cycle:

1. Arbitrating for the Read Return Bus
2. Selecting Read Data
3. Acknowledging a Slave Read

Figure 3.5 shows typical Quick Bus Read Return logic, and Figure 3.6 provides a diagram of the Read Return timing.

Figure 3.5 Typical Quick Bus Read Return Logic



3.6.2.1 Arbitrating for the Read Return Bus

To determine which slave's read return data to select, the Quick Bus arbitrates the <Sn>_RDRDYP signals, which are sent from slave devices. Typically, the arbitration logic uses a simple, fixed-priority encoder. The slave that wins the arbitration receives a read grant, which is incorporated in the QB_RDACK_<Sn>P signal.

If a slave asserts the optional read lock signal (<Sn>_RDLOCKP), the lock signal overrides the arbitration logic and forces a bus grant to that slave. Slaves are not permitted to use the read lock option unless they won the read grant on the previous bus cycle. Once a slave wins the read grant, it can use the read lock with each subsequent bus cycle indefinitely.

3.6.2.2 Selecting Read Data

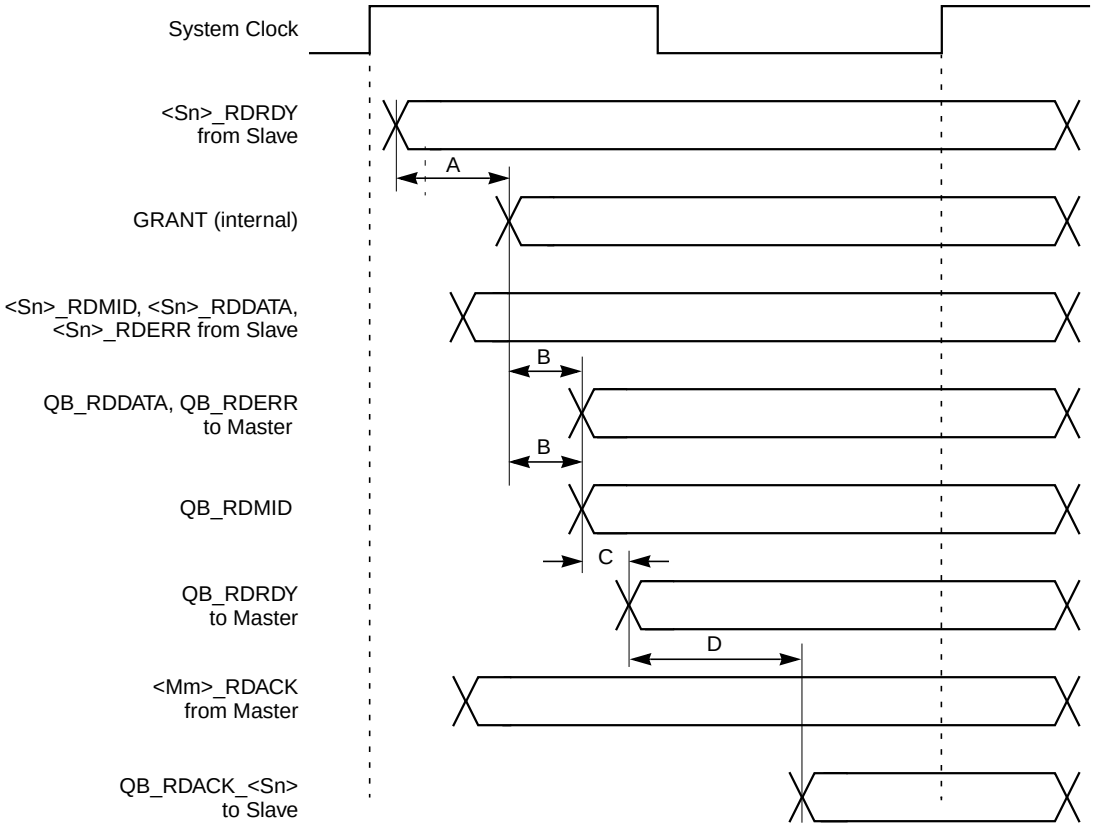
The Read Return Arbitration logic generates the read grant. The Read Grant signal determines which inputs are steered through the read muxes (master ID, read data, and read error muxes).

The single Master ID signal output from the Read Master ID Mux serves two purposes. One, it is decoded to generate a Read Ready (QB_RDRDY_<Mm>P) signal that is forwarded to the master along with the 64 bits of returning read data, QB_RDDATAP[63:0]. Second, the Master ID signal controls the Master Read Acknowledge Mux to select the appropriate incoming Master Read Acknowledge signal (<Mm>_RDACKP).

3.6.2.3 Acknowledging a Slave Read Return

The Quick Bus ANDs the master read acknowledge signal with the Read Grant signal from the arbitration logic to produce a Read Acknowledge (QB_RDACK_<Sn>P). The QB_RDACK_<Sn>P signal is sent to the slave device that initiated the Read Return.

Figure 3.6 Quick Bus Read Return Timing



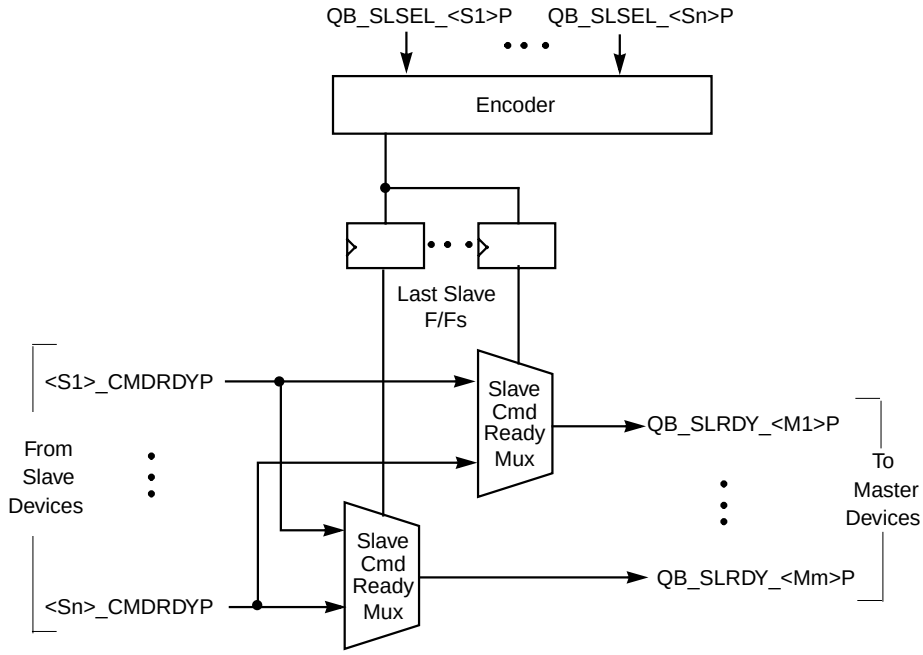
A = Arbitration Delay
B = Mux Delay
C = Decode Read Master ID
D = Mux Delay

3.6.3 Slave Ready Logic

The Slave Ready Logic (see Figure 3.7) allows a bus master to monitor a busy slave until the slave becomes ready. Monitoring the slave allows the master to avoid wasting time arbitrating repeatedly for access to the bus. One command cycle is required before the master can begin monitoring the slave. After a master wins a bus grant, the Quick Bus decodes the slave address and sends the command to the selected slave. If the slave is busy, as indicated by <Sn>_CMDRDYP being LOW, the Quick Bus forwards this information to the master by driving QB_CMDRDYP LOW.

When a master receives the slave busy response, it can handle it in a variety of ways. One method is to set a busy flip-flop and then monitor the QB_SLRDY_<Mm>P signal. The Quick Bus Slave Ready logic (see Figure 3.7) drives QB_SLRDY_<Mm>P LOW as long as the slave is busy. There is one QB_SLRDY_<Mm>P for each master, and the signal indicates the status of the last slave the master communicated with.

Figure 3.7 Slave Ready Logic



3.6.4 Burst Transactions

A standard burst reads or writes four 64-bit words. To initiate a burst transaction, the master should send the burst request and a word address along with the read/write request. Read and write burst transactions differ because reads are split transactions. A read burst request initiates four read returns. A write burst request is followed by three normal write commands.

The word order in burst transactions is the requested word first with wraparound. Thus, words are transferred in one of four orders: 0, 1, 2, 3; 1, 2, 3, 0; 2, 3, 0, 1; or 3, 0, 1, 2.

3.6.5 Bus Locking

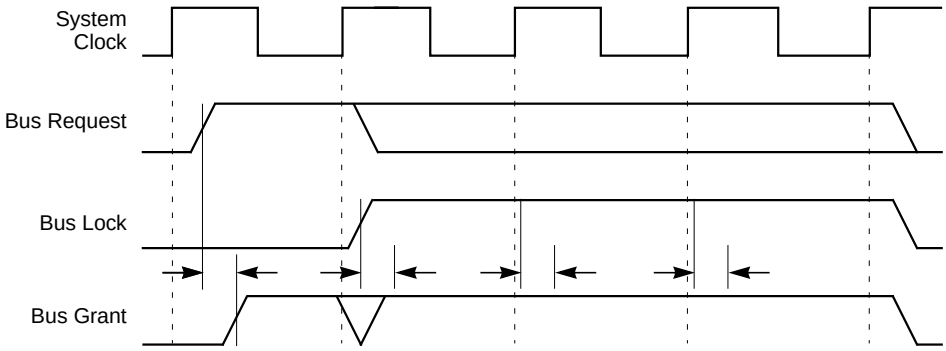
A lock signal overrides the arbitration logic and forces a bus grant (see Figure 3.8) over any other device currently requesting the bus. For that reason, the lock signal should not be used unless the bus was previously granted through normal arbitration. After bus access is granted through normal arbitration, a bus device can assert its lock signal for consecutive bus cycles. A bus master can use the <M>_CMDLOCKP signal to lock the command bus, and a slave can use the <S>_RDLOCKP signal to lock the read return bus.

Command bus locking is primarily used to prevent rearbitration in the middle of a burst transaction.

Command bus locking can also be used to do atomic read-modify-write transactions. For example, in a read-modify-write transaction, the bus is typically locked immediately after the read command occurs and remains locked (but idle) until the modified read data is written back. This method should be avoided if the length of the read latencies causes excessively long bus locks.

Read return bus locking typically is not used. It can be used if a slave needs to prevent rearbitration in the middle of a burst transaction.

Figure 3.8 Four-Word Burst Lock Example



3.6.6 Bus Snooping

The Quick Bus Snoop Interface provides signals that the master can use to monitor write commands sent to slave devices. If a write to a slave device occurs and the address matches data held in a master's internal cache, the master can invalidate the cache data for that address. Typically, a master ignores the following:

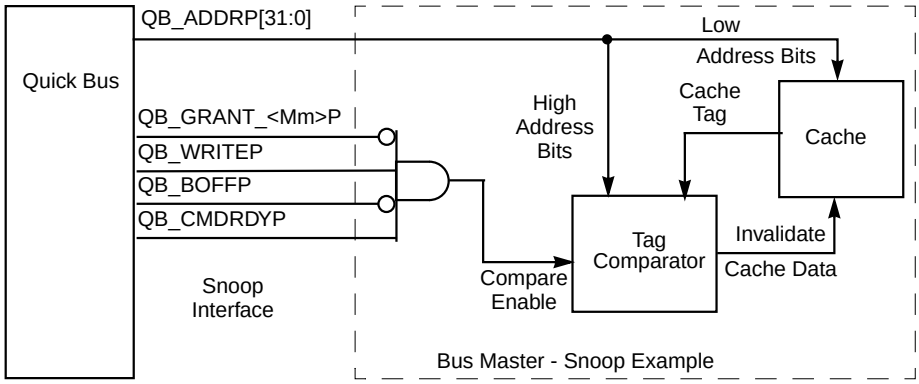
- the master's own write commands (indicated when its own Bus Grant signal is asserted),
- any write when back-off is asserted (QB_BOFFP HIGH)
- any write whose associated command ready signal is not asserted (QB_CMDRDYP LOW)

The Snoop Interface signals are a subset of the command signals sent to masters and slaves during a command bus cycle. The signals shown in the list below are used for snooping in addition to their normal function. Note that the QB_ADDR and QB_WRITE signals go to the master for snooping only.

QB_ADDRP[31:0]	Address	QB->M
QB_BOFFP	Back Off	QB->M
QB_CMDRDYP	Command Ready	QB->M
QB_GRANT_<Mm>P	Bus Grant	QB->M
QB_WRITEP	Write Request	QB->M

Figure 3.9 shows the Quick Bus Snoop interface signals, and it includes an example of how a bus master might use the signals. The low address bits access the Cache RAM. Then the high address bits are compared to the Cache Tag bits. If there is a match, the cache data for that address is invalidated.

