IMPORTANT: See the document titled, *PCI 9056 Blue Book Revision 0.91 Correction,* before designing to this Blue Book. That correction document includes details on features omitted from this Blue Book and it corrects incorrect information in this Blue Book. Please download the correction document from

http://www.plxtech.com/products/9056/

or contact your local PLX Sales Representative for a hard copy.

PLX Technology, Inc.
870 Maude Avenue, Sunnyvale, CA USA 94085

Tel: 408 774-9060
Fax: 408 774-2169

CONFIDENTIAL & PROPRIETARY
PRELIMINARY INFORMATION

PCI 9056 Data Book

Document #: PCI 9056 - _____

Issued To: _____
Name: _____
Company: _____
Date: _____

Authorized: _____

PLX TECHNOLOGY®
THE I/O INTERCONNECT SOLUTION

PCI 9056 Data Book

Preliminary Information

PCI 9056 Data Book

Version 0.91b

February 2002

Website: http://www.plxtech.com
Email: 9056@plxtech.com
Phone: 408 774-9060
800 759-3735
Fax: 408 774-2169

Preliminary Information

Order Number: 9056-SIL-DB-P1-0.91b

Printed in the USA, February 2002

Contents

5. C and J Modes Functional Description 5-1

FIGURES

TABLES

REGISTERS

TIMING DIAGRAMS

PREFACE

The information contained in this document is subject to change without notice. Although an effort has been made to keep the information accurate, there may be misleading or even incorrect statements made herein.

SUPPLEMENTAL DOCUMENTATION

The following is a list of additional documentation to provide the reader with further information:

- *PCI Local Bus Specification, Revision 2.1*,
  PCI Special Interest Group (PCI SIG)
  5440 SW Westgate Drive #217, Portland, OR 97221 USA
  Tel: 800 433-5177 (domestic only) or 503 693-6232, Fax: 503 693-8344, http://www.pcisig.com
- *PCI Local Bus Specification, Revision 2.2*, December 18, 1998
  PCI Special Interest Group (PCI SIG)
  5440 SW Westgate Drive #217, Portland, OR 97221 USA
  Tel: 800 433-5177 (domestic only) or 503 693-6232, Fax: 503 693-8344, http://www.pcisig.com
- *PCI Hot-Plug Specification, Revision 1.0*
  PCI Special Interest Group (PCI SIG)
  5440 SW Westgate Drive #217, Portland, OR 97221 USA
  Tel: 800 433-5177 (domestic only) or 503 693-6232, Fax: 503 693-8344, http://www.pcisig.com
- *PCI Bus Power Management Interface Specification, Revision 1.1*, December 18, 1998
  PCI Special Interest Group (PCI SIG)
  5440 SW Westgate Drive #217, Portland, OR 97221 USA
  Tel: 800 433-5177 (domestic only) or 503 693-6232, Fax: 503 693-8344, http://www.pcisig.com
- *PICMG 2.1, R2.0, Hot Swap Specification,* January 2001
  PCI Industrial Computer Manufacturers Group (PICMG)
  c/o Virtual Inc., 401 Edgewater Place, Suite 500, Wakefield, MA 01880, USA
  Tel: 781 224-1100, Fax: 781 224-1239, http://www.picmg.org
- *Intelligent I/O (I_2O) Architecture Specification, Revision 1.5*, 1997
  I_2O Special Interest Group (I_2O SIG)
  404 Balboa Street, San Francisco, CA 94118, USA
  Tel: 415 750-8352, Fax: 415 751-4829, http://www.i2osig.org
- IEEE Standard 1149.1-1990, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, 1990
  The Institute of Electrical and Electronics Engineers, Inc.
  345 East 47th Street, New York, NY 10017-2394, USA
  Tel: 732 562-3800, Fax: 732 562-1571, http://www.ieee.org

Note: *In the text of this data book, shortened titles are given to the works listed above. The following table lists these abbreviations.*

Supplemental Documentation Abbreviations

Abbreviation	Document
PCI r2.1	*PCI Local Bus Specification, Revision 2.1*
PCI r2.2	*PCI Local Bus Specification, Revision 2.2*
Hot-Plug r1.0	*PCI Hot-Plug Specification, Revision 1.0*
PCI Power Mgmt. r1.1	*PCI Bus Power Management Interface Specification, Revision 1.1*
PICMG 2.1, R2.0	*PICMG 2.1, R2.0, CompactPCI Hot Swap Specification*
I_2O r1.5	*Intelligent I/O (I_2O) Architecture Specification, Revision 1.5*
IEEE Standard 1149.1-1990	*IEEE Standard Test Access Port and Boundary-Scan Architecture*

TERMS AND DEFINITIONS

- Direct Master

  External Local Bus Master initiates Data write/read to/from the PCI Bus

- Direct Slave

  External PCI Bus Master initiates Data write/read to/from the Local Bus

Data Assignment Conventions

Data Width	PCI 9056 Convention
1 byte (8 bits)	Byte
2 bytes (16 bits)	Word
4 bytes (32 bits)	Lword
8 bytes (64 bits)	Qword

REVISION HISTORY

Date	Revision	Comments
01/2000	0.11	Initial release Red Book.
12/2000	0.90	Initial release Blue Book.
01/2001	0.91	Blue Book update. Incorporate *PICMG 2.1, R2.0* Hot Swap Silicon.
02/2002	0.91a	Blue Book update. Only affected pages list the revision and date change.
02/2002	0.91b	Blue Book update. Only affected pages list the revision and date change. Revised pages 2-8, 2-9, 4-8—4-10, 11-3, 11-31, 12-1, 13-2, and 13-3. For corrections to the content on the listed pages, refer to the document, *PCI 9056 Blue Book Revision 0.91 Correction*.

PCI 9056

32-Bit, 66 MHz PCI Bus Mastering I/O Accelerator

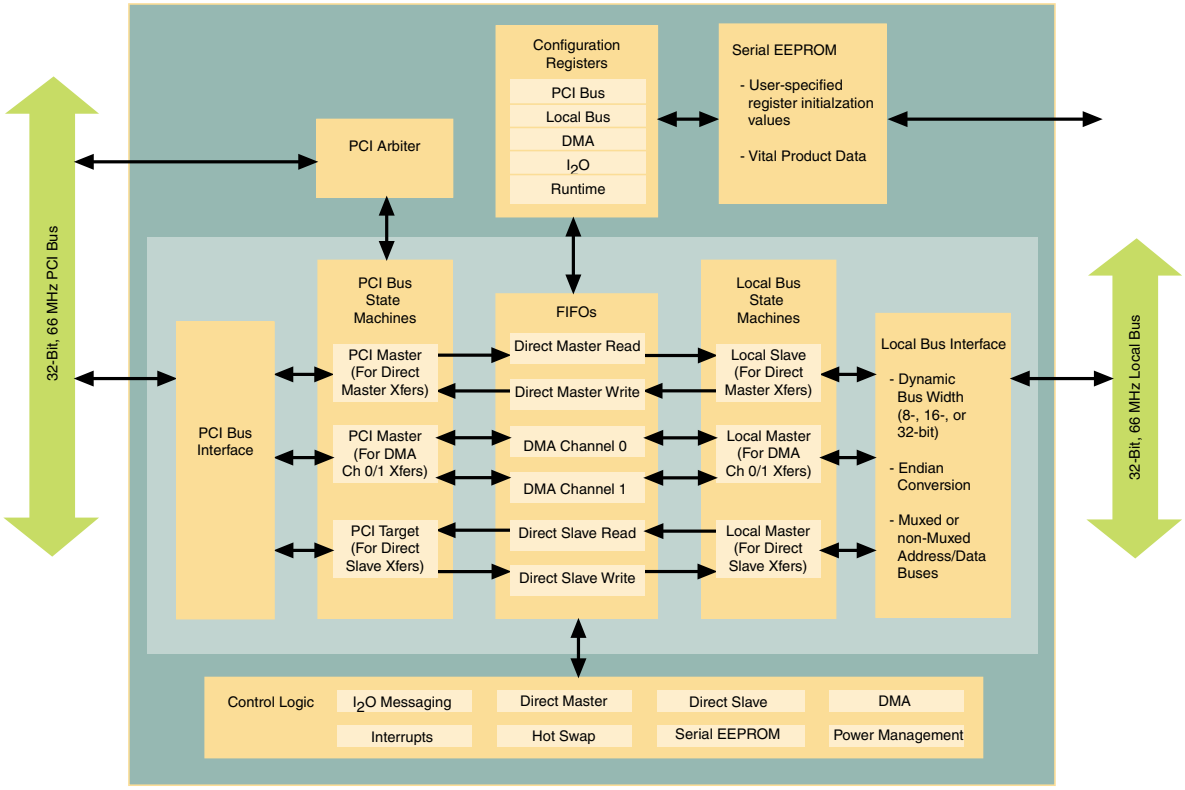
for PowerQUICC and Generic
32-Bit, 66 MHz Local Bus Designs

February 2002
Version 0.91b

FEATURE SUMMARY

- PCI Bus Mastering I/O Accelerator between a 32-bit, 66 MHz PCI Bus and 32-bit, 66 MHz processor Local Bus
  - *PCI r2.2*-compliant
    - Supports Vital Product Data (VPD)
    - Supports PCI Power Management r1.1, including D_{3cold} PME generation for PC 2001 modem and network communications adapter compliance
  - *PICMG 2.1, R2.0*, Hot Swap Silicon
    - Programming Interface 0 (PI = 0)
    - BIAS Voltage Support
    - Early Power Support
    - Initially Not Respond Support
  - *PCI Hot-Plug Specification, Revision 1.0*-compatible
  - Direct connection to three processor Local Bus types
    - **M Mode**—Motorola MPC850, MPC860, PowerPC 801
    - **C Mode (non-multiplexed address/data)**—Intel i960, DSPs, custom ASICs and FPGAs, and others
    - **J Mode (multiplexed address/data)**—Intel i960, IBM PowerPC 401, DSPs, PLX IOP 480, and others
  - Asynchronous clock inputs for PCI and Local Buses
- 256-pin, 17 x 17 mm, 1.00 mm ball pitch PBGA
  - Low-power CMOS 2.5V core, 3.3V I/O
  - 3.3, 5.0V tolerant PCI and Local Bus operation
  - Industrial Temperature Range operation
  - IEEE 1149.1 JTAG boundary scan
- Three data transfer modes—Direct Master, Direct Slave, and DMA
  - **Direct Master**—Transfer data between a Master on the Local Bus and a PCI Bus device
    - Two Local Bus address spaces map to the PCI Bus—one to PCI memory and one to PCI I/O
    - Generates all PCI memory and I/O transaction types, including MWI and Type 0 and Type 1 configuration
    - Read Ahead, Programmable Read Prefetch Counter (all modes)
    - MPC850/MPC860 Delayed Read and IDMA support (M mode)
  - **Direct Slave**—Transfer data between a Master on the PCI Bus and a 32-, 16-, or 8-bit Local Bus device
    - Two general-purpose address spaces to the Local Bus and one expansion ROM address space
    - Delayed Read, Delayed Write, Read Ahead, Posted Write, Programmable Read Prefetch counter
    - Programmable READY# timeout and recovery
  - **DMA**—PCI 9056 services data transfer descriptors, mastering on both buses during transfer
    - Two independent channels
    - Block Mode—Single descriptor execution
    - Scatter/Gather Mode
      - Descriptors in PCI or Local Bus memory
      - Linear descriptor list execution
      - Dynamic DMA descriptor Ring Management with Valid bit semaphore control
      - Burst descriptor loading
    - Hardware EOT/Demand controls to stop/pause DMA in any mode
    - Programmable Local Bus burst lengths, including infinite burst

- Six independent, programmable FIFOs—Direct Master Read and Write, Direct Slave Read and Write, DMA Channel 0 and 1
- Advanced features common to Direct Master, Direct Slave, and DMA
  - Zero wait state burst operation
  - 264 MB/s bursts on PCI Bus
  - 264 MB/s bursts on Local Bus
  - Deep FIFOs prolong fast PCI bursts
  - Unaligned transfers on both buses
  - On-the-fly Local Bus Endian conversion
  - Programmable Local Bus wait states
  - Parity checking on both buses
- *I_2O™ r1.5*-Ready Messaging Unit
- Eight 32-bit Mailbox and two 32-bit Doorbell registers enable general-purpose messaging
- PCI arbiter supports seven external masters
- Reset and interrupt signal directions configurable for host and peripheral applications
- Programmable Interrupt Generator
- Serial EEPROM interface
  - Store user-specified power-on/reset configuration register values
  - Store Vital Product Data (VPD)
- Register compatible with PCI 9060, PCI 9080, PCI 9054, and PCI 9656



PCI 9056 Block Diagram

1 INTRODUCTION

1.1 COMPANY AND PRODUCT BACKGROUND

PLX Technology, Inc. is the leading supplier of high-speed, interconnect silicon and software solutions for the networking and communications industry. These include high-speed silicon, reference design tools that minimize design risk, and software for managing data throughout the PCI Bus, as well as third-party development tool support through the PLX Partner Program, further extending our complete solution.

The PLX solution enables hardware designers and software developers to maximize system input/output (I/O), lower development costs, minimize system design risk, and accelerate time to market.

PLX PCI I/O Accelerator chips and I/O Processor devices are designed in a wide variety of embedded PCI communication systems, including switches, routers, media gateways, base stations, access multiplexors, and remote access concentrators. PLX customers include many of the leading communications equipment companies, including 3Com, Cisco Systems, Compaq Computer, Ericsson, Hewlett-Packard, Intel, IBM, Lucent Technologies, Marconi, Nortel Networks, and Siemens.

Founded in 1986, PLX has developed products based on the PCI industry standard since 1994.

PLX is publicly-traded (NASDAQ: PLXT) and headquartered in Sunnyvale, California, USA, with operations in the United Kingdom, Japan, and China.

1.2 DATA PIPE ARCHITECTURE TECHNOLOGY

PLX I/O accelerators feature PLX proprietary Data Pipe Architecture® technology. This technology consists of powerful, flexible engines for high-speed data transfers, as well as intelligent messaging units for managing distributed I/O functions.

1.2.1 High-Speed Data Transfers

Data Pipe Architecture technology provides independent methods for moving data—Direct Transfers and DMA.

Regardless of the method chosen, Data Pipe Architecture technology data transfers support the following:

- PCI ↔ Local Bus burst transfers at the maximum bus rates
- Unaligned transfers on both buses
- On-the-fly Local Bus Endian conversion
- Programmable Local Bus wait states
- Parity checking on both buses

1.2.1.1 Direct Transfers

Data Pipe Architecture technology Direct Transfers are used by a master on either the PCI or Local Bus to move data through the I/O accelerator to a device on the other bus. The master takes responsibility for moving the data either into the I/O accelerator on a write or out of the I/O accelerator on a read. The I/O accelerator is responsible for moving the data out to the target device on a write, or in from the target device on a read.

1.2.1.1.1 Direct Master

When a master on the local processor bus uses Direct Transfer, this is a *Direct Master* transfer. The I/O accelerator is a master on the PCI Bus. Data Pipe Architecture technology provides independent FIFOs for Direct Master Read and Write transfers. It also supports multiple independent Direct Master Local Bus address spaces for mapping to PCI addresses, as illustrated in Figure 1-1.

Direct Master transfers support generation of all PCI memory and I/O transaction types, including Configuration Type 0 and Type 1 cycles for system configuration.

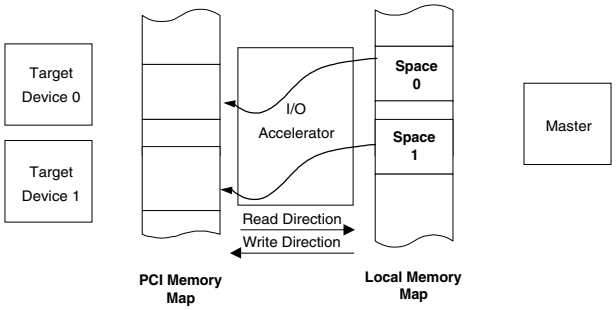


Figure 1-1. Direct Master Address Mapping

1.2.1.1.2 Direct Slave

When a master on the PCI Bus uses Direct Transfer, this is a *Direct Slave* transfer. The I/O accelerator is a slave (technically, a target) on the PCI Bus. Data Pipe Architecture technology provides independent FIFOs for Direct Slave Read and Write transfers. It also supports multiple independent Direct Slave PCI address spaces for mapping to Local Bus addresses, as illustrated in Figure 1-2.

Under software control, Direct Slave transfers support Local Bus data transfers of 16 and 32 bits. Direct Slave read transfers also support PCI delayed reads.

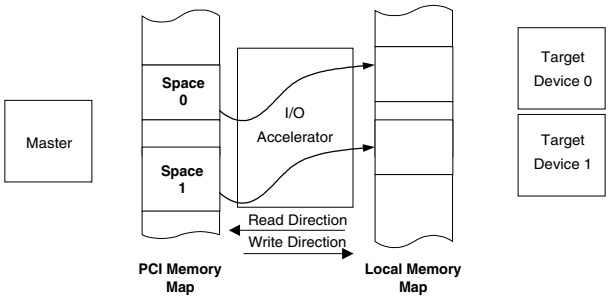


Figure 1-2. Direct Slave Address Mapping

1.2.1.2 DMA

When a Master on either bus uses Data Pipe Architecture technology DMA transfers, instead of the Master moving data, it places a description of the entire transfer in I/O accelerator registers and allows the I/O accelerator to perform the entire data transfer with its DMA engine. This offers two main benefits:

1. Data movement responsibilities are off loaded from the master. A transfer descriptor is short and takes little effort on the master's part to load. Once the descriptor is loaded into the I/O accelerator, the master is free to spend its time and resources elsewhere.

2. Because the I/O accelerator supports multiple DMA channels, each with its own FIFO, it can service multiple PCI and processor Local Bus masters simultaneously. During DMA transfers, the I/O accelerator masters each bus. Consequently, during DMA, there are no external masters to Retry. During DMA, if the I/O accelerator is retried on either bus, it can simply change context to another transfer and continue. Furthermore, DMA can run simultaneously with Direct Master and Direct Slave transfers, providing support for several simultaneous data transfers. Direct Master and Direct Slave transfers have higher priority than DMA.

Data Pipe Architecture technology supports two DMA transfer modes—Block mode and Scatter/Gather mode.

1.2.1.2.1 Block Mode

Block mode is the simplest DMA mode. The master simply programs the description of a single transfer into the I/O accelerator and sets the Start bit. The I/O accelerator signals DMA completion to the master, either by setting a bit in one of its registers that the master polls or by asserting an interrupt.

1.2.1.2.2 Scatter/Gather Mode

In most cases, however, one descriptor is not sufficient. The master typically generates a list of several descriptors in its memory before submitting them to the I/O accelerator. For these cases, *S*catter/Gather mode is used to enable I/O accelerator list processing with minimal master intervention.

With Scatter/Gather mode, the master simply tells the I/O accelerator the location of the first descriptor in its list, sets the Start bit, then waits for the I/O accelerator to service the entire list. This off loads both data and DMA descriptor transfer responsibilities from the master.

Data Pipe Architecture technology supports Scatter/Gather mode descriptor lists in either PCI or Local Bus memory. It also supports linear and circular Ring mode descriptor lists.

Ring mode uses a Valid bit in each descriptor to enable dynamic list management. In this case, the master and I/O accelerator continuously "walk" the descriptor list, the master in the lead filling invalid descriptors, setting the Valid bit when done, and the I/O accelerator following behind servicing valid descriptors, and finally resetting the Valid bit when done. The I/O accelerator supports write back to serviced descriptors, allowing status and actual transfer counts to be posted prior to resetting the Valid bit.

1.2.1.2.3 Hardware DMA Controls—EOT and Demand Mode

To optimize DMA transfers in datacom/telecom and other applications, Data Pipe Architecture technology supports hardware controls for data transfer.

With End of Transfer (EOT), an EOT# signal is asserted to the I/O accelerator to end a transfer. Whenever EOT# is asserted, the I/O accelerator immediately aborts the current DMA transfer and writes back to the current DMA descriptor the actual number of bytes transferred. Data Pipe Architecture technology also supports unlimited bursting. EOT and unlimited bursting are especially useful in applications such as Ethernet adapter cards where the lengths of read packets are not known until the packets are read.

With *Demand mode*, a hardware DREQ#/DACK# signal pair is used to pause and resume the DMA transfer. Data Pipe Architecture technology provides one DREQ#/DACK# signal pair for each DMA channel. Demand mode provides a means for a peripheral device with a FIFO to control DMA transfers. The peripheral device uses Demand mode to pause the transfer when the FIFO is full on a write or empty on a read. Demand mode also resumes the transfer when the FIFO condition changes to allow the data transfer to continue.

1.2.2 Intelligent Messaging Unit

Data Pipe Architecture technology provides two methods for managing system I/O through messaging.

The first method is provided through support for Intelligent I/O (I_2O). As the device independent, industry standard method for I/O control, I_2O is the easiest way to obtain interoperability of all PCI-based components in the system. I_2O is the recommended method for messaging (especially for systems that include PCI or CompactPCI expansion slots).

The second method is provided through general-purpose mailbox and doorbell registers. When all PCI-based components are under direct control of the system designer (*for example*, an embedded system, such as a set-top box), it is often desirable to implement an application-specific messaging unit through general-purpose mailbox and doorbell registers.

1.3 PCI 9056 I/O ACCELERATOR

The PCI 9056, a 32-bit 66 MHz PCI Bus Master I/O Accelerator, extends the PLX family of advanced general-purpose bus master devices to 66 MHz operation. (Refer to Table 1-3 for a detailed comparison of the PCI 9056 with other PLX bus mastering I/O accelerators.)

The PCI 9056 register set is backward-compatible with the previous generation PCI 9054 and PCI 9080 I/O accelerators, and offers a robust *PCI r2.2* implementation, enabling burst transfers up to 264 MB/s. It incorporates the industry-leading PLX Data Pipe Architecture technology, including programmable Direct Master and Direct Slave transfer modes, intelligent DMA engines, and PCI messaging functions.

1.3.1 Applications

The PCI 9056 continues the PLX tradition of expanding its product capabilities to meet the leading edge requirements of I/O intensive embedded-processor applications. The PCI 9056 builds upon the industry-leading PLX PCI 9080 and PCI 9054 products, providing an easy upgrade path to 32-bit, 66 MHz PCI Bus operation and 32-bit, 66 MHz Local Bus operation. The PCI 9056 supports all legacy processors and designs using the M, C and J Local Bus interfaces. Additionally, the PCI 9056 adds several important new features that expand its applicability and performance.

1.3.1.1 High-Performance Motorola MPC850/MPC860 PowerQUICC Designs

A key application for the PCI 9056 is Motorola MPC850- or MPC860-based adapters for telecom

and networking applications. These applications include high-performance communications, such as WAN/LAN controller cards, high-speed modem cards, Frame Relay cards, routers, and switches.

The PCI 9056 simplifies these designs by providing an industry-leading enhanced direct-connect interface to the MPC850 or MPC860 processor. The flexible PCI 9056 3.3V, 5V tolerant I/O buffers, combined with Local Bus operation up to 66 MHz, are ideally suited for current and future PowerQUICC processors.

The PCI 9056 supports the MPC850 and MPC860 IDMA channels for movement of data between the integrated MPC850 or MPC860 communication channels and the PCI Bus.

In addition, the PCI 9056 makes use of the advanced Data Pipe Architecture technology, allowing unlimited burst capability, as illustrated in Figure 1-3.

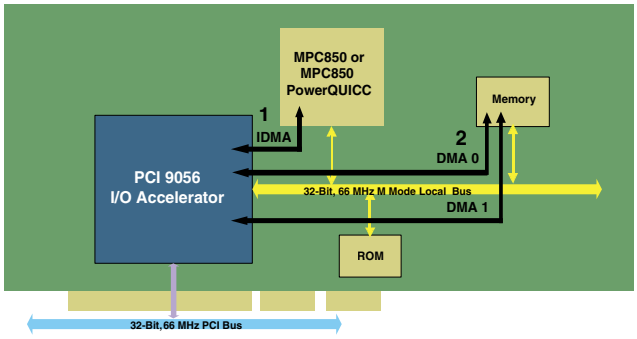


Figure 1-3. High-Performance MPC850 or MPC860 PowerQUICC Adapter Design

1. For PowerQUICC IDMA operation, the PCI 9056 transfers data to the PCI Bus under the control of the IDMA handshake protocol using Direct Master transfers (1).
2. Simultaneously, the PCI 9056 DMA can be operated bi-directionally, with the PCI 9056 as the master for both buses, to manage transfers of data between the Local Bus and the PCI Bus (2).

This is a prime example of how the PCI 9056 provides superior general-purpose bus master performance and provides designers using the PowerQUICC processor with greater flexibility in implementing multiple simultaneous I/O transfers. The PCI 9056 has unlimited bursting capability, which enhances most MPC850 or MPC860 PowerQUICC designs.

1.3.1.2 High-Performance CompactPCI Adapter Cards

Another key application for the PCI 9056 is CompactPCI adapters for telecom and networking applications. These applications include high-performance communications, such as WAN/LAN controller cards, high-speed modem cards, Frame Relay cards, telephony cards for telecom switches, and remote-access systems.

Many processors have integrated communication channels that support ATM, T1/E1, Ethernet, and other high-speed communication standards for communications add-in cards. Today, CompactPCI is the standard choice for the system interconnect of these add-in cards. The PCI 9056 is the perfect choice for adding CompactPCI connection capabilities to a variety of processor platforms.

The PCI 9056 has integrated key features to enable live insertion of Hot Swap CompactPCI adapters. The PCI 9056 Hot Swap Silicon includes the following features:

- Compliant with *PCI r2.2*
- Tolerant of V_{CC} from early power, including support for pin bounce, 2.5 and 3.3V appearing in any order, I/O cell stability within 4 ms, and low current drain during insertion
- Tolerant of asynchronous reset
- Tolerant of pre-charge voltage
- I/O buffers meet modified V/I requirements in *PICMG 2.1, R2.0*
- Limited I/O pin leakage at precharge voltage
- Incorporates the Hot Swap Control/Status register (HS_CSR)
- Incorporates an Extended Capability Pointer (ECP) to the Hot Swap Control/Status register
- Incorporates added resources for software control of the ejector switch, ENUM#, and the blue status LED which indicates insertion and removal to the user

- BIAS voltage support with integrated 10K ohm precharge resistors eliminates the need for an external resistor network
- Early power support allows transition between the operating and powered down states without external circuitry
- Programming Interface 0 (PI = 0)
- Initially Not Respond Support

Figure 1-4 illustrates a CompactPCI peripheral card that utilizes an MPC860 CPU for communication I/O and the PCI 9056 for PCI-based I/O.

The PCI 9056, with its internal PCI arbiter, reset signal direction control, and Type 0 and Type 1 PCI configuration support, is an ideal choice for CompactPCI system cards.

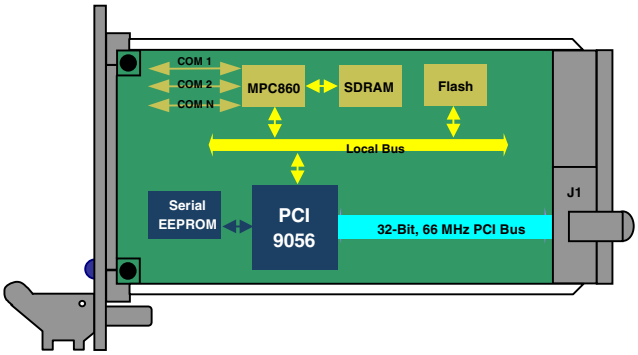


Figure 1-4. PCI 9056 CompactPCI Peripheral Card

1.3.1.3 High-Performance PC Adapter Cards

The PCI 9056 is also designed for traditional PCI adapter card applications requiring 32-bit, 66 MHz PCI operation and bandwidth. Specific applications include high-performance communications, networking, disk control, and data encryption adapters. As such, the PCI 9056 enables easy migration of existing 32-bit, 33 MHz PCI I/O accelerator designs to 32-bit, 66 MHz capability.

Today, Power Management and Green PCs are major initiatives in traditional PCI applications. The PCI 9056 supports PCI power management, including generation of PME in the D_{3cold} state. This is especially useful in applications such as modem cards that are responsible for waking up the system (as a result of an external event), such as a telephone line ringing.

Figure 1-5 illustrates the PCI 9056 in a PCI adapter card application with a CPU, using the C or J Local Bus modes.

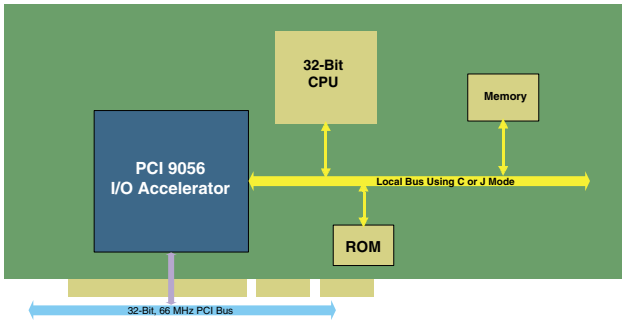


Figure 1-5. PCI 9056 PC Adapter Card with C or J Mode Processor

The C and J Local Bus modes, in addition to supporting Intel's i960 processors, have been adopted by designers of a wide variety of devices, ranging from DSPs to custom ASICs, because of their high-speed, low overhead, and relative simplicity.

For applications using I/O types not supported directly by the processor (such as, SCSI for storage applications), the PCI 9056 provide a high-speed interface between the processor and PCI-based I/O chips. Furthermore, its Local Bus interface supports processors that do not include integrated I/O.

Typically, a PCI-to-PCI bridge chip is used to isolate the add-in card's local PCI Bus and its I/O chips from the system bus. Figure 1-6 illustrates a typical PCI add-in card with local PCI I/O using the PCI 9056 and a PCI-to-PCI bridge interfacing to the system PCI Bus. The PCI 9056 internal PCI arbiter provides arbitration services to the devices on the local PCI Bus.

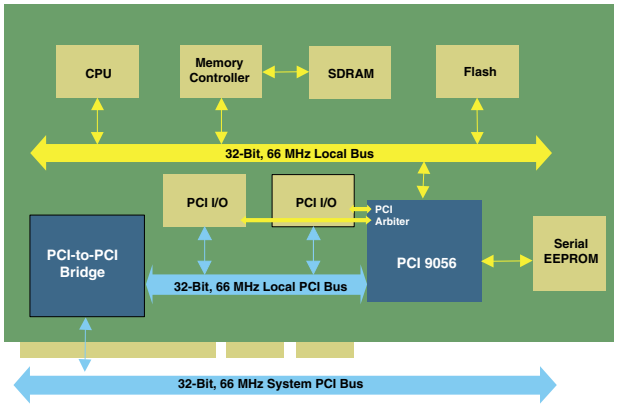


Figure 1-6. PCI 9056 PC Adapter Card with Local PCI I/O and PCI-to-PCI Bridge

The PCI 9056 directly interfaces to the PLX IOP 480, which includes an integrated PowerPC 401 processor, and PCI Bus. The IOP 480 can control the local PCI Bus while the PCI 9056 provides 32-bit, 66 MHz, mastering on the system PCI Bus, as illustrated in Figure 1-7.

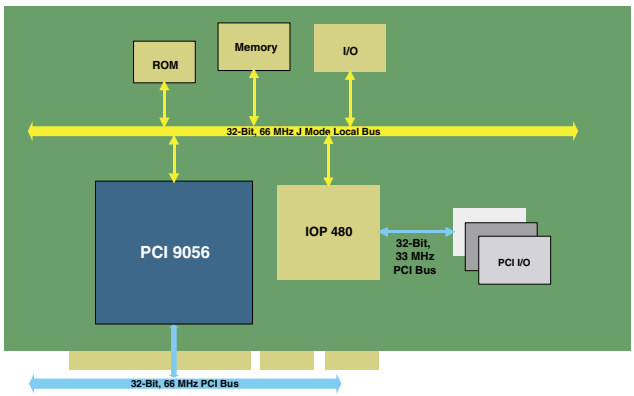


Figure 1-7. PCI 9056 PC Adapter Card with Local PCI I/O and IOP 480

1.3.1.4 High-Performance Embedded Host Designs

I/O intensive embedded host designs are another major application of the PCI 9056. These applications include network switches and routers, printer engines, set-top boxes, CompactPCI system cards, and industrial equipment.

While the support requirements of these embedded host designs share many similarities with peripheral card designs (such as, their requirement for intelligent management of high-performance I/O), there are three significant differences.

First, the host is responsible for configuring the system PCI Bus. The PCI 9056 supports PCI Configuration Type 0 and Type 1 cycles to accomplish this.

Second, the host is responsible for providing PCI Bus arbitration services. The PCI 9056 includes an internal PCI arbiter that supports seven external PCI masters in addition to the PCI 9056. This is sufficient for a standard 33 MHz CompactPCI backplane with seven peripheral slots and one system slot.

Third, for hosts, the directions of the reset and interrupt signals reverse. The PCI 9056 includes a strapping option for reversing the directions of the PCI and Local Bus reset and interrupt signals. In one setting, the directions are appropriate for a peripheral. In the other setting, they are appropriate for a host.

Figure 1-8 illustrates the PCI 9056 in an embedded host system. Figure 1-9 illustrates the PCI 9056 and IOP 480 in an embedded host system.

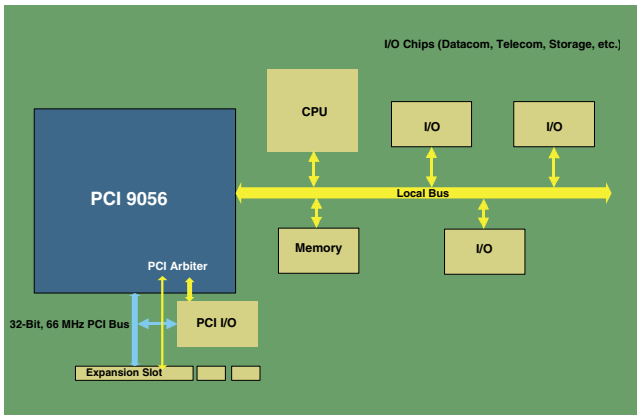


Figure 1-8. PCI 9056 Embedded Host System with Generic Host CPU

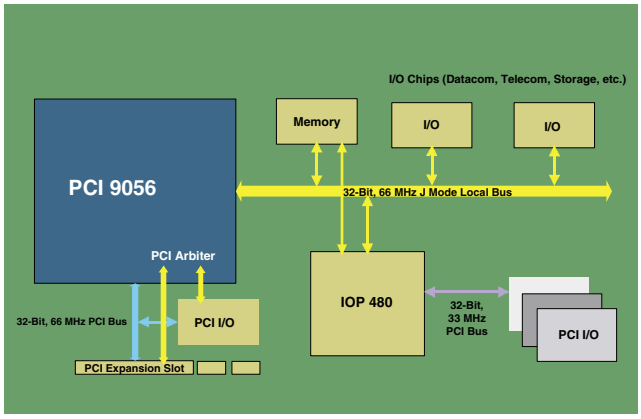


Figure 1-9. PCI 9056 Embedded Host System with IOP 480

1.4 PCI 9056 MAJOR FEATURES

1.4.1 Interfaces

The PCI 9056 is a PCI Bus Master interface chip that connects a 32-bit, 66 MHz PCI Bus to one of three 32-bit, 66 MHz Local Bus types.

PCI r2.1 and r2.2 Compliant. Compliant with *PCI r2.1 and PCI r2.2*, including 66 MHz operation.

New Capabilities Structure. Supports New Capabilities registers to define additional capabilities of the PCI functions.

VPD Support. Supports the Vital Product Data (VPD) PCI extension through its serial EEPROM interface, providing an alternate to Expansion ROM for VPD access.

Power Management. Supports all five power states for PCI Power Management functions (D_0, D_1, D_2, D_{3hot}, and D_{3cold}) and Power Management Event interrupt (PME#) generation in all five states, including D_{3cold}.

PICMG 2.1, R2.0 Hot Swap Silicon. Compliant with *PICMG 2.1, R2.0*, including support for Programming Interface (PI = 0), BIAS Voltage and Early Power Support, and an option to Initially Not Respond while the chip is initializing.

PCI Hot-Plug Compliant. Compliant with *Hot-Plug r1.0*.

Subsystem ID and Subsystem Vendor ID. Contains Subsystem ID and Subsystem Vendor ID in the PCI Configuration register space, in addition to System and Vendor IDs. The PCI 9056 also contains a permanent Vendor ID (10B5h) and Device ID (9056h).

RST# Timing. Supports response to first configuration accesses after de-assertion of RST# under 2^{25} clocks.

Clocks. The PCI and Local Bus clocks are independent and asynchronous. The Local Bus interface runs from an external clock to provide the necessary internal clocks.

Local Bus Direct Interface. 32-bit, 66 MHz Local Bus interface supports direct connection to the Motorola 801 PowerPC and MPC850/MPC860 PowerQUICC families, the Intel i960 family, the IBM PowerPC 401 family, the PLX IOP 480, Texas Instruments DSPs, and other similar bus-protocol devices.

Local Bus Types. Local bus type selected through a pin strapping option, as listed in the following table.

Table 1-1. Local Bus Types

Mode	Description
M	32-bit address/32-bit data, non-multiplexed direct connect interface to MPC850 or MPC860 PowerQUICC
C	32-bit address/32-bit data, non-multiplexed
J	32-bit address/32-bit data, multiplexed

1.4.2 Data Transfer

PCI ↔ Local Burst Transfers up to 264 MB/s.

Six Programmable FIFOs for Zero Wait State Burst Operation. The following table enumerates the FIFO depth.

Table 1-2. FIFO Depth

FIFO	Depth
Direct Master Read	32 Lwords
Direct Master Write	64 Lwords
Direct Slave Read	32 Lwords
Direct Slave Write	64 Lwords
DMA Channel 0	64 Lwords
DMA Channel 1	64 Lwords

Unaligned Transfer Support. Capable of transferring data on any byte-boundary combination of the PCI and Local Address Bus addresses.

Big/Little Endian Conversion. Supports dynamic switching between Big Endian (Address Invariance) and Little Endian (Data Invariance) operations for Direct Slave, Direct Master, DMA, and internal register accesses on the Local Bus.

Supports on-the-fly Endian conversion of Local Bus data transfers. The Local Bus can be Big/Little Endian by using the BIGEND# input pin or programmable internal register configuration. When BIGEND# is asserted, it overrides the internal register configuration during Direct Master, and internal register accesses on the Local Bus.

Note: *The PCI Bus is always Little Endian.*

Keep Bus Mode (M Mode). Supports program control to retain the PCI Bus by generating wait state(s) if the Direct Slave Write FIFO becomes full. Can also be programmed to retain the Local Bus (BB# asserted) if the Direct Slave Write FIFO becomes empty or the Direct Slave Read FIFO becomes full. The Local Bus is dropped in either case when the Local Bus Latency Timer is enabled and expires.

Keep Bus Mode (C and J Modes). The PCI 9056 can be programmed to retain the PCI Bus by generating one or more wait states if the Direct Slave Write FIFO becomes full. The PCI 9056 can also be programmed to retain the Local Bus (LHOLD asserted) if the Direct Slave Write FIFO becomes empty or the Direct Slave Read FIFO becomes full. The Local Bus is dropped in either case when the Local Bus Latency Timer is enabled and expires.

M Mode Data Transfers. Communicates with the MPC850 or MPC860, using five possible data transfer modes:

- Configuration Register Access
- Direct Master Operation
- Direct Slave Operation
- DMA Operation
- IDMA/SDMA Operation

C and J Mode Data Transfers. Communicates with these processors, using four possible data transfer modes:

- Configuration Register Access
- Direct Master Operation
- Direct Slave Operation
- DMA Operation

Direct Master. Supports PCI accesses from a Local Bus master. Burst transfers are supported for memory-mapped devices. Single transfers are supported for Memory- and I/O-Mapped devices. Also supports PCI Bus interlock (LOCK#) cycles.

Direct Slave. Supports Burst Memory-Mapped and single I/O-Mapped accesses to the Local Bus. Supports 8-, 16-, and 32-bit Local Bus data transfers. The Read and Write FIFOs enable high-performance bursting.

Three PCI-to-Local Address Spaces. Supports three PCI-to-Local Address spaces in Direct Slave mode—Space 0, Space 1, and Expansion ROM. These spaces allow any PCI Bus master to access the Local Bus memory spaces with programmable wait states, bus width, burst capabilities, and so forth.

Read Ahead Mode. Supports Read Ahead mode, where prefetched data can be read from the internal Read FIFO instead of the external bus. The address must be subsequent to the previous address and 32-bit aligned (next address = current address + 4). This feature allows for increased bandwidth and reduced data latency.

Programmable Prefetch Counter. Includes programmable controls to prefetch data during Direct Master and Direct Slave accesses. To perform burst reads, prefetching must be enabled. The prefetch size can be programmed to match the master burst length, or can be used as Read Ahead mode data. Reads single data (8, 16, or 32 bit) if the Master initiates a single cycle; otherwise, prefetches the programmed size.

Posted Memory Writes. Supports the Posted Memory Writes (PMW) for maximum performance and to avoid potential deadlock situations.

Two DMA Channels with Independent FIFOs. Provides two independently programmable DMA controllers with independently programmable FIFOs. Each channel supports Block and Scatter/Gather DMA modes, including ring management, as well as EOT mode. Supports Demand mode DMA for both channels.

PCI Dual-Address Cycle (DAC) Support (32-bit Address Space). Supports PCI Dual Address Cycle beyond the low 4-GB Address space. PCI DAC can be used during PCI 9056 PCI Bus Master operation (DMA and Direct Master).

1.4.3 Messaging Unit

I_2O-Ready Messaging Unit. Incorporates the I_2O-Ready Messaging Unit, which enables the adapter or embedded system to communicate with other I_2O-supported devices. The I_2O Messaging Unit is fully compatible with the PCI extension of *I_2O r1.5*.