Figure 3.9 Snoop Interface



3.6.6.1 Snooping in 1:1 Mode

When the system clock (SCLKP) is running at the same speed as the processor clock (PCLKP), the second of two consecutive writes may not snoop properly. To properly snoop the second write, the Quick Bus generates a back off signal to allow the write to be backed off to a third cycle.

This is only necessary if the following conditions are met:

- SCLKP_DIVP[1:0] = 0b01

  The EZ4021-FC must run in 1:1 mode (PCLKP = SCLKP).
- SNOOP_ENABLEP must be asserted

  Data cache invalidate Quick Bus snooping must be enabled.
- Last write must be from a non-EZ4021-FC master and be valid.

  The signals on the last cycle were:

  – QB_WRITEP = HIGH

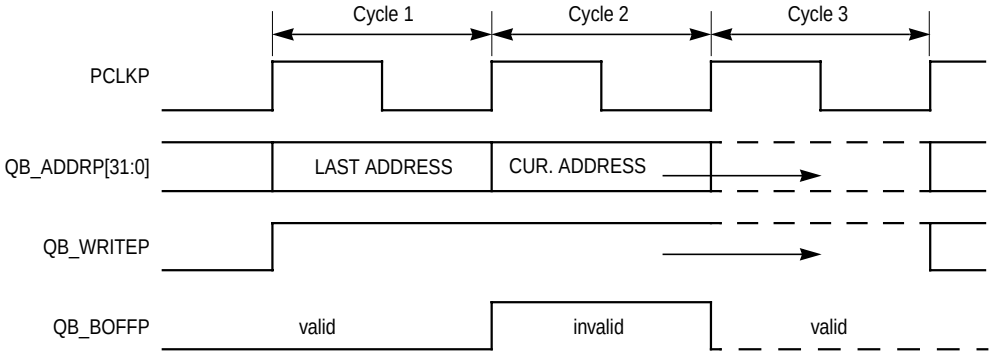
  – QB_CMDRDYP = HIGH

  – QB_BADADDRP = LOW

  – QB_BOFFP = LOW
- The current request must be a write from a non-EZ4021-FC master to a different address than the previous request.

Figure 3.10 shows the waveforms for the 1:1 snooping condition.

Figure 3.10 Snooping Back-to-Back Writes in 1:1 Mode



Last Address and Current Address are distinct

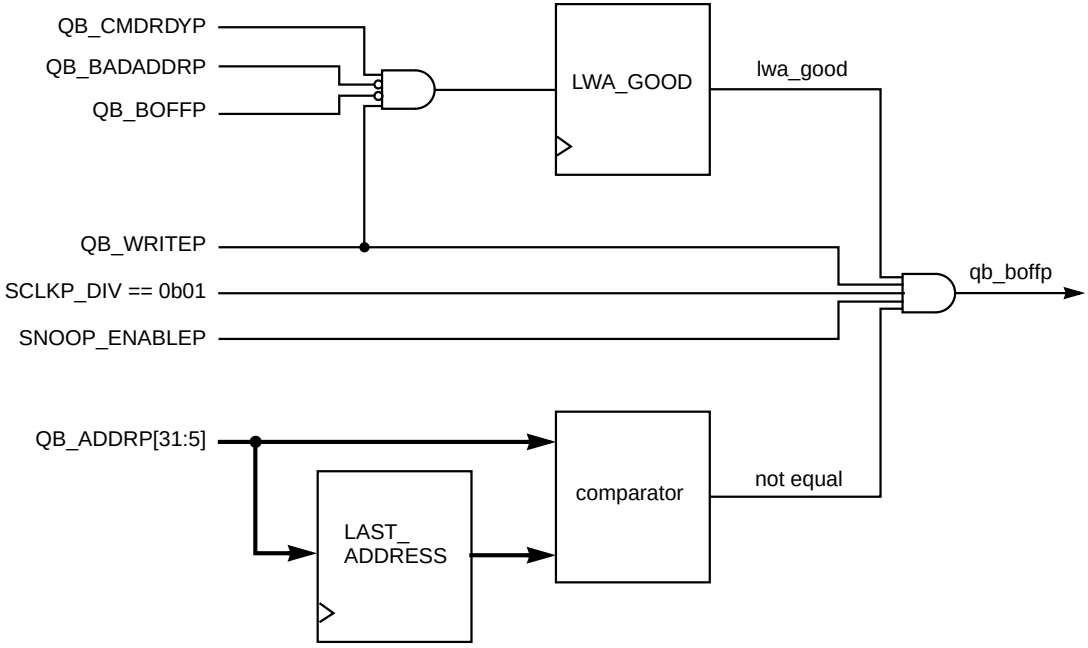
As shown in Figure 3.10, there are two consecutive writes, neither of which is initiated by the EZ4021-FC. The addresses of the two writes are assumed to be different.

To guarantee that the second write (Cycle 2) is snooped correctly, the write of Cycle 2 must be repeated in a subsequent cycle (for example, Cycle 3).

In burst writes, the EZ4021-FC must only snoop the first write cycle to verify a cache line hit. It is not required to snoop the subsequent three write cycles.

If the above conditions are met, the Quick Bus generates a back off signal for the Quick Bus masters and slaves. Figure 3.11 shows the logic for generating the backoff signal.

Figure 3.11 Logic for Generating Backoff



The LWA_GOOD register stores the information that the previous cycle was a valid write cycle. The LAST_ADDRESS register stores the upper address bits of the previous Quick Bus write cycle.

The registers are active at the rising edge of the clock.

Chapter 4
External Bus Controller

This chapter explains the use of the EZ4021-FC External Bus Controller (EBC). The EBC interfaces the EZ4021-FC to an off-chip local bus (LBus) that contains system peripherals.

This chapter includes the following sections:

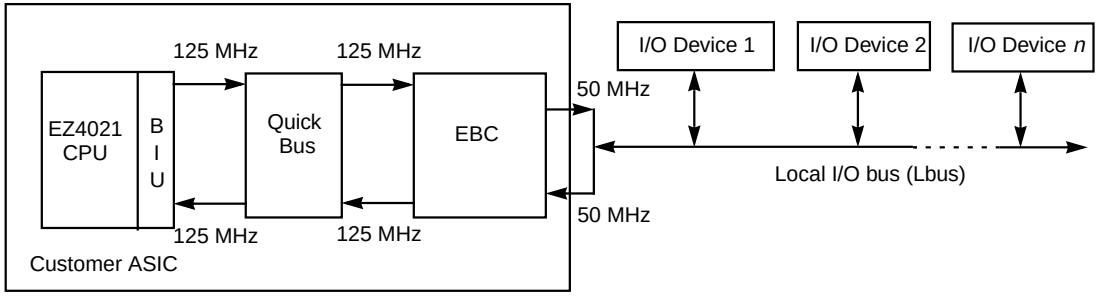
4.1 Overview

The EZ4021-FC External Bus Controller (EBC) interfaces the EZ4021-FC on-chip Quick Bus to an off-chip Local Bus (LBus). The EBC functions as either a master or slave on both the Quick Bus and the LBus. The maximum EBC clock frequency is 125 MHz.

The EBC requires an external I/O controller on the Lbus. The external I/O controller decodes addresses from the EBC and asserts the appropriate chip select.

Figure 4.1 shows a block diagram of the EBC in a typical EZ4021-FC design.

Figure 4.1 External Local Bus Controller (EBC) Block Diagram



The EBC offers the following features:

- Clock synchronization

  The EBC synchronizes requests between the Quick Bus, which operates at up to 125 MHz, and the LBus, which operates at a maximum frequency of 50 MHz (assuming LCLK is a multiple of SCLK).

- 64-bit to 32-bit conversion

  The Quick Bus supports a 64-bit data path between the EZ4021-FC CPU and the EBC, but the Lbus supports only a 32-bit data path between the EBC and external Lbus devices. The EBC handles the task of converting between the two buses.

- Split transaction optimization

  The EZ4021-FC's Quick Bus supports split transactions (requests are decoupled from returns) to prevent low-latency devices from tying up the Quick Bus. The EBC registers data, address, byte enables, and read or write signals for requests to the LBus to allow the Quick Bus to continue serving other devices in the system.

- Watchdog timer

  If the EZ4021-FC attempts to access an invalid address on the local bus, no device responds, causing the bus to hang. The EBC uses a programmable watchdog timer to recover from this condition. If the timer expires before the target device on the Lbus responds, the EBC generates a bus error (on read) or saves the address and generates an interrupt to the EZ4021-FC (upon write).

- Local Bus Retry

  The EBC registers the address, data, byte enables, and read or write signals of all requests. If an Lbus device cannot respond to an EBC request (for instance, the device may be waiting for data from other components in the system), it can request that the EBC reassert the request. The EBC requires the Lbus device to wait 128 Quick Bus cycles before asserting retry. This prevents the device from locking the LBus. This feature is only available when the EZ4021-FC is master on the Lbus.

4.2 Local Bus Overview

The Local Bus (Lbus) uses a 32-bit demultiplexed address bus and a 32-bit data bus. It shares many features with VLbus (or 486 bus) used by Intel 80486 microprocessors. Table 4.1 summarizes the LBus features.

Table 4.1 Lbus Features

Feature	Lbus
I/O space	No
Interrupt acknowledge cycle	No
Support for single transaction	Yes
Support for burst transaction	No
HOLD/HLDA bus arbitration	Yes
Bus retry input	Yes

4.3 EBC Signals

This section describes the External Bus Controller signals, which are listed in Table 4.2. Figure 4.2 shows the connections between the Quick Bus controller, the EBC, and the Lbus.

Figure 4.2 EBC Controller Connection Diagram

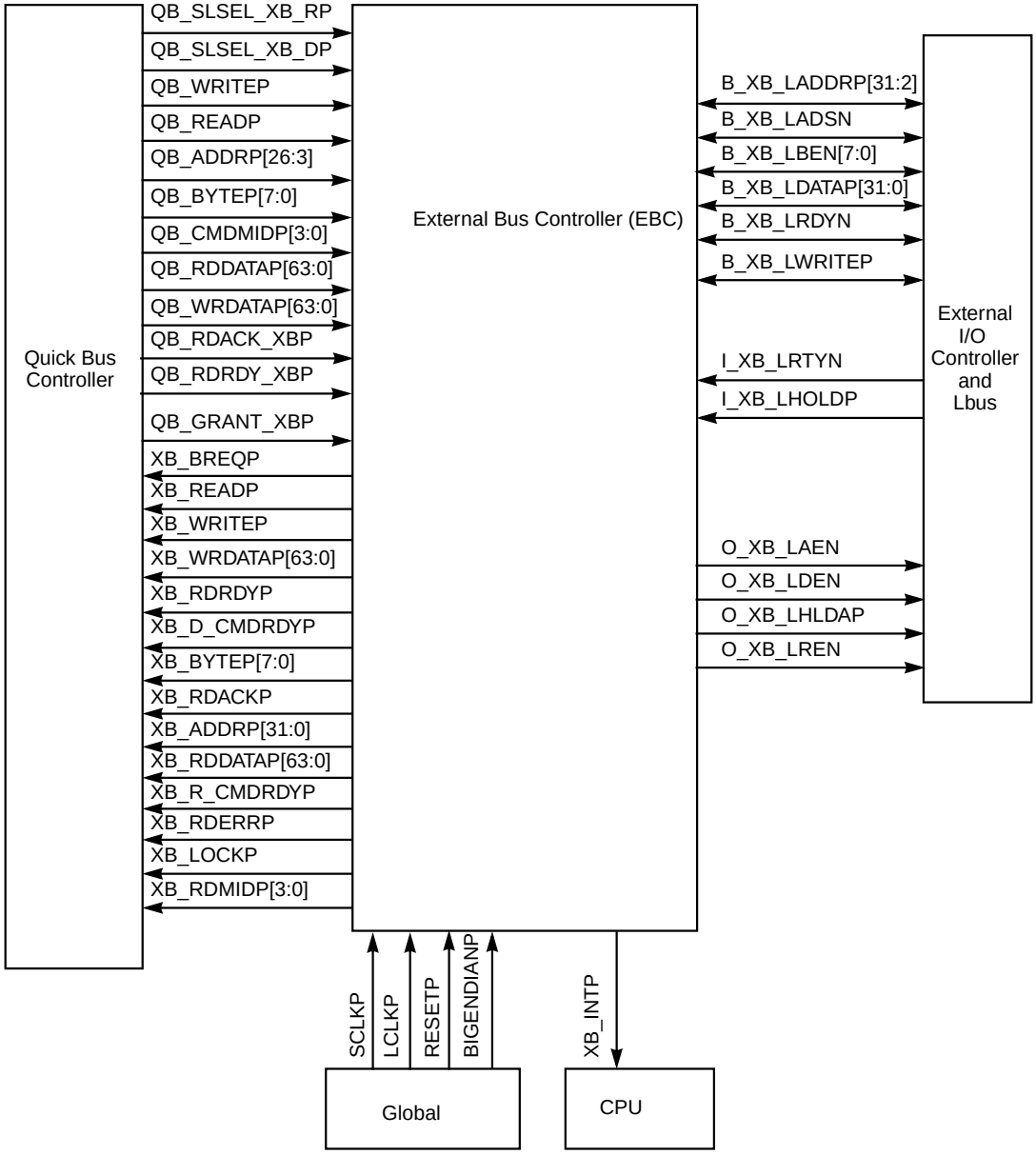


Table 4.2 summarizes the EBC signals. All signals are driven on the rising edge of the clock.