Mailbox Registers. Includes eight 32-bit Mailbox registers that may be accessed from the PCI or Local Bus.

Doorbell Registers. Includes two 32-bit doorbell registers. One asserts interrupts from the PCI Bus to the Local Bus. The other asserts interrupts from the Local Bus to the PCI Bus.

1.4.4 Hosting Features

Type 0 and Type 1 Configuration. In Direct Master mode, supports Type 0 and Type 1 PCI Configuration cycles.

Internal PCI Arbiter. Includes integrated PCI arbiter that supports seven external masters in addition to the PCI 9056.

Reset and Interrupt Signal Directions. Includes a strapping option to reverse the directions of the PCI and Local Bus reset and interrupt signals.

1.4.5 Electrical/Mechanical

Packaging. Available in a 256-pin, 17 x 17 mm PBGA package.

2.5V Core/3.3V I/O. Low power CMOS 2.5V core with 3.3V I/O.

5V Tolerant Operation. Provides 3.3V signaling with 5V I/O tolerance on both the PCI and Local Buses.

Industrial Temperature Range Operation. The PCI 9056 works in a -40 to +85 °C temperature range.

JTAG. Supports IEEE 1149.1 JTAG boundary scan.

1.4.6 Miscellaneous

Serial EEPROM Interface. Includes a serial EEPROM interface (optional only if using a Local processor) that can be used to load configuration information. This is useful for loading information that is unique to a particular adapter (such as, the Network or Vendor ID).

Interrupt Generator. Can assert PCI and Local interrupts from external and internal sources.

1.5 COMPATIBILITY WITH OTHER PLX CHIPS

1.5.1 Pin Compatibility

The PCI 9056 is ***not*** pin compatible with other PLX chips.

1.5.2 Register Compatibility

All registers implemented in the PCI 9060, PCI 9080, PCI 9054, and PCI 9656 are implemented in the PCI 9056. The PCI 9056 includes many new bit definitions and several new registers. (Refer to Section 11 for details.)

The PCI 9056 is ***not*** register-compatible with the PCI 9050.

1.5.3 PCI 9056 Comparison with Other PLX Chips

Table 1-3. Bus Master I/O Accelerator PLX Product Comparison

Features	PCI 9054-AB50PI PCI 9054-AB50BI	PCI 9056-AA66BI	PCI 9656-AA66BI
Interfaces			
Host Bus Type	32-Bit, 33 MHz *PCI r2.2*	32-Bit, 66 MHz *PCI r2.2*	64-Bit, 66 MHz *PCI r2.2*
Processor Local Bus Type(s): A = Address Bus D = Data Bus Mux = Multiplexed A/D Buses Non-Mux = Non-Multiplexed A/D Buses	C: Generic, 32-Bit A, 32-Bit D, non-mux J: Generic, 32-Bit A, 32-Bit D, mux M: PowerPC® PowerQUICC®, 32-Bit A, 32-Bit D, non-mux	C: Generic, 32-Bit A, 32-Bit D, non-mux J: Generic, 32-Bit A, 32-Bit D, mux M: PowerPC® PowerQUICC®, 32-Bit A, 32-Bit D, non-mux	C: Generic, 32-Bit A, 32-Bit D, non-mux J: Generic, 32-Bit A, 32-Bit D, mux M: PowerPC® PowerQUICC®, 32-Bit A, 32-Bit D, non-mux
Maximum Processor Local Bus Speed	50 MHz	66 MHz	66 MHz
Core Voltage	3.3V	2.5V	2.5V
I/O Ring Voltage	3.3V	3.3V	3.3V
3.3V PCI Bus Signalling	✓	✓	✓
5V PCI Bus Signalling	✓	✓	✓
3.3V Tolerant Local Bus	✓	✓	✓
5V Tolerant Local Bus	✓	✓	✓
PICMG 2.1, R2.0	Programming Interface (PI = 0)	Programming Interface (PI = 0) Bias Voltage Support Early Power Support Initially Not Respond Support	Programming Interface (PI = 0) Bias Voltage Support Early Power Support 64-Bit Initialization
Package Size/Type(s): Pin/Ball Count External Dimensions (mm) Pin/Ball Pitch (mm) Package Type	176-Pin, 26 x 26, .5 PQFP 225-Pin, 27 x 27, 1.5 PBGA	256-Pin, 17 x 17, 1.00 PBGA	272-Pin, 27 x 27, 1.27 PBGA
Industrial Temperature Range Operation	✓	✓	✓
Data Transfer			
Direct Slave Address Spaces	Two General-Purpose One Expansion ROM	Two General-Purpose One Expansion ROM	Two General-Purpose One Expansion ROM
Direct Slave Read FIFO Depth	16 Lwords (64 bytes)	32 Lwords (128 bytes)	16 Qwords (128 bytes)
Direct Slave Write FIFO Depth	32 Lwords (128 bytes)	64 Lwords (256 bytes)	32 Qwords (256 bytes)
Delayed Read Support	✓	✓	✓
Programmable READY# Timeout	—	✓	✓
Direct Master Address Spaces	1	1	1
Direct Master Read FIFO Depth	16 Lwords (64 bytes)	32 Lwords (128 bytes)	16 Qwords (128 bytes)
Direct Master Write FIFO Depth	32 Lwords (128 bytes)	64 Lwords (256 bytes)	32 Qwords (256 bytes)

Table 1-3. Bus Master I/O Accelerator PLX Product Comparison (Continued)

Features	PCI 9054-AB50PI PCI 9054-AB50BI	PCI 9056-AA66BI	PCI 9656-AA66BI
Data Transfer (Continued)			
DMA Channels	2	2	2
DMA Channel 0 FIFO Depth	32 Lwords (128 bytes) Bi-directional	64 Lwords (256 bytes) Bi-directional	32 Qwords (256 bytes) Bi-directional
DMA Channel 1 FIFO Depth	16 Lwords (64 bytes) Bi-directional	64 Lwords (256 bytes) Bi-directional	32 Qwords (256 bytes) Bi-directional
DMA Demand Mode Hardware Control	✓ (Channel 0 Only)	✓	✓
DMA EOT Mode Hardware Control	✓	✓	✓
DMA Block Mode	✓	✓	✓
DMA Scatter/Gather Mode	✓	✓	✓
DMA Ring Management Mode	—	✓	✓
Programmable Prefetch Counter	✓	✓	✓
Dual Address Cycle Generation	—	✓	✓
Big Endian/Little Endian Conversion	✓	✓	✓
Control			
Mailbox Registers	Eight 32-Bit	Eight 32-Bit	Eight 32-Bit
Doorbell Registers	Two 32-Bit	Two 32-Bit	Two 32-Bit
I_2O Messaging Unit	✓ *r1.5*	✓ *r1.5*	✓ *r1.5*
PCI Arbiter	—	✓ Seven external masters	✓ Seven external masters
PCI Type 0 and Type 1 Configuration	✓	✓	✓
PCI Power Management	—	✓ *r1.1*	✓ *r1.1*
D_{3cold} PME Generation	—	✓	✓
PCI r2.2 VPD Support	—	✓	✓
Serial EEPROM Support	2K bit, 4K bit Microwire® devices with sequential read support	2K bit, 4K bit Microwire® devices with sequential read support	2K bit, 4K bit Microwire® devices with sequential read support
JTAG Boundary Scan	—	✓	✓
Register Compatibility	—	Backward compatible with PCI 9054	Backward compatible with PCI 9054

2 M MODE BUS OPERATION

2.1 PCI BUS CYCLES

The PCI 9056 is compliant with *PCI r2.2*. Refer to *PCI r2.2* for specific PCI Bus functions.

2.1.1 Direct Slave Command Codes

As a Target, the PCI 9056 allows access to the PCI 9056 internal registers and the Local Bus, using the commands listed in Table 2-1.

All Read or Write accesses to the PCI 9056 can be Byte, Word, or Long-Word (Lword) accesses, defined as 32 bit. All memory commands are aliased to basic memory commands. All I/O accesses to the PCI 9056 are decoded to an Lword boundary. Byte enables are used to determine which bytes are read or written. An I/O access with illegal byte enable combinations is terminated with a Target Abort.

Table 2-1. Direct Slave Command Codes

Command Type	Code (C/BE[3:0]#)
I/O Read	0010 (2h)
I/O Write	0011 (3h)
Memory Read	0110 (6h)
Memory Write	0111 (7h)
Configuration Read	1010 (Ah)
Configuration Write	1011 (Bh)
Memory Read Multiple	1100 (Ch)
Memory Read Line	1110 (Eh)
Memory Write and Invalidate	1111 (Fh)

2.1.2 PCI Master Command Codes

The PCI 9056 can access the PCI Bus to perform DMA or Direct Master Local-to-PCI Bus transfers. During a Direct Master or DMA transfer, the command code assigned to the PCI 9056 internal register location (CNTRL[15:0]) is used as the PCI command code (except for Memory Write and Invalidate mode for DMA cycles where DMPBAM[9]=1).

Notes: *Programmable internal registers determine PCI command codes when the PCI 9056 is the Master.*

DMA cannot perform I/O or Configuration accesses.

2.1.2.1 DMA Master Command Codes

The PCI 9056 DMA controllers can assert the Memory Command cycles listed in Table 2-2.

Table 2-2. DMA Master Command Codes

Command Type	Code (C/BE[3:0]#)
Memory Read	0110 (6h)
Memory Write	0111 (7h)
Memory Read Multiple	1100 (Ch)
PCI Dual Address Cycle	1101 (Dh)
Memory Read Line	1110 (Eh)
Memory Write and Invalidate	1111 (Fh)

2.1.2.2 Direct Master Local-to-PCI Command Codes

For Direct Master Local-to-PCI Bus accesses, the PCI 9056 asserts the cycles listed in Table 2-3 through Table 2-5.

Table 2-3. Local-to-PCI Memory Access

Command Type	Code (C/BE[3:0]#)
Memory Read	0110 (6h)
Memory Write	0111 (7h)
Memory Read Multiple	1100 (Ch)
PCI Dual Address Cycle	1101 (Dh)
Memory Read Line	1110 (Eh)
Memory Write and Invalidate	1111 (Fh)

Table 2-4. Local-to-PCI I/O Access

Command Type	Code (C/BE[3:0]#)
I/O Read	0010 (2h)
I/O Write	0011 (3h)

Table 2-5. Local-to-PCI Configuration Access

Command Type	Code (C/BE[3:0]#)
Configuration Memory Read	1010 (Ah)
Configuration Memory Write	1011 (Bh)

2.1.3 PCI Arbitration

The PCI 9056 asserts REQ# to request the PCI Bus. The PCI 9056 can be programmed using the PCI Request Mode bit (MARBR[23]) to de-assert REQ# when it asserts FRAME# during a Bus Master cycle, or to keep REQ# asserted for the entire Bus Master cycle. The PCI 9056 always de-asserts REQ# for a minimum of two PCI clocks after a bus ownership that sustains a Target disconnect.

The Direct Master Write Delay bits (DMPBAM[15:14]) can be programmed to delay the PCI 9056 from asserting PCI REQ# during a Direct Master Write cycle. DMPBAM can be programmed to wait 0, 4, 8, or 16 PCI Bus clocks after the PCI 9056 has received its first Write data from the Local Bus Master, and is ready to begin the PCI Write transaction. This function is useful in applications where a Local Master is bursting and a Local Bus clock is slower than the PCI Bus clock. This allows Write data to accumulate in the PCI 9056 Direct Master Write FIFO, which provides for better use of the PCI Bus.

2.2 LOCAL BUS CYCLES

The PCI 9056 interfaces a PCI Host bus to several Local Bus types, as listed in Table 2-6. It operates in one of three modes (selected through the MODE[1:0] pins), corresponding to the three bus types——M, C, and J.

Table 2-6. Local Bus Types

MODE1	MODE0	Bus Mode	Bus Type
1	1	M	32-bit non-multiplexed
1	0	Reserved	—
0	0	C	32-bit non-multiplexed
0	1	J	32-bit multiplexed

In M mode, the PCI 9056 provides a direct connection to the MPC850 or MPC860 address and data lines, regardless of the PCI 9056 Little Endian or Big Endian modes.

2.2.1 Local Bus Arbitration

The PCI 9056 asserts BR# to request the Local Bus. It owns the Local Bus when BG# is asserted. Upon receiving BG#, the PCI 9056 waits for BB# to de-assert. The PCI 9056 then asserts BB# at the next rising edge of the Local clock after acknowledging BB# is de-asserted (no other device is acting as the Local Bus Master). The PCI 9056 continues to assert BB# while acting as the Local Bus Master (*that is*, it holds the bus until instructed to release BB#) when the Local Bus Latency Timer is enabled and expires (MARBR[7:0]) or the transaction is complete.

Note: *The Local Bus Pause Timer applies only to DMA operation. It does* ***not*** *apply to Direct Slave operation.*

2.2.2 Direct Master

Local Bus cycles can be single or Burst cycles. As a Local Bus Target, the PCI 9056 allows access to the PCI 9056 internal registers and the PCI Bus.

Local Bus Direct Master accesses to the PCI 9056 must be for a 32-bit non-pipelined bus. Non-32-bit Direct Master accesses to the PCI 9056 require simple external logic (latch array to combine data into a 32-bit bus).

2.2.3 Direct Slave

The PCI Bus Master reads from and writes to the Local Bus (the PCI 9056 is a PCI Bus Target and a Local Bus Master).

2.2.4 Wait State Control

The TA# signal overwrites the programmable wait state counter, and can be used to introduce additional wait states. The following figure illustrates the PCI 9056 wait states for M mode.

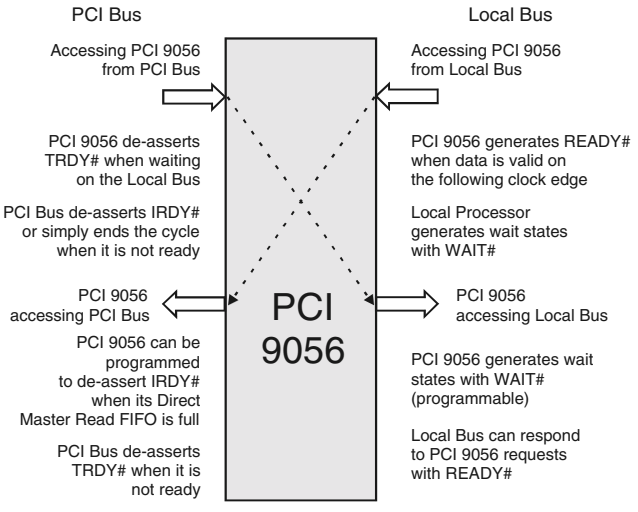


Figure 2-1. Wait States

Note: *Figure 2-1 represents a sequence of Bus cycles.*

2.2.4.1 Wait States—Local Bus

In Direct Master mode, when accessing the PCI 9056 registers, the PCI 9056 acts as a Local Bus Slave. The PCI 9056 asserts external wait states with the TA# signal.

In Direct Slave and DMA modes, the PCI 9056 acts as a Local Bus Master. The Internal Wait States bit(s) (LBRD0[21:18, 5:2], LBRD1[5:2], DMAMODE0[5:2], and/or DMAMODE1[5:2]) can be used to program the number of internal wait states between the first address-to-data (and subsequent data-to-data in Burst mode).

In Direct Slave and DMA modes, if TA# is enabled and active, it continues the Data transfer, regardless of the wait state counter.

2.2.4.2 Wait States—PCI Bus

To insert PCI Bus wait state(s), the PCI Bus Master throttles IRDY# and the PCI Bus Slave throttles TRDY#.

2.2.5 Burst Mode and Continuous Burst Mode (Bterm "Burst Terminate" Mode)

Note: *In the following sections, Bterm refers to the PCI 9056 internal register bit, and BTERM# refers to the PCI 9056 external signal.*

2.2.5.1 Burst and Bterm Modes

Table 2-7. Burst and Bterm on the Local Bus

Mode	Burst	Bterm	Result
Single Cycle	0	0	One TS# per data (default).
	0	1	One TS# per data.
Burst-4	1	0	One TS# per four data (recommended for MPC850 or MPC860).
Burst Forever	1	1	Direct Slave or DMA—One TS# per Burst data or until BI# is asserted. (Refer to Section 2.2.5.2.1.)

Note: *BI# is supported in Burst-4 mode. Refer to the MPC850 or MPC860 data manual.*

On the Local Bus, BTERM# is not supported, but the Bterm bit can be used to gain maximum performance and data throughput.

- If the Burst Mode bit is enabled, but the Bterm Mode bit is disabled, then the PCI 9056 bursts four Lwords. BDIP# is de-asserted at the last Lword transfer before its completion (LA[2:3]=11) and a new TS# is asserted at the first Lword (LA[2:3]=00) of the next burst.
- If the Burst Mode and Bterm Mode bits are both enabled, then the PCI 9056 bursts until the transfer counter counts to "0", the Local Latency Timer is enabled and expires, the EOT function is introduced, or DREQ0# is de-asserted during DMA transactions. For Direct Slave transactions, the PCI 9056 bursts until BI# is asserted, implying a new TS# is required, or the Local Latency Timer is enabled and expires. The PCI 9056 does not release bus ownership during BI# assertion. BDIP# output is supported in Burst Forever mode with a different behavior then MPC860 protocol. Refer to Section 2.2.5.2.

Notes: *If Address Increment is disabled, the DMA transaction bursts beyond four Lwords.*

If the Bterm Mode bit is disabled, the PCI 9056 performs the following:

- ***32-bit Local Bus****—Bursts up to four Lwords*
- ***16-bit Local Bus****—Bursts up to four Lwords*
- ***8-bit Local Bus****—Bursts up to four Lwords*

In every case, it transfers 16 bytes of data.

2.2.5.2 Burst-4 Lword Mode

If the Burst Mode bit is enabled and the Bterm Mode bit is disabled, bursting can start only on a 16-byte boundary and continue up to the next 16-byte address boundary. After data before the boundary is transferred, the PCI 9056 asserts a new Address cycle (TS#).

Table 2-8. Burst-4 Lword Mode

Bus Width	Burst
32 bit	Four Lwords or up to a quad Lword boundary (LA3, LA2 = 11)
16 bit	Eight words or up to a quad Lword boundary (LA2, LA1 = 11)
8 bit	Sixteen bytes or up to a quad Lword boundary (LA1, LA0 = 11)

2.2.5.2.1 Continuous Burst Mode (Bterm "Burst Terminate" Mode)

If both the Burst and Bterm Mode bits are enabled, the PCI 9056 can operate beyond the Burst-4 Lword mode.

Bterm mode enables the PCI 9056 to perform long bursts to special external M mode interface devices that can accept bursts of longer than four Lwords. The PCI 9056 asserts one Address cycle and continues to burst data. The external address is incremented during bursts. If a device requires a new Address cycle, it can assert BI# input anywhere after the first Data phase to cause the PCI 9056 to assert a new Address cycle (TS#). The BI# input acknowledges the current Data transfer and requests that a new Address cycle be asserted (TS#), for the next Data transfer. If the Bterm Mode bit is enabled, the PCI 9056 de-asserts BURST# only if its Read FIFO is full, its Write FIFO is empty, or if a transfer is complete. If the transfer starts on a non-Qword-aligned address, the PCI 9056 single cycles the data until the next Qword-aligned address and bursts forever the remainder of the data.

The PCI 9056 supports the BDIP# signal for continuous bursts greater than four Lwords, which differs from MPC850 and MPC860 protocol. When Bterm and Burst functions are enabled for Direct Slave and/or DMA transactions, and Slow Terminate mode is enabled for DMA, the PCI 9056 asserts the BDIP# signal low until the last Burst Data transfer. On the last Data transfer, the PCI 9056 de-asserts BDIP#, indicating the last transfer of the Burst transaction.

During Burst Forever Write transactions, the PCI 9056 passes all bytes from the PCI Bus to the Local Bus, if C/BE# begins to toggle on the nonquad-aligned address by keeping TSIZ[0:1] at a constant value of 0 and issues TS# for the toggled address. However, if C/BE# toggles on the Qword-aligned address, the PCI 9056 begins the Local Bus Burst and toggles TSIZ[0:1], along with TS#, for all data that follows when a burst resumes. It is recommended to keep all bytes enabled during a PCI Write Burst transaction.

2.2.5.3 Partial Lword Accesses

Lword accesses, in which not all byte enables are asserted, are broken into single Cycle accesses. Burst start addresses can be any Qword boundary. The PCI 9056 first performs a single cycle, if the Burst Start Address in a Direct Slave or DMA transfer is not aligned to a Qword or Lword boundary. It then starts to burst on the Qword boundary if there is remaining data that is not a whole Lword during DMA (*for example*, it results in a single cycle at the end).

2.2.6 Local Bus Read Accesses

For all single cycle Local Bus Read accesses, the PCI 9056 reads only bytes corresponding to byte enables requested by the Direct Master. For all Burst Read cycles, the PCI 9056 passes all the bytes and can be programmed to:

- Prefetch
- Perform Read Ahead mode
- Generate internal wait states
- Enable external wait control (TA# input)
- Enable type of Burst mode to perform

2.2.7 Local Bus Write Accesses

For Local Bus writes, only bytes specified by a PCI Bus master or the PCI 9056 DMA controller are written.

2.2.8 Direct Slave Accesses to 8- or 16-Bit Local Bus

Direct Slave PCI accesses to an 8- or 16-bit Local Bus results in the PCI Bus Lword being broken into multiple Local Bus transfers. For each transfer, byte enables are encoded to provide the Transfer Size bits (TSIZ[0:1]).

2.2.9 Local Bus Data Parity

Generation or use of Local Bus data parity is optional. Signals on the data parity pins do not affect operation of the PCI 9056. The PCI Bus parity checking and generation is independent of the Local Bus parity checking and generation. PCI Bus parity checking may result in assertion of PERR#, a PCI Bus system error (SERR#), or other means of PCI Bus transfer termination as a result of the parity error on the PCI data address, command code, and byte enables. The Local Bus Parity Check is passive and only provides parity information to the Local processor during Direct Master, Direct Slave, and DMA transfers.

There is one data parity pin for each byte lane of the PCI 9056 data bus (DP[0:3]). "Even data parity" is

asserted for each lane during Local Bus reads from the PCI 9056 and during PCI 9056 Master writes to the Local Bus.

Even data parity is checked during Local Bus writes to the PCI 9056 and during PCI 9056 reads from the Local Bus. Parity is checked for each byte lane with an asserted byte enable. If a parity error is detected, TEA# is asserted in the Clock cycle following the data being checked.

Parity is checked for Direct Slave reads, Direct Master writes, and DMA Local Bus reads. The PCI 9056 sets a status bit and asserts an interrupt (TEA#) in the clock cycle following data being checked if a parity error is detected. However, the Data Parity Error Status bit and interrupt are never set or asserted unless the TA# signal is active and asserted low. This applies only when the TA# signal is disabled in the PCI 9056 register. A workaround for this is to disable the TA# Enable bit and externally pull TA# low.

2.3 BIG ENDIAN/LITTLE ENDIAN

2.3.1 PCI Bus Little Endian Mode

PCI Bus is a Little Endian bus (*that is*, the address is invariant and data is Lword-aligned to the lowermost byte lane).

Table 2-9. PCI Bus Little Endian Byte Lanes

Byte Number	Byte Lane
0	AD[7:0]
1	AD[15:8]
2	AD[23:16]
3	AD[31:24]

2.3.2 Local Bus Big/Little Endian Mode

The PCI 9056 Local Bus can be programmed to operate in Big or Little Endian mode.

Table 2-10. Byte Number and Lane Cross-Reference

Byte Number		Byte Lane
Big Endian	Little Endian	
3	0	LD[24:31]
2	1	LD[16:23]
1	2	LD[8:15]
0	3	LD[0:7]

Table 2-11. Big/Little Endian Program Mode

BIGEND# Pin	BIGEND Register (1=Big, 0=Little)	Endian Mode
0	0	Big
0	1	Big
1	0	Little
1	1	Big

Table 2-12 lists register bits associated with the following cycles.

Table 2-12. Cycle Reference

Cycle	Register Bits
Local access to the Configuration registers	BIGEND[0]
Direct Master, Memory, and I/O	BIGEND[1]
Direct Slave	BIGEND[2], Space 0, and BIGEND[3], Expansion ROM

In Big Endian mode, the PCI 9056 transposes data byte lanes. Data is transferred as listed in Table 2-13 through Table 2-17.

2.3.2.1 32-Bit Local Bus—Big Endian Mode

Data is Lword-aligned to the uppermost byte lane (Data Invariance).

Table 2-13. Upper Lword Lane Transfer—32-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 3 appears on Local Data [24:31]
	Byte 2 appears on Local Data [16:23]
	Byte 1 appears on Local Data [8:15]
	Byte 0 appears on Local Data [0:7]

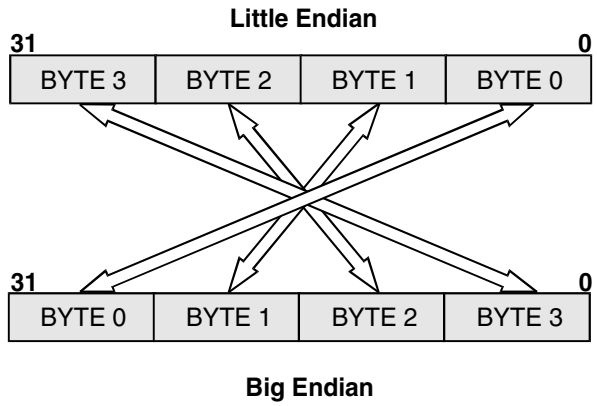


Figure 2-2. Big/Little Endian—32-Bit Local Bus

2.3.2.2 16-Bit Local Bus—Big Endian Mode

For a 16-bit Local Bus, the PCI 9056 can be programmed to use the upper or lower word lanes.

Table 2-14. Upper Word Lane Transfer—16-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [24:31]
	Byte 1 appears on Local Data [16:23]
Second transfer	Byte 2 appears on Local Data [24:31]
	Byte 3 appears on Local Data [16:23]

Table 2-15. Lower Word Lane Transfer—16-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [8:15]
	Byte 1 appears on Local Data [0:7]
Second transfer	Byte 2 appears on Local Data [8:15]
	Byte 3 appears on Local Data [0:7]

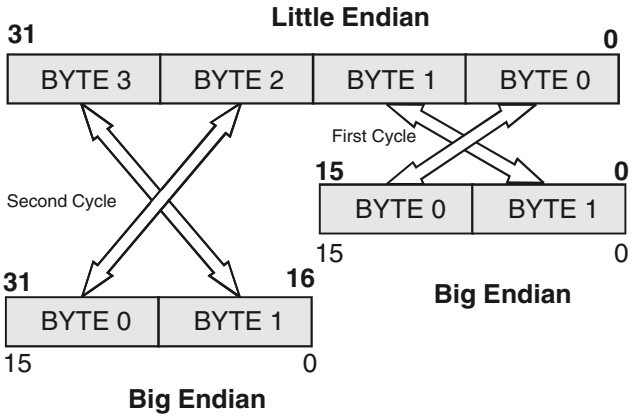


Figure 2-3. Big/Little Endian—16-Bit Local Bus

2.3.2.3 8-Bit Local Bus—Big Endian Mode

For an 8-bit Local Bus, the PCI 9056 can be programmed to use upper or lower byte lanes.

Table 2-16. Upper Byte Lane Transfer—8-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [24:31]
Second transfer	Byte 1 appears on Local Data [24:31]
Third transfer	Byte 2 appears on Local Data [24:31]
Fourth transfer	Byte 3 appears on Local Data [24:31]

Table 2-17. Lower Byte Lane Transfer—8-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [0:7]
Second transfer	Byte 1 appears on Local Data [0:7]
Third transfer	Byte 2 appears on Local Data [0:7]
Fourth transfer	Byte 3 appears on Local Data [0:7]

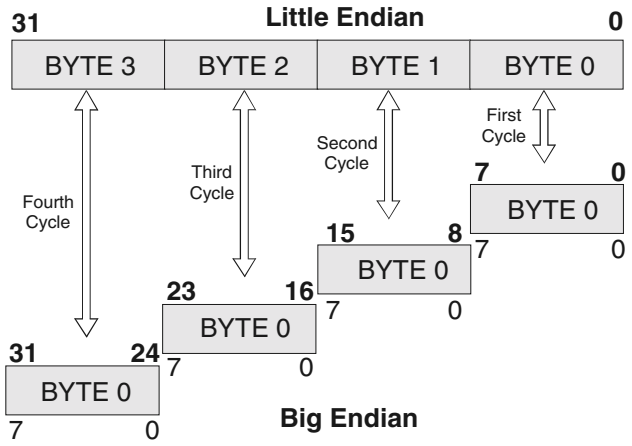


Figure 2-4. Big/Little Endian—8-Bit Local Bus

2.3.2.4 Local Bus Big/Little Endian Mode Accesses

For each of the following transfer types, the PCI 9056 Local Bus can be independently programmed to operate in Little Endian or Big Endian mode:

- Local Bus accesses to the PCI 9056 Configuration registers
- Direct Slave PCI accesses to Local Address Space 0
- Direct Slave PCI accesses to Local Address Space 1
- Direct Slave PCI accesses to the Expansion ROM
- DMA Channel 0 accesses to the Local Bus
- DMA Channel 1 accesses to the Local Bus
- Direct Master accesses to the PCI Bus

For Local Bus accesses to the Internal Configuration registers and Direct Master accesses, use BIGEND# to dynamically change the Endian mode.

Notes: *The PCI Bus is always Little Endian.*
Only byte lanes are swapped, not individual bits.

2.4 SERIAL EEPROM

Functional operation described can be modified through the PCI 9056 programmable internal registers.

2.4.1 Vendor and Device ID Registers

Three Vendor and Device ID registers are supported:

- **PCIIDR**—Contains normal Device and Vendor IDs. Can be loaded from the serial EEPROM or by the Local processor(s).
- **PCISVID**—Contains Subsystem and Subvendor IDs. Can be loaded from the serial EEPROM or by the Local processor(s).
- **PCIHIDR**—Contains hardwired PLX Vendor and Device IDs.

2.4.1.1 Serial EEPROM Initialization

During serial EEPROM initialization, the PCI 9056 responds to Direct Slave accesses with a Retry. During serial EEPROM initialization, the PCI 9056 responds to a Local processor access by delaying acknowledgement of the cycle (TA#).

2.4.1.2 Local Initialization

Refer to the document, *PCI 9056 Blue Book*

***Revision 0.91 Correction,* for the corrected version of this section.**

2.4.2 Serial EEPROM Operation

After reset, the PCI 9056 attempts to read the serial EEPROM to determine its presence. An active Start bit set to 0 indicates a serial EEPROM is present. The PCI 9056 supports 2K bit (FM93CS56L or compatible) or 4K bit (FM93CS66L or compatible) devices. (Refer to manufacturer's data sheet for the particular serial EEPROM being used.) The first Lword is then checked to verify that the serial EEPROM is programmed. If the first Lword (33 bits) is all ones (1), a blank serial EEPROM is present. If the first Lword (33 bits) is all zeros, no serial EEPROM is present. For both conditions, the PCI 9056 reverts to the default values. (Refer to Table 2-18.) The Programmed Serial EEPROM Present bit is set (CNTRL[28]=1) if the serial EEPROM is programmed (real or random data if a serial EEPROM is detected).

The 3.3V serial EEPROM clock (EESK) is derived from the PCI clock. The PCI 9056 generates the serial EEPROM clock by internally dividing the PCI clock by 268. For a 66.6 MHz PCI Bus, EESK is 248.7 kHz; for a 33.3 MHz PCI Bus, EESK is 124.4 kHz.

The serial EEPROM can be read or written from the PCI or Local Buses. The Serial EEPROM Control Register bits (CNTRL[31, 27:24]) control the PCI 9056 pins that enable reading or writing of serial EEPROM data bits. (Refer to manufacturer's data sheet for the particular serial EEPROM being used.)

The PCI 9056 provides the ability to manually access the serial EEPROM interface by using CNTRL[31,

27:24] (EESK, EECS, and EEDI/EEDO, controlled by software). Bit 24 is used to generate EESK (clock), bit 25 controls the chip select, and bit 31 enables the EEDO input buffer. Bit 27, when read, returns the value of EEDO.

Setting bits [31, 25, 24] to 1 causes the EEDI output to go high. A pull-up resistor is required on EEDO to go high when bit 31 is set. When reading the serial EEPROM, bit 31 must be set to 1.

To perform the read, the basic approach is to set the EECS and EEDO bits (bits 25 and 31, respectively) to the desired level and then toggle EESK high and low until done. *For example*, reading the serial EEPROM at location 0 involves the following steps:

1. Clear EESK, EEDO and EECS bits.
2. Set EECS high.
3. Toggle EESK high, then low.
4. Set EEDO bit high (start bit).
5. Toggle EESK high, then low.
6. Repeat step 5.
7. Clear EEDO.
8. Toggle EESK high, then low.
9. Toggle EESK bit high, then low 8 times (clock in serial EEPROM Address 0).
10. Set bit 31 to float the EEDO pin for reading.
11. Toggle EESK high, then low 16 times (clock in one word from serial EEPROM).
12. After each clock pulse, read bit 27 and save.
13. Clear EECS bit.
14. Toggle EESK high, then low.
15. Read is now complete.

The serial EEPROM can also be read or written, using the VPD function. (Refer to Section 10.)

The PCI 9056 has two serial EEPROM load options:

- **Long Load Mode**—Default. The PCI 9056 loads 17 Lwords from the serial EEPROM if the Extra Long Load from the Serial EEPROM bit is clear (LBRD0[25]=0)
- **Extra Long Load Mode**—The PCI 9056 loads 23 Lwords from the serial EEPROM if the Extra Long Load from the Serial EEPROM bit is set (LBRD0[25]=1) during a Long Load

Table 2-18. Serial EEPROM Guidelines

Local Processor	Serial EEPROM	System Boot Condition
None	None	The PCI 9056 uses default values. The EEDI/EEDO pin must be **pulled low**—a 1K ohm resistor is required (rather than pulled high, which is typically done for this pin). If the PCI 9056 detects all zeros, it reverts to default values.
None	Programmed	Boot with serial EEPROM values. The Local Init Status bit (LMISC1[2]) must be set by the serial EEPROM. A 3K to 10K ohm pull-up resistor is required on EDDI/EEDO.
None	Blank	The PCI 9056 detects a blank device and reverts to default values. A 3K to 10K ohm pull-up resistor is required on EDDI/EEDO.
Present	None	**Refer to the document, *PCI 9056 Blue Book Revision 0.91 Correction,* for the corrected version of this table entry.**
Present	Programmed	Load serial EEPROM, but the Local processor can reprogram the PCI 9056. Either the Local processor or the serial EEPROM must set the Local Init Status bit (LMISC1[2]=done). A 3K to 10K ohm pull-up resistor is required on EDDI/EEDO.
Present	Blank	The PCI 9056 detects a blank serial EEPROM and reverts to default values. A 3K to 10K ohm pull-up resistor is required on EDDI/EEDO. ***Notes:*** *In some systems, the Local processor may be overly late to reconfigure the PCI 9056 registers before the BIOS configures them.* *The serial EEPROM can be programmed through the PCI 9056 after the system boots in this condition.*

Note: *If the serial EEPROM is missing and a Local Processor is present with blank Flash, the condition None/None (as seen in Table 2-18) applies, until the Processor's Flash is programmed.*

2.4.2.1 Long Serial EEPROM Load

The registers listed in Table 2-19 are loaded from the serial EEPROM after a reset is de-asserted if the Extra Long Load from Serial EEPROM bit is not set (LBRD0[25]=0). The serial EEPROM is organized in words (16 bit). The PCI 9056 first loads the Most Significant Word bits (MSW[31:16]), starting from the Most Significant bit (MSB[31]). The PCI 9056 then loads the Least Significant Word bits (LSW[15:0]), starting again from the Most Significant bit (MSB[15]). Therefore, the PCI 9056 loads the Device ID, Vendor ID, Class Code, and so forth.

The serial EEPROM values can be programmed using an EEPROM programmer. The values can also be programmed using the PCI 9056 VPD function (refer to Section 10) or through the Serial EEPROM Control register (CNTRL).

The CNTRL register allows programming of the serial EEPROM, one bit at a time. To read back the value from the serial EEPROM, the CNTRL[27] bit (refer to Section 2.4.2) or the VPD function should be utilized. With full utilization of VPD, the designer can perform reads and writes from/to the serial EEPROM, 32 bits at a time. Values should be programmed in the order listed in Table 2-19. The 34, 16-bit words listed in the table are stored sequentially in the serial EEPROM.

Table 2-19. Long Serial EEPROM Load Registers

Serial EEPROM Offset	Description	Register Bits Affected
0h	Device ID	PCIIDR[31:16]
2h	Vendor ID	PCIIDR[15:0]
4h	Class Code	PCICCR[23:8]
6h	Class Code / Revision	PCICCR[7:0] / PCIREV[7:0]
8h	Maximum Latency / Minimum Grant	PCIMLR[7:0] / PCIMGR[7:0]
Ah	Interrupt Pin / Interrupt Line Routing	PCIIPR[7:0] / PCIILR[7:0]
Ch	MSW of Mailbox 0 (User Defined)	MBOX0[31:16]
Eh	LSW of Mailbox 0 (User Defined)	MBOX0[15:0]
10h	MSW of Mailbox 1 (User Defined)	MBOX1[31:16]
12h	LSW of Mailbox 1 (User Defined)	MBOX1[15:0]
14h	MSW of Range for PCI-to-Local Address Space 0	LAS0RR[31:16]
16h	LSW of Range for PCI-to-Local Address Space 0	LAS0RR[15:0]
18h	MSW of Local Base Address (Remap) for PCI-to-Local Address Space 0	LAS0BA[31:16]
1Ah	LSW of Local Base Address (Remap) for PCI-to-Local Address Space 0	LAS0BA[15:0]
1Ch	MSW of Mode/DMA Arbitration Register	MARBR[31:16]
1Eh	LSW of Mode/DMA Arbitration Register	MARBR[15:0]
20h	MSW of Local Miscellaneous Control Register 2 / MSW of Serial EEPROM Write-Protected Address	LMISC2[7:0] / PROT_AREA[7:0]
22h	LSW of Local Miscellaneous Control Register 1/ LSW of Local Bus Big/Little Endian Descriptor Register	LMISC1[7:0] / BIGEND[7:0]
24h	MSW of Range for PCI-to-Local Expansion ROM	EROMRR[31:16]
26h	LSW of Range for PCI-to-Local Expansion ROM	EROMRR[15:0]
28h	MSW of Local Base Address (Remap) for PCI-to-Local Expansion ROM	EROMBA[31:16]
2Ah	LSW of Local Base Address (Remap) for PCI-to-Local Expansion ROM	EROMBA[15:0]
2Ch	MSW of Bus Region Descriptors for PCI-to-Local Accesses	LBRD0[31:16]
2Eh	LSW of Bus Region Descriptors for PCI-to-Local Accesses	LBRD0[15:0]
30h	MSW of Range for Direct Master-to-PCI	DMRR[31:16]
32h	LSW of Range for Direct Master-to-PCI	DMRR[15:0]
34h	MSW of Local Base Address for Direct Master-to-PCI Memory	DMLBAM[31:16]
36h	LSW of Local Base Address for Direct Master-to-PCI Memory	DMLBAM[15:0]
38h	MSW of Local Bus Address for Direct Master-to-PCI I/O Configuration	DMLBAI[31:16]
3Ah	LSW of Local Bus Address for Direct Master-to-PCI I/O Configuration	DMLBAI[15:0]
3Ch	MSW of PCI Base Address (Remap) for Direct Master-to-PCI	DMPBAM[31:16]
3Eh	LSW of PCI Base Address (Remap) for Direct Master-to-PCI	DMPBAM[15:0]
40h	MSW of PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration	DMCFGA[31:16]
42h	LSW of PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration	DMCFGA[15:0]

2.4.2.2 Extra Long Serial EEPROM Load

The registers listed in Table 2-19 and Table 2-20 are loaded from serial EEPROM after a reset is de-asserted if the Extra Long Load from Serial EEPROM bit is set (LBRD0[25]=1). The serial EEPROM is organized in words (16 bit). The PCI 9056 first loads the Most Significant Word bits (MSW[31:16]), starting from the Most Significant bit (MSB[31]). It then loads the Least Significant Word bits (LSW[15:0]), restarting from the Most Significant bit (MSB[15]). Therefore, the PCI 9056 loads Device ID, Vendor ID, class code, and so forth.

The serial EEPROM values can be programmed using a Data I/O programmer. The values can also be programmed using the PCI 9056 VPD function or through the Serial EEPROM Control register (CNTRL).

Values should be programmed in the order listed in Table 2-20. The 46 16-bit words listed in Table 2-19 and Table 2-20 should be stored sequentially in the serial EEPROM.

Table 2-20. Extra Long Serial EEPROM Load Registers

Serial EEPROM Offset	Description	Register Bits Affected
44h	Subsystem ID	PCISID[15:0]
46h	Subsystem Vendor ID	PCISVID[15:0]
48h	MSW of Range for PCI-to-Local Address Space 1 (1 MB)	LAS1RR[31:16]
4Ah	LSW of Range for PCI-to-Local Address Space 1 (1 MB)	LAS1RR[15:0]
4Ch	MSW of Local Base Address (Remap) for PCI-to-Local Address Space 1	LAS1BA[31:16]
4Eh	LSW of Local Base Address (Remap) for PCI-to-Local Address Space 1	LAS1BA[15:0]
50h	MSW of Bus Region Descriptors (Space 1) for PCI-to-Local Accesses	LBRD1[31:16]
52h	LSW of Bus Region Descriptors (Space 1) for PCI-to-Local Accesses	LBRD1[15:0]
54h	MSW of Hot Swap Control/Status	***Reserved***
56h	LSW of Hot Swap Control / Hot Swap Next Capability Pointer	HS_NEXT[7:0] / HS_CNTL[7:0]
58h	PCI Arbiter Control	PCIARB[3:0]
5Ah	***Reserved***	***Reserved***

2.4.2.3 New Capabilities Function Support

The New Capabilities Function Support includes PCI Power Management, Hot Swap, and VPD features, as listed in Table 2-21.

Table 2-21. New Capabilities Function Support Features

New Capability Function	PCI Register Offset Location
First (Power Management)	40h, if the New Capabilities Function Support bit (PCISR[4]) is enabled (PCISR[4] is enabled, by default).
Second (Hot Swap)	48h, which is pointed to from PMNEXT[7:0].
Third (VPD)	4Ch, which is pointed to from HS_NEXT[7:0]. Because PVPD_NEXT[7:0] defaults to zero (0), this indicates that VPD is the last New Capability Function Support feature of the PCI 9056.

2.4.2.4 Recommended Serial EEPROMs

The PCI 9056 is designed to use either a 2K bit (FM93CS56L or compatible) or 4K bit (FM93CS66L or compatible) device.

Note: *The PCI 9056 does not support serial EEPROMs that do not support sequential reads (such as the FM93C56L).*

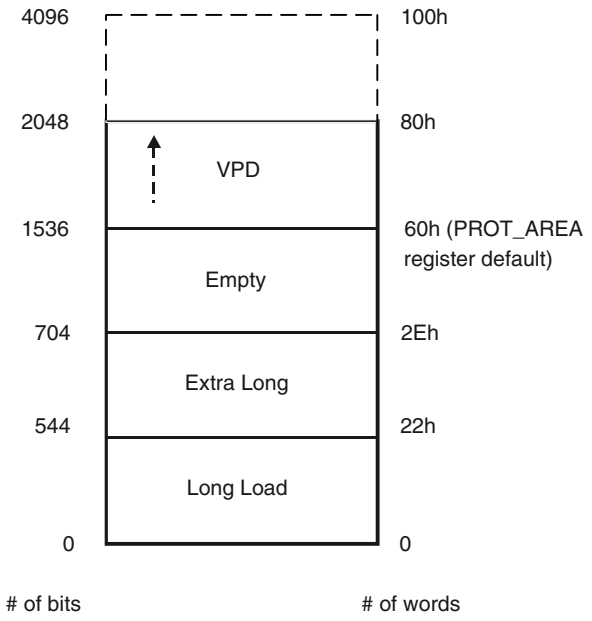


Figure 2-5. Serial EEPROM Memory Map

2.4.3 Internal Register Access

The PCI 9056 provides several internal registers, which allow for maximum flexibility in the bus interface design and performance. These registers are accessible from the PCI and Local Buses (refer to Figure 2-6) and include the following:

- PCI and Local Configuration registers
- DMA registers
- Mailbox registers
- PCI-to-Local and Local-to-PCI Doorbell registers
- Messaging Queue registers (I_2O)
- Power Management registers
- Hot Swap registers
- VPD registers

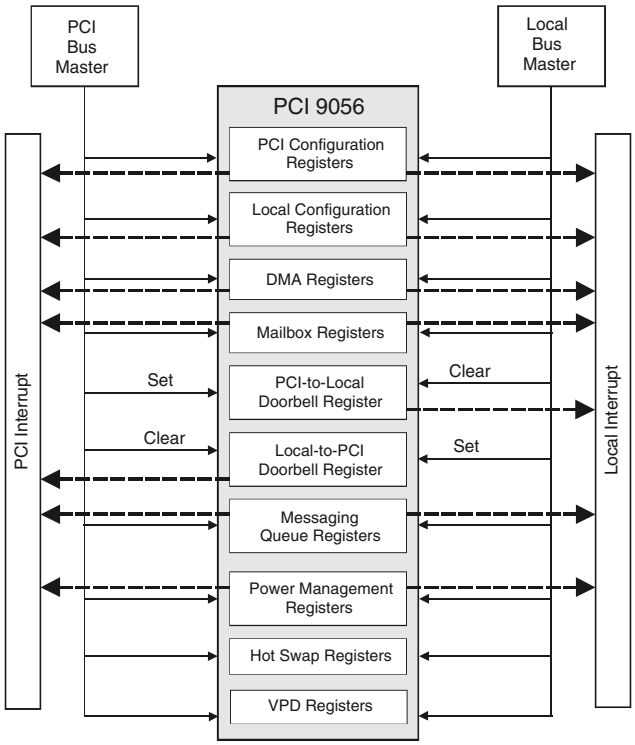


Figure 2-6. PCI 9056 Internal Register Access

2.4.3.1 PCI Bus Access to Internal Registers

The PCI 9056 PCI Configuration registers can be accessed from the PCI Bus with a Configuration Type 0 cycle.

All other PCI 9056 internal registers can be accessed by a Memory cycle, with the PCI Bus address that matches the base address specified in PCI Base Address 0 (PCIBAR0[31:8]) for the PCI 9056 Memory-Mapped Configuration register. These registers can also be accessed by an I/O cycle, with the PCI Bus address matching the base address specified in PCI Base Address 1 for the PCI 9056 I/O-Mapped Configuration register (PCIBAR1).

All PCI Read or Write accesses to the PCI 9056 registers can be Byte, Word, or Lword accesses. All PCI Memory accesses to the PCI 9056 registers can be Burst or Non-Burst accesses. The PCI 9056 responds with a PCI disconnect for all Burst I/O accesses (PCIBAR1[31:8]) to the PCI 9056 Internal registers.

2.4.3.2 Local Bus Access to Internal Registers

The Local processor can access all PCI 9056 internal registers through an external chip select. The PCI 9056 responds to a Local Bus access when the PCI 9056 Configuration Chip Select input (CCS#) is asserted low. Figure 2-7 illustrates how the Configuration Chip Select logic works.

Notes: *CCS# must be decoded while TS# is low.*
Accesses must be for a 32-bit non-pipelined bus.

Local Read or Write accesses to the PCI 9056 internal registers can be Byte, Word, or Lword accesses. The Local Bus width must be 32-bit to access the internal registers. Eight and 16-bit data buses require external latches to form a 32-bit data path for Local Bus access to internal registers. Local accesses to the PCI 9056 internal registers can be Burst or Non-Burst accesses.

The PCI 9056 TA# signal indicates that Data transfer is complete.

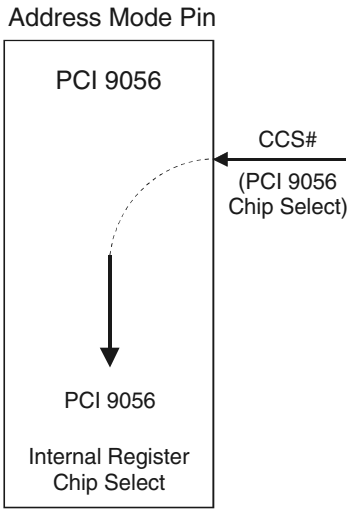
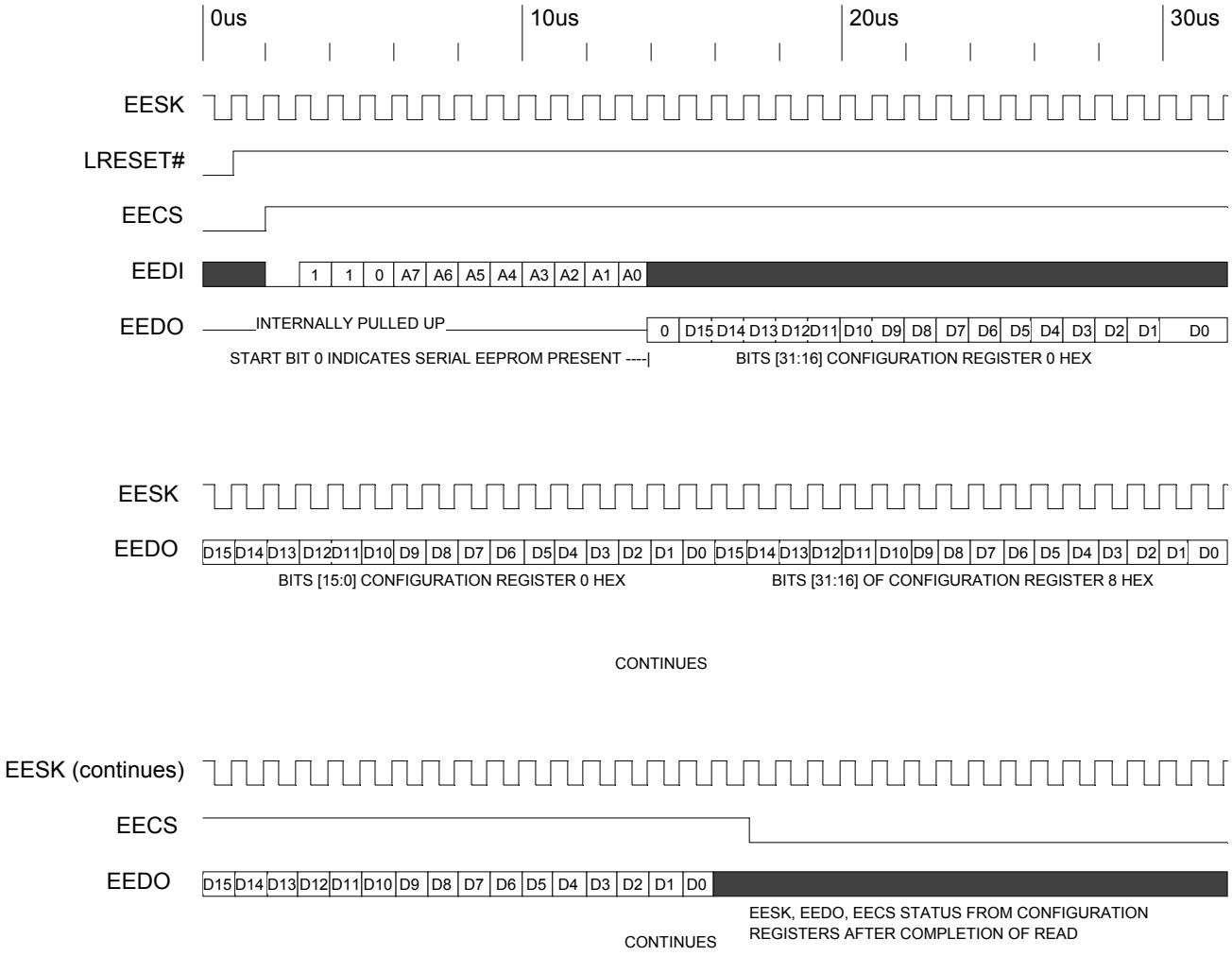


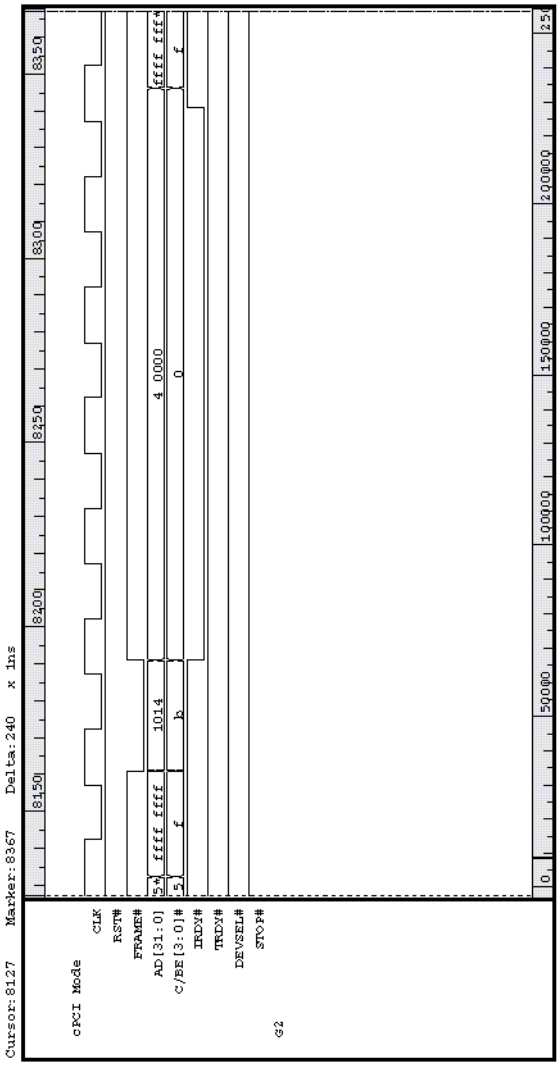
Figure 2-7. Address Decode Mode

2.4.4 Serial EEPROM and Configuration Initialization Timing Diagrams

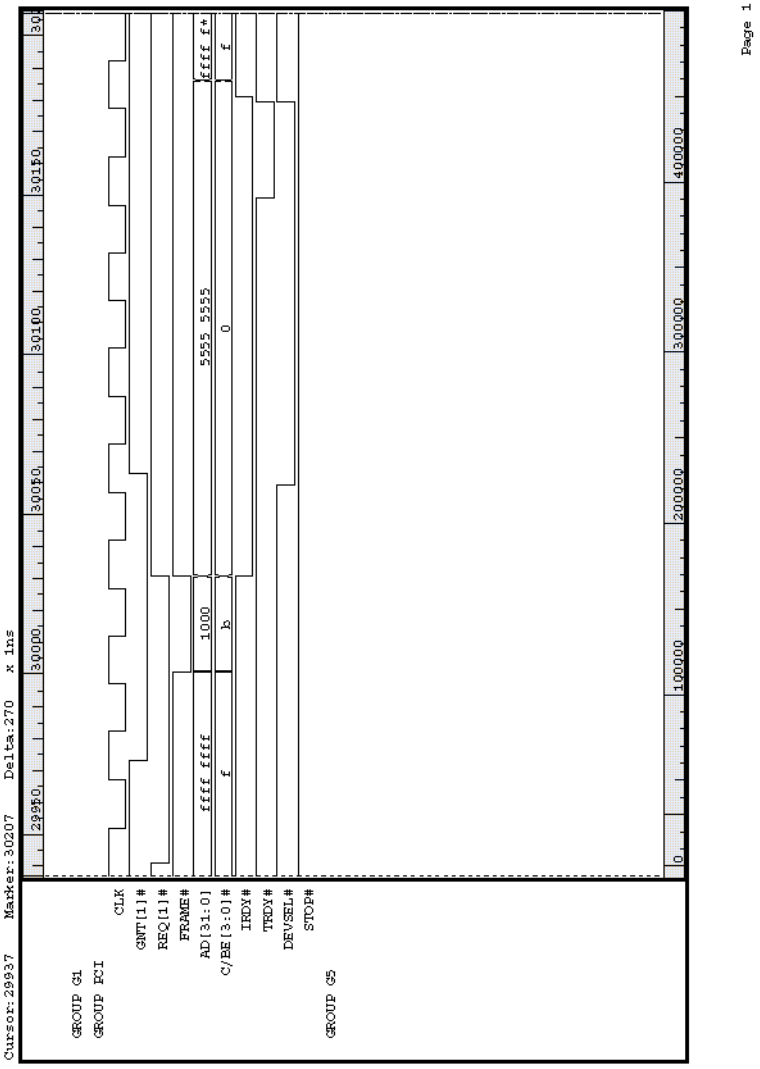
Note: *In the timing diagrams that follow, the "_" symbol at the end of the signal names represents the "#" symbol.*



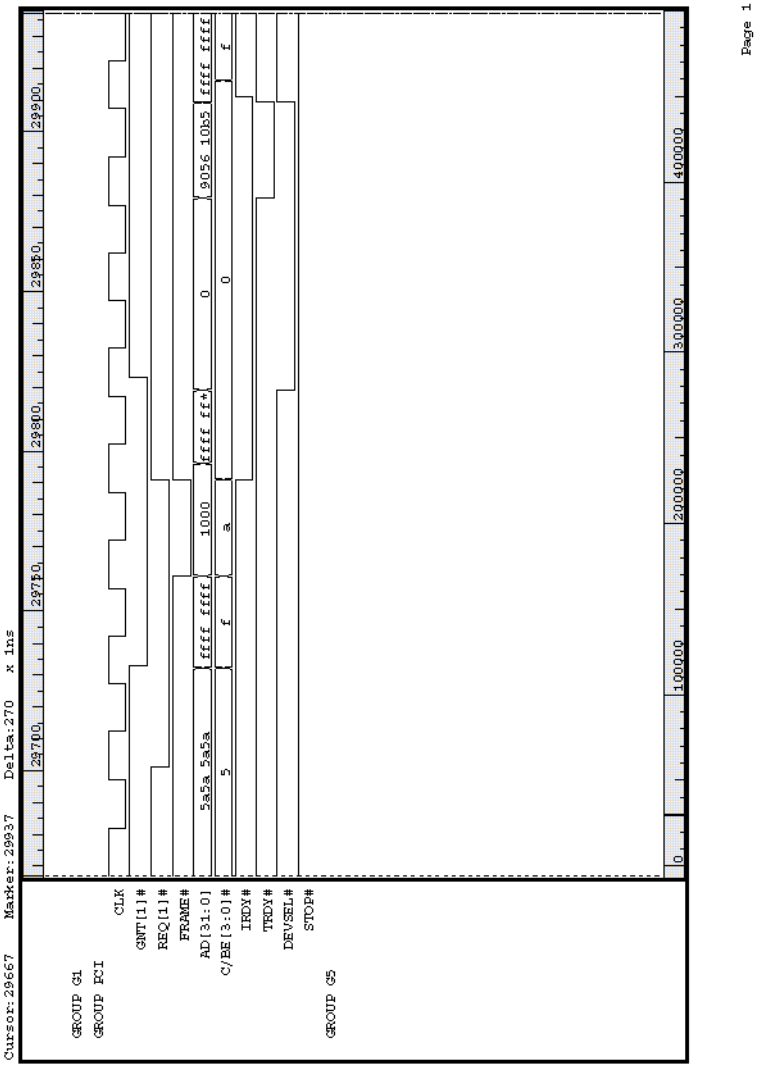
Timing Diagram 2-1. Initialization from Serial EEPROM (2K or 4K Bit)



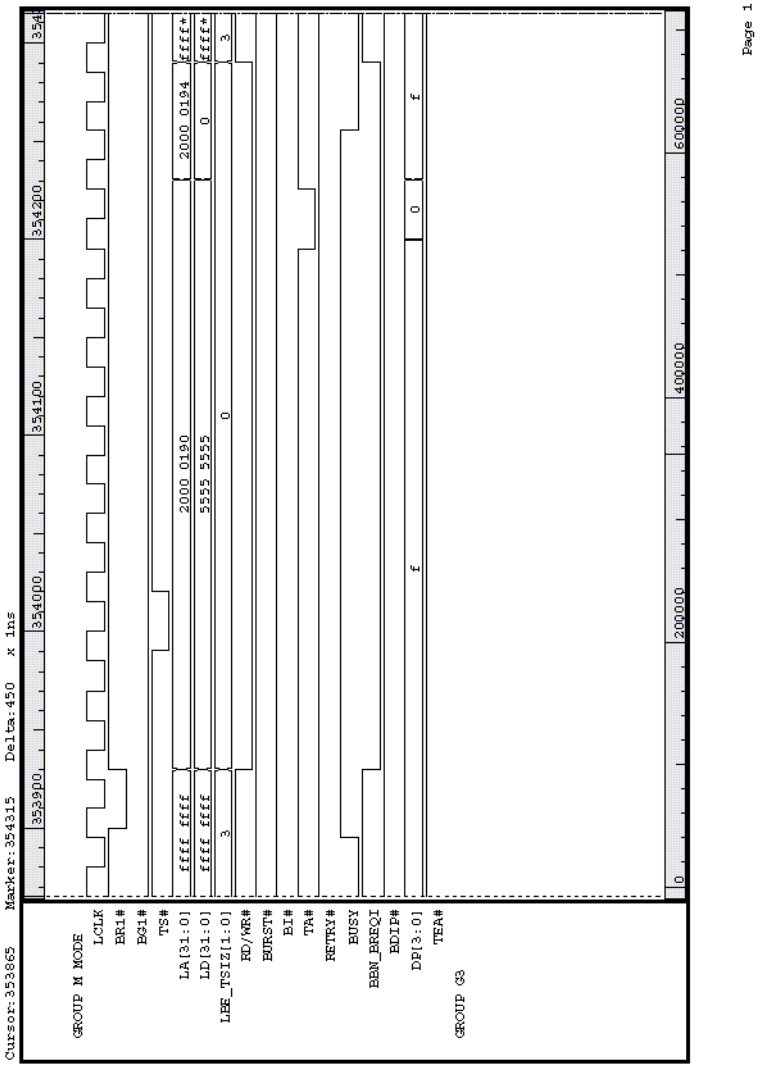
Timing Diagram 2-2. Local Interrupt Asserting PCI Interrupt



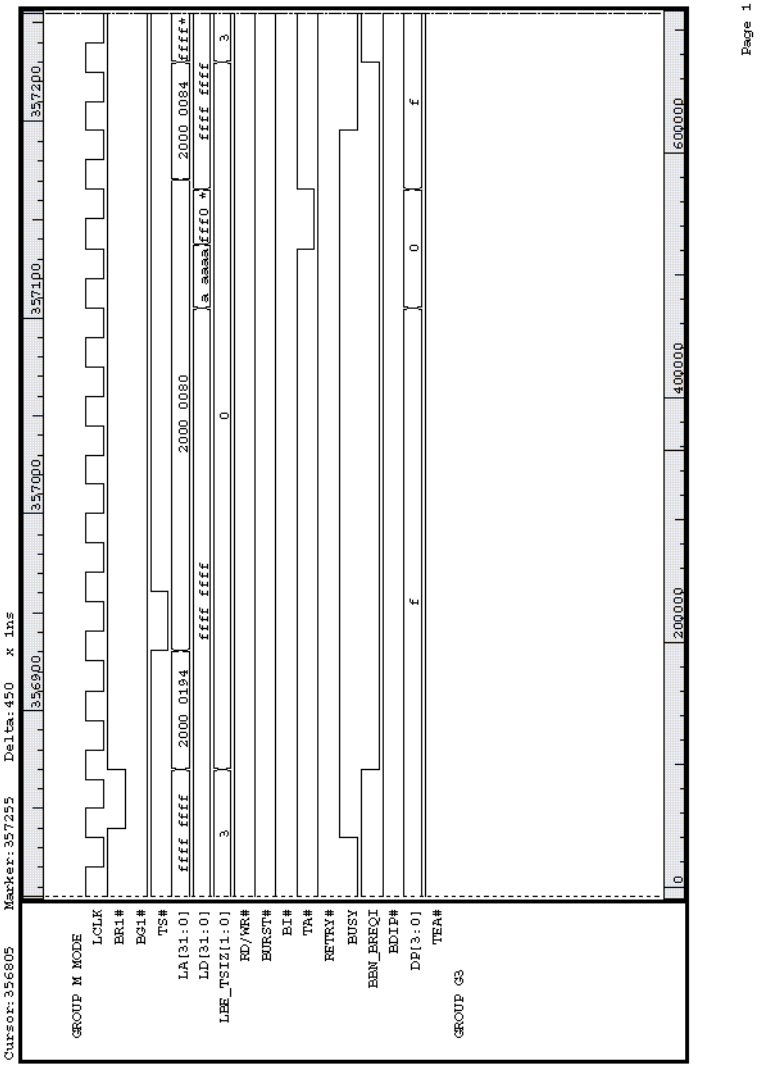
Timing Diagram 2-3. PCI Configuration Write to PCI Configuration Register



Timing Diagram 2-4. PCI Configuration Read to PCI Configuration Register



Timing Diagram 2-5. Local Configuration Write to Configuration Register (M Mode)



Timing Diagram 2-6. Local Configuration Read from Configuration Register (M Mode)

3 M MODE FUNCTIONAL DESCRIPTION

The functional operation described in this chapter can be modified through the PCI 9056 programmable internal registers.

3.1 RESET OPERATION

3.1.1 Adapter Mode

3.1.1.1 PCI Bus Input RST#

The PCI Bus RST# input pin is a PCI Host reset. It causes all PCI Bus outputs to float, resets the entire PCI 9056 and causes the Local LRESET# signal to be asserted.

3.1.1.2 Software Reset

A Host on the PCI Bus can set the PCI Adapter Software Reset bit (CNTRL[30]=1) to reset the PCI 9056 and assert LRESET# output. All Local Configuration registers are reset; however, the PCI Configuration DMA and Shared Runtime registers and the Local Init Status bit (LMISC1[2]) are not reset. When the Software Reset bit (CNTRL[30]) is set, the PCI 9056 responds to PCI accesses, but not to Local Bus accesses. The PCI 9056 remains in this reset condition until the PCI Host clears the bit. The serial EEPROM is reloaded, if the Reload Configuration Registers bit is set (CNTRL[29]=1).

Note: *The Local Bus cannot clear this reset bit because the Local Bus is in a reset state, even if the Local processor does not use LRESET# to reset.*

3.1.1.3 Power Management Reset

When the power management reset is asserted (transition from D_3 to any other state), the PCI 9056 resets as if a PCI reset was asserted. (Refer to Section 8, “PCI Power Management.”)

3.1.2 Host Mode

3.1.2.1 PCI Reset

The PCI Bus RST# output is driven when the Local LRESET# signal is asserted, the Software Reset bit is set (CNTRL[30]=1), or the PCI 9056 initiates an external reset.

3.1.2.2 Local LRESET#

When the Local LRESET# pin is asserted by an external source, the Local Bus interface circuitry, the configuration registers, and the PCI 9056 are reset. The PCI 9056 drives the Local LRESET# pin after it detects a reset for 62 clocks.

3.1.2.3 Software Reset

When the Software Reset bit is set (CNTRL[30]=1), the following occurs:

- PCI Master logic is held reset
- PCI 9056 PCI Configuration registers held in reset
- FIFOs are reset
- PCI RST# pin is asserted

Only the PCI Configuration registers are in reset. A software reset can only be cleared from another Host on the Local Bus, and the PCI 9056 remains in this reset condition until a Local Host clears the bit.

Note: *The PCI Bus cannot clear this reset bit because the PCI Bus is in a reset state.*

3.1.2.4 Power Management Reset

Power Management reset is not applicable for Host mode.

3.2 PCI 9056 INITIALIZATION

The PCI 9056 Configuration registers can be programmed by an optional serial EEPROM and/or by a Local processor, as listed in Table 2-18, “Serial EEPROM Guidelines,” on page 2-9. The serial EEPROM can be reloaded by setting the Reload Configuration Registers bit (CNTRL[29]).

The PCI 9056 retries all PCI cycles until the Local Init Status bit is set to “done” (LMISC1[2]=1).

Note: *The PCI Host processor can also access Internal Configuration registers after the Local Init Status bit is set.*

If a PCI Host is present, the Master Enable, Memory Space, and I/O Space bits (PCICR[2:0], respectively) are programmed by that Host after initialization completes (LMISC1[2]=1).

3.3 RESPONSE TO FIFO FULL OR EMPTY

Table 3-1 lists the PCI 9056 response to full and empty FIFOs.

3.4 DIRECT DATA TRANSFER MODES

The PCI 9056 supports three direct transfer modes:

- **Direct Master**—Local CPU accesses PCI memory or I/O
- **Direct Slave**—PCI Master accesses Local memory or I/O
- **DMA**—PCI 9056 DMA controller reads/writes PCI memory to/from Local memory

3.4.1 Direct Master Operation (Local Master-to-Direct Slave)

The PCI 9056 supports a direct access to the PCI Bus by the Local processor or an intelligent controller. Master mode must be enabled in the PCI Command register. The following registers define Local-to-PCI accesses:

- Direct Master Memory and I/O Range (DMRR)
- Local Base Address for Direct Master-to-PCI Memory (DMLBAM)
- Local Base Address for Direct Master-to-PCI I/O and Configuration (DMLBAI)
- PCI Base Address (DMPBAM)
- Direct Master Configuration (DMCFGA)
- Direct Master PCI Dual Address Cycles (DMDAC)
- Master Enable (PCICR)
- PCI Command Code (CNTRL)

Table 3-1. Response to FIFO Full or Empty

Mode	Direction	FIFO	PCI Bus	Local Bus
Direct Master Write	Local-to-PCI	Full	Normal	De-assert TA#, RETRY#[1]
		Empty	De-assert REQ# (off the PCI Bus)	Normal
Direct Master Read	PCI-to-Local	Full	De-assert REQ# or throttle IRDY#[2]	Normal
		Empty	Normal	De-assert TA#
Direct Slave Write	PCI-to-Local	Full	Disconnect or throttle TRDY#[3]	Normal
		Empty	Normal	De-assert BB#[4]
Direct Slave Read	Local-to-PCI	Full	Normal	De-assert BB#[4]
		Empty	Throttle TRDY#[3]	Normal
DMA	Local-to-PCI	Full	Normal	De-assert BB#[4]
		Empty	De-assert REQ#	Normal
	PCI-to-Local	Full	De-assert REQ#	Normal
		Empty	Normal	De-assert BB#[4]

1. Issue RETRY# depends upon the Direct Master Write FIFO Almost Full RETRY# Output Enable bit (LMISC1[6]).
2. Throttle IRDY# depends upon the Direct Master PCI Read Mode bit (DMPBAM[4]).
3. Throttle TRDY4 depends upon the Direct Slave PCI Write Mode bit (LBRD0[27]).
4. BB# de-assert depends upon the Local Bus Direct Slave Release Bus Mode bit (MARBR[21]).

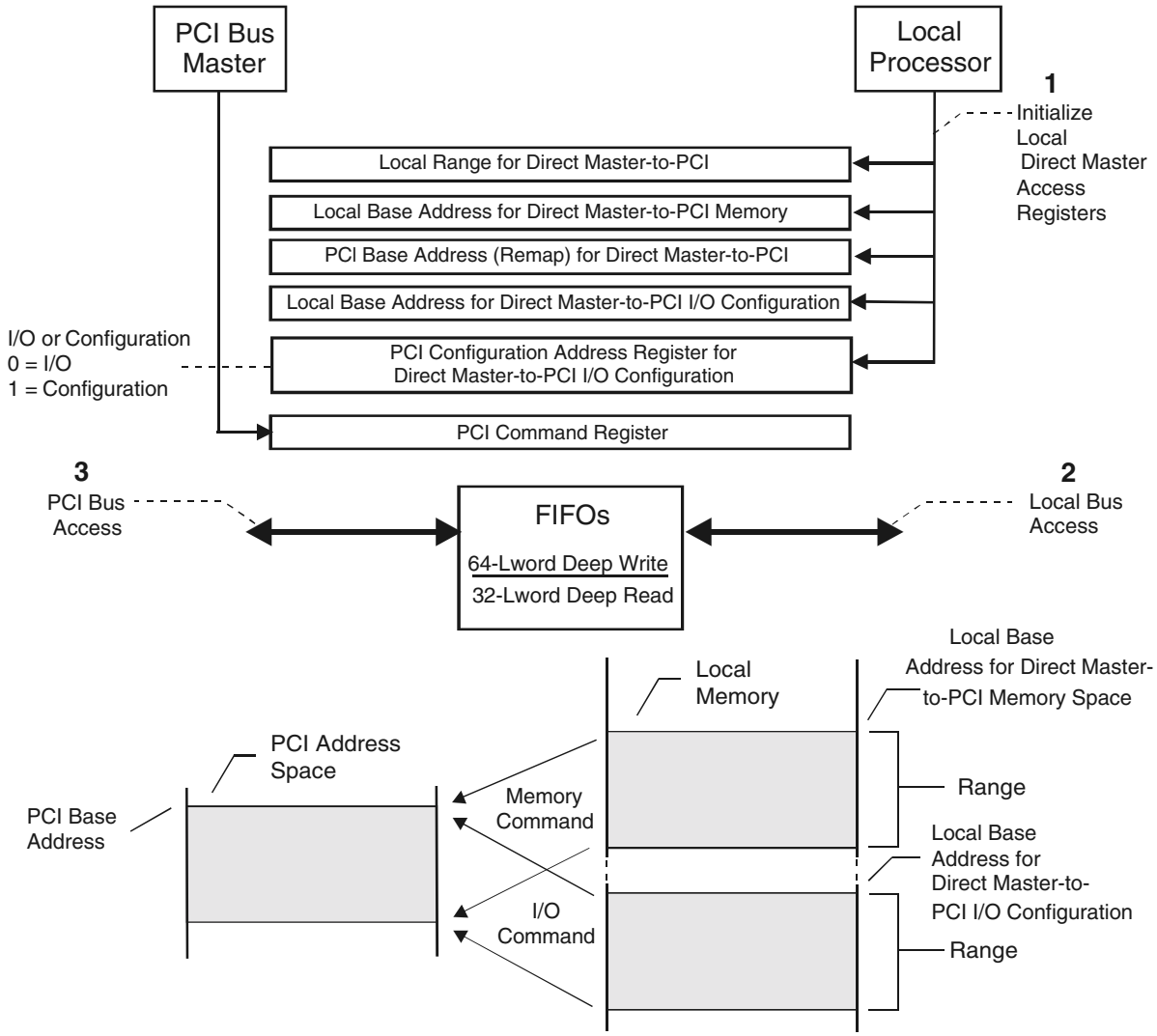


Figure 3-1. Direct Master Access to the PCI Bus

3.4.1.1 Direct Master Memory and I/O Decode

The Direct Master Range register and the Local Base Address register specifies the Local Address bits to use for decoding a Local-to-PCI access (Direct Master). The range of Memory or I/O space must be a power of 2 and the Range register value must be the 2's complement of the range value. In addition, the Local Base Address must be a multiple of the range value.

Any Local Master Address starting from the Direct Master Local Base Address (Memory or I/O) to the range value is recognized as a Direct Master access by the PCI 9056. All Direct Master cycles are then decoded as PCI Memory, I/O, or Configuration Type 0 or Type 1. Moreover, a Direct Master Memory or I/O cycle is remapped according to the Remap register value. The Remap Register value must be a multiple of the Direct Master Range value (not the Range register value).

The PCI 9056 can only accept Memory cycles from a Local processor. The Local Base Address and/or the range determine whether PCI Memory or PCI I/O transactions occur.

3.4.1.2 Direct Master FIFOs

For Direct Master Memory access to the PCI Bus, the PCI 9056 has a 64-Lword (256-byte) Write FIFO and a

32-Lword (128-byte) Read FIFO. The FIFOs enable the Local Bus to operate independent of the PCI Bus and allows high-performance bursting on the PCI and Local Buses. In a Direct Master write, the Local processor (Master) writes data to the PCI Bus (Slave). In a Direct Master read, the Local processor (Master) reads data from the PCI Bus (Slave). The FIFOs that function during a Direct Master write and read are illustrated in Figure 3-2 and Figure 3-3.

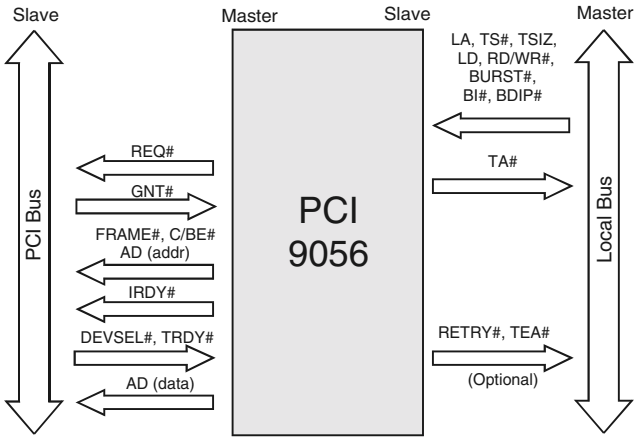


Figure 3-2. Direct Master Write

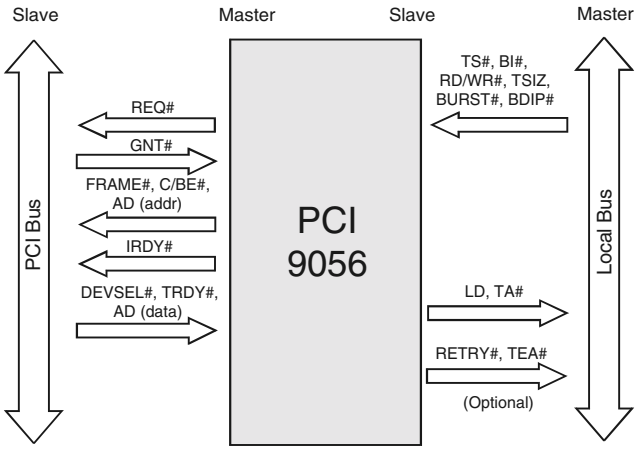


Figure 3-3. Direct Master Read

Note: *Figures 3-2 and 3-3 represent a sequence of Bus cycles.*

3.4.1.3 Direct Master Memory Access

The MPC850 or MPC860 transfers data through a single or burst Read/Write Memory transaction, or through SDMA channels to the PCI 9056 and PCI Bus. The MPC850 or MPC860 IDMA/SDMA accesses to the PCI 9056 appear as Direct Master operations. (Refer to Section 3.4.2 for further information.)

Transactions are initiated by the MPC850 or MPC860 (a Local Bus Master) when the memory address on the Local Bus matches the Memory space decoded for Direct Master operations. Upon a Local Bus Read, the PCI 9056 becomes a PCI Bus Master, arbitrates for the PCI Bus, and reads data from the PCI Slave device directly into the Direct Master Read FIFO. When sufficient data is placed into the FIFO, it asserts the Transfer Acknowledge (TA#) signal onto the Local Bus to indicate that the requested data is on the Local Bus.

The Local processor can read or write to PCI memory. The PCI 9056 converts the Local Read/Write access. The Local Address space starts from Direct Master Local Base Address up to the range. Remap (PCI Base Address) defines the PCI starting address.

The PCI 9056 supports single and Burst cycles performed by the MPC850 or MPC860 processor.

An MPC850 or MPC860 single cycle causes a single cycle PCI transaction. An MPC850 or MPC860 Burst cycle asserts a Burst cycle PCI transaction. Bursts are limited to 16 bytes (four Lwords) in the MPC850 or MPC860 bus protocol.

The PCI 9056 supports bursts beyond the 16-byte boundary (Continuous Burst) when the BDIP# input signal remains asserted beyond a 16-byte boundary by an external Local Bus Master. To finish, the continuing burst and external Master should de-assert the BDIP# signal on the last Data phase.

Writes—Upon a Local Bus Write, the Local Bus Master writes data to the Direct Master Write FIFO. When the first data is in the FIFO, the PCI 9056 becomes the PCI Bus Master, arbitrates for the PCI Bus, and writes data to the PCI Slave device. The PCI 9056 continues to accept writes and returns TA# until the Write FIFO is full. It then holds off TA# until space becomes available in the Write FIFO. A programmable Direct Master “almost full” status output is provided (MDREQ#/DMPAF). The PCI 9056 asserts RETRY# whenever the Direct Master Write FIFO is full, implying that the Local Master can relinquish the bus and finish the Write operation at a later time (LMISC1[6]).

MPC850 or MPC860 single cycle Write transactions result in PCI 9056 transfers of one Lword of data onto a 32-bit PCI Bus.

MPC850 or MPC860 Burst Cycle Write transactions of four Lwords result in PCI 9056 Burst transfers of four Lwords to a 32-bit PCI Bus.

A Local processor (MPC850 or MPC860), with no burst limitations and a Burst Cycle Write transaction of two Lwords, results in PCI 9056 burst transfers of two Lwords to a 32-bit PCI Bus.

Three Lword (or more) Burst cycles of any type result in the PCI 9056 bursting data onto the PCI Bus.

Reads—The PCI 9056 holds off TA# while gathering an Lword from the PCI Bus. Programmable Prefetch modes are available if prefetch is enabled to prefetch, 4, 8, 16, or continuous data until the Direct Master cycle ends. The Read cycle is terminated when the Local BDIP# input is de-asserted. Unused Read data is flushed from the FIFO.

The PCI 9056 does not prefetch Read PCI data for single cycle Direct Master reads (Local BURST# input is not asserted during the first Data phase). In this case, for the 32-bit PCI Bus, the PCI 9056 reads a single PCI Lword unless Direct Master Read Ahead mode is enabled.

For single cycle Direct Master reads, the PCI 9056 passes the corresponding PCI Bus byte enables from the Local Bus address and the TSIZ[0:1] signal.

For Burst Cycle reads, the PCI 9056 reads entire Lwords (all PCI Bus byte enables are asserted).

If the Direct Master Prefetch Limit bit is enabled (DMPBAM[11]=1), the PCI 9056 terminates a read prefetch at 4-KB boundaries and restarts it as a new PCI Read Prefetch cycle at the start of a new boundary. If the bit is disabled, the prefetch crosses the 4-KB boundaries.

3.4.1.4 Direct Master I/O Configuration Access

When a Local Direct Master I/O access to the PCI Bus occurs, the PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration Enable bit (DMCFGA[31]) determines whether an I/O or Configuration access is to be made to the PCI Bus.

Local Burst accesses are broken into single PCI I/O (address/data) cycles. The PCI 9056 does not prefetch Read data for I/O and Configuration reads.

For Direct Master I/O or Configuration cycles, the PCI 9056 asserts the same PCI Bus byte enables as set on the Local Bus.

3.4.1.5 Direct Master I/O

If the Configuration Enable bit is cleared (DMCFGA[31]=0), a single I/O access is made to the PCI Bus. The Local Address, Remapped Decode Address bits, and Local byte enables are encoded to provide the address and are output with an I/O Read or Write command during a PCI Address cycle.

For writes, data is loaded into the Write FIFO and TA# is returned to the Local Bus. For reads, the PCI 9056 holds off TA# while receiving an Lword from the PCI Bus.