Table 4.2 External Bus Controller Alphabetical Signal List

Signal Name	I/O/B[1]	Source	Destination	Transaction	Description
B_XB_LADDRP[31:2]	B	Lbus EBC	EBC LBus	Lbus R/W Quick Bus R/W	Address to Quick Bus Address to Lbus
B_XB_LADSN	B	Lbus EBC	EBC Lbus	Lbus R/W Quick Bus R/W	Address Strobe Address Strobe
B_XB_LBEN[3:0]	B	Lbus EBC	EBC Lbus	Lbus W Quick Bus W	Byte Enable Byte Enable
B_XB_LDATAP[31:0]	B	Lbus EBC	EBC Lbus	Lbus R/W Quick Bus R/W	Data Data
B_XB_LRDYN	B	Lbus EBC	EBC LBus	Lbus R/W Quick Bus R/W	Request Ready Request Ready
B_XB_LWRITEP	B	Lbus EBC	EBC Lbus	Lbus R/W Quick Bus R/W	Write Enable (HIGH = Write, LOW = Read)
BIGENDIANP	I	Global	EBC	–	Endianness Select
I_XB_LHOLDP	I	Lbus	EBC	Lbus R/W	Lbus Bus Hold
I_XB_LRTYN	I	Lbus	EBC	Lbus R/W	Request Retry
LCLKP	I	EZ4021-FC	EBC	–	Lbus Clock
O_XB_LAEN	O	EBC	Lbus	Quick Bus R/W	Address Enable
O_XB_LDEN	O	EBC	Lbus	Quick Bus W Lbus R	Data Enable
O_XB_LHLDAP	O	EBC	Lbus	Lbus R/W	Hold Acknowledge
O_XB_LREN	O	EBC	Lbus	Lbus R/W	Ready Enable
QB_ADDRP[26:3]	I	Quick Bus	EBC	Quick Bus R/W	Address
QB_BYTEP[7:0]	I	Quick Bus	EBC	Quick Bus R/W	Byte Enable
QB_CMDMIDP[3:0]	I	Quick Bus	EBC	Quick Bus R	Command ID

(Sheet 1 of 3)

Table 4.2 External Bus Controller Alphabetical Signal List (Cont.)

Signal Name	I/O/B[1]	Source	Destination	Transaction	Description
QB_GRANT_XBP	I	Quick Bus	EBC	Lbus R/W	Quick Bus Grant
QB_RDACK_XBP	I	Quick Bus	EBC	Quick Bus R	Read Ready Acknowledge
QB_RDDATAP[63:0]	I	Quick Bus	EBC	Lbus R	Read Data
QB_RDRDY_XBP	I	Quick Bus	EBC	Lbus R	Read Data Ready
QB_READP	I	Quick Bus	EBC	Quick Bus R/W	Read Transaction
QB_SLSEL_XB_DP	I	Quick Bus	EBC	Quick Bus R/W	Lbus Data Select
QB_SLSEL_XB_RP	I	Quick Bus	EBC	Quick Bus R/W	EBC Module Register Select
QB_WRDATAP[63:0]	I	Quick Bus	EBC	Quick Bus W	Write Data
QB_WRITEP	I	Quick Bus	EBC	Quick Bus R/W	Write Transaction
RESETP	I	Global	EBC	–	System Reset
SCLKP	I	EZ4021-FC	All	–	Quick Bus Clock
XB_ADDRP[31:0]	O	EBC	Quick Bus	Lbus R/W	Address
XB_BREQP	O	EBC	Quick Bus	Lbus R/W	Quick Bus Ownership Request
XB_BYTEP[7:0]	O	EBC	Quick Bus	Lbus R/W	Byte Enable
XB_D_CMDRDYP	O	EBC	Quick Bus	Quick Bus R/W	Data Command Ready
XB_INTP	O	EBC	EZ4021-FC CPU	Quick Bus W	Watch Dog Timeout (Write Error Interrupt)
XB_R_CMDRDYP	O	EBC	Quick Bus	Quick Bus R/W	Register Command Ready
XB_RDACKP	O	EBC	Quick Bus	Lbus R	Read Acknowledge
XB_RDDATAP[63:0]	O	EBC	Quick Bus	Quick Bus R	Read Data
XB_RDERRP	O	EBC	Quick Bus	Quick Bus R	Watch Dog Timeout (Read Error)

(Sheet 2 of 3)

Table 4.2 External Bus Controller Alphabetical Signal List (Cont.)

Signal Name	I/O/B[1]	Source	Destination	Transaction	Description
XB_LOCKP[2]	O	EBC	Quick Bus	Lbus R/W	Lock Quick Bus
XB_RDMIDP[3:0]	O	EBC	Quick Bus	Quick Bus R	Read Master ID
XB_RDRDYP	O	EBC	Quick Bus	Quick Bus R	Read Ready
XB_READP	O	EBC	Quick Bus	Lbus R	Quick Bus Read Request
XB_WRDATAP[63:0]	O	EBC	Quick Bus	Lbus W	Write Data
XB_WRITEP	O	EBC	Quick Bus	Lbus W	Write Request
(Sheet 3 of 3)					

1. I = Input, O = Output, B = Bidirectional
2. This signal is tied LOW. The EBC cannot lock the Quick Bus.

4.3.1 EBC Signal Descriptions

This section lists detailed descriptions of all External Bus Controller signals. Direction of bidirectional signals is with respect to the EBC.

B_XB_LADDRP[31:2]
Address **Bidirectional**
This 30-bit bus carries addresses between the Lbus and the EBC. As an input, it carries an address from the Lbus to the Quick Bus. As an output, it carries an address from the Quick Bus to the Lbus. The O_XB_LAEN signal controls the direction of B_XB_LADDRP[31:2].

B_XB_LADSN Address Strobe **Bidirectional**
All Lbus transactions are initiated with this signal. The Lbus asserts this signal for one cycle to initiate a transaction to the Quick Bus. The EBC asserts this signal for one cycle to initiate a transaction on the Lbus. The O_XB_LAEN signal controls the direction of B_XB_LADSN.

B_XB_LBEN[3:0] Byte Enable **Bidirectional**
This 4-bit bus operates as a byte mask for Lbus transactions. The Lbus asserts this signal during transactions to the Quick Bus. The EBC asserts this signal during read and write transactions to the Lbus. The

O_XB_LAEN signal controls the direction of B_XB_LBEN. The following table shows the encoding for this bus:

Byte Enable	Valid Data Bits
B_XB_LBEN[3]	[24:31]
B_XB_LBEN[2]	[16:23]
B_XB_LBEN[1]	[8:15]
B_XB_LBEN[0]	[0:7]

B_XB_LDATAP[31:2]

Read/Write Data **Bidirectional**

This 30-bit bus contains read or write data from the Lbus. The O_XB_LDEN signal controls the direction of B_XB_LDATAP[31:2].

B_XB_LRDYN

Request Ready **Bidirectional**

An Lbus device asserts this signal to indicate it has completed a transaction. The EBC asserts this signal to indicate it has completed a transaction. On read transactions, read data is driven simultaneously with this signal. On write transactions, assertion of B_XB_LDRYN indicates that write data has been taken. The O_XB_LREN signal controls the direction of B_XB_LRDYN.

B_XB_LWRITEP

Write Enable **Bidirectional**

When an Lbus device is bus master on the Lbus, an Lbus device asserts this signal to indicate a write request to the Quick Bus. When deasserted, this signal indicates a read request to the Quick Bus. When the EZ4021-FC is the Lbus master, assertion of B_XB_LWRITEP indicates an Lbus write request. When deasserted, this signal indicates a read request to the Lbus. The O_XB_LAEN signal controls the direction of B_XB_LWRITEP.

BIGENDIANP **Endianness Select** **Input**

Assertion of this signal indicates big endian addressing. Deassertion of this signal indicates little endian addressing.

I_XB_LHOLDP

Lbus Hold **Input**

An Lbus device asserts this signal to hold the Lbus. The EBC grants this request with the O_XB_LHLDAP signal.

I_LRTYN **Retry Request** **Input**

An Lbus device asserts this signal to request a retry. The EBC waits until the retry counter decrements and then reissues the request. Refer to Section 4.4.1.1, “Transaction Retry,” page 4-15 for more information on the retry counter.

LCLKP **Lbus Clock** **Input**

This is the Lbus clock input from the EZ4021-FC clock generator to the EBC module. The maximum Lbus clock frequency is 50 MHz. All transactions happen on the rising clock edge.

O_XB_LAEN **Address Enable** **Output**

This signal controls the direction of the following signals: B_XB_LADDRP[31:2], B_XB_LADSN, B_XB_LBEN, and B_LWRITEP signals. When O_XB_LAEN is asserted, these signals operate as outputs. When O_XB_LAEN is deasserted, they operate as inputs.

O_XB_LDATAP[31:0]

Data **Output**

This 32-bit bus contains write data for the current transaction. The EBC drives this data to the Lbus.

O_XB_LDEN **Data Enable** **Output**

This signal controls the direction of the B_XB_LDATAP[31:2] bus. When O_XB_LDEN is asserted, the B_XB_LDATAP[31:2] bus operates as an output. When O_XB_LDEN is deasserted, the B_XB_LDATAP[31:2] bus operates as an input.

O_XB_LREN **Lbus Ready Enable** **Output**

This signal controls the directionality of the B_XB_LRDYN signal. When O_XB_LREN is asserted, B_XB_LRDYN operates as an output. When O_XB_LREN is deasserted, B_XB_LRDYN operates as an input.

QB_ADDRP[26:3]

Address **Input**

This 24-bit bus carries address information from the Quick Bus to the EBC.

QB_BYTEP[7:0]

Byte Enable **Input**

This 8-bit bus operates as a byte mask on data from the Quick Bus to the EBC. The following table shows the correspondence between the byte enable signals and the valid data bits:

Byte Enable	Valid Data Bits	Byte Enable	Valid Data Bits
QB_BYTEP[7]	[63:56]	QB_BYTEP[3]	[31:24]
QB_BYTEP[6]	[55:48]	QB_BYTEP[2]	[23:16]
QB_BYTEP[5]	[47:40]	QB_BYTEP[1]	[15:8]
QB_BYTEP[4]	[39:32]	QB_BYTEP[0]	[7:0]

QB_CMDIDP[3:0]

Command ID **Input**

This 4-bit bus carries the command ID for each request from the Quick Bus. On read returns, the EBC returns the command ID on the XB_RDMIDP[3:0] bus.

QB_GRANT_XBP

Quick Bus Grant **Input**

The Quick Bus asserts this signal to grant access to the EBC. The EBC requests Quick Bus access using the XB_BREQP signal.

QB_RDACK_XBP

Read Ready Acknowledge **Input**

The Quick Bus asserts this signal to indicate it has accepted the data the EBC placed on the Quick Bus.

QB_RDDATAP[63:0]

Read Data **Input**

This 64-bit bus carries read data from the Quick Bus to the EBC.

QB_RDRDY_XBP

Read Data Ready **Input**

The Quick Bus asserts this signal to indicate that valid read data is on the Quick Bus.

QB_READP **Read Transaction** **Input**
The Quick Bus asserts this signal to indicate that the current request is a read transaction.

QB_SLSEL_XB_DP
Lbus Data Select **Input**
The Quick Bus asserts this signal to request access to the Lbus device at the address currently on the QB_ADDRP[26:3] bus.

QB_SLSEL_XB_RP
EBC Register Select **Input**
The Quick Bus asserts this signal to access the internal EBC registers. The address of the desired register must be signaled on the QB_ADDRP[26:3] bus. Refer to Section 4.5, “EBC Registers,” page 4-16, for information on the EBC registers.

QB_WRDATAP[63:0]
Write Data **Input**
This 64-bit bus carries write data from the Quick Bus to the EBC.

QB_WRITEP **Write Transaction** **Input**
The Quick Bus asserts this signal to indicate that the current request is a write transaction.

RESETP **System Reset** **Input**
Master system reset input. The EBC is idle after reset.

SCLKP **System Clock** **Input**
Master system clock input. All transactions occur on the rising edge of the clock.

XB_ADDRP[26:3]
Address **Output**
This 24-bit bus carries address information from the EBC to the Quick Bus.

XB_BREQP **Quick Bus Ownership Request** **Output**
The EBC asserts this signal to request ownership of the Quick Bus. The Quick Bus responds by asserting QB_GRANT_XBP.

XB_BYTEP[7:0]

Byte Enable **Input**

This 8-bit bus operates as a byte mask on data from the EBC to the Quick Bus. The following table shows the correspondence between the byte enable signals and the valid data bits:

Byte Enable	Valid Data Bits	Byte Enable	Valid Data Bits
XB_BYTEP[7]	[63:56]	XB_BYTEP[3]	[31:24]
XB_BYTEP[6]	[55:48]	XB_BYTEP[2]	[23:16]
XB_BYTEP[5]	[47:40]	XB_BYTEP[1]	[15:8]
XB_BYTEP[4]	[39:32]	XB_BYTEP[0]	[7:0]

XB_D_CMDRDYP

Data Command Ready **Output**

The EBC asserts this signal to indicate it can accept data access (read/write) requests.

XB_INTP **Write Error (Watchdog Timer Timeout)** **Output**

The EBC asserts this signal to indicate a write error due to the expiration of the watchdog timer. On expiration of the watchdog timer, the EBC saves the failing address in the Watchdog Timer Failing Address Register and sets the ERR bit in the Watchdog Timer Error Register. Refer to Section 4.5, “EBC Registers,” page 4-16, for more information.

XB_RDACKP **Read Acknowledge** **Output**

The EBC asserts this signal to acknowledge it has taken the read data currently on the Lbus.

XB_RDDATAP[63:0]

Read Data **Output**

This 64-bit bus carries read data from the EBC to the Quick Bus.

XB_RDERRP **Read Error (Watchdog Timer Timeout)** **Output**

The EBC asserts this signal to indicate a read error due to the expiration of the watchdog timer. Refer to Section 4.5, “EBC Registers,” page 4-16, for more information.

XB_LOCKP **Quick Bus Lock Request** **Output**
This signal is tied LOW. The EBC cannot request a Quick Bus lock.

XB_RDMIDP[3:0]
Read Master ID **Output**
This 4-bit bus returns the command ID for each request from the Quick Bus. This command ID matches the one sent by the Quick Bus on the QB_CMDIDP[3:0] bus.

XB_RDRDYP **Read Ready** **Output**
The EBC asserts this signal to indicate valid read data is on the Quick Bus.

XB_READP **Quick Bus Read Request** **Output**
The EBC asserts this signal to indicate the current transaction is a read request.

XB_R_CMDRDYP
Register Command Ready **Output**
The EBC asserts this signal to indicate it can accept register access requests.

XB_WRDATAP[63:0]
Write Data **Output**
This 64-bit bus carries write data to the Quick Bus when an Lbus device is Lbus master.

XB_WRITEP **Write Request** **Output**
The EBC asserts this signal to indicate the current transaction is a write request when an Lbus device is Lbus master.

4.4 EBC Transactions

This section explains the interactions between the EBC and the Lbus. The EBC interfaces the Quick Bus to the Lbus. As such, each EBC transaction is made up of two parts: a transaction between the requesting bus and the EBC, and a transaction between the EBC and the target bus. For clarity, these transactions are divided into two types:

- Transactions where the EZ4021-FC is the Lbus Master
- Transactions where the EZ4021-FC is the Lbus Slave

Both transaction types support read and write operations.

In addition, the EBC handles conversion between 32- and 64-bit formats when moving data between the Quick Bus and the Lbus. Refer to Section 4.6, “Timing Waveforms,” page 4-17, to see how 32- and 64-bit requests are handled.

4.4.1 EZ4021-FC as Lbus Master

The EZ4021-FC is the default Lbus master device, and assumes ownership unless the I_XB_LHOLDP signal is asserted. As Lbus master, the EZ4021-FC drives address and data information to the EBC on the Quick Bus.

When the EZ4021-FC is the Lbus master (that is, when the Quick Bus is the requesting bus), the EBC registers the information associated with the request. This allows the Quick Bus to respond to other devices and prevents stalling while waiting for a high-latency device on the Lbus to respond. The EBC deasserts XB_R_CMDRDYP once it has accepted the request.

The EBC initiates an Lbus transaction by asserting the Lbus address strobe signal. Along with the address strobe, the EBC also drives the address, byte enable, and write enable signals. During a read transaction, the EBC drives address and control information and waits for the peripheral device to return data. During a write transaction, the EBC drives the write data at the same time as address and control information.

An Lbus transaction is completed when the Lbus slave drives the I_XB_LRDYN signal for one cycle. During a read transaction, the Lbus slave drives data and I_XB_LRDYN simultaneously to indicate that valid data is on the bus. During a write transaction the Lbus slave device asserts I_XB_LRDYN to indicate that it has accepted the data on the bus.

When the EBC samples I_XB_LRDYN active, it deasserts the address information associated with the request. This causes the external I/O controller to deassert the chip select signal to the Lbus slave device.

If the Lbus slave cannot complete the transaction, it can assert the I_XB_LRTYN signal, causing the EBC to abort the transaction and retry it at a later time. Refer to Section 4.4.1.1, “Transaction Retry,” page 4-15, for more information.

On read transactions, the EBC then asserts XB_RDRDYP on the Quick Bus and drives the return data on XB_RDDATAP[63:0].

4.4.1.1 Transaction Retry

In cases where the transaction must be terminated prior to completion of the data transfer, the Lbus slave can assert I_XB_LRTYN (retry request) instead of I_XB_LRDYN. Assertion of I_XB_LRTYN causes the master device to abort the transaction and retry it at a later time. This feature is used to avoid a dead lock condition. A transaction retry can only be performed when the EZ4021-FC is the Lbus master.

To issue the same request at a later time, the EBC saves the address, data, byte enable, and the read/write request—the EZ4021-FC does not reissue the request itself. The EBC contains a hard-wired timeout counter with a value of 0x7F (128 Quick Bus clock cycles). The counter begins decrementing once I_XB_LRTYN is asserted. When the counter reaches its terminal count of zero, the EBC regenerates the request. If the I_XB_LHOLDP signal is asserted when the counter reaches zero (an Lbus device is master on the Lbus), the pending request is serviced before the transaction is generated again.

Note: I_XB_LRDYN has higher priority than I_XB_LRTYN if both are asserted at the same time.

4.4.2 EZ4021-FC as Lbus Slave

Certain types of devices can function as Lbus masters. An Lbus device must gain ownership of the Lbus before initiating a transfer. The Lbus device asserts the Lbus hold signal (I_XB_LHOLDP) to request Lbus ownership. This signal is never 3-stated; it is always in either the asserted or deasserted state. After asserting the hold signal, the Lbus device waits for the EBC to grant Lbus ownership.

The EBC allows any outstanding transactions to complete before surrendering Lbus ownership. The EBC asserts O_XB_LHLDAP to acknowledge giving Lbus ownership to the Lbus device. While O_XB_LHLDAP is asserted, the EZ4021-FC cannot initiate an Lbus

transaction. The EBC asserts O_XB_LHLDAP continuously until the Lbus device deasserts the I_XB_LHOLDP signal.

Once granted the bus, the Lbus device drives the address strobe signal for one cycle, indicating to the EBC that a transaction is in progress. Along with the address strobe, the Lbus device also drives the address, byte enable, and write enable signals.

During a read operation, the Lbus device waits for the EBC to return data. The EBC asserts the O_XB_LRDYN signal at the same time it drives data onto the bus.

During a write operation, the Lbus device drives data at the same time as the address. The EBC asserts O_XB_LRDYN to indicate that it has accepted the incoming data from the Lbus master.

4.5 EBC Registers

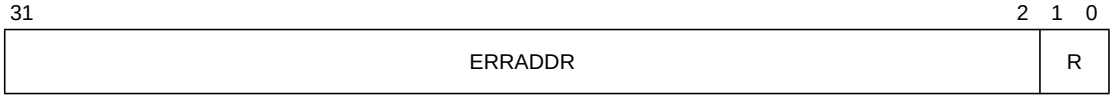
There are two EBC control registers for configuring and controlling the Lbus. The registers are located at the following virtual addresses:

- EBC Watchdog Timer Failing Address register (0xBEFF.FFD8)
- EBC Watchdog Timer Error register (0xBEFF.FFDC)

4.5.1 EBC Watchdog Timer Failing Address Register

This register records the failing address of an Lbus device that does not respond prior to expiration of the watchdog timer.

Figure 4.3 EBC Watchdog Timer Failing Address Register



ERRADDR **Watchdog Timer Error Address** **[31:2]**
This field contains the address of the Lbus target device that caused the watchdog timer to expire. When the Lbus target device does not respond to the transaction by asserting either I_XB_LRDYN or I_XB_LRTYN before the

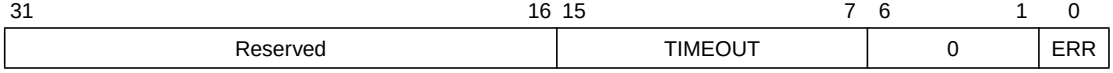
watchdog timer reaches zero, the EBC records the failing address and stores it in this field. This register is read only by the EZ4021-FC.

R **Reserved** **[1:0]**
This field must be set to zero.

4.5.2 EBC Watchdog Timer Error Register

This register contains the timeout value for the watchdog timer, and the error bit for watchdog timer expiration on writes.

Figure 4.4 Watchdog Timer Error Register



R **Reserved** **[31:16]**
This field must be set to zero.

TIMEOUT **Timeout Value** **[15:7]**
This 8-bit field specifies the upper half of the watchdog timeout value. The minimum timeout value is 0x100. The maximum timeout value is 0xFF00. Setting this field to 0x0 disables the watchdog timer.

0 **Preset Field** **[6:0]**
This field contains the lower half of the 16-bit timeout value. This field is preset to 0x0.

ERR **Watchdog Timer Error** **0**
This bit is set whenever the watchdog timer expires on a Quick Bus to Lbus write transaction. The EBC writes the failing address to the EBC Watchdog Timer Failing Address register. It is cleared by the EZ4021-FC.

4.6 Timing Waveforms

This section shows waveforms for EBC transactions. The examples show read and write operations when the EZ4021-FC is both a master and a slave device on the Lbus, and waveforms for transaction termination by retry and watchdog timer expiration.

Signal names for bidirectional signals such as B_XB_LADSN are shown explicitly as input or output signals. For example, B_XB_LADSN is shown as I_XB_LADSN when it operates as an input signal and O_XB_LADSN when it operates as an output signal.

Signals associated specifically with read or write transactions are annotated.

4.6.1 EZ4021-FC as Lbus Master

This section shows waveforms for read and write transactions when the EZ4021-FC is master on the Lbus. Transaction termination by retry request is also shown.

To allow communication between the 32-bit Lbus and the 64-bit Quick Bus, the EBC handles bit-length conversion. This action is transparent to the user and no special handling is required.

The EBC uses the byte enable signals to determine if a request is a 32- or 64-bit request. For 64-bit requests, the EBC issues two 32-bit requests to the Lbus and concatenates the results prior to signaling request ready on the Quick Bus.