3.4.1.6 Direct Master Delayed Write Mode

The PCI 9056 supports Direct Master Delayed Write mode transactions, where posted Write data accumulates in the Direct Master Write FIFO before the PCI 9056 requests the PCI Bus. Direct Master Delayed Write mode is programmable to delay REQ# assertion for the number of PCI clocks specified in DMPBAM[15:14]. This feature is useful for gaining higher throughput during Direct Master Write Burst transactions for conditions in which the Local clock frequency is slower than the PCI clock frequency.

The PCI 9056 only utilizes the delay counter and accumulates data in the Direct Master Write FIFO for burst transactions on the Local Bus. Otherwise, an immediate single cycle PCI transfer occurs.

3.4.1.7 Direct Master Read Ahead Mode

The PCI 9056 also supports Direct Master Read Ahead mode (DMPBAM[2]), where prefetched data can be read from the internal FIFO of the PCI 9056 instead of from the Local Bus. The address must be subsequent to the previous address and 32-bit aligned (next address = current address + 4) for 32-bit Direct Slave transfers. Read Ahead mode functions can be used with or without Delayed Read mode.

A Local Bus single cycle Direct Master transaction, with Read Ahead Mode (DMPBAM[2]) enabled results in the PCI 9056 processing continuous PCI Bus Read burst data with all bytes enabled (C/BE# = 0h).

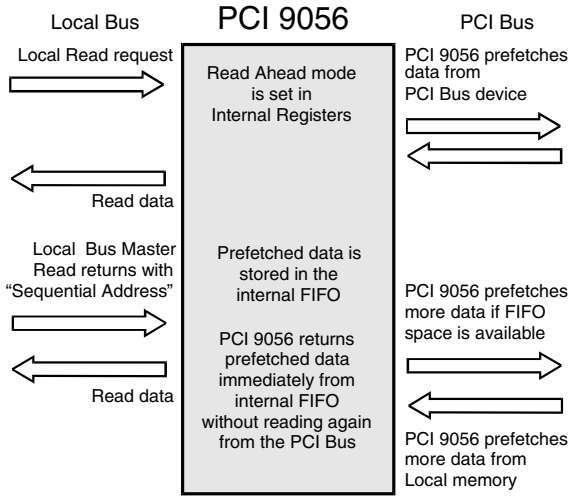


Figure 3-4. Direct Master Read Ahead Mode

Note: *Figure 3-4 represents a sequence of Bus cycles.*

3.4.1.8 RETRY# Capability

3.4.1.8.1 Direct Master Write FIFO Full

The PCI 9056 supports the Direct Master Write FIFO full condition. When enabled (LMISC1[6]=1), the PCI 9056 asserts the RETRY# signal to the Local Bus Master to relinquish ownership of the bus and return to finish the initial write at a later time.

In a Direct Master Write FIFO full condition, the PCI 9056 asserts the RETRY# signal. Otherwise, the Direct Master Write transfer goes through successfully.

3.4.1.8.2 Direct Master Delayed Read

The PCI 9056 supports Direct Master Delayed Read transactions. When the M Mode Direct Master Deferred Read Enable bit is set (LMISC1[4]=1), the PCI 9056 asserts RETRY# and prefetches Read data every time the Local Master requests a read. During a PCI data prefetch, the Local Master is capable of doing other transactions and free to return for requested data at a later time. When Delayed Direct Master Read mode is disabled, the Local Master must retain the Local Bus and wait for the requested data (TA# is not asserted until data is available to the Local Bus). In this mode, it is required that a Local Processor returns and reads at least one data. Otherwise, the PCI 9056 indefinitely retries a Local Bus.

3.4.1.9 Direct Master Configuration (PCI Configuration Type 0 or Type 1 Cycles)

If the Configuration Enable bit is set (DMCFGA[31]=1), and a Direct Master access is made to the Local Bus address programmed in DMLBAM, a Configuration access is made to the PCI Bus. In addition to enabling configuration of this bit, the user must provide all register information. The Register Number and Device Number bits (DMCFGA[7:2] and DMCFGA[15:11], respectively) must be modified and a new Configuration Read/Write cycle must be performed before accessing other registers or devices.

If the PCI Configuration Address register selects a Type 0 command, register bits [10:0] are copied to address bits [10:0]. Bits [15:11] (device number) are translated into a single bit being set in the PCI Address bits [31:11]. The PCI Address bits [31:11] can be used as a device select. For a Type 1 command, bits [23:0] are copied from the register to PCI address bits [23:0]. The PCI Address bits [31:24] are set to 0. A configuration Read or Write command code is output with the address during the PCI Address cycle. (Refer to the DMCFGA register.)

For writes, Local data is loaded into the Write FIFO and TA# is returned. For reads, the PCI 9056 holds off TA# while gathering an Lword from the PCI Bus.

3.4.1.9.1 Direct Master Configuration Cycle Example

To perform a Configuration Type 0 cycle to PCI device on AD[21]:

1. The PCI 9056 must be configured to allow Direct Master access to the PCI Bus. The PCI 9056 must also be set to respond to I/O Space accesses. These bits must be set (PCICR[2:0]=111b).

   In addition, Direct Master memory and I/O access must be enabled (DMPBAM[1:0]=11).

2. The Local Memory map selects the Direct Master range. For this example, use a range of 1 MB:

   1 MB = 2^{20} = 00100000h

3. The value to program into the Range register is the 2's complement of 00100000h (FFF00000h):

   DMRR = FFF00000h

4. The Local Memory map determines the Local Base Address for the Direct Master-to-PCI I/O Configuration register. For this example, use 40000000h:

   DMLBAI = 40000000h

5. The PCI Address (Remap) for Direct Master-to-PCI Memory register must enable the Direct Master I/O access. The Direct Master I/O Access Enable bit must be set (DMPBAM[1]=1).

6. The user must know which PCI device and PCI Configuration register the PCI Configuration cycle is accessing. This example assumes the IDSEL signal of the Target PCI device is connected to AD[21] (logical device #10=0Ah). It also assumes access is to PCIBAR0 (the fourth register, counting from 0. Use Table 11-2 for reference). Set DMCFGA[31, 23:0] as follows:

Bit	Description	Value
1:0	Configuration Type 0.	00b
7:2	Register Number. Fourth register. Must program a "4" into this value, beginning with bit 2.	000100b
10:8	Function Number.	000b
15:11	Device Number n-11, where n is the value in AD[n]=21-11 = 10.	01010b
23:16	Bus Number.	00000000b
31	Configuration Enable.	1

After these registers are configured, a simple Local Master Memory cycle to the I/O Base Address is necessary to generate a PCI Configuration Read or Write cycle. An offset to the Base Address is not necessary because the register offset for the read or write is specified in the Configuration register. The PCI 9056 takes the Local Bus Master Memory cycle and checks for the Configuration Enable bit (DMCFGA[31]). If set, the PCI 9056 converts the current cycle to a PCI Configuration cycle, using the DMCFGA register and the Write/Read signal (RD/WR#).

The Register Number and Device Number bits (DMCFGA[7:2] and DMCFGA[15:11], respectively) must be modified and a new Configuration Read/Write cycle must be performed before accessing other registers or devices.

3.4.1.10 Direct Master PCI Dual Address Cycle

The PCI 9056 supports PCI Dual Address Cycle (DAC) when it is a PCI Bus Master using the DMDAC register for Direct Master transactions. The DAC command is used to transfer a 32-bit address to devices that support 32-bit addressing when the address is not in the low 4-GB address space. The PCI 9056 performs the address portion of a DAC in two PCI clock periods, where the first PCI address is a Lo-Addr with the command (C/BE[3:0]#) "D" and the second PCI address will be a Hi-Addr with the command (C/BE[3:0]#) "6" or "7", depending upon it being a PCI Read or a PCI Write cycle. Whenever the DMDAC register contains a value of 0x00000000, the PCI 9056 performs a Single Address Cycle (SAC) on the PCI Bus. (Refer to Figure 3-5.)

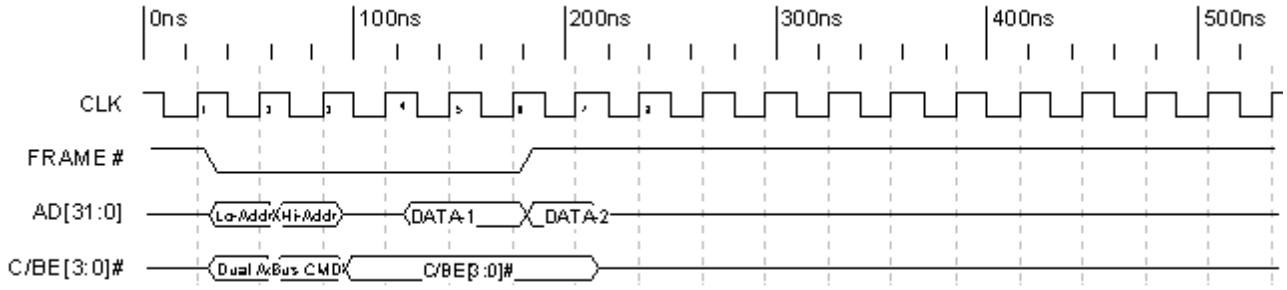


Figure 3-5. Dual Address Timing

3.4.1.11 PCI Master/Target Abort

The PCI 9056 PCI Master/Target Abort logic enables a Local Bus Master to perform a Direct Master Bus poll of devices to determine whether devices exist (typically when the Local Bus performs Configuration cycles to the PCI Bus). When a PCI Master device attempts to access and does not receive DEVSEL# within six PCI clocks, it results in a Master Abort. The Local Bus Master must clear the Received Master Abort bit or Target Abort bit (PCISR[13 or 11]=0, respectively) and continue by processing the next task.

If a PCI Master/Target Abort, or Retry Timeout is encountered during a transfer, the PCI 9056 asserts TEA# if enabled [INTCSR[1:0]=1, which can be used as a Non-Maskable Interrupt (NMI)]. If a Local Bus Master is waiting for TA#, it is asserted along with TEA#. The interrupt handler of the Local Bus Master can take the appropriate application-specific action. It can then clear the Target Abort bit (PCISR[11]) to clear the TEA# interrupt and re-enable Direct Master transfers.

If a Local Bus Master is attempting a Burst read from a nonresponding PCI device (Master/Target Abort), it receives TA# and BI# for the first cycle only. In addition, the PCI 9056 asserts TEA# if the Enable Local Bus TEA# bits are enabled (INTCSR[1:0], which can be used as an NMI). If the Local processor cannot terminate its Burst cycle, it may cause the Local processor to hang. The Local Bus must then be reset from the PCI Bus. If the Local Bus Master cannot terminate its cycle with TEA# output, it should not perform Burst cycles when attempting to determine whether a PCI device exists.

If a PCI Master/Target Abort is encountered during a Direct Master transfer, the PCI 9056 stores the PCI Abort address into the PCI Abort Address register bits (PABTADR[31:0]).

3.4.1.12 Direct Master Memory Write and Invalidate

The PCI 9056 can be programmed to perform Memory Write and Invalidate cycles to the PCI Bus for Direct Master transfers, as well as for DMA transfers. (Refer to Section 3.5.4.) The PCI 9056 supports Memory Write and Invalidate transfers for cache line sizes of 8 or 16 Lwords. Size is specified in the System Cache Line Size bits (PCICLSR[7:0]). If a size other than 8 or 16 is specified, the PCI 9056 performs Write transfers rather than Memory Write and Invalidate transfers.

Direct Master Memory Write and Invalidate transfers are enabled when the Invalidate Enable and the Memory Write and Invalidate Enable bits are set (DMPBAM[9]) and (PCICR[4], respectively).

In Memory Write and Invalidate mode, if the start address of the Direct Master transfer is on a cache line boundary, the PCI 9056 waits until the number of Lwords required for the specified cache line size are written from the Local Bus before starting a PCI Memory Write and Invalidate access. This ensures a complete cache line write can complete in one PCI Bus ownership.

If the start address is not on a cache line boundary, the PCI 9056 starts a normal PCI Write access (PCI command code = 7h). The PCI 9056 does not terminate a normal PCI Write at an MWI cache boundary. The normal PCI Write transfer continues until the Data transfer is complete. If a Target disconnects before a cache line is completed, the PCI 9056 completes the remainder of that cache line, using normal writes.

3.4.2 IDMA/SDMA Operation

3.4.2.1 IDMA Operation

The PCI 9056 supports the MPC850 or MPC860 Independent DMA (IDMA) mode, using the MDREQ# signal and operating in Direct Master mode. In M mode, this signal is connected to the MPC850 or MPC860 DREQ0# and/or DREQ1# input pins. After programming the MPC850 or MPC860 IDMA channel, the PCI 9056 uses Direct Master mode to transfer data between the PCI Bus and the MPC850 or MPC860 internal dual-port RAM (or external memory). The data count is controlled by the IDMA Byte counter and throttled by the PCI 9056 MDREQ# signal. When the PCI 9056 FIFO is nearly full, MDREQ# is de-asserted to the MPC850 or MPC860, indicating that it should inhibit transferring further data (the FIFO threshold count in the PCI 9056 must be set to a value of at least five Lwords below the full capacity of the FIFO—27 Lwords) (DMPBAM[10, 8:5]). The Retry function can be used to communicate to the Local Bus Master that it should relinquish ownership of the Local Bus.

Note: *The Direct Master Write FIFO Almost Full RETRY# Output Enable bit (LMISC1[6]) can be disabled to prevent assertion of the RETRY# signal.*

In IDMA reads (PCI 9056 to the Local Bus), the MDREQ# signal is asserted (indicating data is available), although the Read FIFO is empty. Any Local Bus read of the PCI Bus causes the PCI 9056 to become a PCI Bus Master and fills the Direct Master Read FIFO buffer. When sufficient data is in the FIFO, the PCI 9056 completes the Local Bus cycle by asserting Transfer Acknowledge (TA#).

After the IDMA has transferred all required bytes [MPC850 or MPC860 Byte counter decrements to zero (0)], the MPC850 or MPC860 generate an internal interrupt, which in turn should execute the code to disable the IDMA channel (the MDREQ# input signal may still be asserted by the PCI 9056). The SDACK[1:0] signal from the MPC850 or MPC860 is not used by the PCI 9056 (no connection).

Refer to Section 3.4.1 for more information about Direct Master Data transfers.

3.4.2.2 SDMA Operation

The PCI 9056 supports the MPC850 or MPC860 Serial DMA (SDMA) mode, using Direct Master mode. No handshake signals are required to perform the SDMA operation.

The Retry function can be used to communicate to the Local Bus Master it should relinquish ownership of the Local Bus. The Direct Master Write FIFO Almost Full RETRY# Output Enable bit (LMISC1[6]) can be disabled to prevent assertion of the RETRY# signal.

Note: *The Direct Master Write FIFO can be programmed to identify the full status condition (DMPBAM[10, 8:5]). The FIFO Full Status Flag is in MARBR[30].*

3.4.3 Direct Slave Operation (PCI Master-to-Local Bus Access)

The PCI 9056 supports Burst Memory-Mapped Transfer accesses and I/O-Mapped, Single-Transfer PCI-to-Local Bus accesses through a 32-Lword (128-byte) Direct Slave Read FIFO and a 64-Lword (256-byte) Direct Slave Write FIFO. The PCI Base Address registers are provided to set up the location of the adapter in the PCI memory and the I/O space. In addition, Local mapping registers allow address translation from the PCI Address Space to the Local Address Space. Three spaces are available:

- Space 0
- Space 1
- Expansion ROM

Expansion ROM is intended to support a bootable ROM device for the Host.

Writes—Upon a PCI Bus Write, the PCI Bus Master writes data to the Direct Slave Write FIFO. When the first data is in the FIFO, the PCI 9056 becomes the Local Bus Master, arbitrates for the Local Bus, and writes data to a Local Slave device. The PCI 9056 continues to accept writes and returns TRDY# until the Write FIFO is full. It then holds off TRDY# until space becomes available in the Write FIFO or asserts STOP#, and Retries the PCI Bus Master, dependent upon the register bit setting (LBRD0[27]).

A 32-bit PCI Bus Master single cycle Write transaction results in PCI 9056 transfers of one Lword of data onto a Local Bus.

Reads—The PCI 9056 holds off TRDY# while gathering an Lword from the Local Bus, unless the Delayed Read Mode bit is enabled (MARBR[24]=1). (Refer to Section 3.4.3.2.) Programmable Prefetch modes are available, if prefetch is enabled—prefetch, 0-16, or continuous—until the Direct Slave read ends. The Read cycles are terminated on the following clock after FRAME# is de-asserted or the PCI 9056 issues a Retry or disconnect.

For the highest data transfer rate, the PCI 9056 supports posted writes and can be programmed to prefetch data during a PCI Burst read. The Prefetch size, when enabled, can be from one to 16 Lwords or until the PCI Bus stops requesting. When the PCI 9056 prefetches, if enabled, it drops the Local Bus after reaching the prefetch counter limit. In Continuous Prefetch mode, the PCI 9056 prefetches as long as FIFO space is available, and stops prefetching when the PCI Bus terminates the request. If Read prefetching is disabled, the PCI 9056 disconnects after one Read transfer.

In addition to Prefetch mode, the PCI 9056 supports Read Ahead mode. (Refer to Section 3.4.3.3.)

Only 32-bit PCI Bus single cycle Direct Slave Read transactions result in the PCI 9056 passing requested PCI bytes (C/BE#) to a Local Bus Target device by way of TSIZ[0:1] assertion back to a PCI Bus Master. This transaction results in the PCI 9056 reading one Lword or partial Lword data. For other types of Read transactions (Burst transfers or Unaligned), the PCI 9056 reads multiple Local Bus data with all bytes asserted (TSIZ[0:1] =0h).

Each Local space can be programmed to operate in an 8-, 16-, or 32-bit Local Bus width. The PCI 9056 has an internal wait state generator and external wait state input, TA#. TA# can be disabled or enabled with the Internal Configuration registers.

With or without wait state(s), the Local Bus, independent of the PCI Bus, can perform the following:

- Burst as long as data is available (Continuous Burst mode)
- Burst four Lwords at a time (recommended)
- Perform continuous single cycles

A Burst cycle from the PCI Bus through the PCI 9056 asserts an MPC850 or MPC860 Burst transaction, if the following is true:

- The address is quad-Lword-aligned,
- A FIFO contains at least four Lwords, and
- All PCI Bus byte enables are set for writes only and ignored for reads

3.4.3.1 Direct Slave Lock

The PCI 9056 supports direct PCI-to-Local-Bus exclusive accesses (locked atomic operations). A PCI-locked operation to the Local Bus results in the entire address Space 0, Space 1, and Expansion ROM space being locked until they are released by the PCI Bus Master. Locked operations are enabled or disabled with the Direct Slave LOCK# Enable bit (MARBR[22]).

3.4.3.2 Direct Slave Delayed Read Mode

The PCI 9056 can be programmed through the Delayed Read Mode bit (MARBR[24]=1) to perform delayed reads.

PCI Bus single cycle aligned or unaligned 32-bit Direct Slave Delayed Read transactions always result in 1-Lword single cycle transfers on the Local Bus with the corresponding Local Address and TSIZ[0:1] asserted to reflect the PCI byte enables (C/BE#), unless the PCI Read No Flush Mode bit is enabled (MARBR[28]=1). (Refer to Section 3.4.3.3 for further information.) This causes the PCI 9056 to Retry all PCI Bus Read requests that follow, until the original PCI byte enables (C/BE#) are matched.

In addition to delayed reads, the PCI 9056 supports the following Delayed Read mode functions:

- No writes while a read is pending (PCI Retry for writes)
- Write and flush pending read

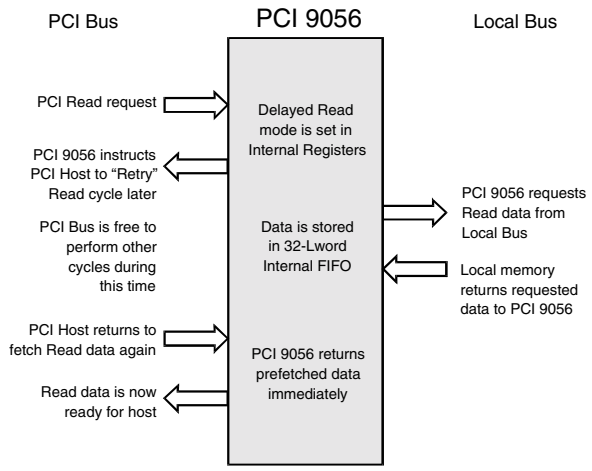


Figure 3-6. Direct Slave Delayed Read

Note: *Figure 3-6 represents a sequence of Bus cycles.*

3.4.3.3 Direct Slave Read Ahead Mode

The PCI 9056 also supports Direct Slave Read Ahead mode (MARBR[28]), where prefetched data can be read from the internal FIFO of the PCI 9056 instead of from the Local Bus. The address must be subsequent to the previous address and 32-bit aligned (next address = current address + 4) for 32-bit Direct Slave transfers.

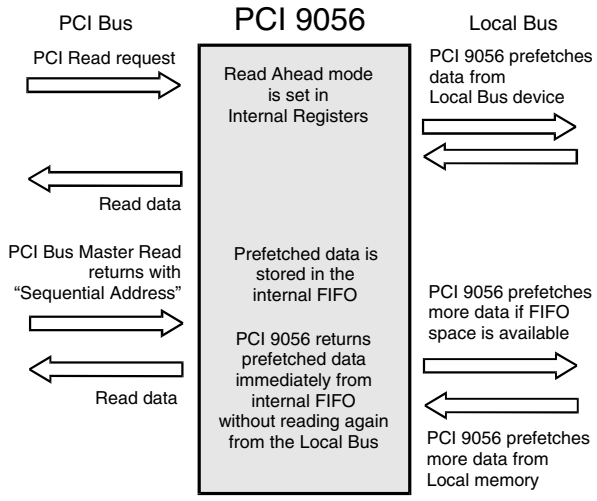


Figure 3-7. Direct Slave Read Ahead Mode

Note: *Figure 3-7 represents a sequence of Bus cycles.*

3.4.3.4 Direct Slave Delayed Write Mode

The PCI 9056 supports Direct Slave Delayed Write mode transactions, where posted Write data accumulates in the Direct Slave Write FIFO before the PCI 9056 requests a Write transaction (TS# assertion) to be performed on the Local Bus. The Direct Slave Delayed Write mode is programmable to delay the TS# assertion in the amount of Local clocks (LMISC2[4:2]). This feature is useful for gaining higher throughput during Direct Slave Write burst transactions for conditions in which the PCI clock frequency is slower than the Local clock frequency.

3.4.3.5 Direct Slave Local Bus TA# Timeout Mode

The PCI 9056 supports Direct Slave Local Bus TA# Timeout mode transactions, where the PCI 9056 asserts an internal TA# signal to recover from stalling the Local and PCI Buses. The Direct Slave Local Bus TA# Timeout mode transaction is programmable to select the amount of Local clocks before TA# times out (LMISC2[1:0]). If a Local Slave stalls with a TA# assertion during Direct Slave Write transactions, the PCI 9056 empties the Write FIFO by dumping the data into the Local Bus and does not pass an error condition to the PCI Bus Initiator. During Direct Slave Read transactions, the PCI 9056 issues a Direct Slave Abort to the PCI Bus Initiator every time the Direct Slave Local Bus TA# Timeout is detected.

3.4.3.6 Direct Slave Transfer

A PCI Bus Master addressing the Memory space decoded for the Local Bus initiates transactions. Upon a PCI Read/Write, the PCI 9056 becomes a Local Bus Master and arbitrates for the Local Bus.

The PCI 9056 then reads data into the Direct Slave Read FIFO or writes data to the Local Bus.

The Direct Slave or Direct Master preempts DMA; however, the Direct Slave does not preempt the Direct Master. (Refer to Section 3.4.4.1)

The PCI 9056 can be programmed to retain the PCI Bus by generating a wait state(s) and de-asserting TRDY#, if the Write FIFO becomes full. The PCI 9056 can also be programmed to retain the Local Bus and continue asserting BB#, if the Direct Slave Write FIFO

becomes empty or the Direct Slave Read FIFO becomes full. In either case, the Local Bus is dropped when the Local Bus Latency Timer is enabled and expires (MARBR[7:0]).

For Direct Slave writes, the PCI Bus writes data to the Local Bus. Direct Slave is the "Command from the PCI Host," which has the highest priority.

For Direct Slave reads, the PCI Bus Master reads data from the Local Bus Slave.

The PCI 9056 supports on-the-fly Endian conversion for Space 0, Space 1, and Expansion ROM space. The Local Bus can be Big/Little Endian (Address/Data Invariance) by using the programmable internal register configuration.

Note: *The PCI Bus is always Little Endian.*

During Direct Slave transactions, the MPC850 or MPC860 user has the option to use the PCI 9056 for maximum Burst transfers, using the BTERM# Input Enable bit(s) (LBRD0[23, 7], LBRD1[7], DMAMODE0[7], and/or DMAMODE1[7]).

In Direct Slave transfers, each Direct Slave space (Space 0, Space 1, and Expansion ROM) has its own BTERM# Input Enable bit (the BTERM# input signal becomes the BI# signal in M mode). Space 0 is in LBRD0[7], Space 1 is in LBRD1[7], and Expansion ROM is in LBRD0[23].

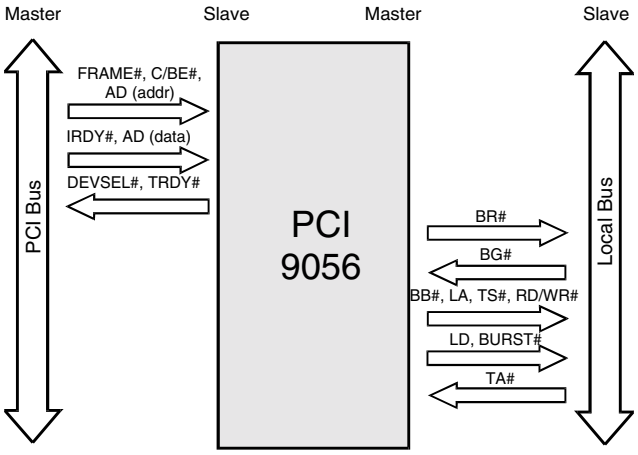


Figure 3-8. Direct Slave Write

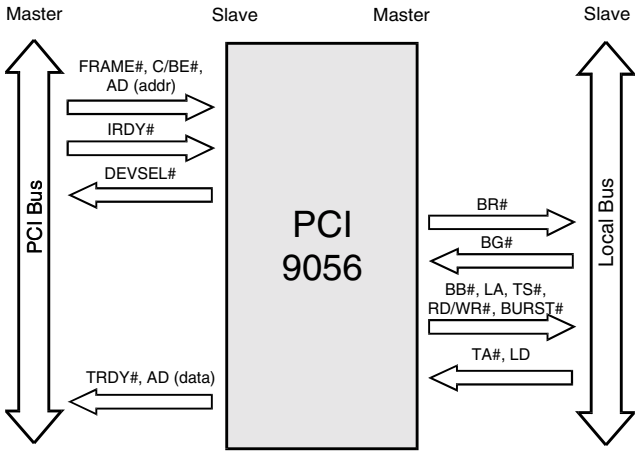


Figure 3-9. Direct Slave Read

Note: *Figures 3-8 and 3-9 represent a sequence of Bus cycles.*

When the Bterm Mode bit is enabled, the PCI 9056 continues to burst on the Local Bus until the BI# signal is asserted for one CLK cycle any time after the first Data phase, implying a new Address cycle (TS#) is needed if there is more data to transfer. If the BI# signal is asserted on the first Data phase, the Burst transfer is broken into single cycle transactions.

When the Bterm Mode bit is enabled and the BI# signal asserted for one CLK cycle any time after the first Data phase, this implies that a new Address cycle (TS#) is needed for more data to transfer. This can be used when crossing memory banks.

Regardless of the Bterm mode setting, if the BI# signal is asserted on the first Data phase, single cycle transfers are performed until the Qword boundary is reached.

Table 3-2. Direct Slave Burst Mode Cycle Detection

Burst Enable Bit	BTERM# Input Enable Bit	BI# Signal	Result
1	0	Not asserted	Burst 16 bytes (MPC850 or MPC860 compatible)
1	0	Asserted during first Data phase	Single cycle
1	1	Asserted after first Data phase	Burst until BI# is asserted for one CLK cycle
0	X	X	Single cycle

Caution: *The MPC850 and MPC860 do not support bursting more than 16 bytes. The BTERM# Input Enable bits should be set only for Local Bus Masters that support continuous bursting.*

Note: *"X" is "Don't Care."*

During burst forever, extended M mode protocol, the PCI 9056 supports BDIP# signal. The BDIP# signal is asserted at the beginning of a Burst transaction and remains active until the last data of the Transfer packet. The BDIP# signal is de-asserted on the last Data Transfer phase, indicating the end of transfer.

The PCI 9056 supports Local Bus error conditions using TEA#. TEA# may be asserted by a device on the Local Bus, either before or simultaneously with TA#. In either case, the PCI 9056 tries to complete the current transaction by transferring data and then asserting TS# for every address that follows, waiting for another TA# or TEA# to be issued (used to flush Direct Slave FIFOs). After acknowledging TEA# is asserted, the PCI 9056 asserts PCI SERR# and sets an error flag, using the Signaled System Error bit (PCISR[14]=1). When set, this indicates a catastrophic error occurred on the Local Bus. SERR# may be masked off by resetting the TEA# Input Interrupt Mask bit (LMISC1[5]=0).

The PCI 9056 Local Bus Latency Timer (MARBR[7:0]) can be used to better utilize the Local Bus.

3.4.3.7 Direct Slave PCI-to-Local Address Mapping

Note: *In I_2O mode (QSR[0]=1), Memory-Mapped Local Configuration registers and Space 1 share the PCIBAR0 Base Address (refer to Section 7.1.10 for further details).*

Three Local Address spaces—Space 0, Space 1, and Expansion ROM—are accessible from the PCI Bus. Each is defined by a set of three registers:

- Local Address Range (LAS0RR, LAS1RR, and/or EROMRR)
- Local Base Address (LAS0BA, LAS1BA, and/or EROMBA)
- PCI Base Address (PCIBAR2, PCIBAR3, and/or PCIERBAR)

A fourth register, the Bus Region Descriptor register(s) for PCI-to-Local Accesses (LBRD0 and/or LBRD1), defines the Local Bus characteristics for the Direct Slave regions. (Refer to Figure 3-10.)

Each PCI-to-Local Address space is defined as part of reset initialization, as described in Section 3.4.3.7.1. These Local Bus characteristics can be modified at any time before actual data transactions.

3.4.3.7.1 Direct Slave Local Bus Initialization

Range—Specifies which PCI Address bits to use for decoding a PCI access to Local Bus space. Each bit corresponds to a PCI Address bit. Bit 31 corresponds to address bit 31. Write 1 to all bits that must be included in decode and 0 to all others.

Remap PCI-to-Local Addresses into a Local Address Space—Bits in this register remap (replace) the PCI Address bits used in decode as the Local Address bits.

Local Bus Region Descriptor—Specifies the Local Bus characteristics.

3.4.3.7.2 Direct Slave PCI Initialization

After a PCI reset, the software determines how much address space is required by writing all ones (1) to a PCI Base Address register and then reading back the value. The PCI 9056 returns zeroes (0) in the Don't Care Address bits, effectively specifying the address space required. The PCI software then maps the Local Address space into the PCI Address space by programming the PCI Base Address register. (Refer to Figure 3-10.)

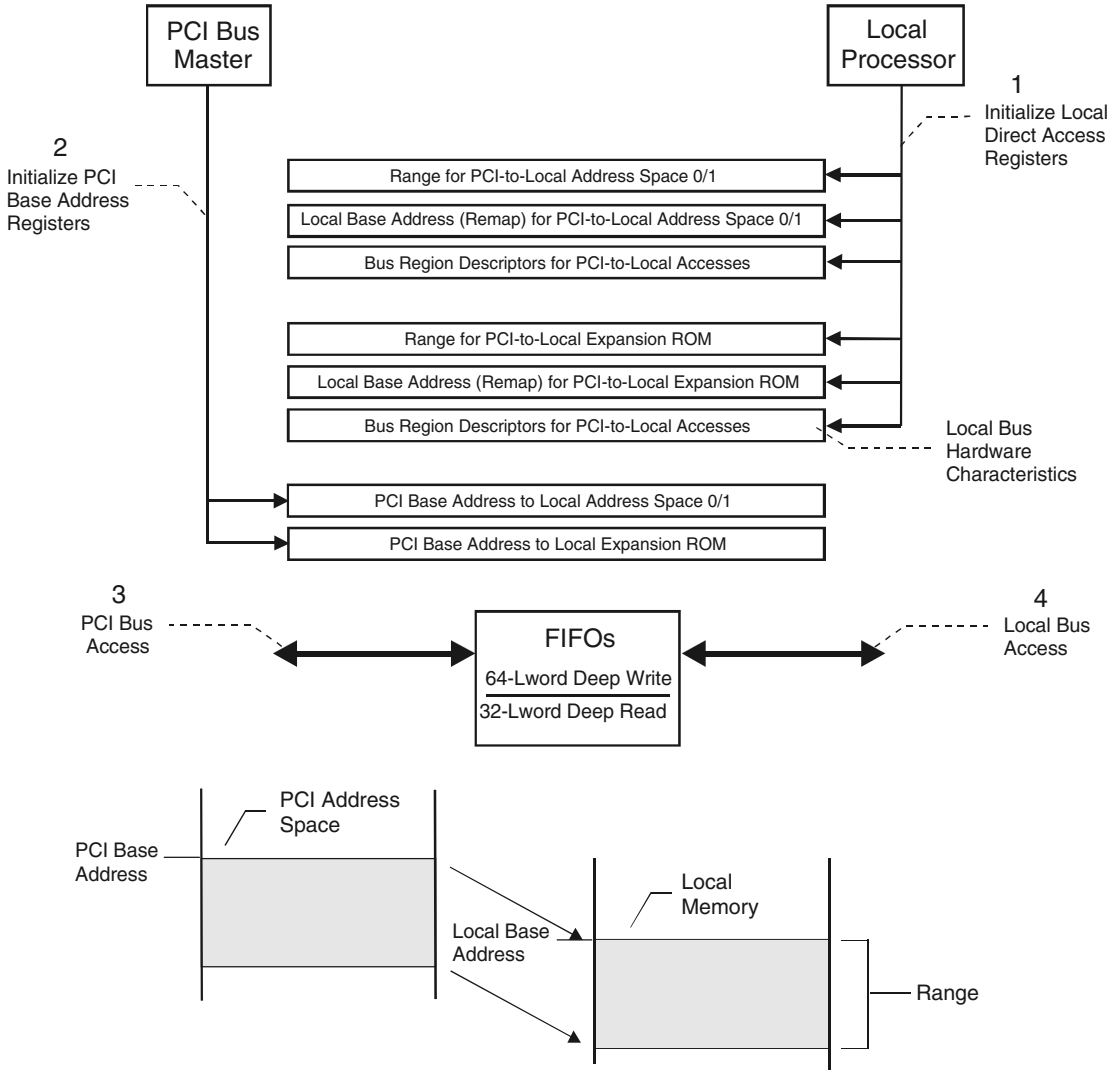


Figure 3-10. Local Bus Direct Slave Access

3.4.3.7.3 Direct Slave Transfer Size

The TSIZ[0:1] pins correspond to the data-transfer size on the Local Bus, as listed in the following tables.

Table 3-3. Data Bus TSIZ[0:1] Contents for Single Write Cycles

Transfer Size	TSIZ [0:1]		Address		External Data Bus Pattern For 32-, 16-, and 8-Bit Port Sizes			
			LA30	LA31	LD[0:7]	LD[8:15]	LD[16:23]	LD[24:31]
Byte	0	1	0	0	OP0	—	—	—
	0	1	0	1	OP1	OP1	—	—
	0	1	1	0	OP2	—	OP2	—
	0	1	1	1	OP3	OP3	—	OP3
Word	1	0	0	0	OP0	OP1	—	—
	1	0	1	0	OP2	OP3	OP2	OP3
Lword	0	0	0	0	OP0	OP1	OP2	OP3

Table 3-4. Data Bus TSIZ[0:1] Requirements for Single Read Cycles

Transfer Size	TSIZ [0:1]		Address		32-Bit Port Size				16-Bit Port Size		8-Bit Port Size
			LA30	LA31	LD[0:7]	LD[8:15]	LD[16:23]	LD[24:31]	LD[0:7]	LD[8:15]	LD[0:7]
Byte	0	1	0	0	OP0	—	—	—	OP0	—	OP0
	0	1	0	1	—	OP1	—	—	—	OP1	OP1
	0	1	1	0	—	—	OP2	—	OP2	—	OP2
	0	1	1	1	—	—	—	OP3	—	OP3	OP3
Word	1	0	0	0	OP0	OP1	—	—	OP0	OP1	OP0
	1	0	1	0	—	—	OP2	OP3	OP2	OP3	OP2
Lword	0	0	0	0	OP0	OP1	OP2	OP3	OP0	OP1	OP0

3.4.3.7.3.1 Direct Slave Transfer Size Example

A 1 MB Local Address Space, 12300000h through 123FFFFFh, is accessible from the PCI Bus at PCI addresses 78900000h through 789FFFFFh.

a. Local initialization software sets the Range and Local Base Address registers as follows:
   - **Range**—FFF00000h (1 MB, decode the upper 12 PCI Address bits)
   - **Local Base Address (Remap)**—123XXXXXh (Local Base Address for PCI-to-Local accesses [Space Enable bit(s) must be set to be recognized by the PCI Host (LAS0BA[0]=1 and/or LAS1BA[0]=1)]

b. PCI Initialization software writes all ones (1) to the PCI Base Address, then reads it back again.
   - The PCI 9056 returns a value of FFF00000h. The PCI software then writes to the PCI Base Address register(s).
   - **PCI Base Address**—789XXXXXh (PCI Base Address for Access to the Local Address Space registers, PCIBAR2 and PCIBAR3).

For a PCI Direct access to the Local Bus, the PCI 9056 has a 64-Lword (256-byte) Write FIFO and a 32-Lword (128-byte) Read FIFO. The FIFOs enable the Local Bus to operate independent of the PCI Bus. The PCI 9056 can be programmed to return a Retry response or to throttle TRDY# for any PCI Bus transaction attempting to write to the PCI 9056 Local Bus when the FIFO is full.

For PCI Read transactions from the Local Bus, the PCI 9056 holds off TRDY# while gathering data from the Local Bus. For Read accesses mapped to PCI Memory space, the PCI 9056 prefetches up to 16 Lwords (has Continuous Prefetch mode) from the Local Bus. Unused Read data is flushed from the FIFO. For Read accesses mapped to PCI I/O space, the PCI 9056 does not prefetch Read data. Rather, it breaks each read of a Burst cycle into a single Address/Data cycle on the Local Bus.

The Direct Slave Retry Delay Clocks bits (LBRD0[31:28]) can be used to program the period of time in which the PCI 9056 holds off TRDY#. The PCI 9056 issues a Retry to the PCI Bus Transaction Master when the programmed time period expires. This occurs when the PCI 9056 cannot gain control of the Local Bus and return TRDY# within the programmed time period.

3.4.3.8 Direct Slave Priority

Direct Slave accesses have a higher priority than DMA accesses, thereby preempting DMA transfers. During a DMA transfer, if the PCI 9056 detects a pending Direct Slave access, it releases the Local Bus within two Data transfers. The PCI 9056 resumes operation after the Direct Slave access completes.

When the PCI 9056 DMA controller owns the Local Bus, its BR# output and BG# input are asserted. When a Direct Slave access occurs, the PCI 9056 releases the Local Bus within two Lword transfers by de-asserting BB# and floating the Local Bus outputs. After the PCI 9056 acknowledges that BG# is de-asserted, it requests the Local Bus for a Direct Slave transfer by asserting BR#. When the PCI 9056 receives BG#, it drives the bus and performs the Direct Slave transfer. Upon completing a Direct Slave transfer, the PCI 9056 releases the Local Bus by de-asserting BB# and floating the Local Bus outputs. After the PCI 9056 acknowledges that BG# is de-asserted and the Local Bus Pause Timer is set to zero (0), it requests a DMA transfer from the Local Bus by re-asserting BR#. When it receives BG#, it drives the bus and continues the DMA transfer.

3.4.4 Deadlock Conditions

Deadlock can occur when a PCI Bus Master must access the PCI 9056 Local Bus at the same time a Master on the PCI 9056 Local Bus must access the PCI Bus.

There are two types of deadlock:

- **Partial Deadlock**—A Local Bus Master is performing a Direct Bus Master access to a PCI Bus device other than the PCI Bus device concurrently trying to access the Local Bus
- **Full Deadlock**—A Local Bus Master is performing a Direct Bus Master access to the same PCI Bus device concurrently trying to access the Local Bus

This applies only to Direct Master and Direct Slave accesses through the PCI 9056. Deadlock does not occur in transfers through the PCI 9056 DMA channels or the PCI 9056 internal registers (such as mailboxes).

For partial deadlock, the PCI access to the Local Bus times out [the Direct Slave Retry Delay Clock (LBRD0[31:28]), which is programmable through the Local Bus Region Descriptor register] and the PCI 9056 responds with a PCI Retry. *PCI r2.2* requires that a PCI Master release its request for the PCI Bus (de-assert REQ#) for a minimum of two PCI clocks after receiving a Retry. This allows the PCI Bus arbiter to grant the PCI Bus to the PCI 9056 so that it can complete its Direct Master access and free up the Local Bus. Possible solutions are described in the following sections for cases in which the PCI Bus arbiter does not function as described (PCI Bus architecture dependent), waiting for a time out is undesirable, or a full deadlock condition exists.

When a full deadlock occurs, the only solution is to back off the Local Bus Master.

3.4.4.1 Backoff

The PCI 9056 Local RETRY# signal indicates whether a possible deadlock condition exists. The PCI 9056 starts the Backoff Timer (programmable through registers) when it detects one of the following conditions:

- A PCI Bus Master is attempting to access memory or an I/O device on the Local Bus and is not gaining access (*for example*, BG# is not received).
- A Local Bus Master is performing a Direct Bus Master Read access to the PCI Bus. Or, a Local Bus Master is performing a Direct Bus Master Write access to the PCI Bus and the PCI 9056 Direct Master Write FIFO cannot accept another Write cycle.

If the Local Bus Backoff Enable bit is enabled (EROMBA[4]=1), the Backoff Timer expires, and the PCI 9056 has not received BG#, the PCI 9056 asserts RETRY#. External bus logic can use this signal to perform backoff.

The Backoff cycle is device/bus architecture dependent. The external logic (arbiter) can assert the necessary signals to cause the Local Bus Master to release the Local Bus (backoff). After the Local Bus Master backs off, it can grant the bus to the PCI 9056 by asserting BG#.

Once RETRY# is asserted, TA# for the current Data cycle is never asserted (the Local Bus Master must perform a backoff). When the PCI 9056 detects BG#, it proceeds with the PCI Master-to-Local-Bus access. When this access completes and the PCI 9056 releases the Local Bus, external logic can then release the backoff and the Local Bus Master can resume the cycle interrupted by the Backoff cycle. The PCI 9056 Write FIFO retains all data acknowledged (*that is*, last data for which TA# was asserted).

After the backoff condition ends, the Local Bus Master restarts the last cycle with TS#. For writes, data following TS# should be the data the PCI 9056 did not acknowledge prior to the Backoff cycle (*for example*, the last data for which TA# is not asserted).

All PCI Read cycles completed before the Local Bus was backed off remain in the Direct Master Read FIFO. Therefore, if the Local Bus Master returns with the same last cycle, the cycle is acknowledged with the data currently in the FIFO (the FIFO data is not read twice). A new PCI read is performed, if the resumed Local Bus cycle is not the same as the Backed Off cycle.

3.4.4.1.1 Software/Hardware Solution for Systems without Backoff Capability

For adapters that do not support backoff, a possible deadlock solution is as follows.

PCI Host software, external Local Bus hardware, general-purpose output USERo and general-purpose input USERi can be used to prevent deadlock. USERo can be asserted to request that the external arbiter not grant the bus to any Local Bus Master except the PCI 9056. Status output from the Local arbiter can be connected to the general purpose input USERi to indicate that no Local Bus Master owns the Local Bus, or the PCI Host to determine that no Local Bus Master that currently owns the Local Bus can read input. The PCI Host can then perform Direct Slave access. When the Host finishes, it de-asserts USERo.

3.4.4.1.2 Preempt Solution

For devices that support preempt, USERo can be used to preempt the current Local Bus Master device. When USERo is asserted, the current Local Bus Master device completes its current cycle and releases the Local Bus, de-asserting BB#.

3.4.4.2 Software Solutions to Deadlock

Both PCI Host and Local Bus software can use a combination of mailbox registers, doorbell registers, interrupts, direct Local-to-PCI accesses and direct PCI-to-Local accesses to avoid deadlock.

3.5 DMA OPERATION

The PCI 9056 supports two independent DMA channels capable of transferring data from the:

- Local-to-PCI Bus
- PCI-to-Local Bus

Each channel consists of a DMA controller and a dedicated, bidirectional FIFO. Both channels support Block transfers, and Scatter/Gather transfers, with or without End of Transfer (EOT#). Only DMA Channel 0 supports Demand mode DMA transfers. Master mode must be enabled with the Master Enable bit

(PCICR[2]) before the PCI 9056 can become a PCI Bus Master. In addition, both DMA channels can be programmed to:

- Operate in 8-, 16-, or 32-bit Local Bus width
- Use zero to 15 internal wait states (Local Bus)
- Enable/disable internal wait states (Local Bus)
- Enable/disable Local Bus Burst capability
- Limit Local Bus bursts to four (BTERM# enable/ disable)
- Hold Local address constant (Local Slave is FIFO) or increment
- Perform PCI Memory Write and Invalidate (command code = Fh) or normal PCI Memory Write (command code = 7h)
- Pause Local transfer with/without BLAST# (DMA Fast/Slow termination)
- Assert PCI interrupt (INTA#) or Local interrupt (LINTo#) when DMA transfer is complete or Terminal Count is reached during Scatter/Gather DMA mode transfers
- Operate in DMA Clear Count mode (only if the descriptor is in Local memory)

The PCI 9056 also supports PCI Dual Address with the upper 32-bit register(s) (DMADAC0 and/or DMADAC1).

The Local Bus Latency Timer determines the number of Local clocks the PCI 9056 can burst data before relinquishing the Local Bus. The Local Bus Pause Timer sets how soon the DMA channel can request the Local Bus.

3.5.1 DMA PCI Dual Address Cycle

The PCI 9056 supports PCI Dual Address Cycles (DAC) when it is a PCI Bus Master using the DMADAC0 and/or DMADAC1 register(s) for Block DMA transactions. Scatter/Gather DMA can utilize the DAC function by way of the DMADAC0 and/or DMADAC1 register(s) or DMAMODE0[18] and/or DMAMODE1[18]. The DAC command is used to transfer a 32-bit address to devices that support 32-bit addressing when the address is above the 4-GB address space. The PCI 9056 performs a DAC within two PCI clock periods, when the first PCI address is a Lo-Addr, with the command (C/BE[3:0]#) “D”, and the second PCI address is a Hi-Addr, with the command (C/BE[3:0]#) “6” or “7”, depending upon whether it is a PCI Read or PCI Write cycle.

3.5.2 Block DMA Mode

The Host processor or the Local processor sets the Local and PCI starting addresses, transfer byte count, and transfer direction. The Host or Local processor then sets the DMA Start bit (DMACSR0[1] and/or DMACSR1[1]) to initiate a transfer. The PCI 9056 requests the PCI and Local Buses and transfers data. Once the transfer completes, the PCI 9056 sets the Channel Done bit(s) (DMACSR0[4]=1 and/or DMACSR1[4]=1) and, if enabled, asserts an interrupt(s) (DMAMODE0[10] and/or DMAMODE1[10]) to the Local processor or the PCI Host (programmable). The Channel Done bit(s) can be polled, instead of interrupt generation, to indicate the DMA transfer status.

DMA registers are accessible from the PCI and Local Buses. (Refer to Figure 3-11.)

During DMA transfers, the PCI 9056 is a Master on both the PCI and Local Buses. For simultaneous access, Direct Slave or Direct Master has a higher priority than DMA.

The PCI 9056 releases the PCI Bus, if one of the following conditions occur (refer to Figure 3-12 and Figure 3-13):

- FIFO is full (PCI-to-Local Bus)
- FIFO is empty (Local-to-PCI Bus)
- Terminal count is reached
- PCI Bus Latency Timer expires (PCILTR[7:0])—normally programmed by the Host PCI BIOS—and PCI GNT# de-asserts
- PCI Host asserts STOP#

The PCI 9056 releases the Local Bus, if one of the following conditions occurs:

- FIFO is empty (PCI-to-Local Bus)
- FIFO is full (Local-to-PCI Bus)
- Terminal count is reached
- Local Bus Latency Timer is enabled and expires (MARBR[7:0])
- Special cycle BI# input is asserted
- Direct Slave request is pending

During DMA transactions, users have the option of using the Burst Forever BTERM# Input Enable bit(s) (DMAMODE0[7] and/or DMAMODE1[7]), if the External Memory Controller is provided. Used in conjunction with the Fast/Slow Terminate Mode Select bit(s) (DMAMODE0[15] and/or DMAMODE1[15]).

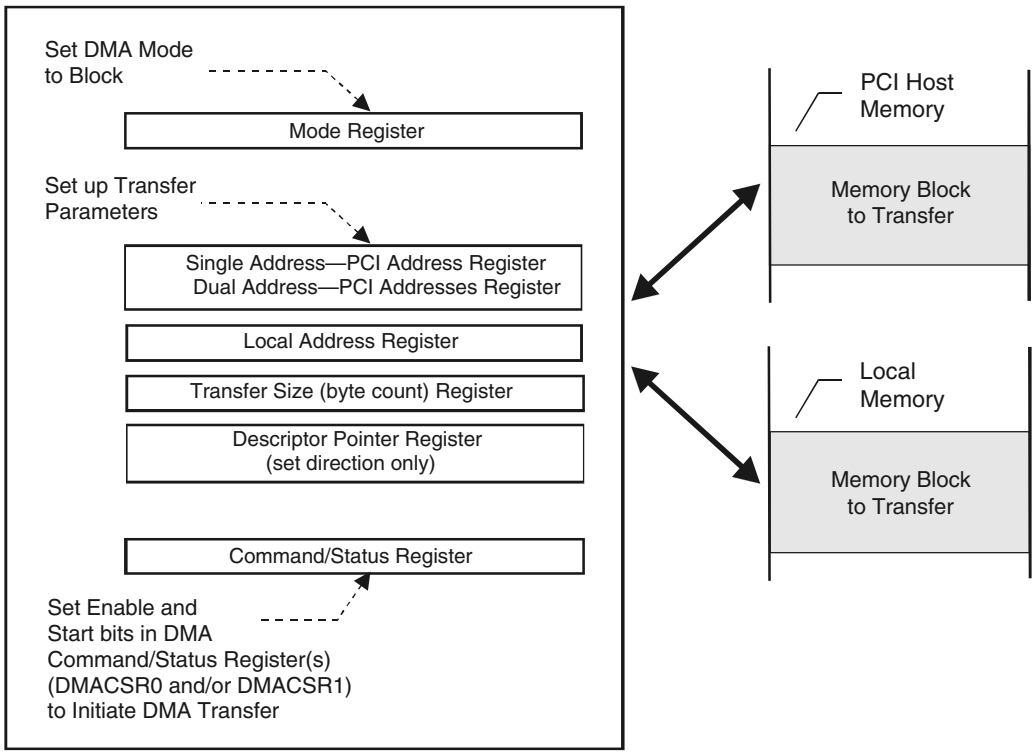


Figure 3-11. Block DMA Mode Initialization (Single Address or Dual Address PCI)

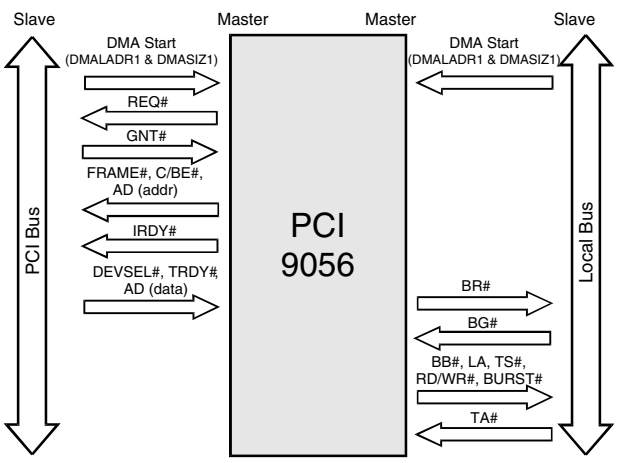


Figure 3-12. DMA, PCI-to-Local Bus

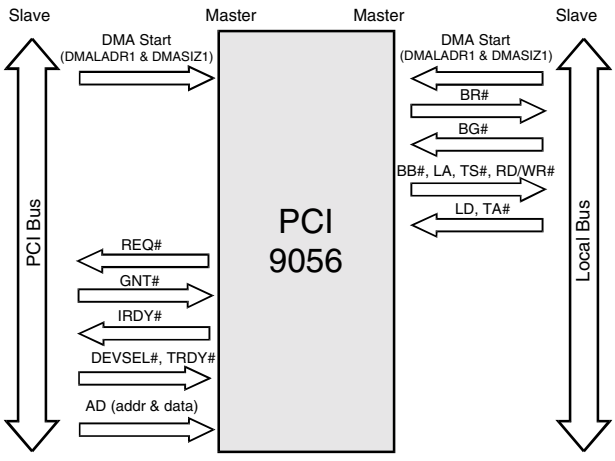


Figure 3-13. DMA, Local-to-PCI Bus

Note: *Figures 3-12 and 3-13 represent a sequence of Bus cycles.*

Table 3-5. DMA

BTERM# Input Enable Bit(s)	Fast/Slow Terminate Mode Select Bit(s)	PCI 9056 BDIP# Output
Enabled (1)	Disabled (1)	BDIP# is not asserted. Burst forever or until BI# asserts for one CLK cycle.
Enabled (1)	Enabled (0)	BDIP# is asserted until the last Data transfer, or until BI# asserts for one CLK cycle Burst forever. (Refer to Section 2.2.5.2.1.)
Disabled (0)	Disabled (1)	BDIP# is not asserted. Burst forever.
Disabled (0)	Enabled (0)	BDIP# is asserted by the PCI 9056. Burst up to 16 bytes (MPC850 or MPC860 compatible).

Table 3-6. Normal DMA with EOT Function

BTERM# Enable Bit(s)	Fast/Slow Terminate Mode Select Bit(s)	PCI 9056 BDIP# Output
Enabled (1)	Disabled (1)	BDIP# is not asserted. Immediate transfer terminated by EOT#. BDIP# is asserted until the last Data transfer, or until BI# asserts for one CLK cycle. (Refer to Section 2.2.5.2.1.)
Enabled (1)	Enabled (0)	BDIP# is asserted until the last Data transfer, or until BI# asserts for one CLK cycle Burst forever. (Refer to Section 2.2.5.2.1.)
Disabled (0)	Disabled (1)	BDIP# is not asserted. Immediate transfer terminated by EOT#.
Disabled (0)	Enabled (0)	BDIP# is asserted by the PCI 9056. Transfers up to the nearest 16-byte boundary, then terminates (MPC850 or MPC860 compatible).

Note: *If the Burst Enable bit is set to 0, the PCI 9056 performs single cycle transfers on the Local Bus.*

3.5.2.1 Block DMA PCI Dual Address Cycle

The PCI 9056 supports the DAC feature in Block DMA mode. Whenever the DMADAC0 and/or DMADAC1 register(s) contain a value of 0x00000000, the PCI 9056 performs a Single Address Cycle (SAC) on the PCI Bus. Any other value causes a Dual Address to appear on the PCI Bus. (Refer to Figure 3-14.)

3.5.3 Scatter/Gather DMA Mode

In Scatter/Gather DMA mode, the Host processor or Local processor sets up descriptor blocks in Local or Host memory composed of PCI and Local addresses, transfer count, transfer direction, and address of next descriptor block. (Refer to Figure 3-15 and Figure 3-16.) The Host or Local processor then:

- Enables the Scatter/Gather mode bit(s) (DMAMODE0[9]=1 and/or DMAMODE1[9]=1)
- Sets up the address of initial descriptor block in the PCI 9056 Descriptor Pointer register(s) (DMADPR0 and/or DMADPR1)
- Initiates the transfer by setting a control bit(s) (DMACSR0[1:0] and/or DMACSR1[1:0])

The PCI 9056 supports zero wait state Descriptor Block bursts from the Local and PCI Bus when the Local Burst Enable bit(s) is enabled (DMAMODE0[8]=1 and/or DMAMODE1[8]=1).

The PCI 9056 loads the first descriptor block and initiates the Data transfer. The PCI 9056 continues to load descriptor blocks and transfer data until it detects the End of Chain bit(s) is set (DMADPR0[1]=1 and/or DMADPR1[1]=1) (these bits are part of each descriptor). When the End of Chain bit(s) is detected, the PCI 9056 completes the current descriptor block and sets the DMA Done bit(s) (DMACSR0[4] and/or DMACSR1[4]). If the End of Chain bit(s) is detected, the PCI 9056 asserts a PCI interrupt (INTA#) and/or Local interrupt (LINTo#).

The PCI 9056 can also be programmed to assert PCI or Local interrupts after each descriptor is loaded, then finish transferring.

If Scatter/Gather descriptors are in Local memory, the DMA controller can be programmed to clear the transfer size at completion of each DMA, using the

DMA Clear Count Mode bit(s) (DMAMODE0[16] and/or DMAMODE1[16]).

Notes: *In Scatter/Gather DMA mode, the descriptor includes the PCI and Local Address Space, transfer size, and next descriptor pointer. It also includes a DAC value, if the DAC Chain Load bit(s) is enabled (DMAMODE0[18]=1 and/or DMAMODE1[18]=1). Otherwise, the register (DMADAC0 and/or DMADAC1) values are used.*

The Descriptor Pointer register(s) (DMADPR0 and/or DMADPR1) contains end of chain (bit 1), direction of transfer (bit 3), next descriptor address (bits [31:4]), interrupt after terminal count (bit 2), and descriptor location (bit 0) bits.

The Local Bus width must be the same as Local Memory Bus width.

A DMA descriptor can be on the Local memory or the PCI memory, or both (for example, one descriptor on Local memory, another descriptor on PCI memory and vice-versa).

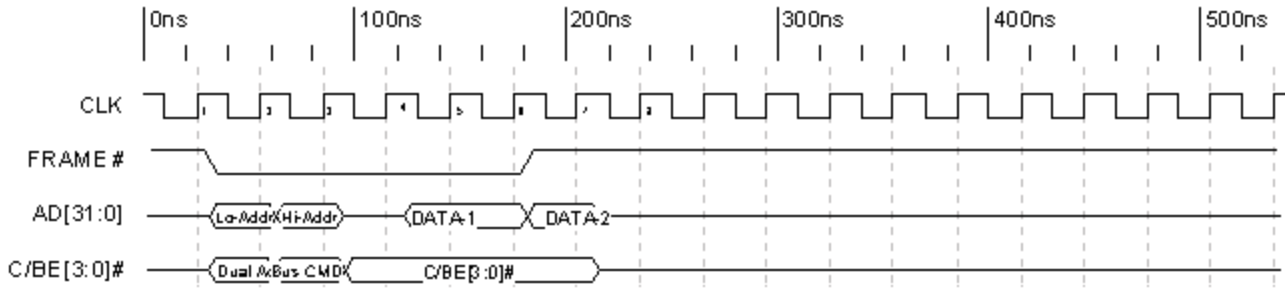


Figure 3-14. Dual Address Timing

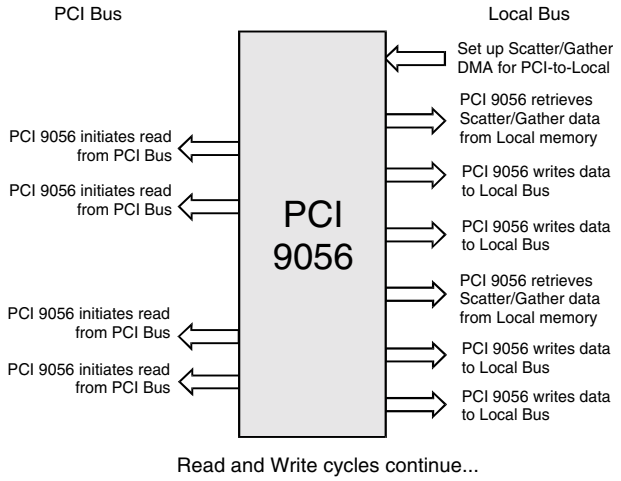


Figure 3-15. Scatter/Gather DMA Mode from PCI-to-Local Bus (Control Access from the Local Bus)

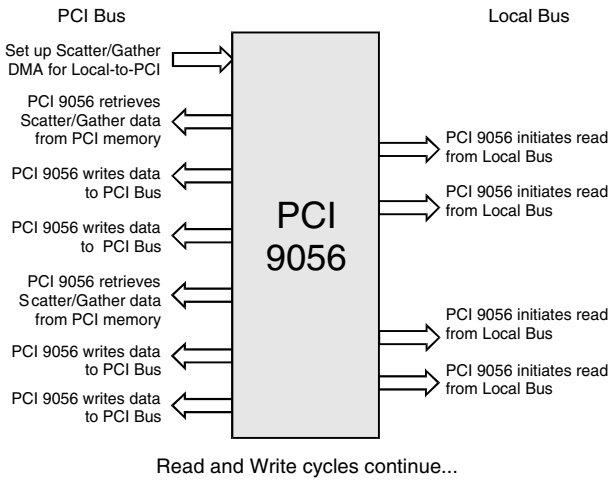


Figure 3-16. Scatter/Gather DMA Mode from Local-to-PCI Bus (Control Access from the PCI Bus)

Note: *Figures 3-15 and 3-16 represent a sequence of Bus cycles.*

3.5.3.1 Scatter/Gather DMA PCI Dual Address Cycle

The PCI 9056 supports the DAC feature in Scatter/Gather DMA mode for Data transfers only. The descriptor blocks should reside below the 4-GB Address space.

The PCI 9056 offers three different options of how PCI DAC Scatter/Gather DMA is utilized. Assuming the descriptor blocks are located on the PCI Bus:

- DMADAC0 and/or DMADAC1 contain(s) a non-zero value. DMAMODE0[18] and/or DMAMODE1[18] is set to 0. The PCI 9056 performs a Single Address Cycle (SAC) four-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. (Refer to Figure 3-17.)
- DMADAC0 and/or DMADAC1 contain(s) an 0x00000000 value. DMAMODE0[18] and/or DMAMODE1[18] is set to 1. The PCI 9056 performs a SAC five-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. (Refer to Figure 3-18.)
- DMADAC0 and/or DMADAC1 contain(s) a non-zero value. DMAMODE0[18] and/or DMAMODE1[18] is set to 1. The PCI 9056 performs a SAC five-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. The fifth descriptor overwrites the value of the DMADAC0 and/or DMADAC1 register(s). (Refer to Figure 3-18.)

3.5.3.2 DMA Clear Count Mode

The PCI 9056 supports DMA Clear Count mode (Write-Back feature, DMAMODE0[16] and/or DMAMODE1[16]). This feature allows users to control the Data transfer blocks during Scatter/Gather DMA operations. The PCI 9056 clears the Transfer Size descriptor to zero (0) by writing to a descriptor memory location at the end of each transfer chain. This feature is available for DMA descriptors located on the Local and PCI Buses.

3.5.3.3 DMA Descriptor Ring Management (Valid Mode)

In Scatter/Gather DMA mode, when the Valid Mode Enable bit(s) is set to 0 (DMAMODE0[20]=0 and/or DMAMODE1[20]=0), the Valid bit (bit 31 of transfer count) is ignored. When the Valid Mode Enable bit(s) is set to 1 (DMAMODE0[20]=1 and/or DMAMODE1[20]=1), the DMA descriptor proceeds only when the Valid bit is set. If the Valid bit is set, the transfer count is 0, and the descriptor is not the last descriptor, then the DMA controller moves on to the next descriptor in the chain.

When the Valid Stop Control bit(s) is set to 0 (DMAMODE0[21]=0 and/or DMAMODE1[21]=0), the DMA Scatter/Gather controller continuously polls the descriptor with the Valid bit set to 0 (invalid descriptor) until the Valid bit is read to be a 1. When the Valid Stop Control bit(s) is set to 1 (DMAMODE0[21]=1 and/or DMAMODE1[21]=1), the DMA Scatter/Gather controller pauses if a Valid bit with a value of 0 is detected. In this case, the PCI 9056 must restart the DMA controller by setting bit 1 of the DMA Control/Status register(s) (DMACSR0[1] and/or DMACSR1[1]). The DMA Clear Count mode bit(s) (DMAMODE0[16] and/or DMAMODE1[16]) must be enabled for the Ring Management Valid bit to be cleared at the completion of each descriptor.

3.5.4 DMA Memory Write and Invalidate

The PCI 9056 can be programmed to perform Memory Write and Invalidate cycles to the PCI Bus for DMA transfers, as well as Direct Master transfers. (Refer to Section 3.4.1.12.) The PCI 9056 supports Memory Write and Invalidate transfers for cache line sizes of 8 or 16 Lwords. Size is specified in the System Cache Line Size bits (PCICLSR[7:0]). If a size other than 8 or 16 is specified, the PCI 9056 performs Write transfers rather than Memory Write and Invalidate transfers.

DMA Memory Write and Invalidate transfers are enabled when the DMA controller Memory Write and Invalidate Enable bit(s) (DMAMODE0[13] and/or DMAMODE1[13]) and the Memory Write and Invalidate Enable bit (PCICR[4]) are set.

In Memory Write and Invalidate mode, the PCI 9056 waits until the number of Lwords required for specified cache line size are read from the Local Bus before starting the PCI access. This ensures a complete cache line write can complete in one PCI Bus ownership. If a target disconnects before a cache line completes, the PCI 9056 completes the remainder of that cache line, using normal writes before resuming Memory Write and

Invalidate transfers. If a Memory Write and Invalidate cycle is in progress, the PCI 9056 continues to burst if another cache line is read from the Local Bus before the cycle completes. Otherwise, the PCI 9056 terminates the burst and waits for the next cache line to be read from the Local Bus. If the final transfer is not a complete cache line, the PCI 9056 completes the DMA transfer, using normal writes.

EOT# signal assertion, in any DMA transfer type, or DREQ0# and/or DREQ1# signal de-assertion in Demand Mode before the cache line is read from the Local Bus, results in the PCI 9056 performing a normal PCI Memory Write to data read into a DMA FIFO.

3.5.4.1 DMA Abort

DMA transfers can be aborted, in addition to the EOT# signal, as follows:

1. Clear the DMA Channel Enable bit(s) (DMACSR0[0]=0 and/or DMACSR1[0]=0).
2. Abort DMA by setting the Channel Abort bit(s) (DMACSR0[2]=1 and/or DMACSR1[2]=1).
3. Wait until the Channel Done bit(s) is set (DMACSR0[4]=1 and/or DMACSR1[4]=1).

Note: *One to two Data transfers occur after the Abort bit is set. Aborting when no DMA cycles are in progress causes the next DMA to abort.*

3.5.5 DMA Priority

The DMA Channel Priority bits (MARBR[20:19]) can be used to specify the following priorities:

- Rotating (MARBR[20:19]=00)
- DMA Channel 0 (MARBR[20:19]=01)
- DMA Channel 1 (MARBR[20:19]=10)

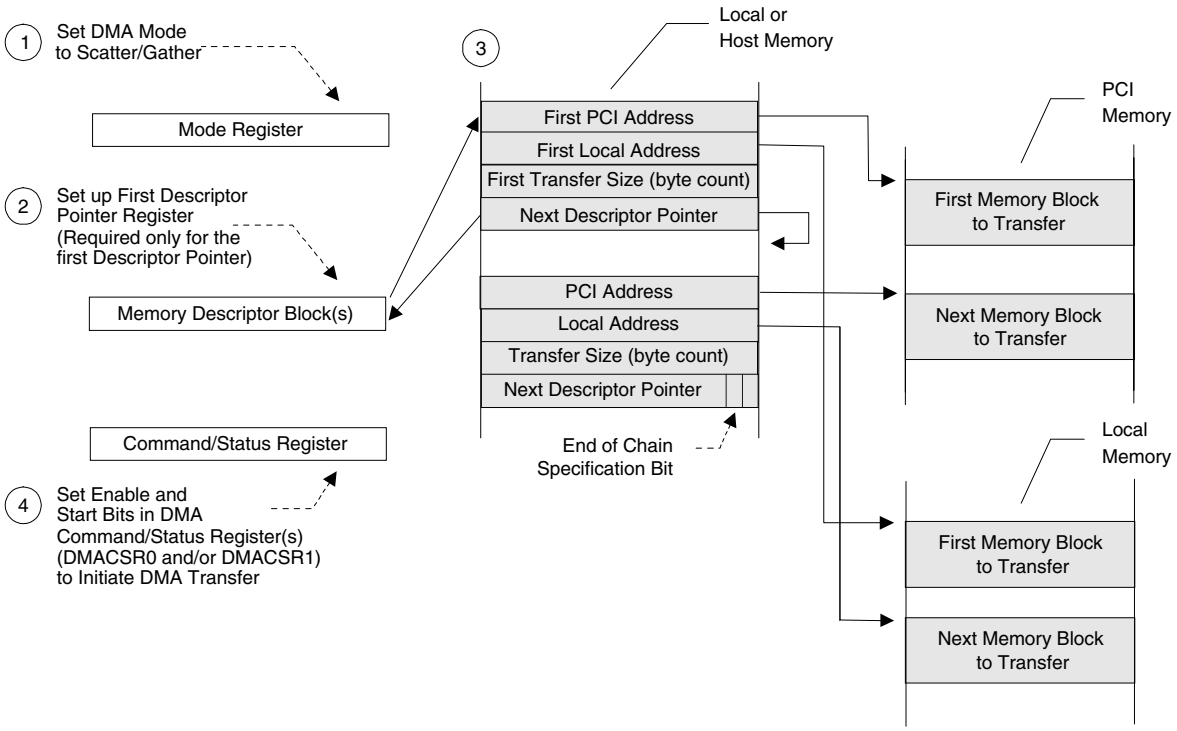


Figure 3-17. Scatter/Gather DMA Mode Descriptor Initialization [PCI SAC/DAC PCI Address (DMADAC0 and/or DMADAC1) Register Dependent]

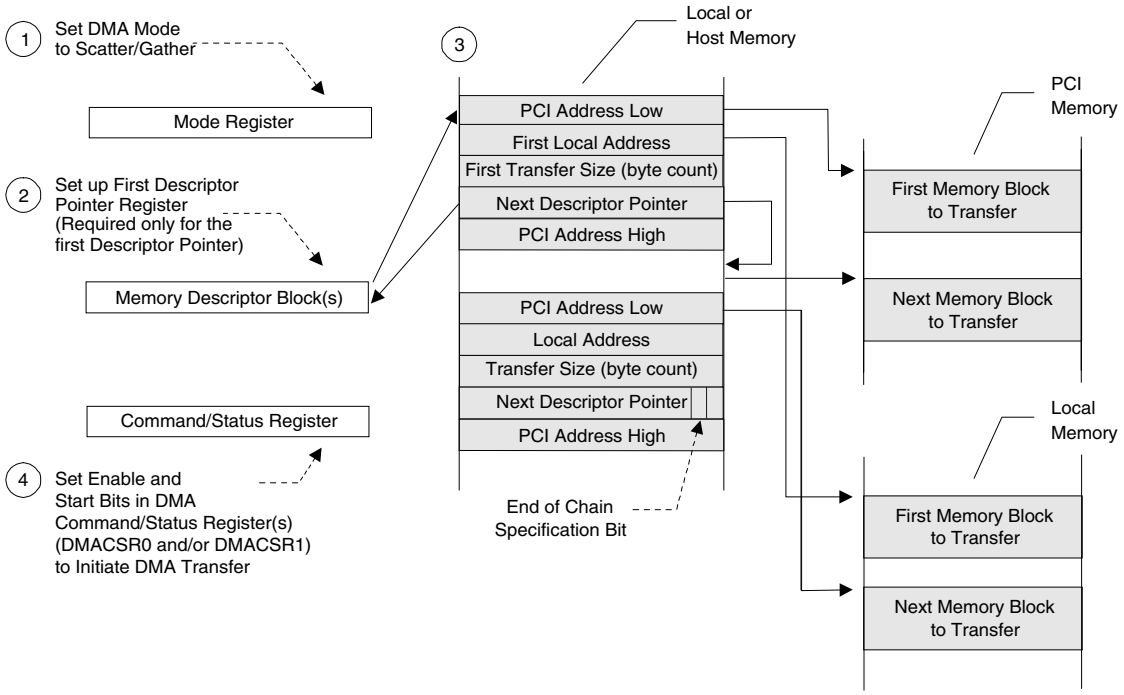


Figure 3-18. Scatter/Gather DMA Mode Descriptor Initialization [DAC PCI Address (DMAMODE0[18] and/or DMAMODE1[18]) Descriptor Dependent (PCI Address High Added)]

3.5.6 DMA Channel 0 and Channel 1 Interrupts

A DMA channel can assert a PCI Bus or Local Bus interrupt when done (transfer complete) or after a transfer is complete for the current descriptor in Scatter/Gather DMA mode. The DMA Channel Interrupt Select bit(s) determine whether to assert a PCI (DMAMODE0[17]=1 and/or DMAMODE1[17]=1) or Local (DMAMODE0[17]=0 and/or DMAMODE1[17]=0) interrupt. The PCI or Local processor can read the DMA Channel 0 Interrupt Active bits to determine whether a DMA Channel 0 (INTCSR[21]) or DMA Channel 1 (INTCSR[22]) interrupt is pending.

The Channel Done bit(s) (DMACSR0[4] and/or DMACSR1[4]) can be used to determine whether an interrupt is:

- DMA Done interrupt
- Transfer complete for current descriptor interrupt

The Done Interrupt Enable bit(s) (DMAMODE0[10] and/or DMAMODE1[10]) enable a Done interrupt. In Scatter/Gather DMA mode, a bit in the Next Descriptor Pointer register of the channel (loaded from Local memory) specifies whether to assert an interrupt at the end of the transfer for the current descriptor.

A DMA Channel interrupt is cleared by the Channel Clear Interrupt bit(s) (DMACSR0[3]=1 and/or DMACSR1[3]=1).

3.5.7 DMA Data Transfers

The PCI 9056 DMA controller can be programmed to transfer data from the Local-to-PCI Bus or from the PCI-to-Local Bus.

3.5.7.1 Local-to-PCI Bus DMA Transfer

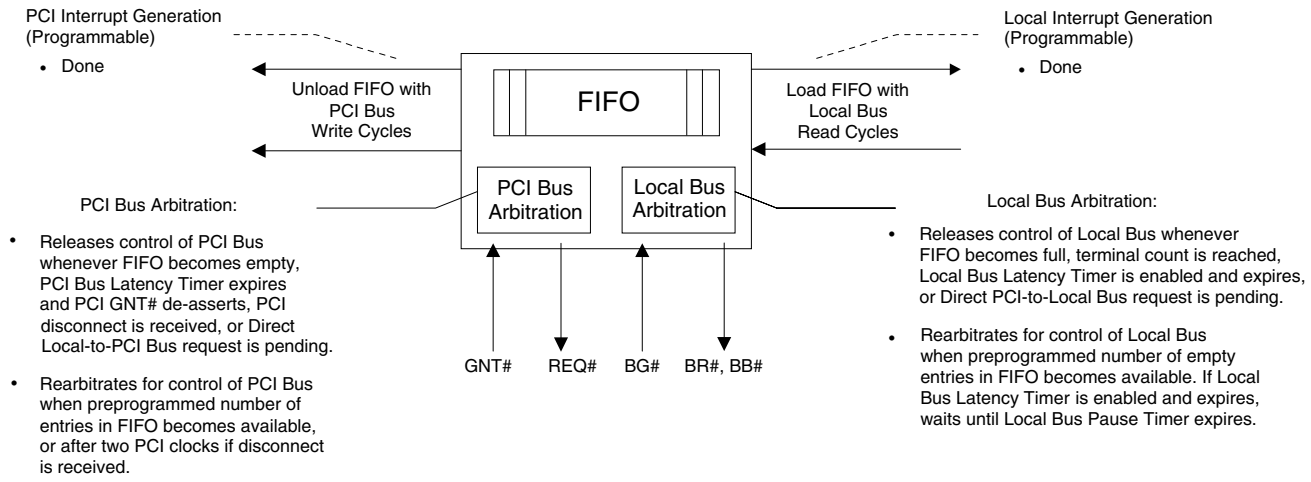


Figure 3-19. Local-to-PCI Bus DMA Data Transfer Operation

3.5.7.2 PCI-to-Local Bus DMA Transfer

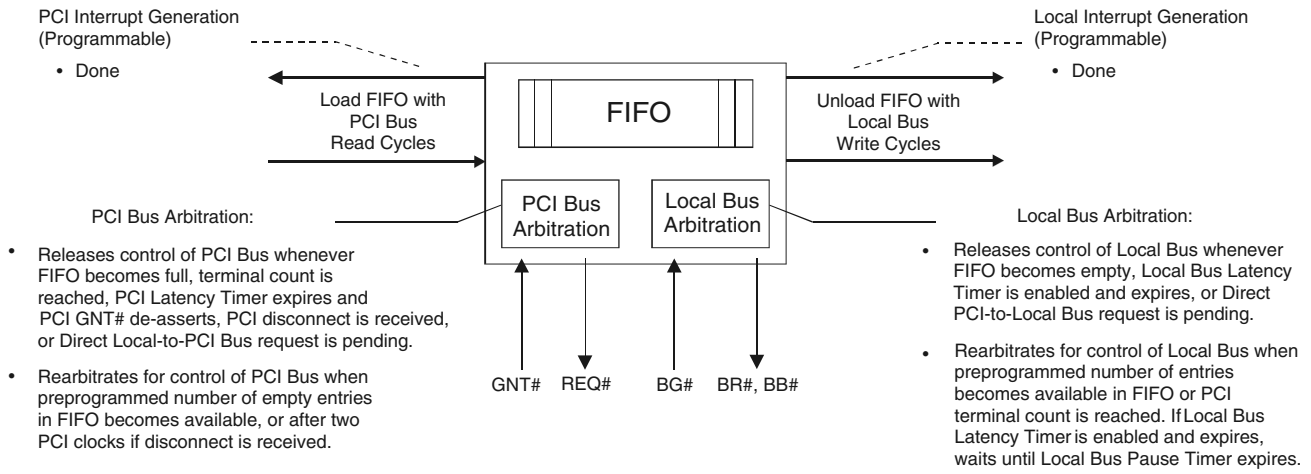


Figure 3-20. PCI-to-Local Bus DMA Data Transfer Operation

3.5.7.3 DMA Local Bus Error Condition

The PCI 9056 supports Local Bus error conditions with the TEA# signal. TEA# may be asserted by a device on the Local Bus, either before or simultaneously with TA#. In either case, the PCI 9056 attempts to finish the current transaction by transferring data and then asserting TS# for every address that follows, waiting for another TA# or TEA# to be issued to flush the FIFOs. After sensing TEA# is asserted, the PCI 9056 asserts PCI SERR# and sets the Signaled System Error bit (PCISR[14], indicating a catastrophic error occurred on the Local Bus. SERR# may be masked by resetting the TEA# Input Interrupt Mask bit (LMISC1[5]=0).

The PCI 9056 Local Bus Latency Timer (MARBR[7:0]), as well as the Local Bus Pause Timer (MARBR[15:8]), can be used to better utilize the Local Bus.

3.5.7.4 DMA Unaligned Transfers

For unaligned Local-to-PCI transfers, the PCI 9056 reads a partial Lword from the Local Bus. It continues to perform a single cycle read (Lwords) from the Local Bus until the nearest 16-byte boundary. If the Burst Mode bit is enabled, the PCI 9056 bursts thereafter. Lwords are assembled, aligned to the PCI Bus address, and loaded into the FIFO until the nearest 16-byte boundary.

For PCI-to-Local transfers, Lwords are read from the PCI Bus and loaded into the FIFO. On the Local Bus, Lwords are assembled from the FIFO, aligned to the Local Bus address and single cycle written to the Local Bus until the nearest 16-byte boundary. If burst functionality is enabled, the PCI 9056 bursts thereafter.

3.5.8 Demand Mode DMA, Channel 0 and Channel 1

The Fast/Slow Terminate Mode Select bit(s) (DMAMODE0[15] and/or DMAMODE1[15]) determines the number of Lwords to transfer after the DMA controller DREQ0# and/or DREQ1# input is de-asserted.

If BDIP# output is not required to be driven by the PCI 9056 for a DMA transfer (bit [15]=1), the DMA controller releases the data bus after it receives an external TA# or the internal wait state counter decrements to 0 for the current Lword.

When the PCI 9056 is in Demand Mode DMA Local-to-PCI Fast Terminate mode (DMAMODE0[15] and/or DMAMODE1[15]) unaligned DMA transfers, it monitors PCI address increments to guarantee Qword PCI data, 32-bit data completion when DREQ0# and/or DREQ1# is de-asserted in the middle of the Data-Pocket transfer, Demand Mode DMA pause.

Due to the nature of unaligned transfers, the PCI 9056 retains partial data, and seven or fewer bytes remaining in the DMA FIFO are not transferred when DREQ0# and/or DREQ1# is de-asserted in the middle of the Data-Pocket transfer.

When DREQ0# and/or DREQ1# resumes, the data is transferred to the PCI Bus. If the DREQ0# and/or DREQ1# assertion does not resume for ongoing transfers, the EOT# signal assertion (along with DREQ0# and/or DREQ1# de-assertion) should be used to ensure the partial data successfully transfers to the PCI Bus.

These same conditions for DMA PCI-to-Local cause the PCI 9056 to immediately pause the DMA transfer on the Local Bus at Lword boundary. EOT# assertion (along with DREQ0# and/or DREQ1# de-assertion) causes the PCI 9056 to immediately terminate the ongoing Data transfer and flush the DMA FIFO.

If BDIP# output must be driven by the PCI 9056 for the DMA transfer (bit [15]=0), the DMA controller continues transferring data up to the nearest 16-byte boundary. If DREQ0# and/or DREQ1# is de-asserted, or the Local Latency Timer expired (MARBR register) during the Address phase of the first transfer in PCI 9056 Local Bus ownership (TS#, BG# asserted), the DMA controller completes a 16-byte transfer. If DREQ0# and/or DREQ1# is de-asserted, or the Local Latency Timer expired (MARBR register) during a Data-Transfer phase, one Lword before the last 16-byte transfer, the PCI 9056 finishes the transfer and performs an additional 16-byte transfer to satisfy BDIP# de-assertion protocol. (Refer to Table 3-7.)