4.6.1.1 32-Bit Requests

For a 32-bit read request, the EBC:

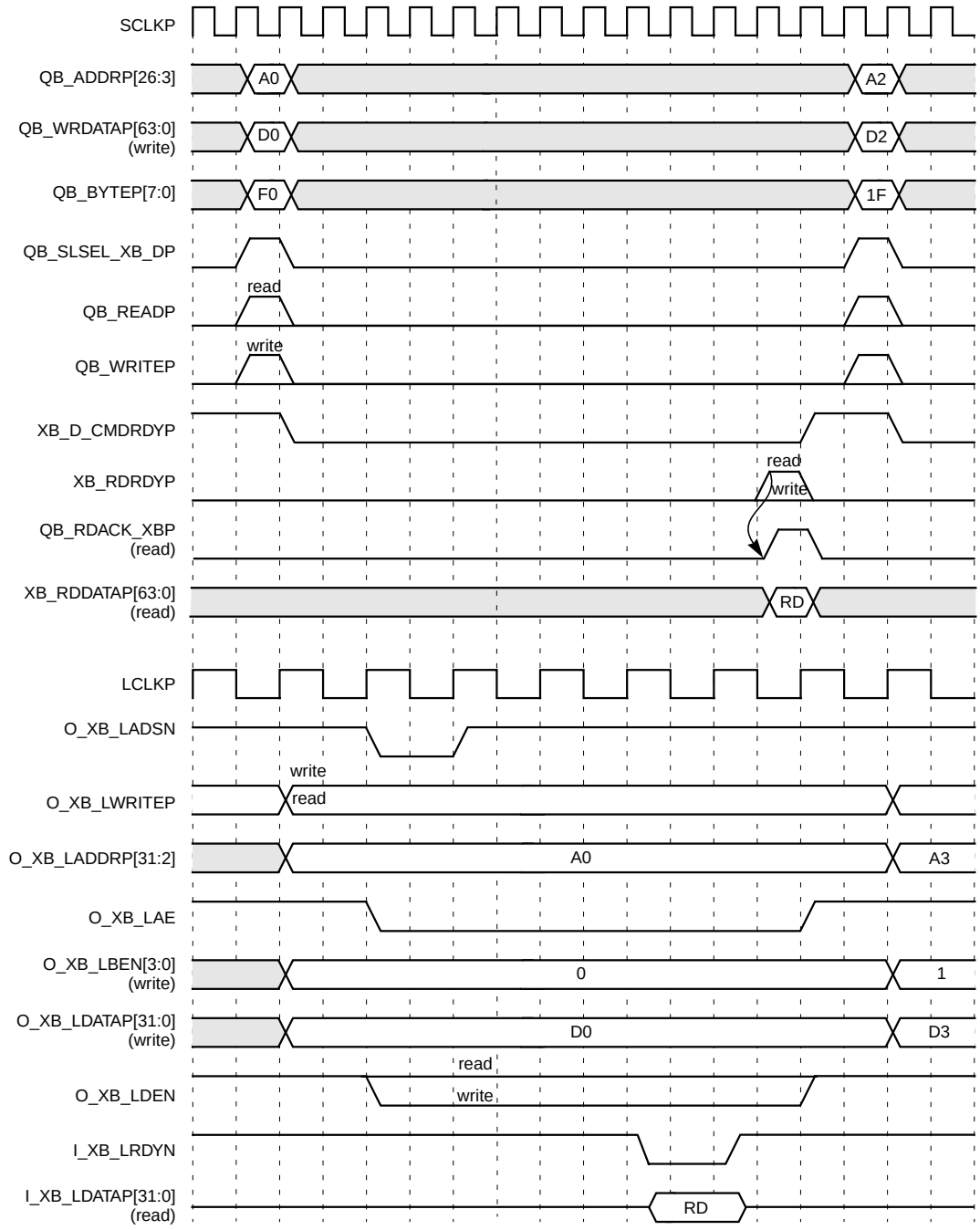
- Takes the read request from the Quick Bus, registers the associated information (address, data, byte enables, and read signal), and deasserts command ready (XB_D_CMDRDYP) on the Quick Bus.
- Issues a 32-bit request to the Lbus.
- Waits for the 32-bit read return from the Lbus (data accompanied by the B_XB_LRDYN signal).
- Asserts XB_RDRDYP and puts the read data on the Quick Bus.
- Returns to the idle state and reasserts XB_D_CMDRDYP on the Quick Bus.

For a 32-bit write request, the EBC:

- Takes the 32-bits of write data from the Quick Bus, registers the associated information (address, data, byte enables, and write signal) and deasserts command ready (XB_D_CMDRDYP) on the Quick Bus.
- Asserts the Lbus address strobe (B_XB_LADSN) and drives the 32-bit address and data to the Lbus slave device.
- Waits for a write acknowledge (B_XB_LRDYN) signal from the Lbus.
- Returns to the idle state and reasserts XB_D_CMDRDYP on the Quick Bus.

Figure 4.5 shows the waveforms associated with Lbus read and write transactions when the EZ4021-FC makes a 32-bit request.

Figure 4.5 Quick Bus Master Read/Write Timing Waveforms (32 Bit Access)



4.6.1.2 64-Bit Request

For a 64-bit read request, the EBC:

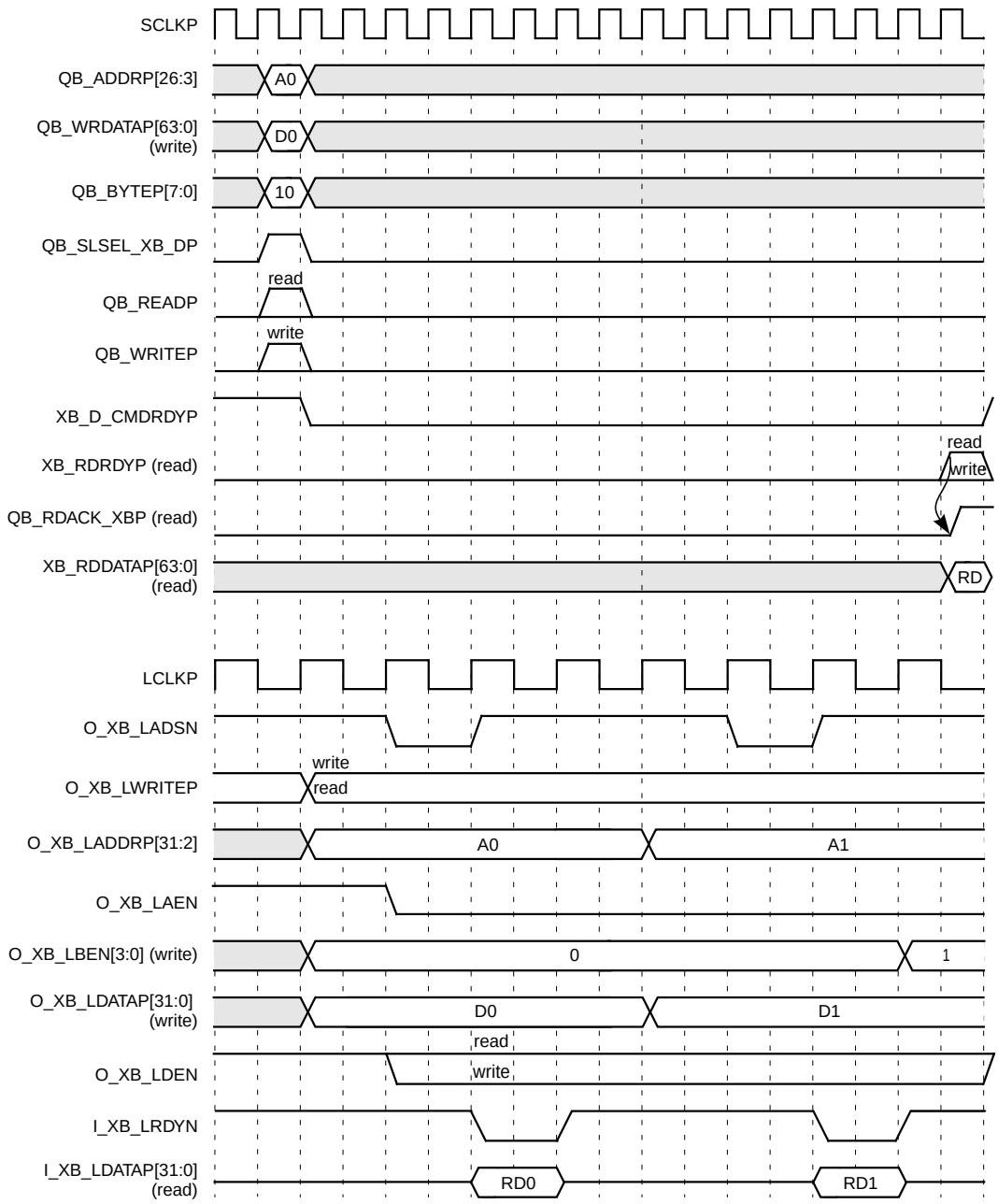
- Takes the read request from the Quick Bus, registers the associated information (address, data, byte enables, and read signal), and deasserts command ready (XB_D_CMDRDYP) on the Quick Bus.
- Issues a 32-bit request to the Lbus.
- Waits for the 32-bit read return from the Lbus (data accompanied by the B_XB_LRDYN signal).
- Registers the first 32-bits of read return data from the Lbus.
- Issues a second 32-bit request to the Lbus.
- Waits for the second 32-bit read return from the Lbus (data accompanied by the B_XB_LRDYN signal).
- Concatenates the second 32-bit read return with the first.
- Asserts XB_RDRDYP and puts the 64-bits of read data on the Quick Bus.
- Returns to the idle state and reasserts XB_D_CMDRDYP on the Quick Bus.

For a 64-bit write request, the EBC:

- Takes the 64-bits of write data from the Quick Bus, registers the associated information (address, data, byte enables, and write signal), and deasserts command ready (XB_D_CMDRDYP) on the Quick Bus.
- Asserts the Lbus address strobe (B_XB_LADSN) and drives the first 32-bit address and data to the Lbus slave device.
- Waits for a write acknowledge (B_XB_LRDYN) signal from the Lbus.
- Asserts the Lbus address strobe (B_XB_LADSN) and drives the second 32-bit address and data to the Lbus slave device.
- Waits for a write acknowledge (B_XB_LRDYN) signal from the Lbus.
- Returns to the idle state and reasserts XB_D_CMDRDYP on the Quick Bus.

Figure 4.6 shows the waveforms associated with Lbus read and write transactions when the EZ4021-FC makes a 64-bit request.

Figure 4.6 Quick Bus Master Read/Write Timing Waveforms (64 Bit Access)



4.6.1.3 Lbus Transaction Terminated by Retry

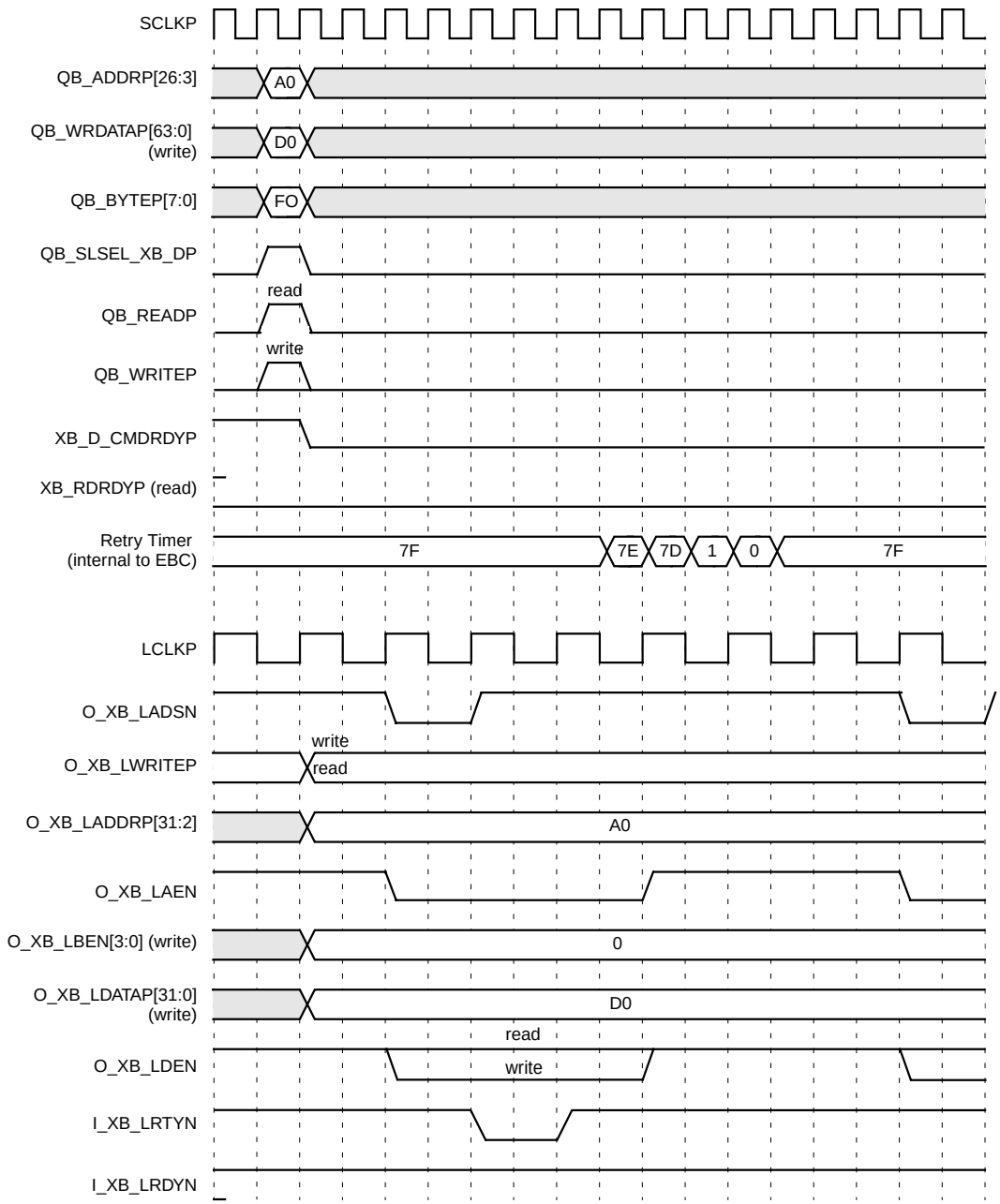
In cases where the transaction must be terminated prior to completion of the data transfer, the Lbus slave can assert I_XB_LRTYN instead of B_XB_LRDYN, causing the master device to abort the transaction and retry it at a later time. Retries are only available when the EZ4021-FC is the Lbus master. This feature avoids a dead lock condition.