Table 3-7. Demand Mode DMA, Channel 0 and Channel 1

BTERM# Input Enable Bit(s)	Fast/Slow Terminate Mode Select Bit(s)	PCI 9056 BDIP# Output
Enabled (1)	Disabled (1)	BDIP# is not asserted. Immediate transfer terminated by EOT#, or until BI# asserts for one CLK cycle. (Refer to Section 2.2.5.2.1.)
Enabled (1)	Enabled (0)	BDIP# is asserted by the PCI 9056 until the last Data transfer, or until BI# asserts for one CLK cycle. (Refer to Section 2.2.5.2.1.)
Disabled (0)	Disabled (1)	BDIP# is not asserted. Immediate transfer terminated by EOT#, or until BI# asserts for one CLK cycle. (Refer to Section 2.2.5.2.1.)
Disabled (0)	Enabled (0)	BDIP# is asserted by the PCI 9056. Transfers up to the nearest 16-byte boundary, then terminates (MPC850 or MPC860 compatible).

When the PCI 9056 is in Demand Mode DMA Local-to-PCI Slow Terminate mode (DMAMODE0[15] and/or DMAMODE1[15]) unaligned DMA transfers, it monitors PCI address increments to guarantee a Qword PCI data, 32-bit data completion when DREQ0# and/or DREQ1# is de-asserted in the middle of the Data-Pocket transfer, Demand Mode DMA pause. Due to the nature of unaligned transfers, the PCI 9056 retains partial Qword data, seven or fewer bytes remain in the DMA FIFO and are not transferred when DREQ0# and/or DREQ1# is de-asserted in the middle of the Data-Pocket transfer. When DREQ0# and/or DREQ1# resumes, the data is transferred to the PCI Bus. If DREQ0# and/or DREQ1# assertion is never resumed for ongoing transfers, the EOT# signal assertion (along with DREQ0# and/or DREQ1# de-assertion) should be used to ensure the partial data successfully transfers to the PCI Bus.

These same conditions for DMA PCI-to-Local cause the PCI 9056 to pause the DMA transfer on the Local Bus at the Qword Address or Lword Data boundary, dependent upon BTERM# Input Enable bit. (Refer to Section 2.2.5.2.1.) EOT# assertion (along with DREQ0# and/or DREQ1# de-assertion) causes the PCI 9056 to terminate the ongoing Data transfer and flush the DMA FIFO.

3.5.9 End of Transfer (EOT#) Input

The DMA EOT# Enable bit(s) (DMAMODE0[14] and/or DMAMODE1[14]) determines the number of Lwords to transfer after a DMA controller asserts EOT# input. EOT# input should be asserted only when the PCI 9056 owns a bus. (Refer to Table 3-8.)

If BDIP# output is not required to be driven by the PCI 9056 for the DMA transfer (DMAMODE0[15]=1 and/or DMAMODE1[15]=1), and the DMA EOT# Enable bit(s) is set (DMAMODE0[14]=1 and/or DMAMODE1[14]=1), the DMA controller releases the data bus and terminates DMA after receiving an external TA# signal. Or, the internal wait state counter decrements to 0 for the current Lword when EOT# is asserted.

If BDIP# output must be driven by the PCI 9056 for the DMA transfer (DMAMODE0[15]=0 and/or DMAMODE1[15]=0), the DMA controller transfers data up to the nearest 16-byte boundary if EOT#, (DMAMODE0[14]=1 and/or DMAMODE1[14]=1) is asserted and enabled.

When the BTERM# Enable bit is disabled, Fast/Slow Terminate is enabled, and EOT# is asserted during the Data-Transfer phase of the last four bytes of a 16-byte transfer, the PCI 9056 completes the transfer and performs an additional 16-byte transfer to satisfy the BDIP# de-assertion protocol. Otherwise, it completes the current 16-byte transfer.

When the BTERM# Enable bit is enabled, or the BTERM# Enable bit is disabled and Fast/Slow Terminate is disabled, the DMA controller terminates a transfer on an Lword boundary after EOT# is asserted. For an 8-bit bus, the PCI 9056 terminates after transferring the last byte for the Lword. For a 16-bit bus, the PCI 9056 terminates after transferring the last word for the Lword. In Single Cycle mode (burst disabled), the transfer is terminated at the next Lword boundary after EOT# occurs. The exception to this is when EOT# occurs on the last four bytes of the Transfer Count setting.

During the descriptor loading on the Local Bus, EOT# assertion causes a complete descriptor load and no subsequent Data transfer; however, this is not recommended. This has no effect when the descriptor is loaded from the PCI Bus.

Table 3-8. Any DMA Transfer Channel 0 and Channel 1 with EOT Functionality

BTERM# Enable Bit(s)	Fast/Slow Terminate Mode Select Bit(s)	PCI 9056 BDIP# Output
Enabled (1)	Disabled (1)	BDIP# is not asserted. Transfer is immediately terminated by EOT# or paused by DREQ# at Lword boundary, or until BI# asserts for one CLK cycle. (Refer to Section 2.2.5.2.1.)
Enabled (1)	Enabled (0)	BDIP# is asserted by the PCI 9056 until last Data transfer. Transfer is immediately terminated by EOT# or paused by DREQ# at Lword boundary.
Disabled (0)	Disabled (1)	BDIP# is not asserted. Transfer is immediately terminated by EOT# or paused by DREQ# at Lword boundary.
Disabled (0)	Enabled (0)	BDIP# is asserted by the PCI 9056. Transfers up to the nearest 16-byte boundary, then terminates (MPC850 or MPC860 compatible).

3.5.10 DMA Arbitration

The PCI 9056 asserts BR# when it needs to be the Local Bus Master. Upon receiving BG#, the PCI 9056 waits for BB# to be de-asserted. The PCI 9056 then asserts BB# at the next rising edge of the Local clock after sensing that BB# is de-asserted (no other device is acting as Local Bus Master). The PCI 9056 continues to assert BB# while acting as the Local Bus Master (*that is*, it holds the bus until instructed to release BB#) under the following conditions:

- Local Bus Latency Timer is enabled and expires (MARBR[7:0])
- Direct Slave access is pending
- EOT# input is received (if enabled)

The DMA controller releases control of the PCI Bus when one of the following conditions occurs:

- FIFOs are full or empty
- PCI Bus Latency Timer expires (PCILTR[7:0])—and loses the PCI GNT# signal
- Target disconnect response is received

The DMA controller de-asserts PCI REQ# for a minimum of two PCI clocks.

3.5.11 Local Bus Latency and Pause Timers

The Local Bus Latency and Pause Timers are programmable with the Mode/DMA Arbitration register (MARBR[7:0, 15:8], respectively). If the Local Bus Latency Timer is enabled and expires, the PCI 9056 completes an Lword transfer up to the nearest 16-byte boundary and releases the Local Bus, de-asserting BB#. After the programmable Pause Timer expires, it arbitrates for the bus by asserting BR#. When it receives BG#, it asserts BB# and continues to transfer until the FIFO is empty for a Local-to-PCI transfer or full for a PCI-to-Local transfer.

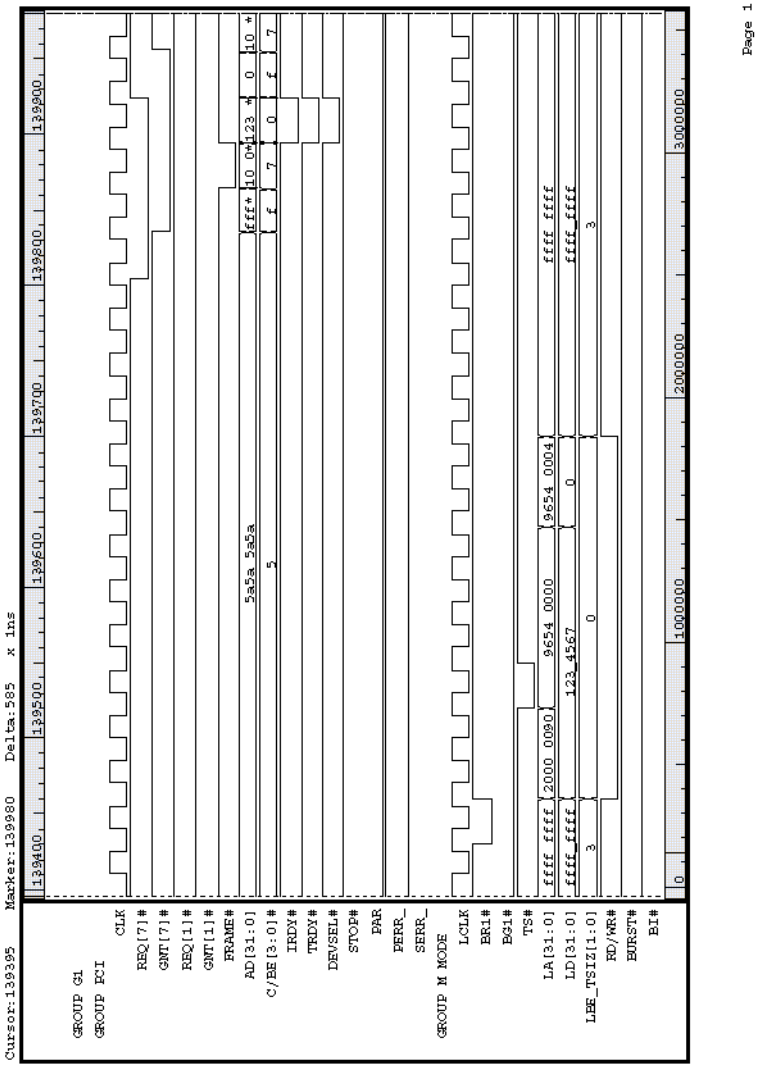
The DMA transfer can be paused by writing a 0 to the Channel Enable bit. To acknowledge the disable, the PCI 9056 gets at least one data from the bus before it stops. However, this is not recommended during a burst.

The DMA Local Bus Timer starts after the Local Bus is granted to the PCI 9056 and the Local Pause Timer starts after BB# is de-asserted.

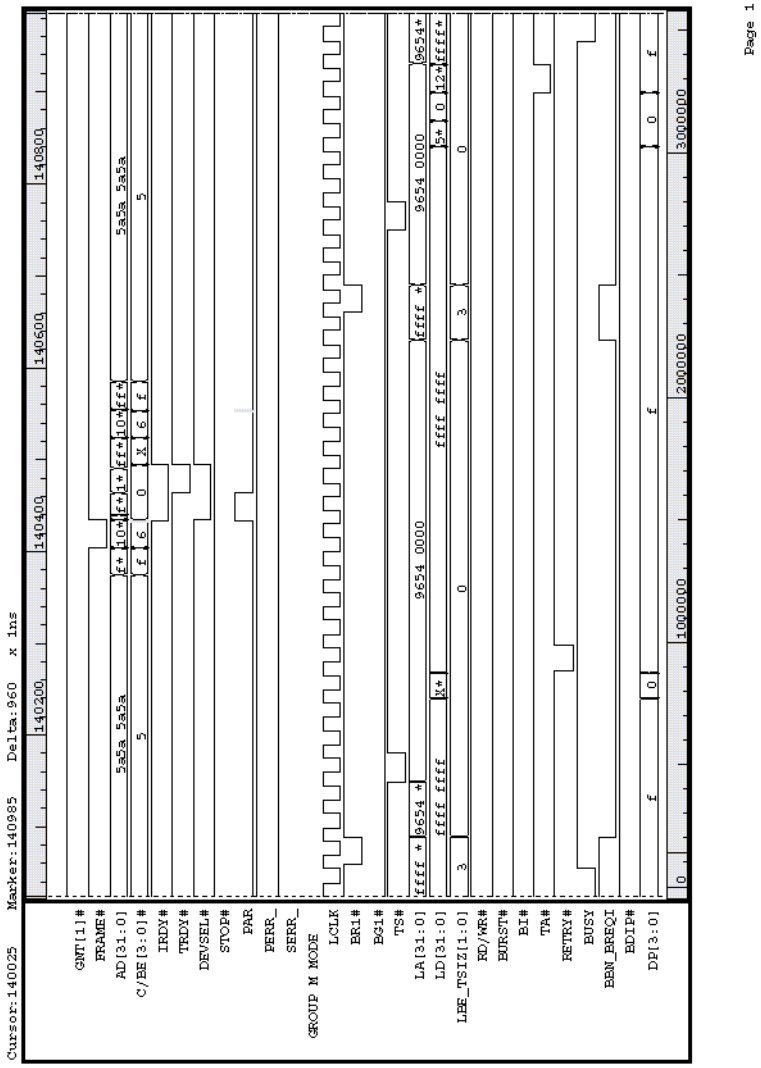
3.6 M MODE TIMING DIAGRAMS

Note: *In the timing diagrams that follow, the "_" symbol at the end of the signal names represents the "#" symbol.*

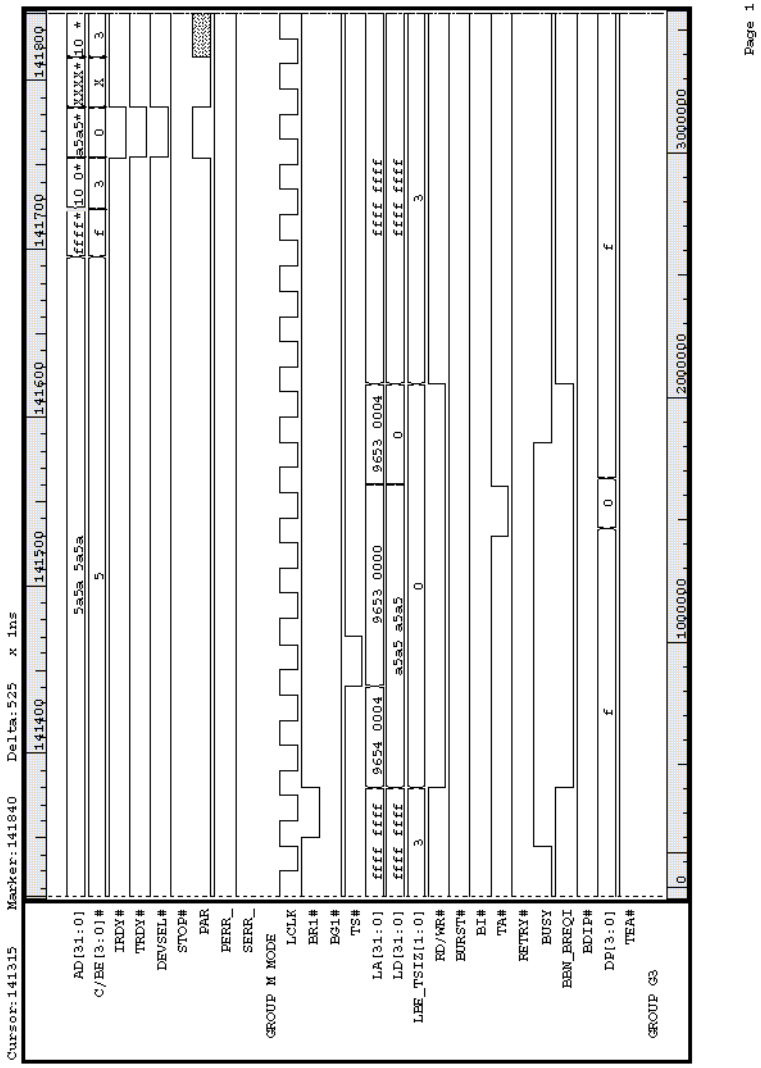
3.6.1 M Mode Direct Master Timing Diagrams



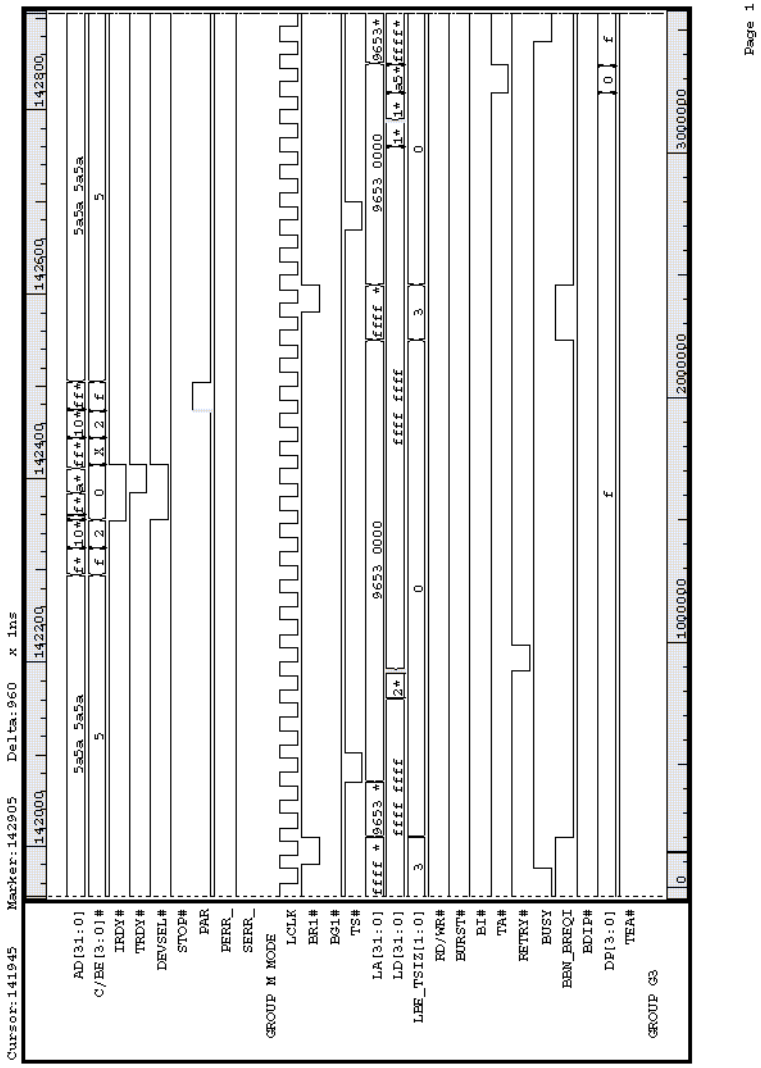
Timing Diagram 3-1. Direct Master Single Write to PCI Memory Space



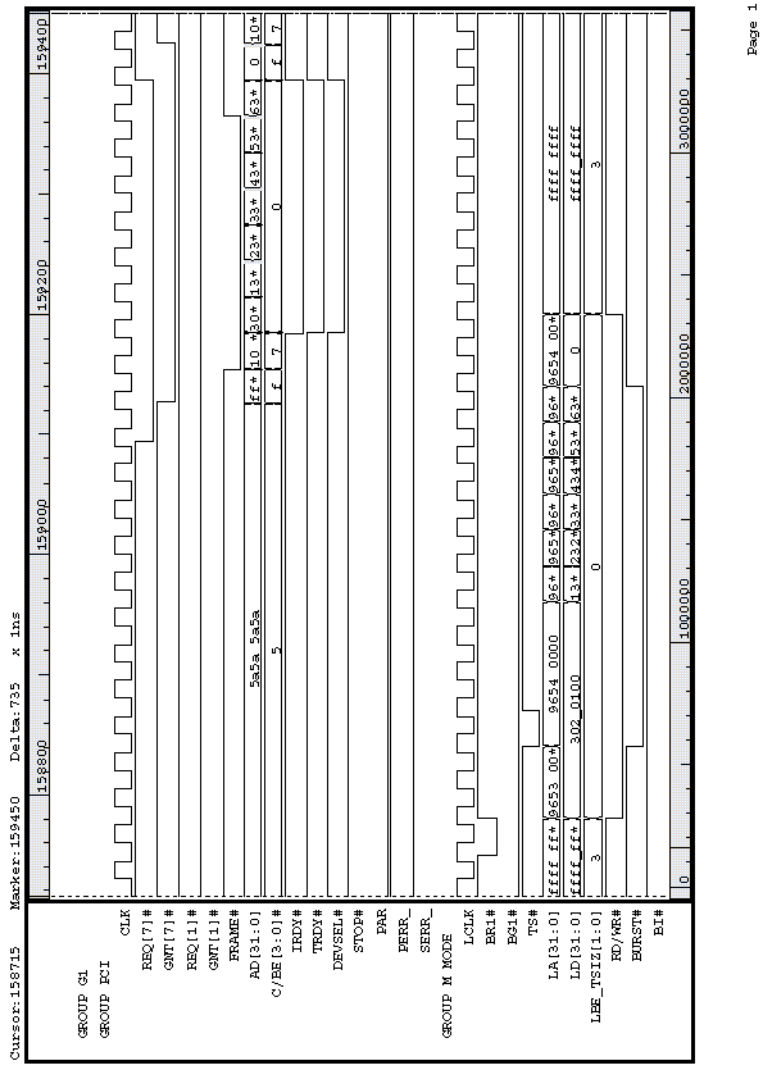
Timing Diagram 3-2. Direct Master Single Read from PCI Memory Space



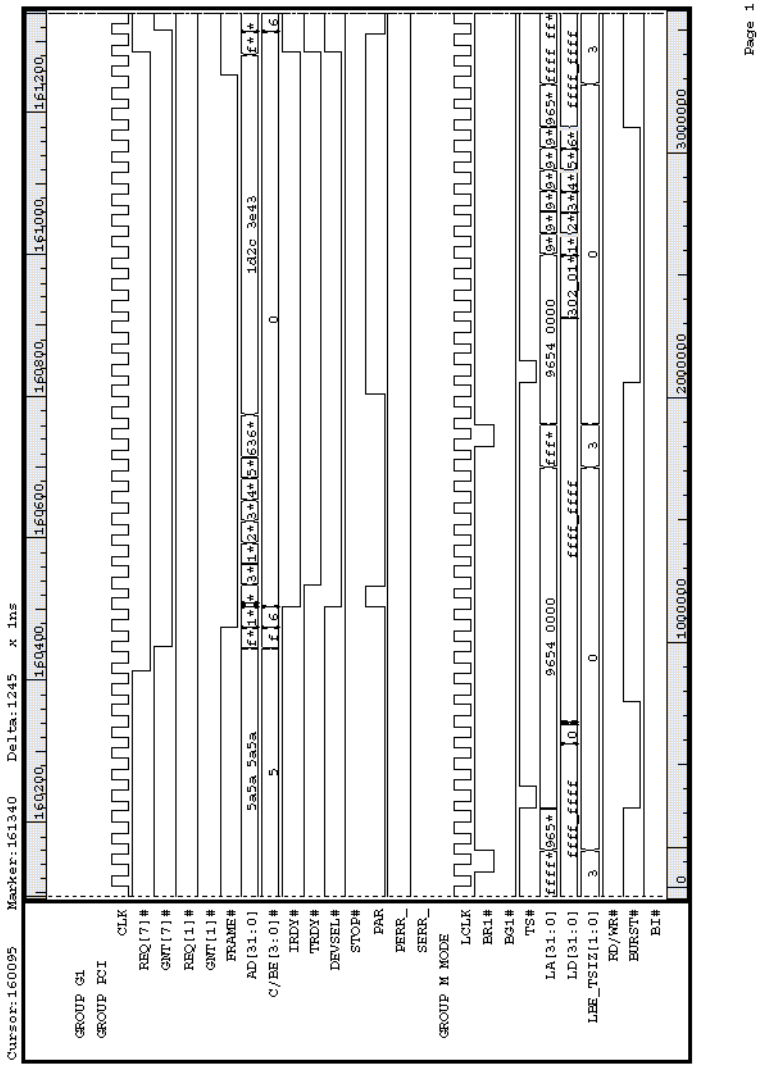
Timing Diagram 3-3. Direct Master Single Write to PCI I/O Space



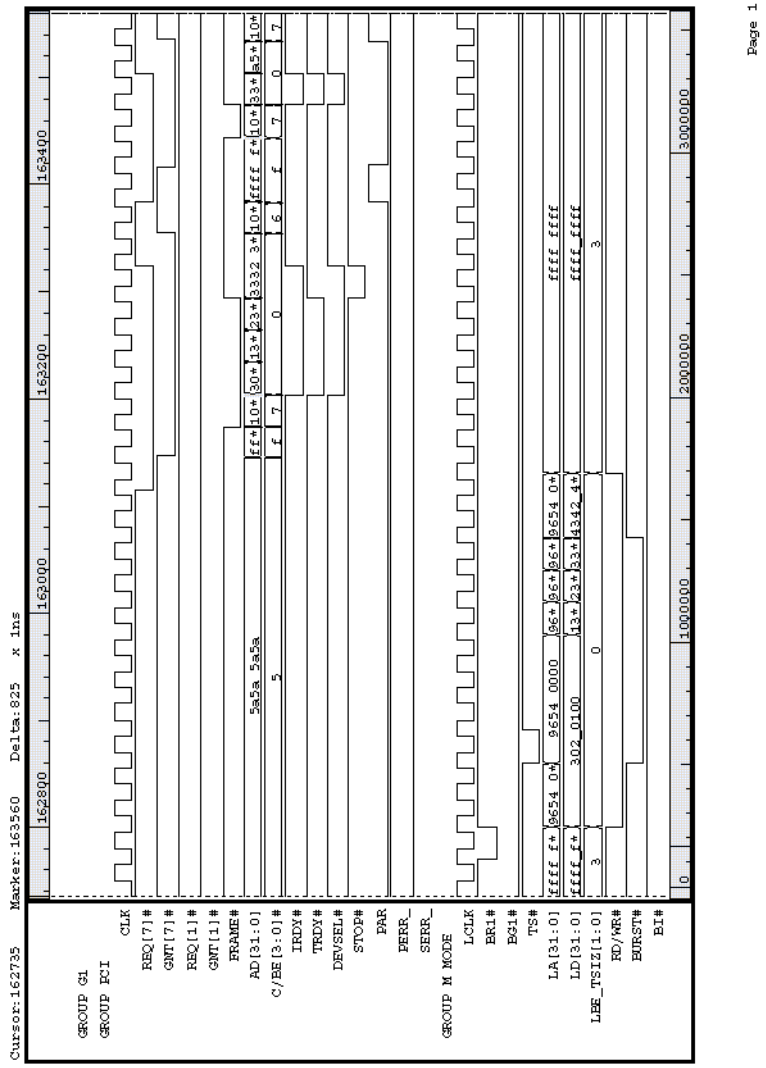
Timing Diagram 3-4. Direct Master Single Read from PCI I/O Space



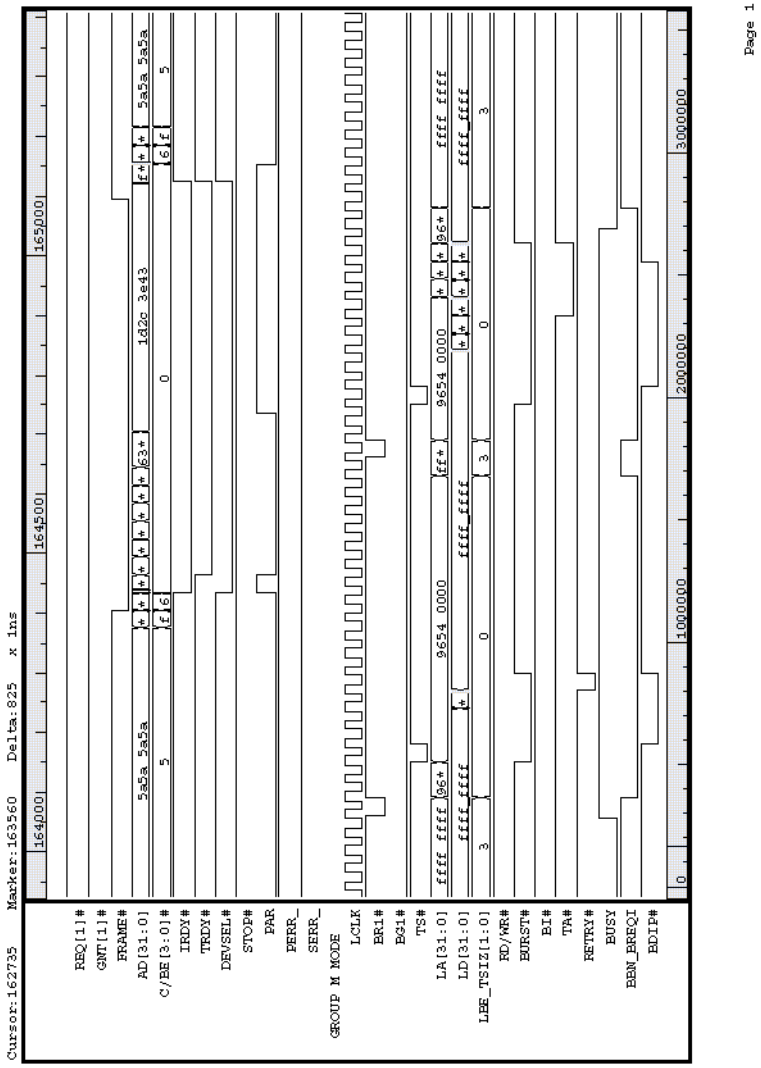
Timing Diagram 3-5. Direct Master Burst Write to PCI Memory Space



Timing Diagram 3-6. Direct Master Burst Read from PCI Memory Space

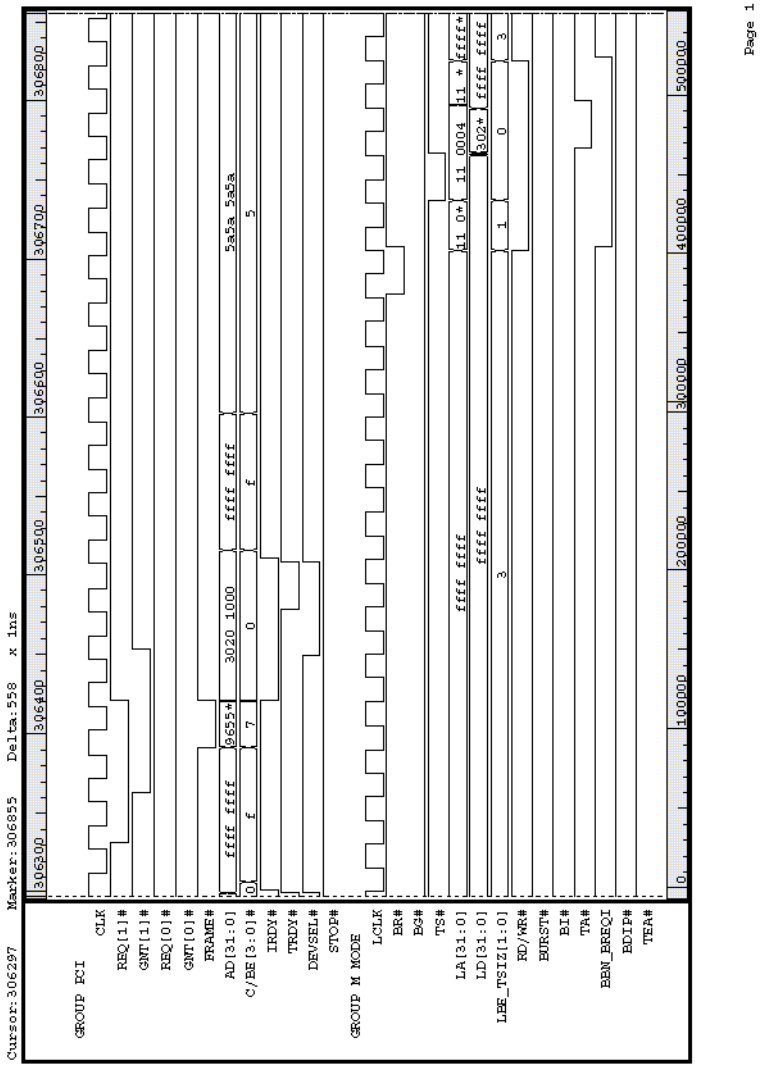


Timing Diagram 3-7. Direct Master Burst Write with a Retry on PCI Bus

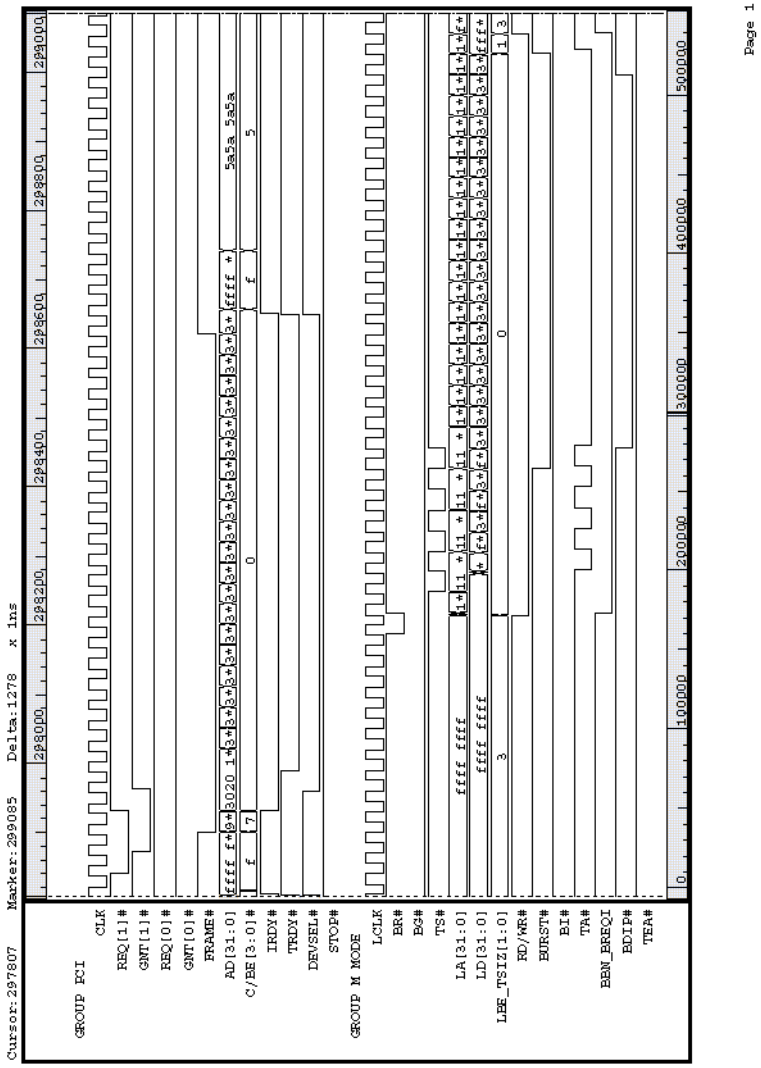


Timing Diagram 3-8. Direct Master Burst Read with a Retry on Local Bus

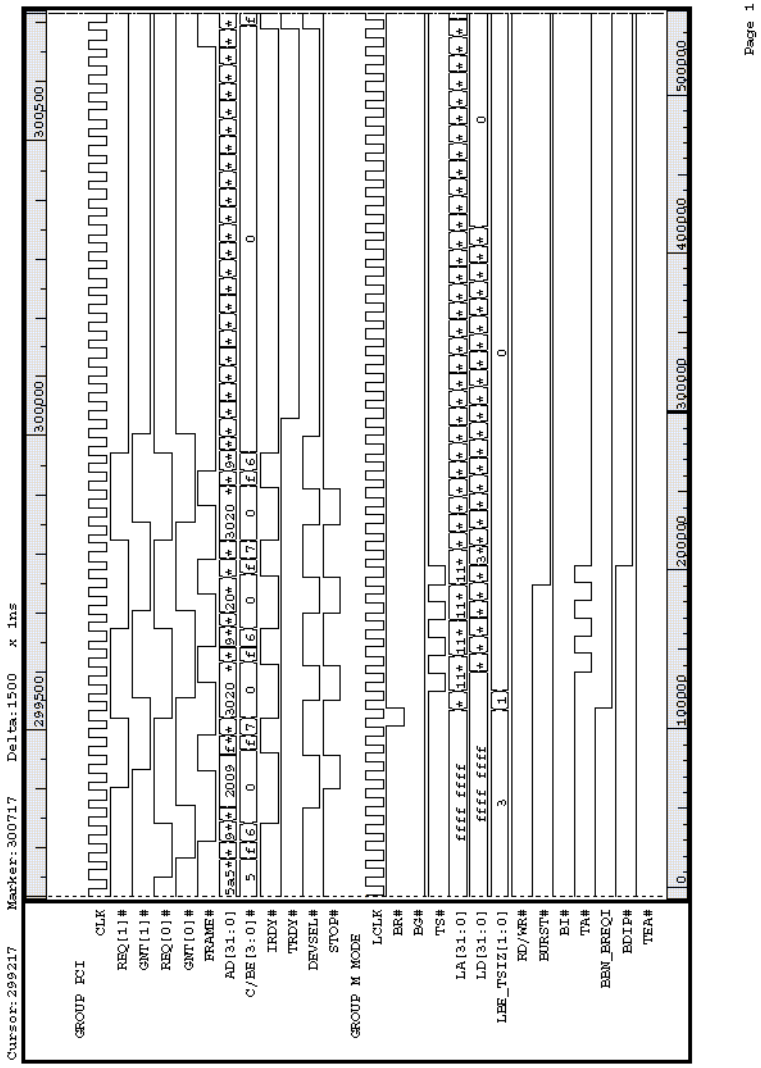
3.6.2 M Mode Direct Slave Timing Diagrams



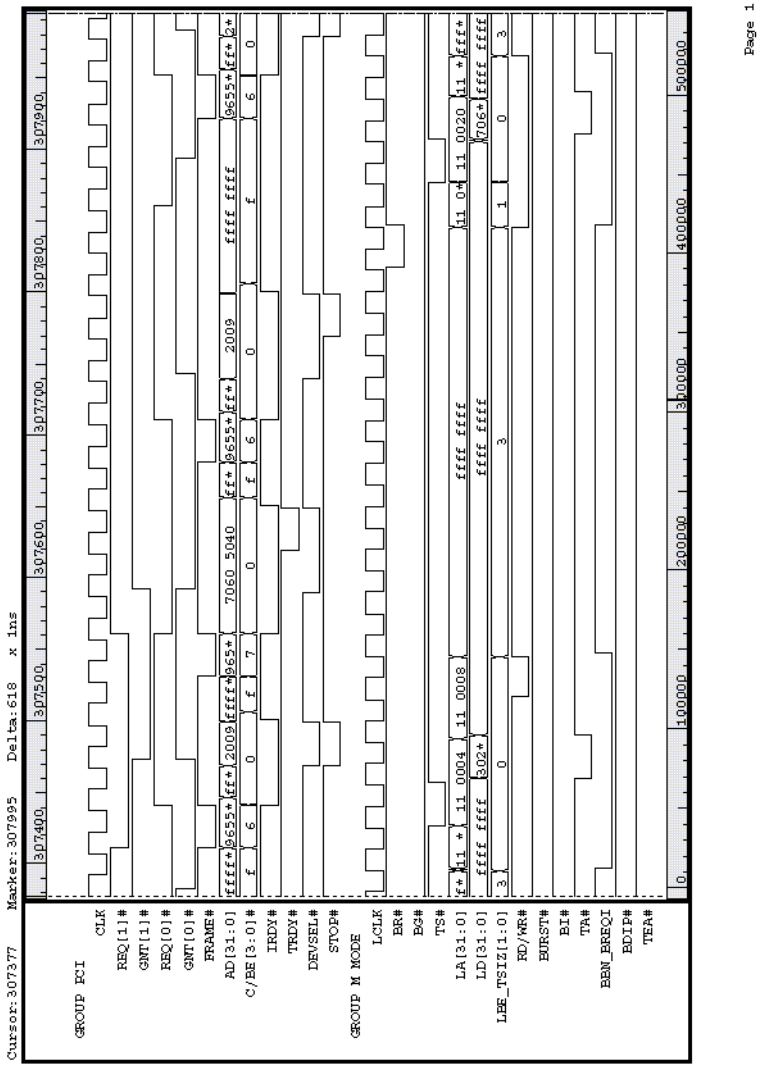
Timing Diagram 3-9. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Single Write



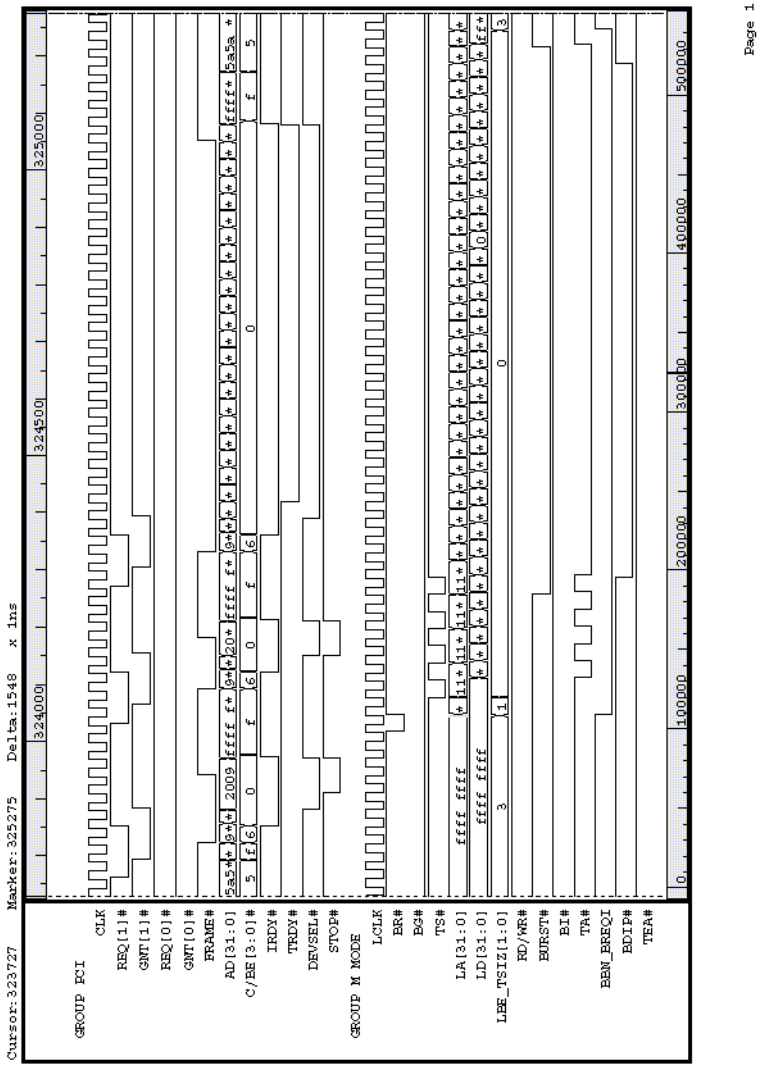
Timing Diagram 3-10. Direct Slave Burst Write