The EBC generates the internal transaction termination at least one LCLKP cycle after the Lbus device asserts I_XB_LRTYN. If the I_XB_LHOLDP signal is asserted, the pending request is serviced before the transaction is generated again.

The EBC contains a hard-wired timeout counter with a value of 0x7F (128 Quick Bus clock cycles). The counter starts counting down when I_XB_LRTYN is asserted. When the counter reaches its terminal count of zero, the EBC regenerates the request.

Figure 4.7 shows the waveforms associated with a retry request.

Figure 4.7 Lbus Transaction Terminated by Retry Request



4.6.2 Lbus Device as Lbus Master

This section gives waveforms for read/write transactions when an Lbus device is master on the Lbus. All Lbus requests to the Quick Bus are 32-bits by definition.

Figure 4.8 shows the waveforms associated with Quick Bus read transactions when an Lbus device is master on the Lbus. Figure 4.9 shows the waveforms associated with Quick Bus write transactions when an Lbus device is master on the Lbus.

Figure 4.8 Lbus Master Read Timing Waveforms

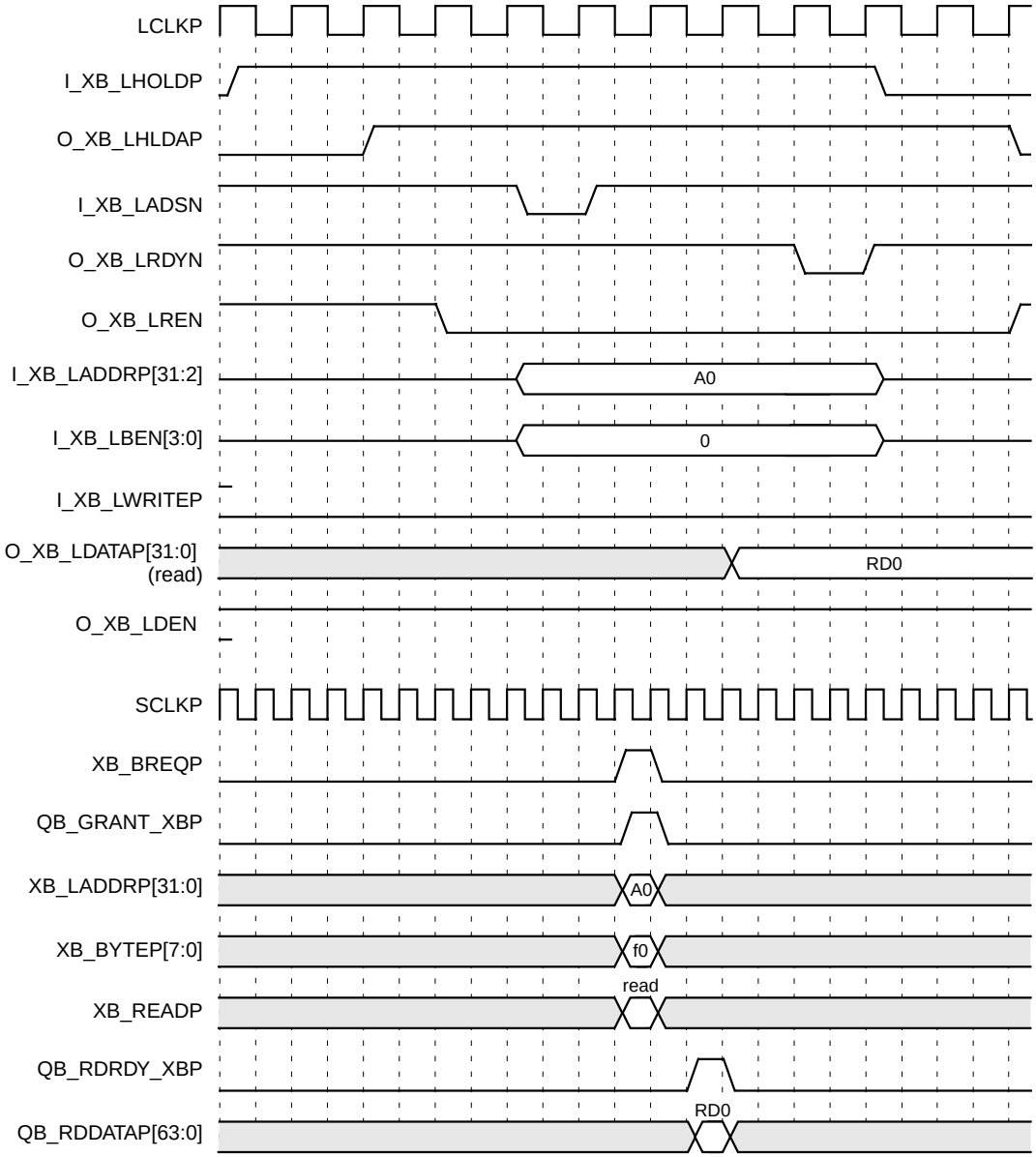
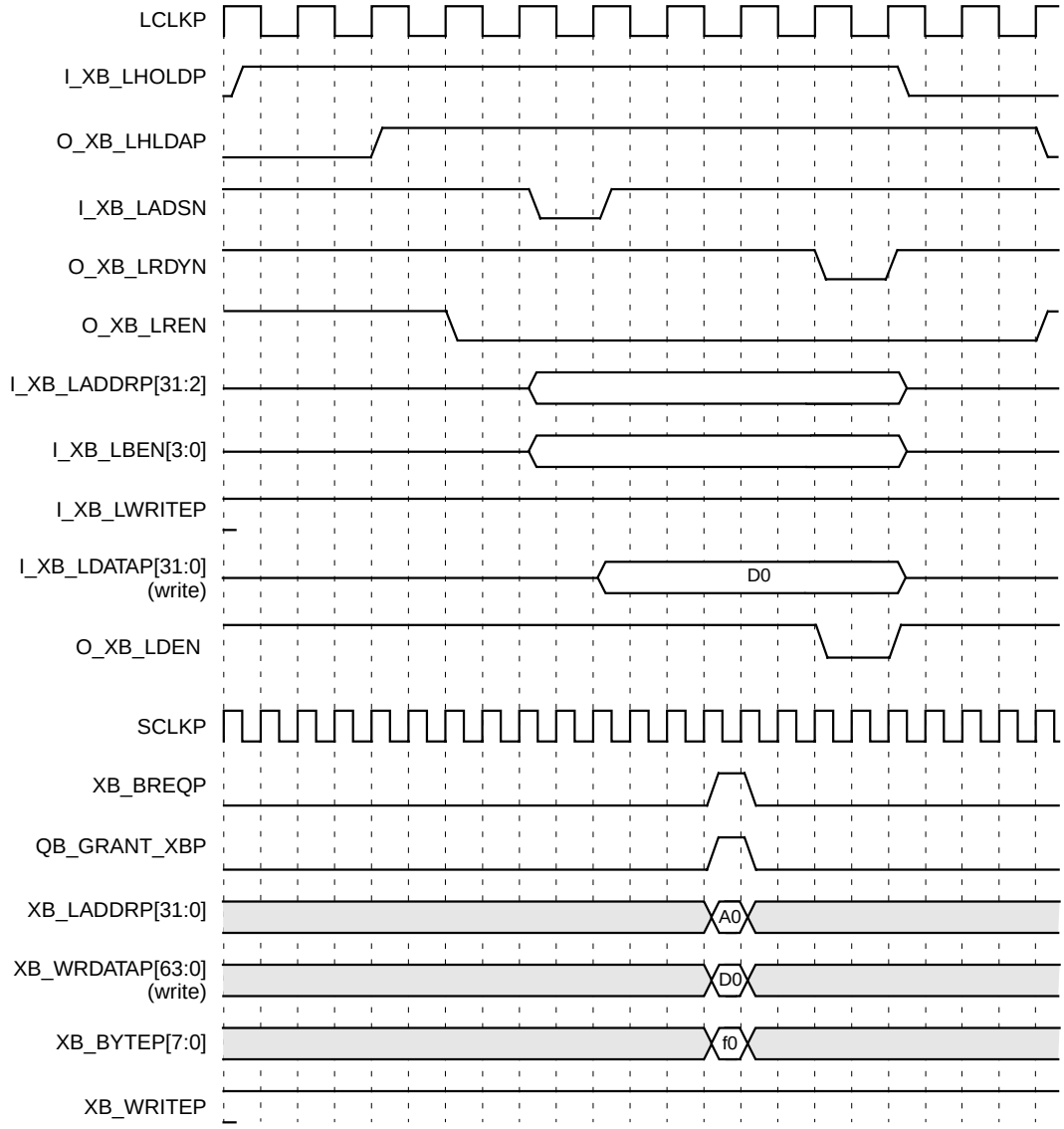


Figure 4.9 Lbus Master Write Timing Waveforms



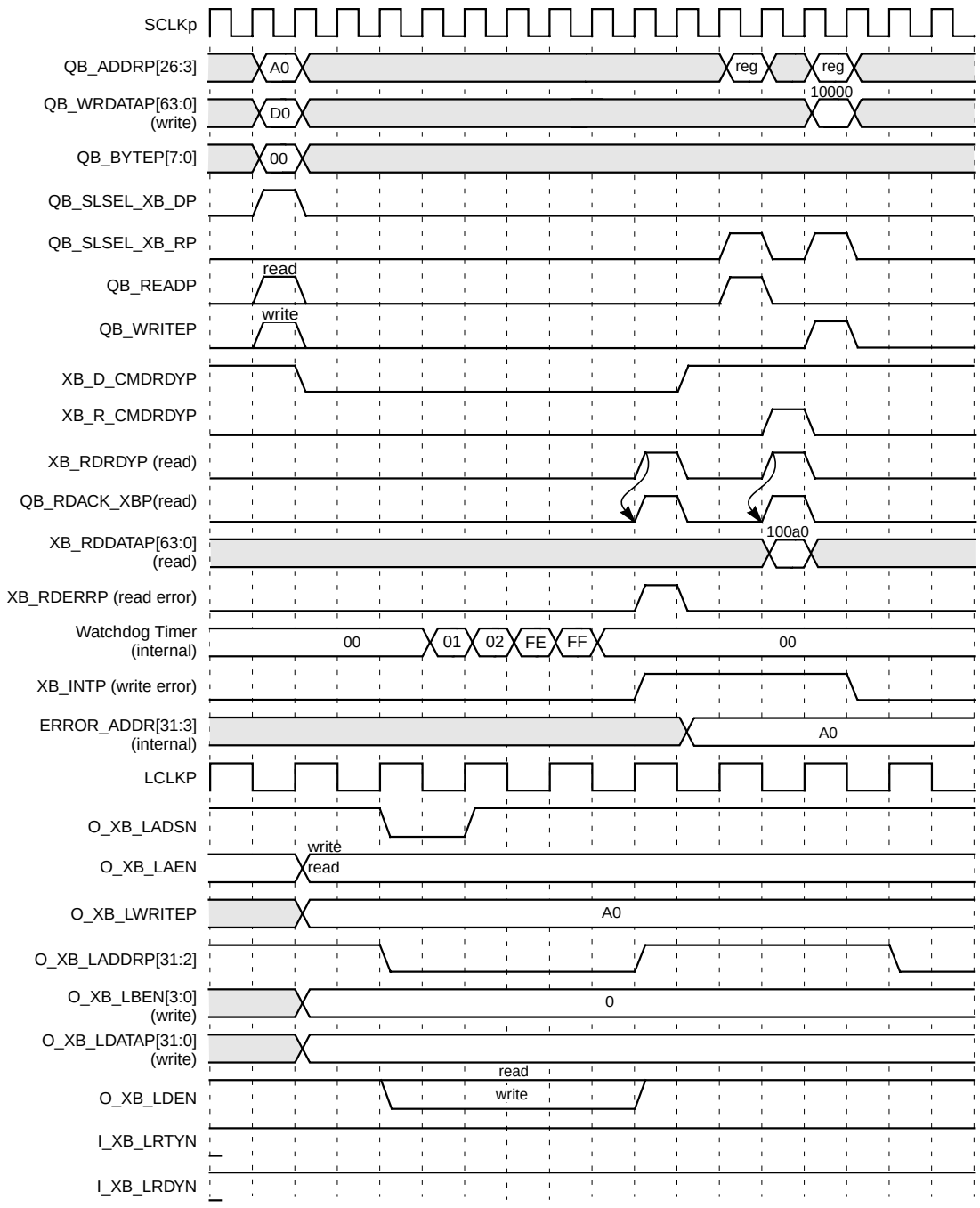
4.6.3 Lbus Transaction Termination by Watchdog Timer Expiration

The EBC provides a watchdog timer to eliminate the possibility of a bus hang condition if a slave device fails to respond. The 32-bit timer begins counting down from its programmed value whenever there is an Lbus

request from the Quick Bus. On a write transaction, if the counter counts down to zero before a slave device asserts either I_XB_LRDYN or I_XB_LRTYN, the EBC generates an interrupt to the EZ4021-FC and saves the failing address. On a read transaction, a local bus read error is generated. If the device responds by asserting either I_XB_LRDYN or I_XB_LRTYN, the counter is reset.