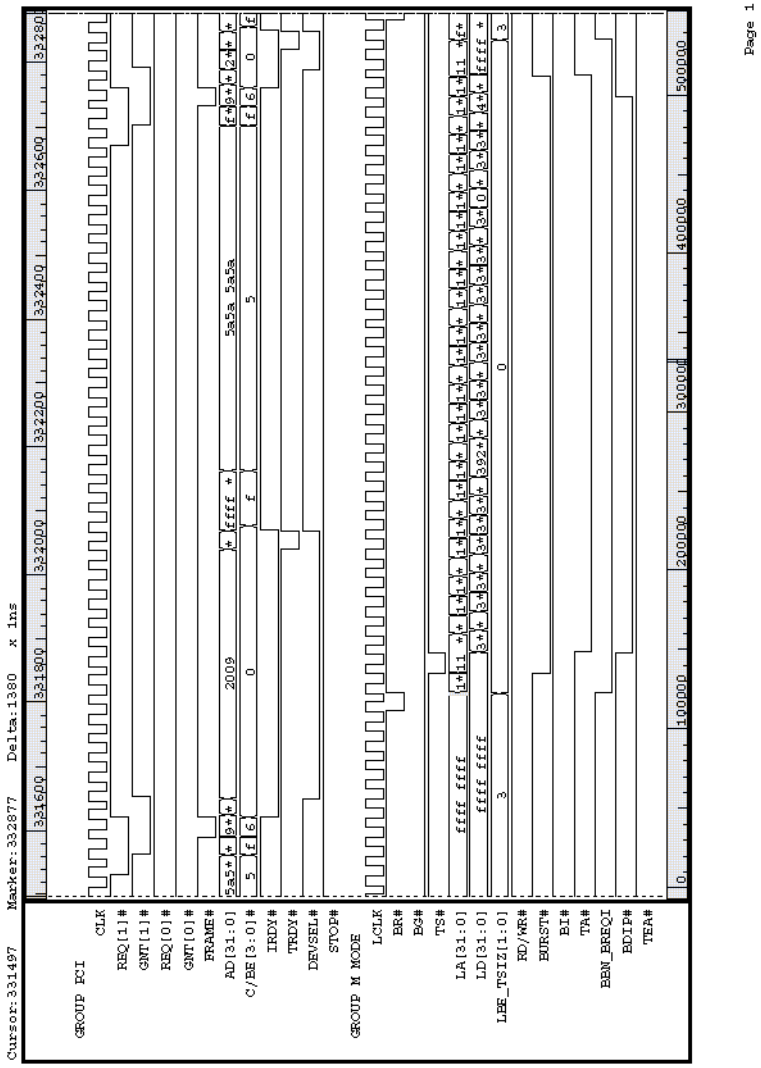
Timing Diagram 3-11. Direct Slave Burst Read



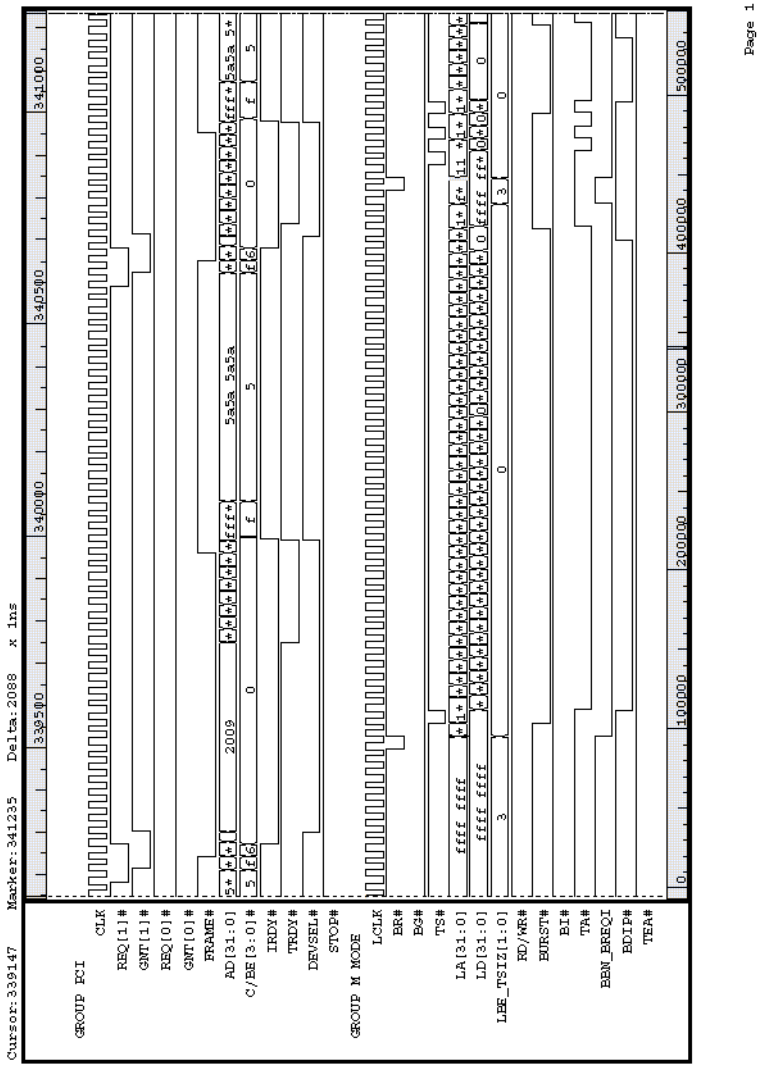
Timing Diagram 3-12. Direct Slave Single Write



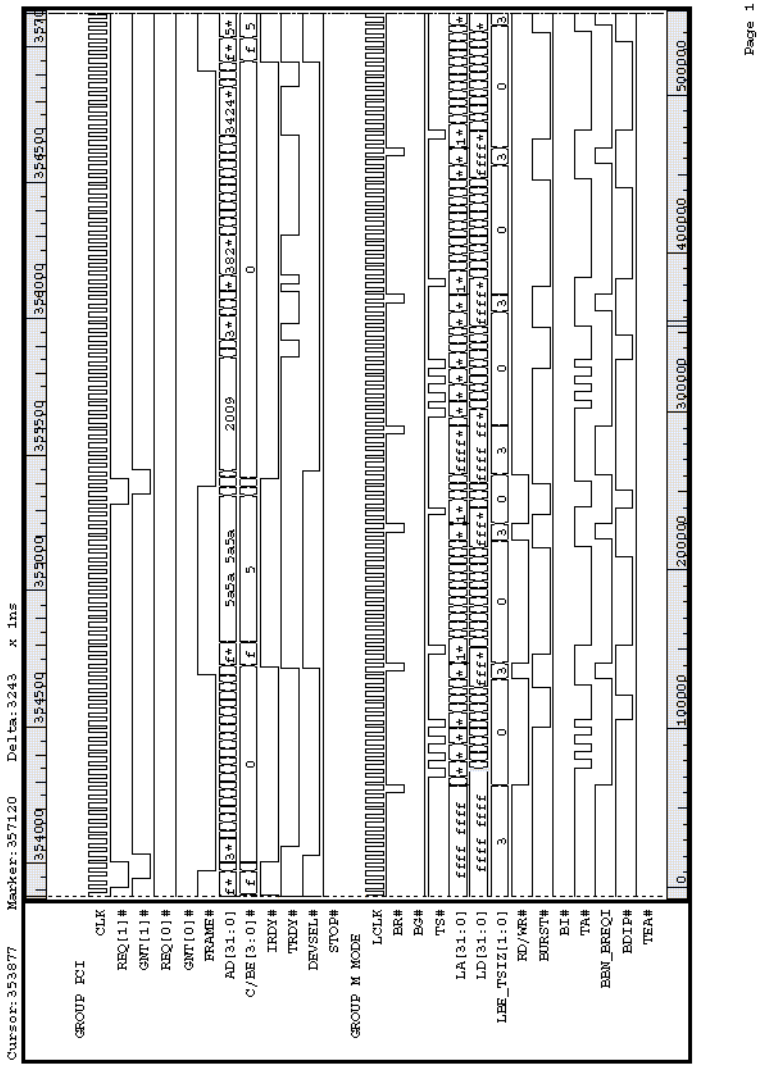
Timing Diagram 3-13. Direct Slave Delay Burst Read



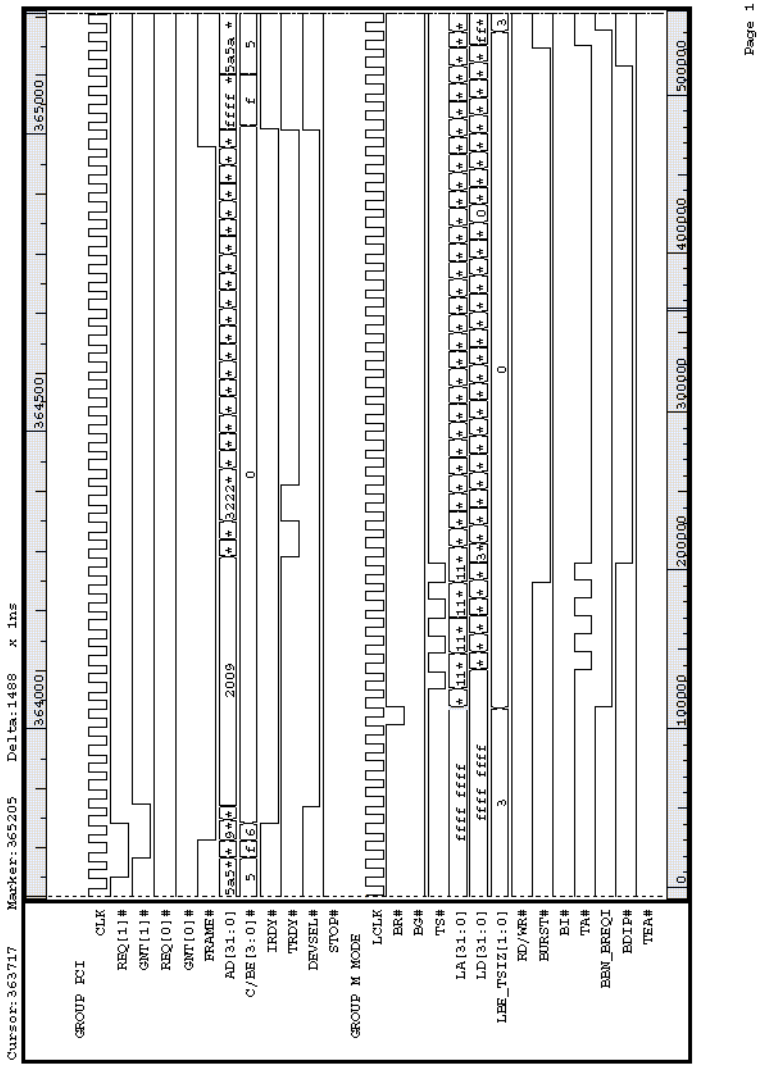
Timing Diagram 3-14. Direct Slave Single Read Ahead



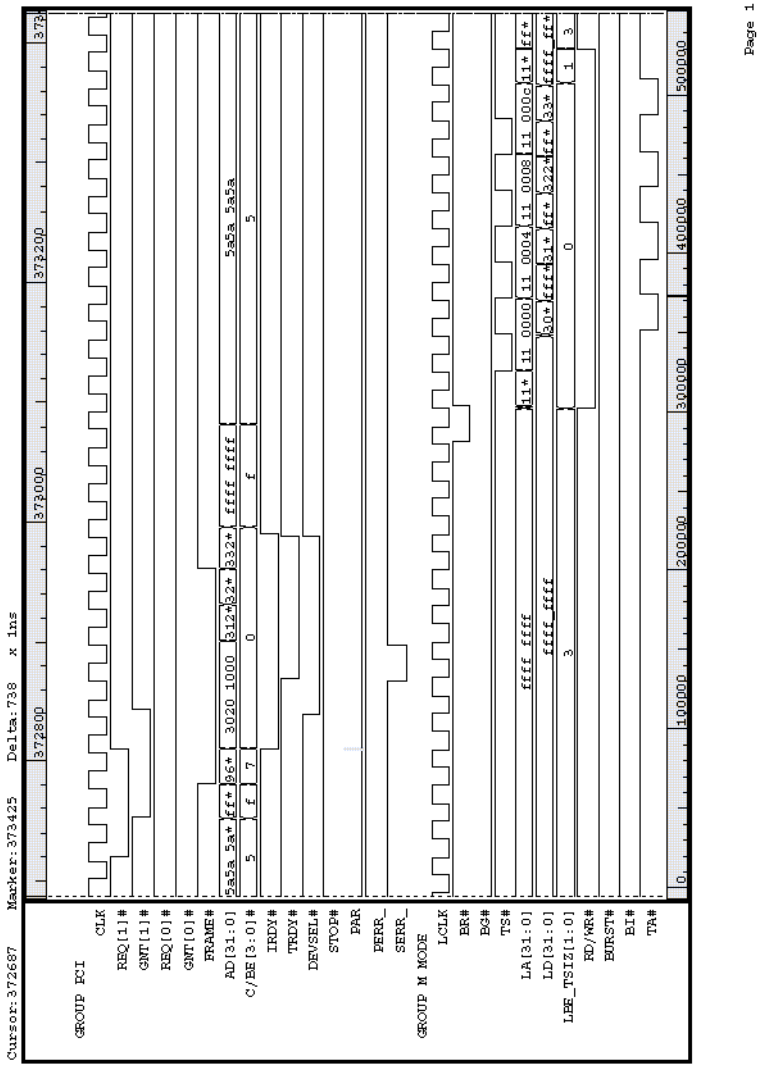
Timing Diagram 3-15. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Burst Read Ahead Enabled



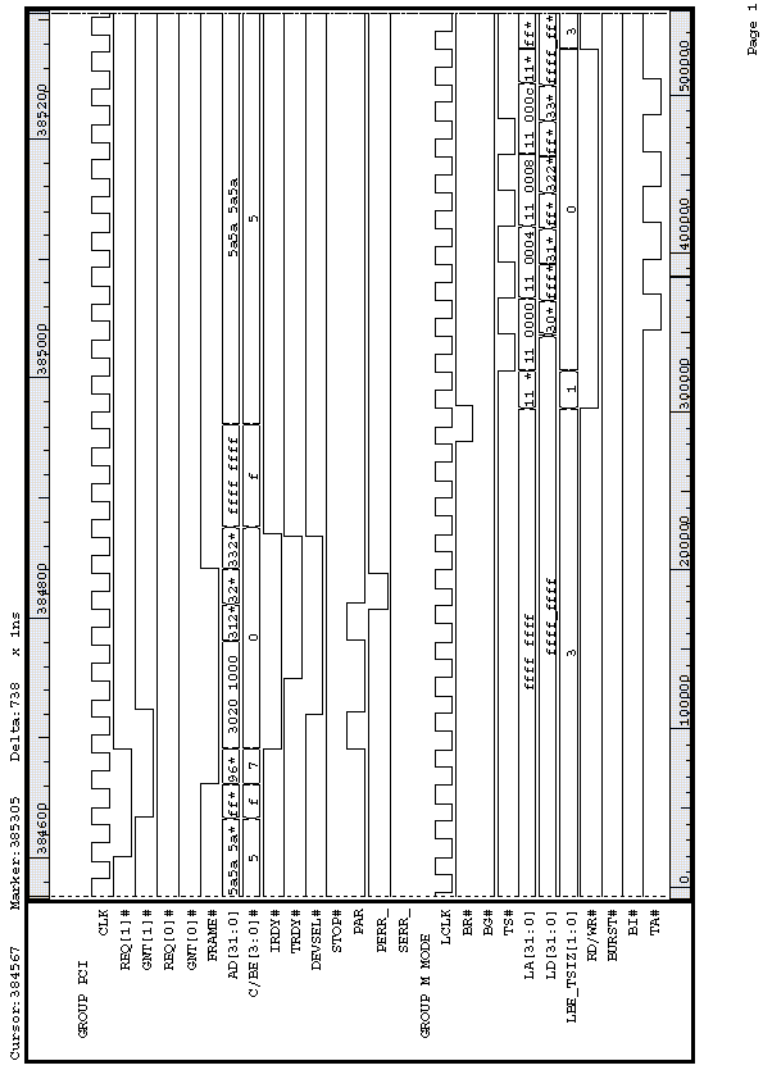
Timing Diagram 3-16. Direct Slave Burst Write and Read with Timer 8



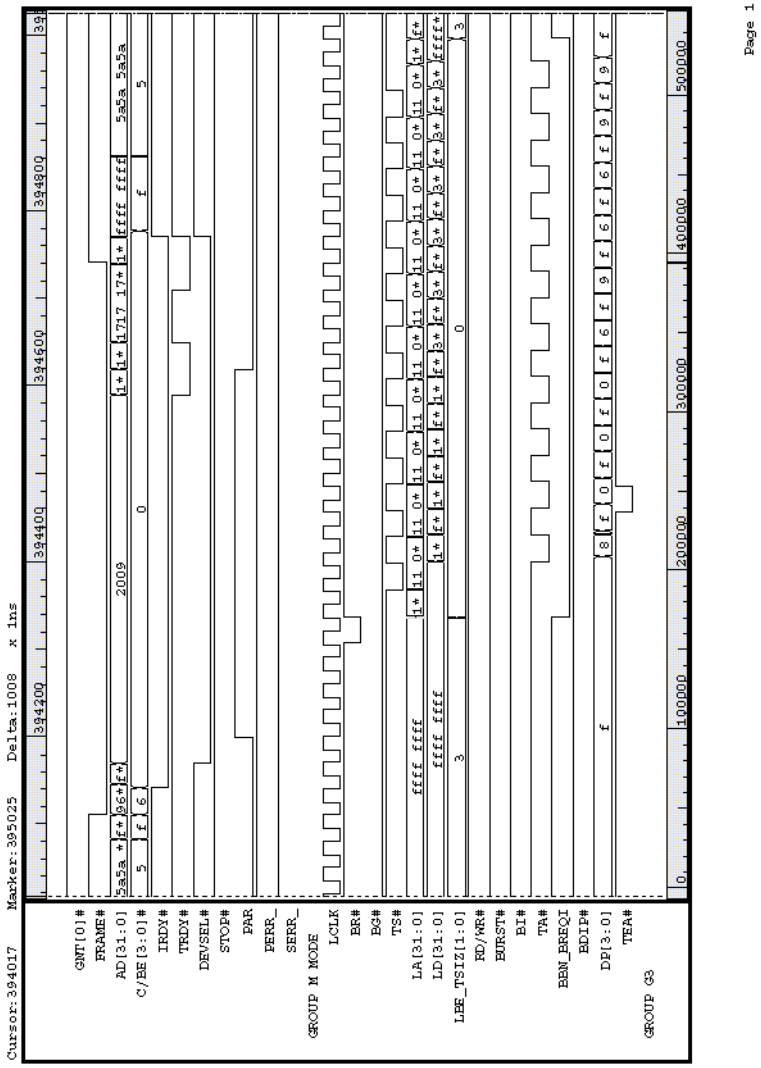
Timing Diagram 3-17. Direct Slave from PCI Bus to 32-Bit Device on Local Bus; Local Bus Latency and Pause Timers Set to 23 to Test LHOLD



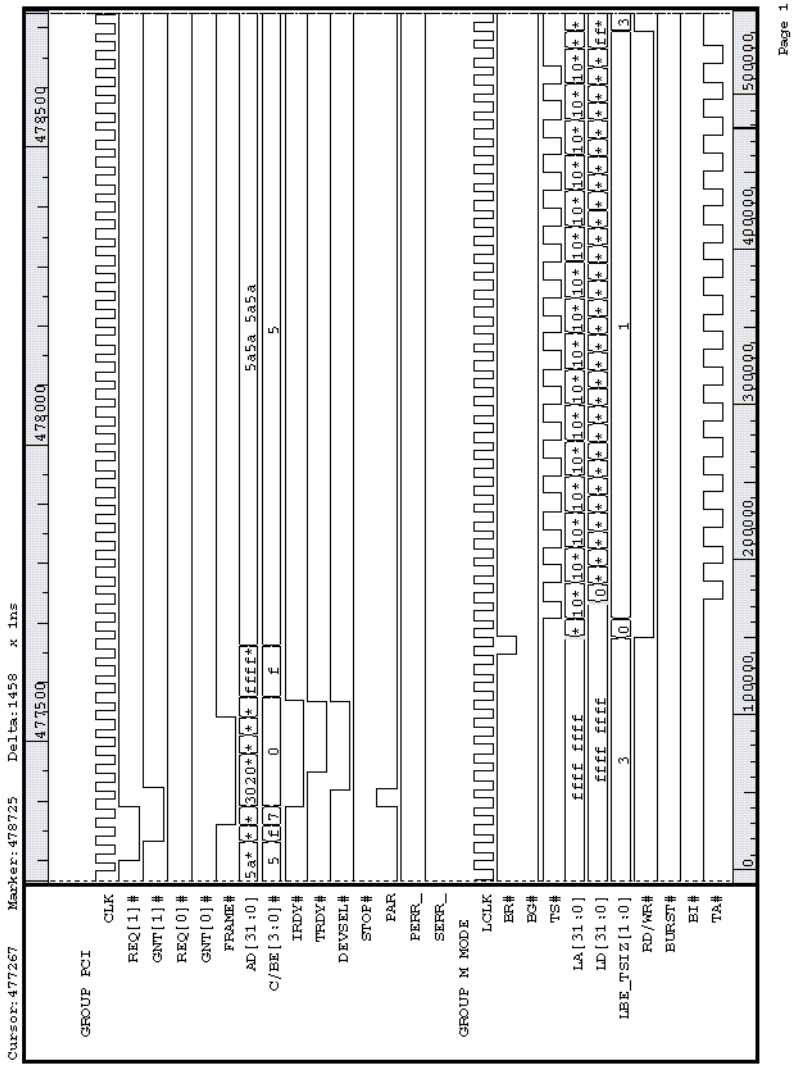
Timing Diagram 3-18. PCI Parity Error (Address Phase), SERR# Asserted



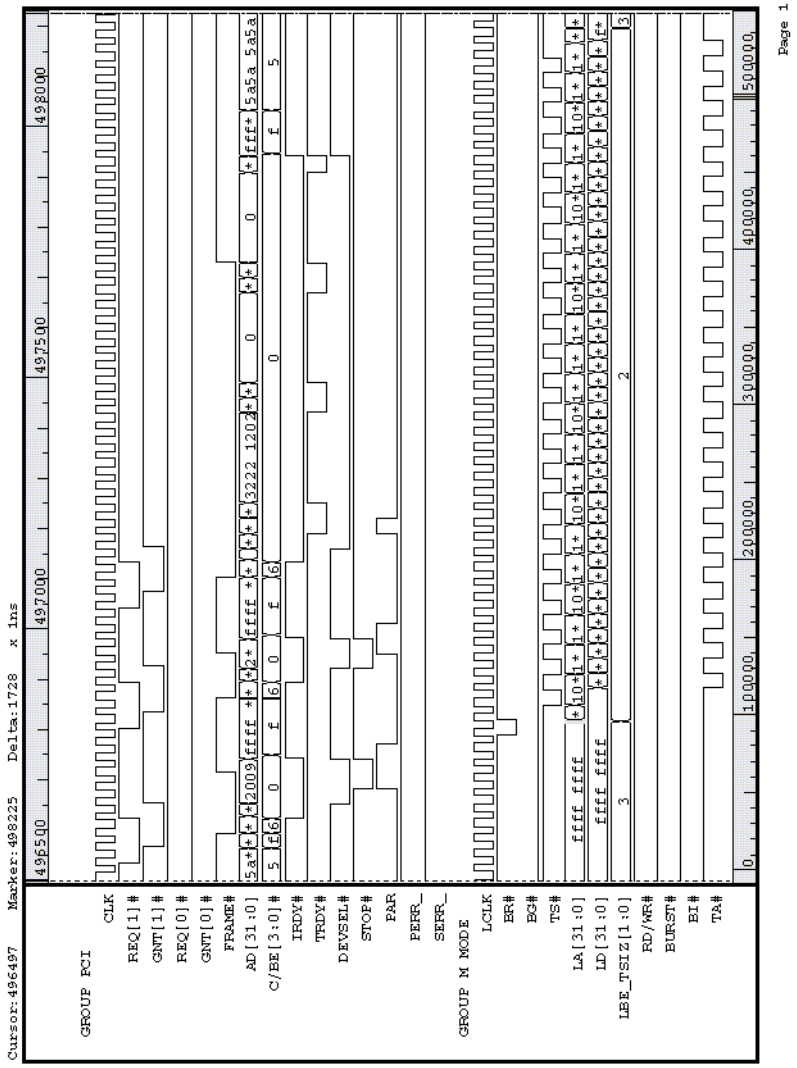
Timing Diagram 3-19. PCI Parity PERR# Direct Slave Write Interrupts, First Data Phase



Timing Diagram 3-20. PCI Parity PERR# on Direct Slave Read Interrupts, First Data, DP0

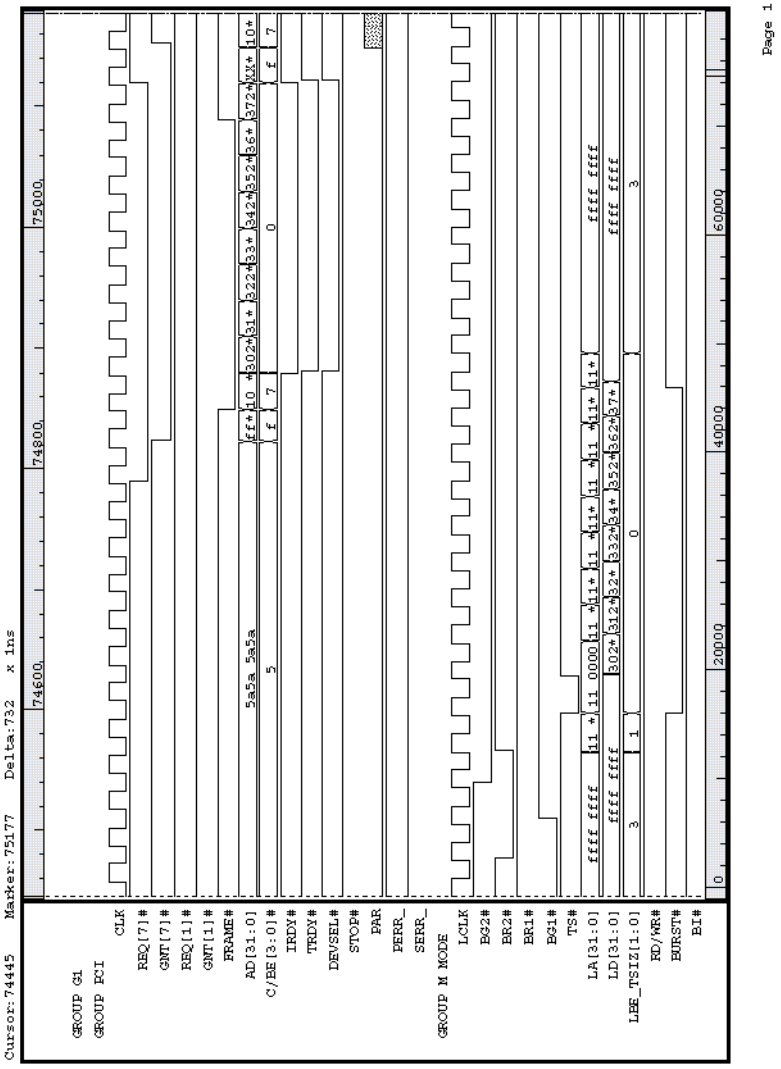


Timing Diagram 3-21. Direct Slave Burst Write 4 Data, Big Endian, Upper Bytes [31:24]

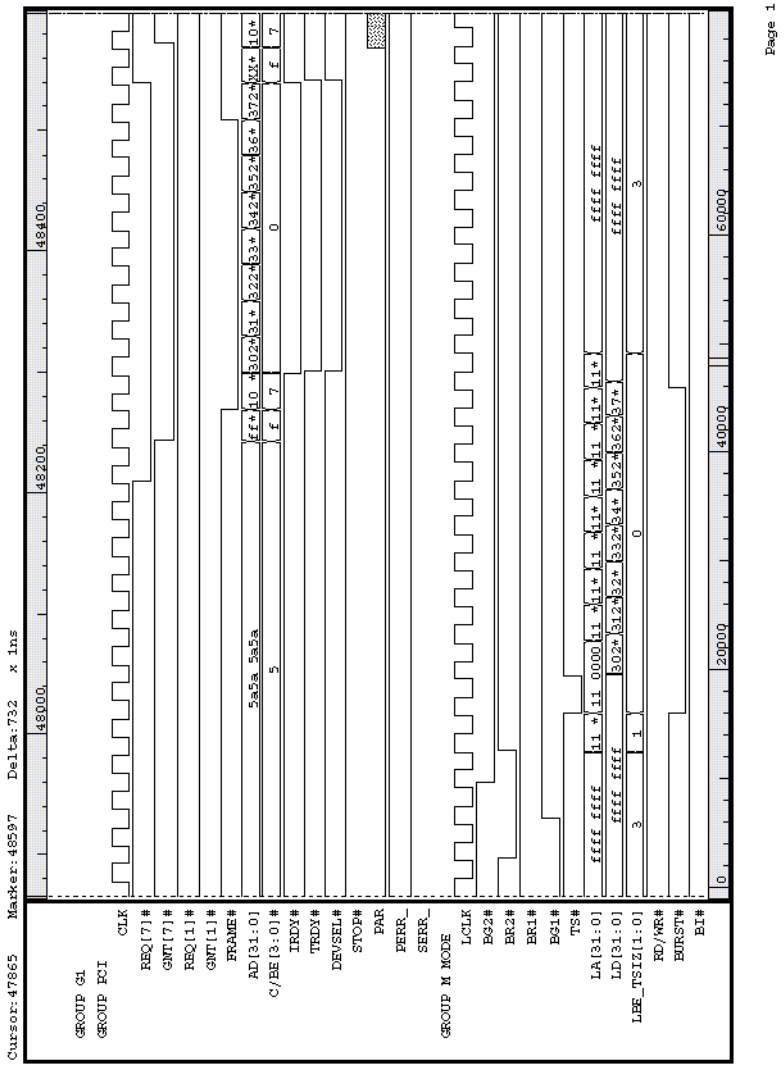


Timing Diagram 3-22. Direct Slave Burst Read 7, Big Endian, Lower Bytes [7:0]

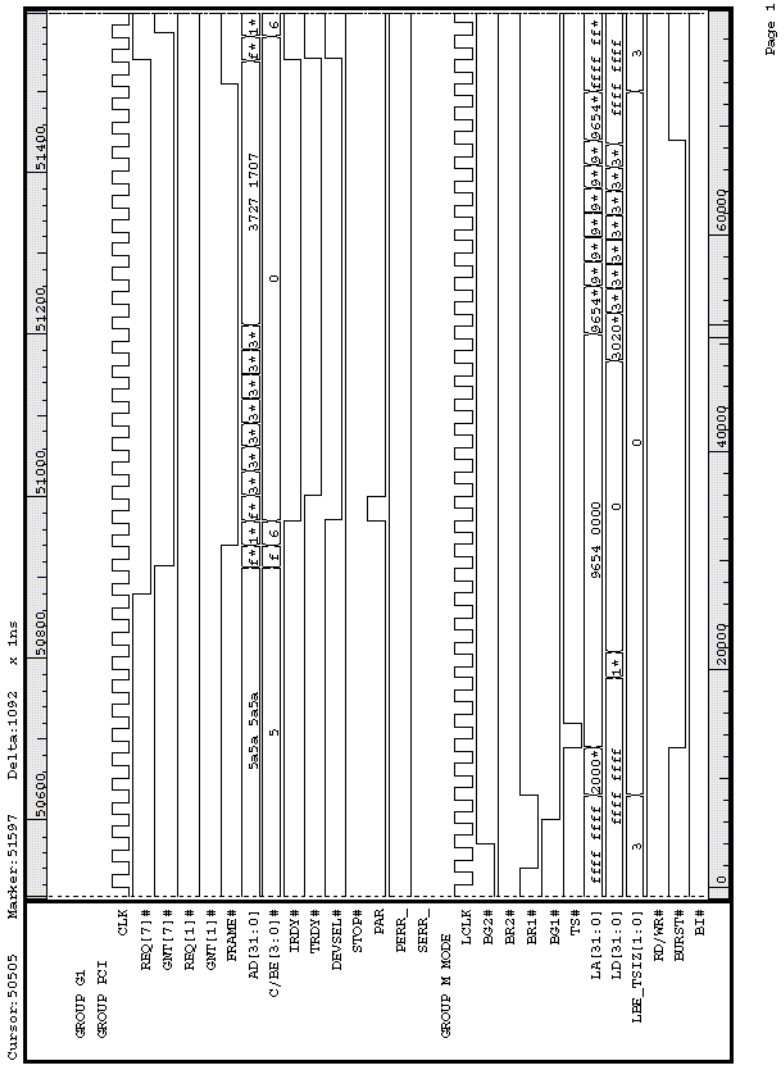
3.6.3 M Mode DMA Timing Diagrams



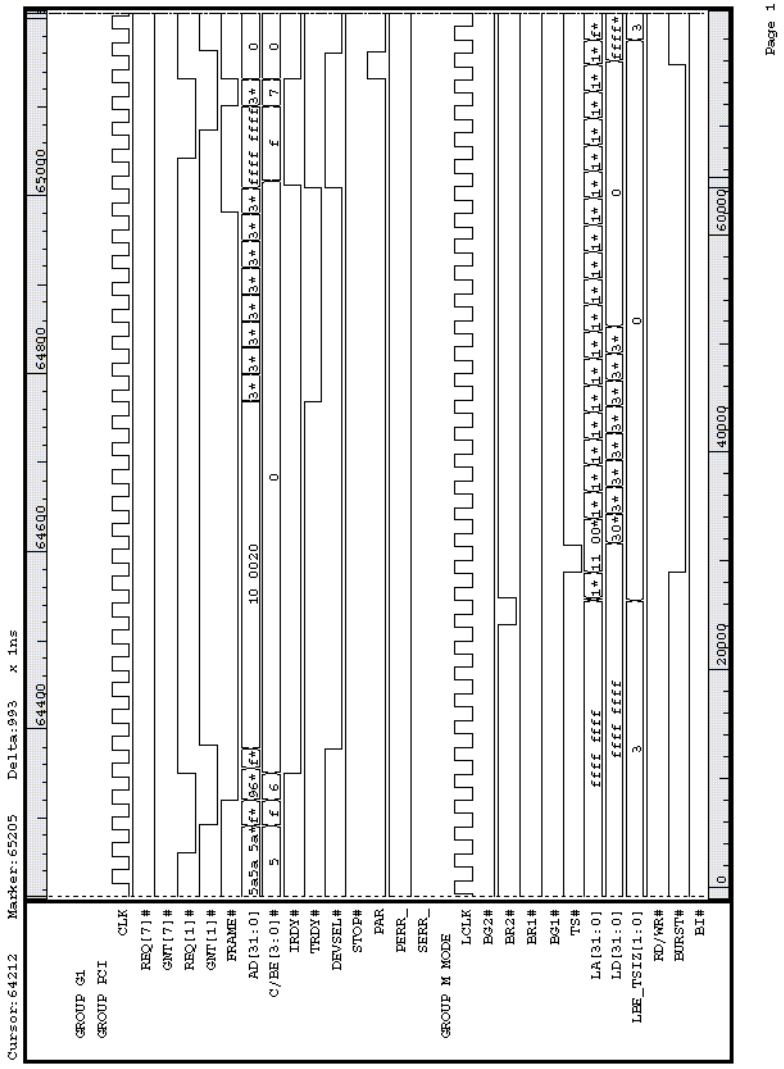
Timing Diagram 3-23. DMA Channel 0 Local-to-PCI (Memory Write Command)



Timing Diagram 3-24. DMA Channel 0 PCI-to-Local (Memory Read Line Command)



Timing Diagram 3-25. DMA Channel 0 PCI-to-Local (Memory Read Command)



Timing Diagram 3-26. DMA Channel 1 PCI-to-Local (Memory Read Command)

4 C AND J MODES BUS OPERATION

4.1 PCI BUS CYCLES

The PCI 9056 is compliant with *PCI r2.2*. Refer to *PCI r2.2* for specific PCI Bus functions.

4.1.1 Direct Slave Command Codes

As a Target, the PCI 9056 allows access to the PCI 9056 internal registers and the Local Bus, using the commands listed in Table 4-1.

All Read or Write accesses to the PCI 9056 can be Byte, Word, or Long-Word (Lword) accesses, defined as 32 bit. All memory commands are aliased to basic memory commands. All I/O accesses to the PCI 9056 are decoded to an Lword boundary. Byte enables are used to determine which bytes are read or written. An I/O access with illegal byte enable combinations is terminated with a Target Abort.

Table 4-1. Direct Slave Command Codes

Command Type	Code (C/BE[3:0]#)
I/O Read	0010 (2h)
I/O Write	0011 (3h)
Memory Read	0110 (6h)
Memory Write	0111 (7h)
Configuration Read	1010 (Ah)
Configuration Write	1011 (Bh)
Memory Read Multiple	1100 (Ch)
Memory Read Line	1110 (Eh)
Memory Write and Invalidate	1111 (Fh)

4.1.2 PCI Master Command Codes

The PCI 9056 can access the PCI Bus to perform DMA or Direct Master Local-to-PCI Bus transfers. During a Direct Master or DMA transfer, the command code assigned to the PCI 9056 internal register location (CNTRL[15:0]) is used as the PCI command code (except for Memory Write and Invalidate mode for DMA cycles where DMPBAM[9]=1).

Notes: *Programmable internal registers determine PCI command codes when the PCI 9056 is the Master.*

DMA cannot perform I/O or Configuration accesses.

4.1.2.1 DMA Master Command Codes

The PCI 9056 DMA controllers can assert the Memory Command cycles listed in Table 4-2.

Table 4-2. DMA Master Command Codes

Command Type	Code (C/BE[3:0]#)
Memory Read	0110 (6h)
Memory Write	0111 (7h)
Memory Read Multiple	1100 (Ch)
PCI Dual Address Cycle	1101 (Dh)
Memory Read Line	1110 (Eh)
Memory Write and Invalidate	1111 (Fh)

4.1.2.2 Direct Master Local-to-PCI Command Codes

For Direct Master Local-to-PCI Bus accesses, the PCI 9056 asserts the cycles listed in Table 4-3 through Table 4-5.

Table 4-3. Local-to-PCI Memory Access

Command Type	Code (C/BE[3:0]#)
Memory Read	0110 (6h)
Memory Write	0111 (7h)
Memory Read Multiple	1100 (Ch)
PCI Dual Address Cycle	1101 (Dh)
Memory Read Line	1110 (Eh)
Memory Write and Invalidate	1111 (Fh)

Table 4-4. Local-to-PCI I/O Access

Command Type	Code (C/BE[3:0]#)
I/O Read	0010 (2h)
I/O Write	0011 (3h)

Table 4-5. Local-to-PCI Configuration Access

Command Type	Code (C/BE[3:0]#)
Configuration Memory Read	1010 (Ah)
Configuration Memory Write	1011 (Bh)

4.1.3 PCI Arbitration

The PCI 9056 asserts REQ# to request the PCI Bus. The PCI 9056 can be programmed using the PCI Request Mode bit (MARBR[23]) to de-assert REQ# when it asserts FRAME# during a Bus Master cycle, or to keep REQ# asserted for the entire Bus Master cycle. The PCI 9056 always de-asserts REQ# for a minimum of two PCI clocks after a bus ownership that sustains that sustains a Target disconnect.

The Direct Master Write Delay bits (DMPBAM[15:14]) can be programmed to delay the PCI 9056 from asserting PCI REQ# during a Direct Master Write cycle. DMPBAM can be programmed to wait 0, 4, 8, or 16 PCI Bus clocks after the PCI 9056 has received its first Write data from the Local Bus Master and is ready to begin the PCI Write transaction. This function is useful in applications where a Local Master is bursting and a Local Bus clock is slower than the PCI Bus clock. This allows Write data to accumulate in the PCI 9056 Direct Master Write FIFO, which provides for better use of the PCI Bus.

4.2 LOCAL BUS CYCLES

The PCI 9056 interfaces a PCI Host bus to several Local Bus types, as listed in Table 4-6. It operates in one of three modes (selected through the MODE[1:0] pins), corresponding to the three bus types—M, C, and J.

Table 4-6. Local Bus Types

MODE1	MODE0	Bus Mode	Bus Type
1	1	M	32-bit non-multiplexed
1	0	Reserved	—
0	0	C	32-bit non-multiplexed
0	1	J	32-bit multiplexed

4.2.1 Local Bus Arbitration

The PCI 9056 asserts LHOLD to request the Local Bus. It owns the Local Bus when LHOLD and LHOLDA are asserted. When the PCI 9056 acknowledges BREQi assertion during DMA or Direct Slave Write transfers, it releases the Local Bus within two Lword transfers by de-asserting LHOLD and floating the Local Bus outputs if either of the following conditions exist:

- BREQi is asserted and enabled
- Gating is enabled and the Local Bus Latency Timer is enabled and expires (MARBR[27, 7:0], respectively)

The Local Arbiter can now grant the Local Bus to another Local Master. After the PCI 9056 acknowledges that LHOLDA is de-asserted and the Local Bus Pause Timer is zero, it re-asserts LHOLD to request the Local Bus. When the PCI 9056 receives LHOLDA, it drives the bus and continues the transfer.