Figure 4.10 shows a timing diagram of an Lbus transaction terminated by expiration of the watchdog timer.

Figure 4.10 Lbus Transaction Terminated by Watchdog Timer Expiration



Customer Feedback

We would appreciate your feedback on this document. Please copy the following page, add your comments, and fax it to us at the number shown.

If appropriate, please also fax copies of any marked-up pages from this document.

Important: Please include your name, phone number, fax number, and company address so that we may contact you directly for clarification or additional information.

Thank you for your help in improving the quality of our documents.

Reader's Comments

Fax your comments to: LSI Logic Corporation
Technical Publications
M/S E-198
Fax: 408.433.4333

Please tell us how you rate this document: *MiniRISC EZ4021-FC Building Blocks Technical Manual.* Place a check mark in the appropriate blank for each category.

	Excellent	Good	Average	Fair	Poor
Completeness of information	____	____	____	____	____
Clarity of information	____	____	____	____	____
Ease of finding information	____	____	____	____	____
Technical content	____	____	____	____	____
Usefulness of examples and illustrations	____	____	____	____	____
Overall manual	____	____	____	____	____

What could we do to improve this document?

_____

_____

_____

_____

If you found errors in this document, please specify the error and page number. If appropriate, please fax a marked-up copy of the page(s).

_____

_____

_____

Please complete the information below so that we may contact you directly for clarification or additional information.

Name _____ Date _____

Telephone _____ Fax _____

Title _____

Department _____ Mail Stop _____

Company Name _____

Street _____

City, State, Zip _____

U.S. Distributors by State

A. E. Avnet Electronics
http://www.hh.avnet.com
B. M. Bell Microproducts, Inc. (for HAB's)
http://www.bellmicro.com
I. E. Insight Electronics
http://www.insight-electronics.com
W. E. Wyle Electronics
http://www.wyle.com

Alabama
Daphne
I. E. Tel: 334.626.6190
Huntsville
A. E. Tel: 256.837.8700
B. M. Tel: 256.705.3559
I. E. Tel: 256.830.1222
W. E. Tel: 800.964.9953

Alaska
A. E. Tel: 800.332.8638

Arizona
Phoenix
A. E. Tel: 480.736.7000
B. M. Tel: 602.267.9551
W. E. Tel: 800.528.4040
Tempe
I. E. Tel: 480.829.1800
Tucson
A. E. Tel: 520.742.0515

Arkansas
W. E. Tel: 972.235.9953

California
Agoura Hills
B. M. Tel: 818.865.0266
Granite Bay
B. M. Tel: 916.523.7047
Irvine
A. E. Tel: 949.789.4100
B. M. Tel: 949.470.2900
I. E. Tel: 949.727.3291
W. E. Tel: 800.626.9953
Los Angeles
A. E. Tel: 818.594.0404
W. E. Tel: 800.288.9953
Sacramento
A. E. Tel: 916.632.4500
W. E. Tel: 800.627.9953
San Diego
A. E. Tel: 858.385.7500
B. M. Tel: 858.597.3010
I. E. Tel: 800.677.6011
W. E. Tel: 800.829.9953
San Jose
A. E. Tel: 408.435.3500
B. M. Tel: 408.436.0881
I. E. Tel: 408.952.7000
Santa Clara
W. E. Tel: 800.866.9953
Woodland Hills
A. E. Tel: 818.594.0404
Westlake Village
I. E. Tel: 818.707.2101

Colorado
Denver
A. E. Tel: 303.790.1662
B. M. Tel: 303.846.3065
W. E. Tel: 800.933.9953
Englewood
I. E. Tel: 303.649.1800
Idaho Springs
B. M. Tel: 303.567.0703

Connecticut
Cheshire
A. E. Tel: 203.271.5700
I. E. Tel: 203.272.5843
Wallingford
W. E. Tel: 800.605.9953

Delaware
North/South
A. E. Tel: 800.526.4812
Tel: 800.638.5988
B. M. Tel: 302.328.8968
W. E. Tel: 856.439.9110

Florida
Altamonte Springs
B. M. Tel: 407.682.1199
I. E. Tel: 407.834.6310
Boca Raton
I. E. Tel: 561.997.2540
Bonita Springs
B. M. Tel: 941.498.6011
Clearwater
I. E. Tel: 727.524.8850
Fort Lauderdale
A. E. Tel: 954.484.5482
W. E. Tel: 800.568.9953
Miami
B. M. Tel: 305.477.6406
Orlando
A. E. Tel: 407.657.3300
W. E. Tel: 407.740.7450
Tampa
W. E. Tel: 800.395.9953
St. Petersburg
A. E. Tel: 727.507.5000

Georgia
Atlanta
A. E. Tel: 770.623.4400
B. M. Tel: 770.980.4922
W. E. Tel: 800.876.9953
Duluth
I. E. Tel: 678.584.0812

Hawaii
A. E. Tel: 800.851.2282

Idaho
A. E. Tel: 801.365.3800
W. E. Tel: 801.974.9953

Illinois
North/South
A. E. Tel: 847.797.7300
Tel: 314.291.5350
Chicago
B. M. Tel: 847.413.8530
W. E. Tel: 800.853.9953
Schaumburg
I. E. Tel: 847.885.9700

Indiana
Fort Wayne
I. E. Tel: 219.436.4250
W. E. Tel: 888.358.9953
Indianapolis
A. E. Tel: 317.575.3500

Iowa
W. E. Tel: 612.853.2280
Cedar Rapids
A. E. Tel: 319.393.0033

Kansas
W. E. Tel: 303.457.9953
Kansas City
A. E. Tel: 913.663.7900
Lenexa
I. E. Tel: 913.492.0408

Kentucky
W. E. Tel: 937.436.9953
Central/Northern/ Western
A. E. Tel: 800.984.9503
Tel: 800.767.0329
Tel: 800.829.0146

Louisiana
W. E. Tel: 713.854.9953
North/South
A. E. Tel: 800.231.0253
Tel: 800.231.5775

Maine
A. E. Tel: 800.272.9255
W. E. Tel: 781.271.9953

Maryland
Baltimore
A. E. Tel: 410.720.3400
W. E. Tel: 800.863.9953
Columbia
B. M. Tel: 800.673.7461
I. E. Tel: 410.381.3131

Massachusetts
Boston
A. E. Tel: 978.532.9808
W. E. Tel: 800.444.9953
Burlington
I. E. Tel: 781.270.9400
Marlborough
B. M. Tel: 800.673.7459
Woburn
B. M. Tel: 800.552.4305

Michigan
Brighton
I. E. Tel: 810.229.7710
Detroit
A. E. Tel: 734.416.5800
W. E. Tel: 888.318.9953
Clarkston
B. M. Tel: 877.922.9363

Minnesota
Champlin
B. M. Tel: 800.557.2566
Eden Prairie
B. M. Tel: 800.255.1469
Minneapolis
A. E. Tel: 612.346.3000
W. E. Tel: 800.860.9953
St. Louis Park
I. E. Tel: 612.525.9999

Mississippi
A. E. Tel: 800.633.2918
W. E. Tel: 256.830.1119

Missouri
W. E. Tel: 630.620.0969
St. Louis
A. E. Tel: 314.291.5350
I. E. Tel: 314.872.2182

Montana
A. E. Tel: 800.526.1741
W. E. Tel: 801.974.9953

Nebraska
A. E. Tel: 800.332.4375
W. E. Tel: 303.457.9953

Nevada
Las Vegas
A. E. Tel: 800.528.8471
W. E. Tel: 702.765.7117

New Hampshire
A. E. Tel: 800.272.9255
W. E. Tel: 781.271.9953

New Jersey
North/South
A. E. Tel: 201.515.1641
Tel: 609.222.6400
Mt. Laurel
I. E. Tel: 856.222.9566
Pine Brook
B. M. Tel: 973.244.9668
W. E. Tel: 800.862.9953
Parsippany
I. E. Tel: 973.299.4425
Wayne
W. E. Tel: 973.237.9010

New Mexico
W. E. Tel: 480.804.7000
Albuquerque
A. E. Tel: 505.293.5119

U.S. Distributors by State (Continued)

New York
Hauppauge
I. E. Tel: 516.761.0960
Long Island
A. E. Tel: 516.434.7400
W. E. Tel: 800.861.9953
Rochester
A. E. Tel: 716.475.9130
I. E. Tel: 716.242.7790
W. E. Tel: 800.319.9953
Smithtown
B. M. Tel: 800.543.2008
Syracuse
A. E. Tel: 315.449.4927

North Carolina
Raleigh
A. E. Tel: 919.859.9159
I. E. Tel: 919.873.9922
W. E. Tel: 800.560.9953

North Dakota
A. E. Tel: 800.829.0116
W. E. Tel: 612.853.2280

Ohio
Cleveland
A. E. Tel: 216.498.1100
W. E. Tel: 800.763.9953
Dayton
A. E. Tel: 614.888.3313
I. E. Tel: 937.253.7501
W. E. Tel: 800.575.9953
Strongsville
B. M. Tel: 440.238.0404
Valley View
I. E. Tel: 216.520.4333

Oklahoma
W. E. Tel: 972.235.9953
Tulsa
A. E. Tel: 918.459.6000
I. E. Tel: 918.665.4664

Oregon
Beaverton
B. M. Tel: 503.524.1075
I. E. Tel: 503.644.3300
Portland
A. E. Tel: 503.526.6200
W. E. Tel: 800.879.9953

Pennsylvania
Mercer
I. E. Tel: 412.662.2707
Philadelphia
A. E. Tel: 800.526.4812
B. M. Tel: 877.351.2355
W. E. Tel: 800.871.9953
Pittsburgh
A. E. Tel: 412.281.4150
W. E. Tel: 440.248.9996

Rhode Island
A. E. 800.272.9255
W. E. Tel: 781.271.9953

South Carolina
A. E. Tel: 919.872.0712
W. E. Tel: 919.469.1502

South Dakota
A. E. Tel: 800.829.0116
W. E. Tel: 612.853.2280

Tennessee
W. E. Tel: 256.830.1119
East/West
A. E. Tel: 800.241.8182
Tel: 800.633.2918

Texas
Arlington
B. M. Tel: 817.417.5993
Austin
A. E. Tel: 512.219.3700
B. M. Tel: 512.258.0725
I. E. Tel: 512.719.3090
W. E. Tel: 800.365.9953
Dallas
A. E. Tel: 214.553.4300
B. M. Tel: 972.783.4191
W. E. Tel: 800.955.9953
El Paso
A. E. Tel: 800.526.9238
Houston
A. E. Tel: 713.781.6100
B. M. Tel: 713.917.0663
W. E. Tel: 800.888.9953
Richardson
I. E. Tel: 972.783.0800
Rio Grande Valley
A. E. Tel: 210.412.2047
Stafford
I. E. Tel: 281.277.8200

Utah
Centerville
B. M. Tel: 801.295.3900
Murray
I. E. Tel: 801.288.9001
Salt Lake City
A. E. Tel: 801.365.3800
W. E. Tel: 800.477.9953

Vermont
A. E. Tel: 800.272.9255
W. E. Tel: 716.334.5970

Virginia
A. E. Tel: 800.638.5988
W. E. Tel: 301.604.8488
Haymarket
B. M. Tel: 703.754.3399
Springfield
B. M. Tel: 703.644.9045

Washington
Kirkland
I. E. Tel: 425.820.8100
Maple Valley
B. M. Tel: 206.223.0080
Seattle
A. E. Tel: 425.882.7000
W. E. Tel: 800.248.9953

West Virginia
A. E. Tel: 800.638.5988

Wisconsin
Milwaukee
A. E. Tel: 414.513.1500
W. E. Tel: 800.867.9953
Wauwatosa
I. E. Tel: 414.258.5338

Wyoming
A. E. Tel: 800.332.9326
W. E. Tel: 801.974.9953

Sales Offices and Design Resource Centers

LSI Logic Corporation
Corporate Headquarters
1551 McCarthy Blvd
Milpitas CA 95035
Tel: 408.433.8000
Fax: 408.433.8989

NORTH AMERICA

California
Irvine
18301 Von Karman Ave
Suite 900
Irvine, CA 92612
♦ Tel: 949.809.4600
Fax: 949.809.4444

Pleasanton Design Center
5050 Hopyard Road, 3rd Floor
Suite 300
Pleasanton, CA 94588
Tel: 925.730.8800
Fax: 925.730.8700

San Diego
7585 Ronson Road
Suite 100
San Diego, CA 92111
Tel: 858.467.6981
Fax: 858.496.0548

Silicon Valley
1551 McCarthy Blvd
Sales Office
M/S C-500
Milpitas, CA 95035
♦ Tel: 408.433.8000
Fax: 408.954.3353
Design Center
M/S C-410
Tel: 408.433.8000
Fax: 408.433.7695

Wireless Design Center
11452 El Camino Real
Suite 210
San Diego, CA 92130
Tel: 858.350.5560
Fax: 858.350.0171

Colorado
Boulder
4940 Pearl East Circle
Suite 201
Boulder, CO 80301
♦ Tel: 303.447.3800
Fax: 303.541.0641

Colorado Springs
4420 Arrowswest Drive
Colorado Springs, CO 80907
Tel: 719.533.7000
Fax: 719.533.7020

Fort Collins
2001 Danfield Court
Fort Collins, CO 80525
Tel: 970.223.5100
Fax: 970.206.5549

Florida
Boca Raton
2255 Glades Road
Suite 324A
Boca Raton, FL 33431
Tel: 561.989.3236
Fax: 561.989.3237

Georgia
Alpharetta
2475 North Winds Parkway
Suite 200
Alpharetta, GA 30004
Tel: 770.753.6146
Fax: 770.753.6147

Illinois
Oakbrook Terrace
Two Mid American Plaza
Suite 800
Oakbrook Terrace, IL 60181
Tel: 630.954.2234
Fax: 630.954.2235

Kentucky
Bowling Green
1262 Chestnut Street
Bowling Green, KY 42101
Tel: 270.793.0010
Fax: 270.793.0040

Maryland
Bethesda
6903 Rockledge Drive
Suite 230
Bethesda, MD 20817
Tel: 301.897.5800
Fax: 301.897.8389

Massachusetts
Waltham
200 West Street
Waltham, MA 02451
♦ Tel: 781.890.0180
Fax: 781.890.6158

Burlington - Mint Technology
77 South Bedford Street
Burlington, MA 01803
Tel: 781.685.3800
Fax: 781.685.3801

Minnesota
Minneapolis
8300 Norman Center Drive
Suite 730
Minneapolis, MN 55437
♦ Tel: 612.921.8300
Fax: 612.921.8399

New Jersey
Red Bank
125 Half Mile Road
Suite 200
Red Bank, NJ 07701
Tel: 732.933.2656
Fax: 732.933.2643

Cherry Hill - Mint Technology
215 Longstone Drive
Cherry Hill, NJ 08003
Tel: 856.489.5530
Fax: 856.489.5531

New York
Fairport
550 Willowbrook Office Park
Fairport, NY 14450
Tel: 716.218.0020
Fax: 716.218.9010

North Carolina
Raleigh
Phase II
4601 Six Forks Road
Suite 528
Raleigh, NC 27609
Tel: 919.785.4520
Fax: 919.783.8909

Oregon
Beaverton
15455 NW Greenbrier Parkway
Suite 235
Beaverton, OR 97006
Tel: 503.645.0589
Fax: 503.645.6612

Texas
Austin
9020 Capital of TX Highway North
Building 1
Suite 150
Austin, TX 78759
Tel: 512.388.7294
Fax: 512.388.4171

Plano
500 North Central Expressway
Suite 440
Plano, TX 75074
♦ Tel: 972.244.5000
Fax: 972.244.5001

Houston
20405 State Highway 249
Suite 450
Houston, TX 77070
Tel: 281.379.7800
Fax: 281.379.7818

Canada
Ontario
Ottawa
260 Hearst Way
Suite 400
Kanata, ON K2L 3H1
♦ Tel: 613.592.1263
Fax: 613.592.3253

INTERNATIONAL

France
Paris
LSI Logic S.A.
Immeuble Europa
53 bis Avenue de l'Europe
B.P. 139
78148 Velizy-Villacoublay
Cedex, Paris
♦ Tel: 33.1.34.63.13.13
Fax: 33.1.34.63.13.19

Germany
Munich
LSI Logic GmbH
Orleansstrasse 4
81669 Munich
♦ Tel: 49.89.4.58.33.0
Fax: 49.89.4.58.33.108

Stuttgart
Mittlerer Pfad 4
D-70499 Stuttgart
♦ Tel: 49.711.13.96.90
Fax: 49.711.86.61.428

Italy
Milan
LSI Logic S.P.A.
Centro Direzionale Colleoni Palazzo
Orione Ingresso 1
20041 Agrate Brianza, Milano
♦ Tel: 39.039.687371
Fax: 39.039.6057867

Japan
Tokyo
LSI Logic K.K.
Rivage-Shinagawa Bldg. 14F
4-1-8 Kounan
Minato-ku, Tokyo 108-0075
♦ Tel: 81.3.5463.7821
Fax: 81.3.5463.7820

Osaka
Crystal Tower 14F
1-2-27 Shiromi
Chuo-ku, Osaka 540-6014
♦ Tel: 81.6.947.5281
Fax: 81.6.947.5287

Sales Offices and Design Resource Centers (Continued)

Korea
Seoul
LSI Logic Corporation of Korea Ltd
10th Fl., Haesung 1 Bldg.
942, Daechi-dong,
Kangnam-ku, Seoul, 135-283
Tel: 82.2.528.3400
Fax: 82.2.528.2250

The Netherlands
Eindhoven
LSI Logic Europe Ltd
World Trade Center Eindhoven
Building 'Rijder'
Bogert 26
5612 LZ Eindhoven
Tel: 31.40.265.3580
Fax: 31.40.296.2109

Singapore
Singapore
LSI Logic Pte Ltd
7 Temasek Boulevard
#28-02 Suntec Tower One
Singapore 038987
Tel: 65.334.9061
Fax: 65.334.4749

Sweden
Stockholm
LSI Logic AB
Finlandsgatan 14
164 74 Kista
♦ Tel: 46.8.444.15.00
Fax: 46.8.750.66.47

Taiwan
Taipei
LSI Logic Asia, Inc.
Taiwan Branch
10/F 156 Min Sheng E. Road
Section 3
Taipei, Taiwan R.O.C.
Tel: 886.2.2718.7828
Fax: 886.2.2718.8869

United Kingdom
Bracknell
LSI Logic Europe Ltd
Greenwood House
London Road
Bracknell, Berkshire RG12 2UB
♦ Tel: 44.1344.426544
Fax: 44.1344.481039

♦ Sales Offices with Design Resource Centers

International Distributors

Australia
New South Wales
Reptechnic Pty Ltd
3/36 Bydown Street
Neutral Bay, NSW 2089
♦ Tel: 612.9953.9844
Fax: 612.9953.9683

Belgium
Acal nv/sa
Lozenberg 4
1932 Zaventem
Tel: 32.2.7205983
Fax: 32.2.7251014

China
Beijing
LSI Logic International Services Inc.
Beijing Representative Office
Room 708
Canway Building
66 Nan Li Shi Lu
Xicheng District
Beijing 100045, China
Tel: 86.10.6804.2534 to 38
Fax: 86.10.6804.2521

France
Rungis Cedex
Azzurri Technology France
22 Rue Saarinen
Sillic 274
94578 Rungis Cedex
Tel: 33.1.41806310
Fax: 33.1.41730340

Germany
Haar
EBV Elektronik
Hans-Pinsel Str. 4
D-85540 Haar
Tel: 49.89.4600980
Fax: 49.89.46009840

Munich
Avnet Emg GmbH
Stahlgruberring 12
81829 Munich
Tel: 49.89.45110102
Fax: 49.89.42.27.75

Wuennenberg-Haaren
Peacock AG
Graf-Zepplin-Str 14
D-33181 Wuennenberg-Haaren
Tel: 49.2957.79.1692
Fax: 49.2957.79.9341

Hong Kong
Hong Kong
AVT Industrial Ltd
Unit 608 Tower 1
Cheung Sha Wan Plaza
833 Cheung Sha Wan Road
Kowloon, Hong Kong
Tel: 852.2428.0008
Fax: 852.2401.2105

Serial System (HK) Ltd
2301 Nanyang Plaza
57 Hung To Road, Kwun Tong
Kowloon, Hong Kong
Tel: 852.2995.7538
Fax: 852.2950.0386

India
Bangalore
Spike Technologies India Private Ltd
951, Vijayalakshmi Complex,
2nd Floor, 24th Main,
J P Nagar II Phase,
Bangalore, India 560078
♦ Tel: 91.80.664.5530
Fax: 91.80.664.9748

Israel
Tel Aviv
Eastronics Ltd
11 Rozanis Street
P.O. Box 39300
Tel Aviv 61392
Tel: 972.3.6458777
Fax: 972.3.6458666

Japan
Tokyo
Daito Electron
Sogo Kojimachi No.3 Bldg
1-6 Kojimachi
Chiyoda-ku, Tokyo 102-8730
Tel: 81.3.3264.0326
Fax: 81.3.3261.3984

Global Electronics Corporation
Nichibei Time24 Bldg. 35 Tansu-cho
Shinjuku-ku, Tokyo 162-0833
Tel: 81.3.3260.1411
Fax: 81.3.3260.7100
Technical Center
Tel: 81.471.43.8200

Marubeni Solutions
1-26-20 Higashi
Shibuya-ku, Tokyo 150-0001
Tel: 81.3.5778.8662
Fax: 81.3.5778.8669

Shinki Electronics
Myuru Daikanyama 3F
3-7-3 Ebisu Minami
Shibuya-ku, Tokyo 150-0022
Tel: 81.3.3760.3110
Fax: 81.3.3760.3101

Yokohama-City
Innotech
2-15-10 Shin Yokohama
Kohoku-ku
Yokohama-City, 222-8580
Tel: 81.45.474.9037
Fax: 81.45.474.9065

Macnica Corporation
Hakusan High-Tech Park
1-22-2 Hadusan, Midori-Ku,
Yokohama-City, 226-8505
Tel: 81.45.939.6140
Fax: 81.45.939.6141

The Netherlands
Eindhoven
Acal Nederland b.v.
Beatrix de Rijkweg 8
5657 EG Eindhoven
Tel: 31.40.2.502602
Fax: 31.40.2.510255

Switzerland
Brugg
LSI Logic Sulzer AG
Mattenstrasse 6a
CH 2555 Brugg
Tel: 41.32.3743232
Fax: 41.32.3743233

Taiwan
Taipei
Avnet-Mercuries Corporation, Ltd
14F, No. 145,
Sec. 2, Chien Kuo N. Road
Taipei, Taiwan, R.O.C.
Tel: 886.2.2516.7303
Fax: 886.2.2505.7391

Lumax International Corporation, Ltd
7th Fl., 52, Sec. 3
Nan-Kang Road
Taipei, Taiwan, R.O.C.
Tel: 886.2.2788.3656
Fax: 886.2.2788.3568

Prospect Technology Corporation, Ltd
4Fl., No. 34, Chu Luen Street
Taipei, Taiwan, R.O.C.
Tel: 886.2.2721.9533
Fax: 886.2.2773.3756

Wintech Microeletronics Co., Ltd
7F., No. 34, Sec. 3, Pateh Road
Taipei, Taiwan, R.O.C.
Tel: 886.2.2579.5858
Fax: 886.2.2570.3123

United Kingdom
Maidenhead
Azzurri Technology Ltd
16 Grove Park Business Estate
Waltham Road
White Waltham
Maidenhead, Berkshire SL6 3LW
Tel: 44.1628.826826
Fax: 44.1628.829730

Milton Keynes
Ingram Micro (UK) Ltd
Garamonde Drive
Wymbush
Milton Keynes
Buckinghamshire MK8 8DF
Tel: 44.1908.260422

Swindon
EBV Elektronik
12 Interface Business Park
Bincknoll Lane
Wootton Bassett,
Swindon, Wiltshire SN4 8SY
Tel: 44.1793.849933
Fax: 44.1793.859555

♦ Sales Offices with Design Resource Centers