Note: *The Local Bus Pause Timer applies only to DMA operation. It does* ***not*** *apply to Direct Slave operation.*

4.2.2 Direct Master

Local Bus cycles can be single or Burst cycles. The BLAST# signal is used to determine whether a single or Burst cycle is to be performed. If BLAST# is asserted at the beginning of the first Data phase, the PCI 9056 performs a single PCI Bus cycle. Otherwise, the PCI 9056 performs a Burst PCI Bus cycle and BLAST# is used to end the cycle. As a Local Bus Target, the PCI 9056 allows access to the PCI 9056 internal registers and the PCI Bus. Non-32-bit Direct Master accesses to the PCI 9056 require simple external logic (latch array to combine data into a 32-bit bus).

Local Bus Direct Master accesses to the PCI 9056 must be for a 32-bit non-pipelined bus.

4.2.3 Direct Slave

The PCI Bus Master reads from and writes to the Local Bus (the PCI 9056 is a PCI Bus Target and a Local Bus Master).

4.2.4 Wait State Control

If READY# mode is disabled, the external READY# input signal has no effect on wait states for a Local access. Wait states between Data cycles are asserted internally by a wait state counter. The wait state counter is initialized with its Configuration register value at the start of each data access.

If READY# mode is enabled, it has no effect until the wait state counter reaches 0. READY# then controls the number of additional wait states.

BTERM# input is not sampled until the wait state counter reaches 0. BTERM# overrides READY# when BTERM# is enabled and asserted.

The following figure illustrates the PCI 9056 wait states for C and J modes.

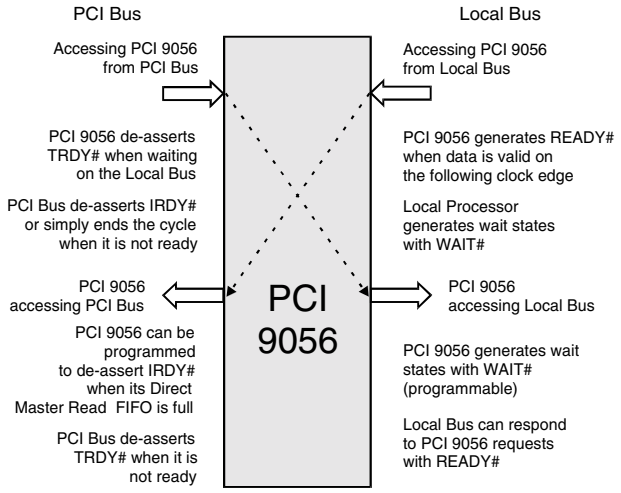


Figure 4-1. Wait States

Note: *Figure 4-1 represents a sequence of Bus cycles.*

4.2.4.1 Wait States—Local Bus

In Direct Master mode and when accessing the PCI 9056 registers, the PCI 9056 acts as a Local Bus Slave. The PCI 9056 asserts wait states by delaying the READY# signal. The Local processor asserts wait states with the WAIT# signal.

In Direct Slave and DMA modes, the PCI 9056 acts as a Local Bus Master. The PCI 9056 inserts internal wait states with the WAIT# signal. The Local processor asserts external wait states by delaying the READY# signal.

The Internal Wait State bit(s) (LBRD0[21:18, 5:2], LBRD1[5:2], DMAMODE0[5:2], and/or DMAMODE1 [5:2]) can be used to program the number of internal wait states between the first address-to-data (and subsequent data-to-data in Burst mode).

During Direct Master accesses, WAIT# signal must be asserted during the ADS phase for the PCI 9056 to sample the wait state phase.

In Direct Slave and DMA modes, the READY# signal has no effect until the wait state counter (LBRD0[21:18, 5:2], LBRD1[5:2], DMAMODE0[5:2], and/or DMAMODE1[5:2]) reaches zero. READY# then controls the number of wait states by being de-asserted in the middle of the Data transaction.

4.2.4.2 Wait States—PCI Bus

The PCI Bus Master throttles IRDY# and the PCI Bus Slave throttles TRDY# to assert PCI Bus wait state(s).

4.2.5 Burst Mode and Continuous Burst Mode (Bterm "Burst Terminate" Mode)

Note: *In the following sections, Bterm refers to the PCI 9056 internal register bit and BTERM# refers to the PCI 9056 external signal.*

4.2.5.1 Burst and Bterm Modes

Table 4-7. Burst and Bterm on the Local Bus

Mode	Burst	Bterm	Result
Single Cycle	0	0	One ADS# per data (default).
	0	1	One ADS# per data.
Burst-4	1	0	One ADS# per four data (recommended for i960 and PPC401 family).
Burst Forever	1	1	One ADS# per BTERM# (refer to Section 4.2.5.2.1).

On the Local Bus, BLAST# and BTERM# perform the following:

- If the Burst Mode bit is enabled, but the Bterm Mode bit is disabled, then the PCI 9056 bursts (up to a Qword boundary) four Lwords. BLAST# is asserted at the beginning of the fourth Lword Data phase (LA[3:2]=11) and a new ADS# is asserted at the first Lword (LA[3:2]=00) of the next burst.
- If BTERM# is enabled and asserted, the PCI 9056 terminates the Burst cycle of the end of the current Data phase without generating BLAST#. The PCI 9056 generates a new burst transfer starting with a new ADS#, terminating it normally using BLAST#.
- BTERM# input is valid only when the PCI 9056 is Master of the Local Bus (Direct Slave or DMA modes).
- As an input, BTERM# is asserted by external logic. It instructs the PCI 9056 to break up a Burst cycle.
- BTERM# is used to indicate a memory access is crossing a page boundary or requires a new Address cycle.

Notes: *If Address Increment is disabled, the DMA transfer bursts beyond four Lwords.*

If the Bterm Mode bit is disabled, the PCI 9056 performs the following:

- ***32-bit Local Bus****—Bursts up to four Lwords*
- ***16-bit Local Bus****—Bursts up to two Lwords*
- ***8-bit Local Bus****—Bursts up to one Lword*

In every case, it performs four transactions.

4.2.5.2 Burst-4 Lword Mode

If the Burst Mode bit is enabled and the Bterm Mode bit is disabled, bursting can start on any Lword boundary and continue up to a 16-byte address boundary. After data up to the boundary is transferred, the PCI 9056 asserts a new Address cycle (ADS#).

Table 4-8. Burst-4 Lword Mode

Bus Width	Burst
32 bit	Four Lwords or up to a quad-Lword boundary (LA3, LA2 = 11)
16 bit	Four words or up to a Qword boundary (LA2, LA1 = 11)
8 bit	Four bytes or up to a quad-byte boundary (LA1, LA0 = 11)

4.2.5.2.1 Continuous Burst Mode (Bterm "Burst Terminate" Mode)

If both the Burst and Bterm Mode bits are enabled, the PCI 9056 can operate beyond the Burst-4 Lword mode.

Bterm mode enables PCI 9056 to perform long bursts to devices that can accept bursts of longer than four Lwords. The PCI 9056 asserts one Address cycle and continues to burst data. If a device requires a new Address cycle (ADS#), it can assert BTERM# input to cause the PCI 9056 to assert a new Address cycle. BTERM# input acknowledges current Data transfer and requests that a new Address cycle be asserted (ADS#). The new address is for the next Data transfer. If the Bterm Mode bit is enabled and the BTERM# signal is asserted, the PCI 9056 asserts BLAST# only if its Read FIFO is full, its Write FIFO is empty, or if a transfer is complete.

4.2.5.3 Partial Lword Accesses

Lword accesses, in which not all byte enables are asserted, are broken into single cycle accesses. Burst start addresses can be any Lword boundary. If the Burst Start Address in a Direct Slave or DMA transfer is not aligned to an Lword boundary, the PCI 9056 first performs a single cycle. It then starts to burst on the Lword boundary if there is remaining data that is not a whole Lword during DMA (*for example*, it results in a single cycle at the end).

4.2.6 Recovery States (J Mode Only)

In J mode, the PCI 9056 inserts one recovery state between the last Data transfer and the next Address cycle.

Note: *The PCI 9056 does not support the i960J function that uses READY# input to add recovery states. No additional recovery states are added if READY# input remains asserted during the last Data cycle.*

4.2.7 Local Bus Read Accesses

For all single cycle Local Bus Read accesses, the PCI 9056 reads only bytes corresponding to byte enables requested by the Direct Master. For all Burst Read cycles, the PCI 9056 passes all the bytes and can be programmed to:

- Prefetch
- Perform Read Ahead mode
- Generate internal wait states
- Enable external wait control (READY# input)
- Enable type of Burst mode to perform

4.2.8 Local Bus Write Accesses

For Local Bus writes, only bytes specified by a PCI Bus master or the PCI 9056 DMA controller are written.

4.2.9 Direct Slave Accesses to 8- or 16-Bit Local Bus

Direct Slave PCI accesses to an 8- or 16-bit Local Bus results in the PCI Bus Lword being broken into multiple Local Bus transfers. For each transfer, byte enables are encoded as in the i960C to provide Local Address bits LA[1:0].

4.2.10 Local Bus Data Parity

Generation or use of Local Bus data parity is optional. Signals on the data parity pins do not affect operation of the PCI 9056. The PCI Bus parity checking and generation is independent of the Local Bus parity checking and generation. PCI Bus parity checking may result in assertion of PERR#, a PCI Bus system error (SERR#), or other means of PCI Bus transfer termination as a result of the parity error on the PCI data address, command code, and byte enables. The Local Bus Parity Check is passive and only provides parity information to the Local processor during Direct Master, Direct Slave, and DMA transfers.

There is one data parity pin for each byte lane of the PCI 9056 data bus (DP[3:0]). “Even data parity” is asserted for each lane during Local Bus reads from the PCI 9056 and during PCI 9056 Master writes to the Local Bus.

Even data parity is checked during Local Bus writes to the PCI 9056 and during PCI 9056 reads from the Local Bus. Parity is checked for each byte lane with an asserted byte enable. If a parity error is detected, LSERR# is asserted in the Clock cycle following the data being checked.

Parity is checked for Direct Slave reads, Direct Master writes, and DMA Local Bus reads. The PCI 9056 sets a status bit and asserts an interrupt (LSERR#) in the clock cycle following data being checked if a parity error is detected. However, the Data Parity Error Status bit and interrupt are never set or asserted unless the READY# signal is active and asserted low. This applies only when the READY# signal is disabled in the PCI 9056 register. A workaround for this is to disable the READY# Enable bit and externally pull READY# low.

4.3 BIG ENDIAN/LITTLE ENDIAN

4.3.1 PCI Bus Little Endian Mode

PCI Bus is a Little Endian bus (*that is*, the address is invariant and data is Lword-aligned to the lowermost byte lane).

Table 4-9. PCI Bus Little Endian Byte Lanes

Byte Number	Byte Lane
0	AD[7:0]
1	AD[15:8]
2	AD[23:16]
3	AD[31:24]

4.3.2 Local Bus Big/Little Endian Mode

The PCI 9056 Local Bus can be programmed to operate in Big or Little Endian mode.

Table 4-10. Byte Number and Lane Cross-Reference

Mode	Byte Number		Byte Lane
	Big Endian	Little Endian	
C	3	0	LD[7:0]
	2	1	LD[15:8]
	1	2	LD[23:16]
	0	3	LD[31:24]
J	3	0	LAD[7:0]
	2	1	LAD[15:8]
	1	2	LAD[23:16]
	0	3	LAD[31:24]

Table 4-11. Big/Little Endian Program Mode

BIGEND# Pin	BIGEND Register (1=Big, 0=Little)	Endian Mode
0	0	Big
0	1	Big
1	0	Little
1	1	Big

Table 4-12 lists register bits associated with the following cycles.

Table 4-12. Cycle Reference

Cycle	Register Bits
Local access to the Configuration registers	BIGEND[0]
Direct Master, Memory, and I/O	BIGEND[1]
Direct Slave	BIGEND[2], Space 0, and BIGEND[3], Expansion ROM

In Big Endian mode, the PCI 9056 transposes data byte lanes. Data is transferred as listed in Table 4-13 through Table 4-18.

4.3.2.1 32-Bit Local Bus—Big Endian Mode

Data is Lword-aligned to the uppermost byte lane (Data Invariance).

Table 4-13. Upper Lword Lane Transfer—32-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [31:24]
	Byte 1 appears on Local Data [23:16]
	Byte 2 appears on Local Data [15:8]
	Byte 3 appears on Local Data [7:0]

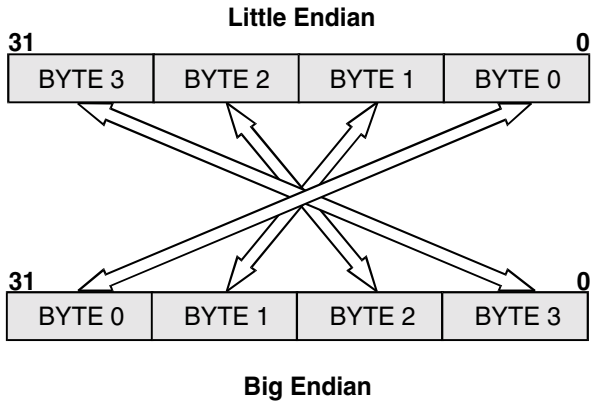


Figure 4-2. Big/Little Endian—32-Bit Local Bus

4.3.2.2 16-Bit Local Bus—Big Endian Mode

For a 16-bit Local Bus, the PCI 9056 can be programmed to use the upper or lower word lanes.

Table 4-14. Upper Word Lane Transfer—16-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [31:24]
	Byte 1 appears on Local Data [23:16]
Second transfer	Byte 2 appears on Local Data [31:24]
	Byte 3 appears on Local Data [23:16]

Table 4-15. Lower Word Lane Transfer—16-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [15:8]
	Byte 1 appears on Local Data [7:0]
Second transfer	Byte 2 appears on Local Data [15:8]
	Byte 3 appears on Local Data [7:0]

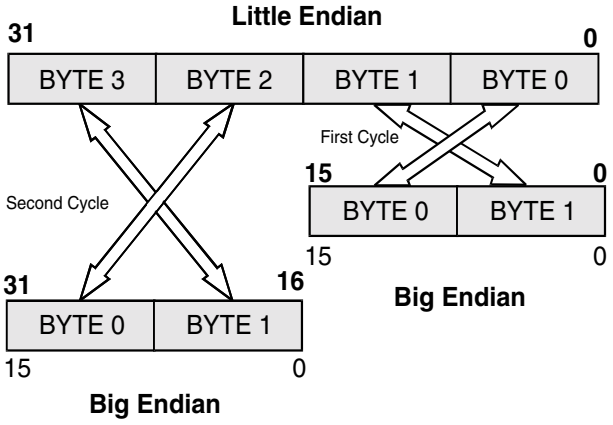


Figure 4-3. Big/Little Endian—16-Bit Local Bus

4.3.2.3 8-Bit Local Bus—Big Endian Mode

For an 8-bit Local Bus, the PCI 9056 can be programmed to use the upper or lower byte lanes.

Table 4-16. Upper Byte Lane Transfer—8-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [31:24]
Second transfer	Byte 1 appears on Local Data [31:24]
Third transfer	Byte 2 appears on Local Data [31:24]
Fourth transfer	Byte 3 appears on Local Data [31:24]

Table 4-17. Lower Byte Lane Transfer—8-Bit Local Bus

Burst Order	Byte Lane
First transfer	Byte 0 appears on Local Data [7:0]
Second transfer	Byte 1 appears on Local Data [7:0]
Third transfer	Byte 2 appears on Local Data [7:0]
Fourth transfer	Byte 3 appears on Local Data [7:0]

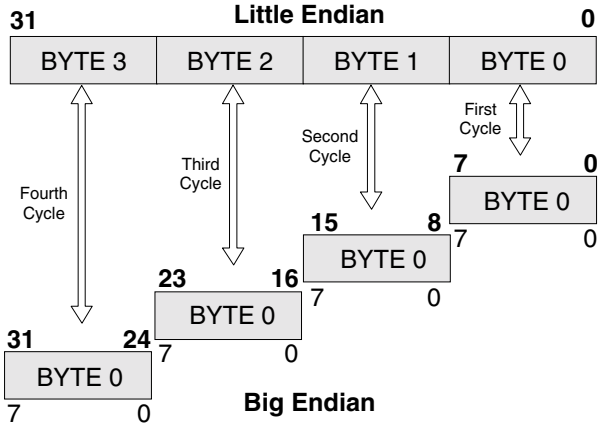


Figure 4-4. Big/Little Endian—8-Bit Local Bus

4.3.2.4 Local Bus Big/Little Endian Mode Accesses

For each of the following transfer types, the PCI 9056 Local Bus can be independently programmed to operate in Big or Little Endian mode:

- Local Bus accesses to the PCI 9056 Configuration registers
- Direct Slave PCI accesses to Local Address Space 0
- Direct Slave PCI accesses to Local Address Space 1
- Direct Slave PCI accesses to the Expansion ROM
- DMA Channel 0 accesses to the Local Bus
- DMA Channel 1 accesses to the Local Bus
- Direct Master accesses to the PCI Bus

For Local Bus accesses to the Internal Configuration registers and Direct Master accesses, use BIGEND# to dynamically change the Endian mode.

Notes: *The PCI Bus is always Little Endian.*
Only byte lanes are swapped, not individual bits.

4.4 SERIAL EEPROM

Functional operation described can be modified through the PCI 9056 programmable internal registers.

4.4.1 Vendor and Device ID Registers

Three Vendor and Device ID registers are supported:

- **PCIIDR**—Contains normal Device and Vendor IDs. Can be loaded from the serial EEPROM or Local processor(s).
- **PCISVID**—Contains Subsystem and Subvendor IDs. Can be loaded from the serial EEPROM or Local processor(s).
- **PCIHIDR**—Contains hardwired PLX Vendor and Device IDs.

4.4.1.1 Serial EEPROM Initialization

During serial EEPROM initialization, the PCI 9056 responds to Direct Slave accesses with a Retry. During serial EEPROM initialization, the PCI 9056 responds to a Local processor access by delaying acknowledgment of the cycle (READY#).

4.4.1.2 Local Initialization

Refer to the document, *PCI 9056 Blue Book*

***Revision 0.91 Correction,* for the corrected version of this section.**

Refer to the document, *PCI 9056 Blue Book Revision 0.91 Correction,* for the corrected version of this section.

4.4.2 Serial EEPROM Operation

After reset, the PCI 9056 attempts to read the serial EEPROM to determine its presence. An active Start bit set to 0 indicates a serial EEPROM is present. The PCI 9056 supports 2K bit (FM93CS56L or compatible) or 4K bit (FM93CS66L or compatible) devices. (Refer to manufacturer's data sheet for the particular serial EEPROM being used.) The first Lword is then checked to verify that the serial EEPROM is programmed. If the first Lword (33 bits) is all ones (1), a blank serial EEPROM is present. If the first Lword (33 bits) is all zeros, no serial EEPROM is present. For both conditions, the PCI 9056 reverts to the default values. (Refer to Table 4-18.) The Programmed Serial EEPROM Present bit is set (CNTRL[28]=1) if the serial EEPROM is programmed (real or random data if a serial EEPROM is detected).

The 3.3V serial EEPROM clock (EESK) is derived from the PCI clock. The PCI 9056 generates the serial EEPROM clock by internally dividing the PCI clock by 268. For a 66.6 MHz PCI Bus, EESK is 248.7 kHz; for a 33.3 MHz PCI Bus, EESK is 124.4 kHz.

The serial EEPROM can be read or written from the PCI or Local Buses. The Serial EEPROM Control Register bits (CNTRL[31, 27:24]) control the PCI 9056 pins that enable reading or writing of serial EEPROM data bits. (Refer to manufacturer's data sheet for the particular serial EEPROM being used.)

The PCI 9056 provides the ability to manually access the serial EEPROM interface by using CNTRL[31, 27:24] (EESK, EECS, and EEDI/EEDO controlled by software). Bit 24 is used to generate EESK (clock), bit 25 controls the chip select, and bit 31 enables EEDO Input buffer. Bit 27, when read, returns the value of EEDI.

Setting bits [31, 25, 24] to 1 causes the EEDI output to go high. A pull-up resistor is required on EEDO to go high when bit 31 is set. When reading the serial EEPROM, bit 31 must be set to 1.

To perform the read, the basic approach is to set the EECS and EEDO bits (bits 25 and 31, respectively) to the desired level and then toggle EESK high and low until done. *For example*, reading the serial EEPROM at location 0 involves the following steps:

1. Clear EESK, EEDO and EECS bits.
2. Set EECS high.
3. Toggle EESK high, then low.
4. Set EEDO bit high (start bit).
5. Toggle EESK high, then low.
6. Repeat step 6.
7. Clear EEDO.
8. Toggle EESK high, then low.
9. Toggle EESK bit high, then low 8 times (clock in serial EEPROM Address 0).
10. Set bit 31 to float the EEDO pin for reading.
11. Toggle EESK high, then low 16 times (clock in one word from serial EEPROM).
12. After each clock pulse, read bit 27 and save.
13. Clear EECS bit.
14. Toggle EESK high, then low.
15. Read is now complete.

The serial EEPROM can also be read or written, using the VPD function. (Refer to Section 10.)

The PCI 9056 has two serial EEPROM load options:

- **Long Load Mode**—Default. The PCI 9056 loads 17 Lwords from the serial EEPROM if the Extra Long Load from the Serial EEPROM bit is clear (LBRD0[25]=0)
- **Extra Long Load Mode**—The PCI 9056 loads 23 Lwords from the serial EEPROM if the Extra Long Load from the Serial EEPROM bit is set (LBRD0[25]=1) during a Long Load

Table 4-18. Serial EEPROM Guidelines

Local Processor	Serial EEPROM	System Boot Condition
None	None	The PCI 9056 uses default values. The EEDI/EEDO pin must be **pulled low**—a 1K ohm resistor is required (rather than pulled high, which is typically done for this pin). If the PCI 9056 detects all zeros, it reverts to default values.
None	Programmed	Boot with serial EEPROM values. The Local Init Status bit (LMISC1[2]) must be set by the serial EEPROM. A 3K to 10K ohm pull-up resistor is required on EDDI/EEDO.
None	Blank	The PCI 9056 detects a blank device and reverts to default values. A 3K to 10K ohm pull-up resistor is required on EDDI/EEDO.
Present	None	**Refer to the document, *PCI 9056 Blue Book Revision 0.91 Correction,* for the corrected version of this table entry.**
Present	Programmed	Load serial EEPROM, but the Local processor can reprogram the PCI 9056. Either the Local processor or the serial EEPROM must set the Local Init Status bit (LMISC1[2]=done). A 3K to 10K ohm pull-up resistor is required on EDDI/EEDO.
Present	Blank	The PCI 9056 detects a blank serial EEPROM and reverts to default values. A 3K to 10K ohm pull-up resistor is required on EDDI/EEDO. ***Notes:*** *In some systems, the Local processor may be too late to reconfigure the PCI 9056 registers before the BIOS configures them.* *The serial EEPROM can be programmed through the PCI 9056 after the system boots in this condition.*

Note: *If the serial EEPROM is missing and a Local Processor is present with blank Flash, the condition None/None (as seen in Table 4-18) applies, until the Processor's Flash is programmed.*

4.4.2.1 Long Serial EEPROM Load

The registers listed in Table 4-19 are loaded from the serial EEPROM after a reset is de-asserted if the Extra Long Load from Serial EEPROM bit is not set (LBRD0[25]=0). The serial EEPROM is organized in words (16 bit). The PCI 9056 first loads the Most Significant Word bits (MSW[31:16]), starting from the Most Significant bit, (MSB[31]). The PCI 9056 then loads the Least Significant Word bits (LSW[15:0]), starting again from the Most Significant bit (MSB[15]). Therefore, the PCI 9056 loads the Device ID, Vendor ID, Class Code, and so forth.

The serial EEPROM values can be programmed using an EEPROM programmer. The values can also be programmed using the PCI 9056 VPD function (refer to Section 10) or through the Serial EEPROM Control register (CNTRL).

The CNTRL register allows programming of the serial EEPROM, one bit at a time. To read back the value from the serial EEPROM, the CNTRL[27] bit (refer to Section 4.4.2) or the VPD function should be utilized. With full utilization of VPD, the designer can perform reads and writes from/to the serial EEPROM, 32 bits at a time. Values should be programmed in the order listed in Table 4-19. The 34, 16-bit words listed in the table are stored sequentially in the serial EEPROM.

4.4.2.2 Extra Long Serial EEPROM Load

The registers listed in the Local Address Space 0/ Expansion ROM Bus Region Descriptor register (LBRD0) are loaded from the serial EEPROM after a reset is de-asserted if the Extra Long Load from Serial EEPROM bit is set (LBRD0[25]=1). The serial EEPROM is organized in words (16 bit). The PCI 9056 first loads the Most Significant Word bits (MSW[31:16]), starting from the Most Significant bit (MSB[31]). It then loads the Least Significant Word bits (LSW[15:0]), restarting from the Most Significant bit (MSB[15]). Therefore, the PCI 9056 loads Device ID, Vendor ID, Class Code, and so forth.

The serial EEPROM values can be programmed using an EEPROM programmer. The values can also be programmed using the PCI 9056 VPD function or through the Serial EEPROM Control register (CNTRL).

Values should be programmed in the order listed in Table 4-20. The 46 16-bit words listed in Table 4-19 and Table 4-20 should be stored sequentially in the serial EEPROM.

Table 4-19. Long Serial EEPROM Load Registers

Serial EEPROM Offset	Description	Register Bits Affected
0h	Device ID	PCIIDR[31:16]
2h	Vendor ID	PCIIDR[15:0]
4h	Class Code	PCICCR[23:8]
6h	Class Code / Revision	PCICCR[7:0] / PCIREV[7:0]
8h	Maximum Latency / Minimum Grant	PCIMLR[7:0] / PCIMGR[7:0]
Ah	Interrupt Pin / Interrupt Line Routing	PCIIPR[7:0] / PCIILR[7:0]
Ch	MSW of Mailbox 0 (User Defined)	MBOX0[31:16]
Eh	LSW of Mailbox 0 (User Defined)	MBOX0[15:0]
10h	MSW of Mailbox 1 (User Defined)	MBOX1[31:16]
12h	LSW of Mailbox 1 (User Defined)	MBOX1[15:0]
14h	MSW of Range for PCI-to-Local Address Space 0	LAS0RR[31:16]
16h	LSW of Range for PCI-to-Local Address Space 0	LAS0RR[15:0]
18h	MSW of Local Base Address (Remap) for PCI-to-Local Address Space 0	LAS0BA[31:16]
1Ah	LSW of Local Base Address (Remap) for PCI-to-Local Address Space 0	LAS0BA[15:0]
1Ch	MSW of Mode/DMA Arbitration Register	MARBR[31:16]
1Eh	LSW of Mode/DMA Arbitration Register	MARBR[15:0]
20h	MSW of Local Miscellaneous Control Register 2 / MSW of Serial EEPROM Write-Protected Address	LMISC2[7:0] / PROT_AREA[7:0]
22h	LSW of Local Miscellaneous Control Register 1 / LSW of Local Bus Big/Little Endian Descriptor Register	LMISC1[7:0] / BIGEND[7:0]
24h	MSW of Range for PCI-to-Local Expansion ROM	EROMRR[31:16]
26h	LSW of Range for PCI-to-Local Expansion ROM	EROMRR[15:0]
28h	MSW of Local Base Address (Remap) for PCI-to-Local Expansion ROM	EROMBA[31:16]
2Ah	LSW of Local Base Address (Remap) for PCI-to-Local Expansion ROM	EROMBA[15:0]
2Ch	MSW of Bus Region Descriptors for PCI-to-Local Accesses	LBRD0[31:16]
2Eh	LSW of Bus Region Descriptors for PCI-to-Local Accesses	LBRD0[15:0]
30h	MSW of Range for Direct Master-to-PCI	DMRR[31:16]
32h	LSW of Range for Direct Master-to-PCI	DMRR[15:0]
34h	MSW of Local Base Address for Direct Master-to-PCI Memory	DMLBAM[31:16]
36h	LSW of Local Base Address for Direct Master-to-PCI Memory	DMLBAM[15:0]
38h	MSW of Local Bus Address for Direct Master-to-PCI I/O Configuration	DMLBAI[31:16]
3Ah	LSW of Local Bus Address for Direct Master-to-PCI I/O Configuration	DMLBAI[15:0]
3Ch	MSW of PCI Base Address (Remap) for Direct Master-to-PCI	DMPBAM[31:16]
3Eh	LSW of PCI Base Address (Remap) for Direct Master-to-PCI	DMPBAM[15:0]
40h	MSW of PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration	DMCFGA[31:16]
42h	LSW of PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration	DMCFGA[15:0]

Table 4-20. Extra Long Serial EEPROM Load Registers

Serial EEPROM Offset	Description	Register Bits Affected
44h	Subsystem ID	PCISID[15:0]
46h	Subsystem Vendor ID	PCISVID[15:0]
48h	MSW of Range for PCI-to-Local Address Space 1 (1 MB)	LAS1RR[31:16]
4Ah	LSW of Range for PCI-to-Local Address Space 1 (1 MB)	LAS1RR[15:0]
4Ch	MSW of Local Base Address (Remap) for PCI-to-Local Address Space 1	LAS1BA[31:16]
4Eh	LSW of Local Base Address (Remap) for PCI-to-Local Address Space 1	LAS1BA[15:0]
50h	MSW of Bus Region Descriptors (Space 1) for PCI-to-Local Accesses	LBRD1[31:16]
52h	LSW of Bus Region Descriptors (Space 1) for PCI-to-Local Accesses	LBRD1[15:0]
54h	MSW of Hot Swap Control	***Reserved***
56h	LSW of Hot Swap Control / Hot Swap Next Capability Pointer	HS_NEXT[7:0] / HS_CNTL[7:0]
58h	PCI Arbiter Control	PCIARB[3:0]
5Ah	***Reserved***	***Reserved***

4.4.2.3 New Capabilities Function Support

The New Capabilities Function Support includes PCI Power Management, Hot Swap, and VPD features, as listed in Table 4-21.

Table 4-21. New Capabilities Function Support Features

New Capability Function	PCI Register Offset Location
First (Power Management)	40h, if the New Capabilities Function Support bit (PCISR[4]) is enabled (PCISR[4] is enabled, by default).
Second (Hot Swap)	48h, which is pointed to from PMNEXT[7:0].
Third (VPD)	4Ch, which is pointed to from HS_NEXT[7:0]. Because PVPD_NEXT[7:0] defaults to zero (0), this indicates that VPD is the last New Capability Function Support feature of the PCI 9056.

4.4.2.4 Recommended Serial EEPROMs

The PCI 9056 is designed to use either a 2K bit (FM93CS56L or compatible) or 4K bit (FM93CS66L or compatible) device.

Note: *The PCI 9056 does not support serial EEPROMs that do not support sequential reads (such as the FM93C56L).*

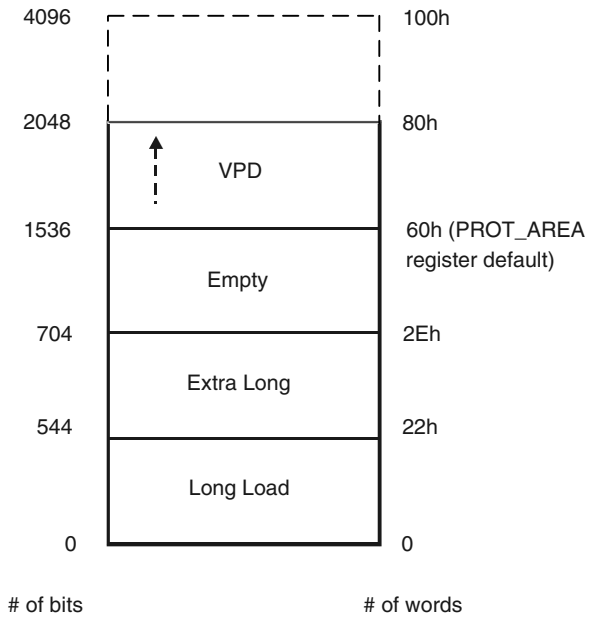


Figure 4-5. Serial EEPROM Memory Map

4.4.3 Internal Register Access

The PCI 9056 provides several internal registers, which allow for maximum flexibility in the bus-interface design and performance. These registers are accessible from the PCI and Local Buses (refer to Figure 4-6) and include the following:

- PCI and Local Configuration registers
- DMA registers
- Mailbox registers
- PCI-to-Local and Local-to-PCI Doorbell registers
- Messaging Queue registers (I_2O)
- Power Management registers
- Hot Swap registers
- VPD registers

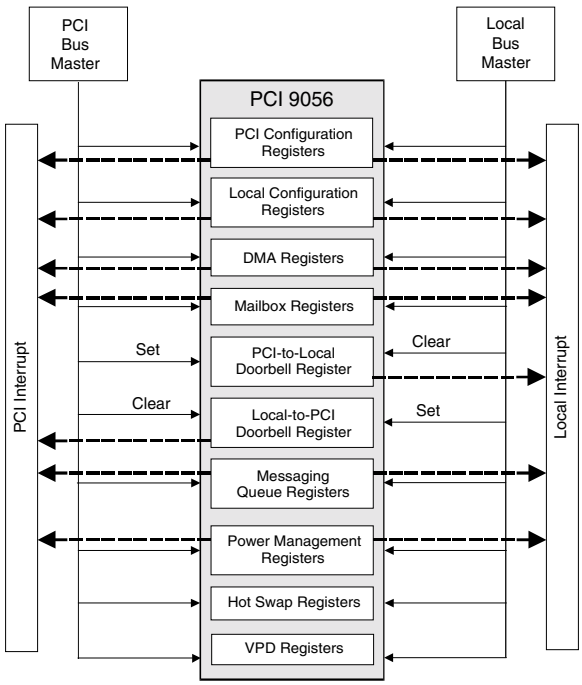


Figure 4-6. PCI 9056 Internal Register Access

4.4.3.1 PCI Bus Access to Internal Registers

The PCI 9056 PCI Configuration registers can be accessed from the PCI Bus with a Configuration Type 0 cycle.

All other PCI 9056 internal registers can be accessed by a Memory cycle, with the PCI Bus address that matches the base address specified in PCI Base Address 0 (PCIBAR0[31:8]) for the PCI 9056 Memory-Mapped Configuration register. These registers can also be accessed by an I/O cycle, with the PCI Bus address matching the base address specified in PCI Base Address 1 for the PCI 9056 I/O-Mapped Configuration register.

All PCI Read or Write accesses to the PCI 9056 registers can be Byte, Word, or Lword accesses. All PCI Memory accesses to the PCI 9056 registers can be Burst or Non-Burst accesses. The PCI 9056 responds with a PCI disconnect for all Burst I/O accesses (PCIBAR1[31:8]) to the PCI 9056 Internal registers.

4.4.3.2 Local Bus Access to Internal Registers

The Local processor can access all PCI 9056 internal registers through an external chip select. The PCI 9056 responds to a Local Bus access when the PCI 9056 Configuration Chip Select input (CCS#) is asserted low. Figure 4-7 illustrates how the Configuration Chip Select logic works.

Notes: *CCS# must be decoded while ADS# is low.*
Accesses must be for a 32-bit non-pipelined bus.

Local Read or Write accesses to the PCI 9056 internal registers can be Byte, Word, or Lword accesses. The Local Bus width must be 32-bit to access internal registers. Eight and 16-bit data buses require external latches to form a 32-bit data path for Local Bus access to internal registers. Local accesses to the PCI 9056 internal registers can be Burst or Non-Burst accesses.

The PCI 9056 READY# signal indicates that Data transfer is complete.

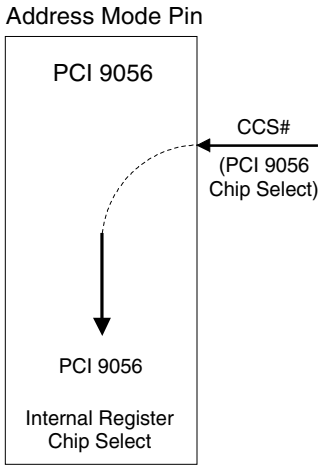
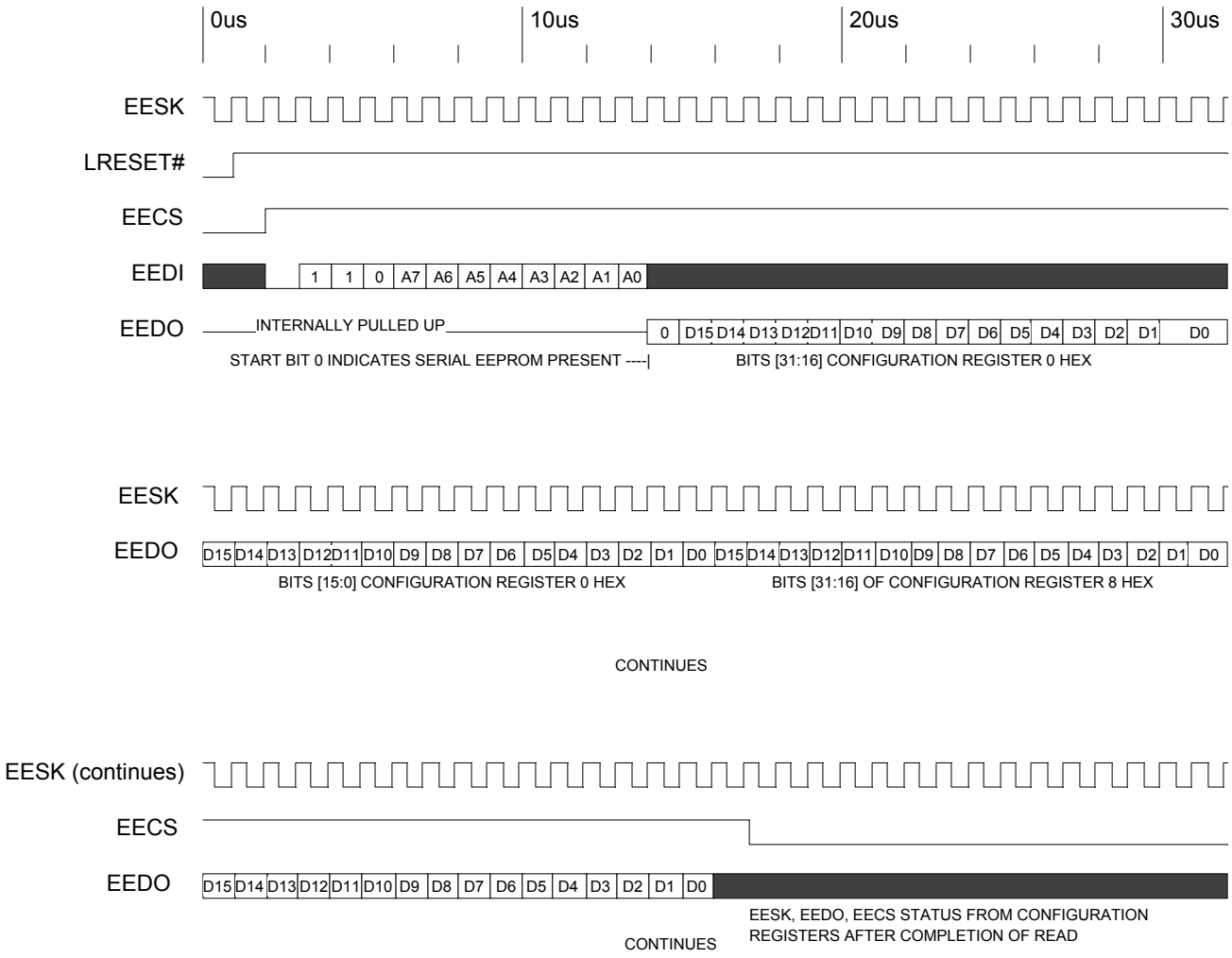


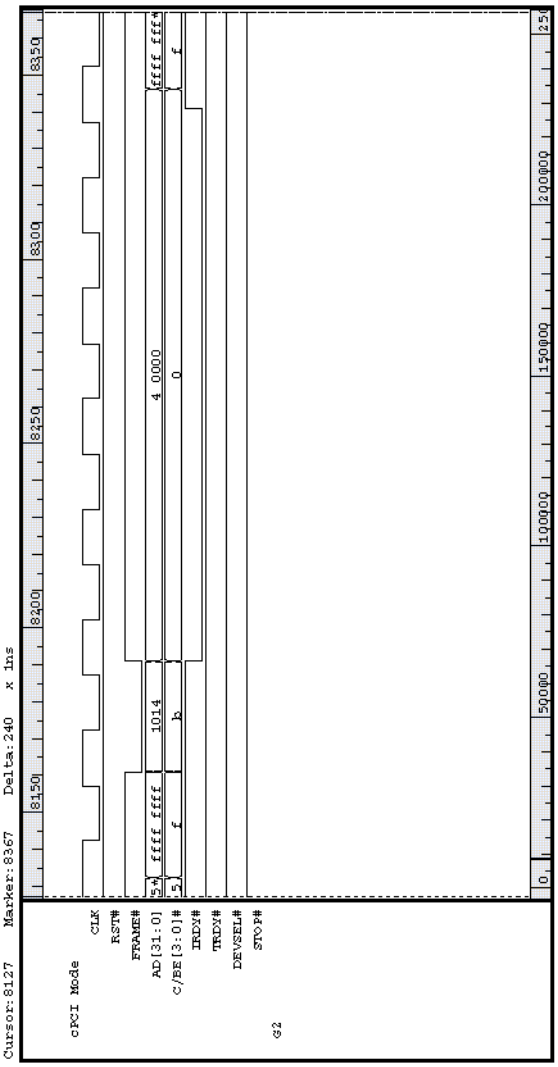
Figure 4-7. Address Decode Mode

4.4.4 Serial EEPROM and Configuration Initialization Timing Diagrams

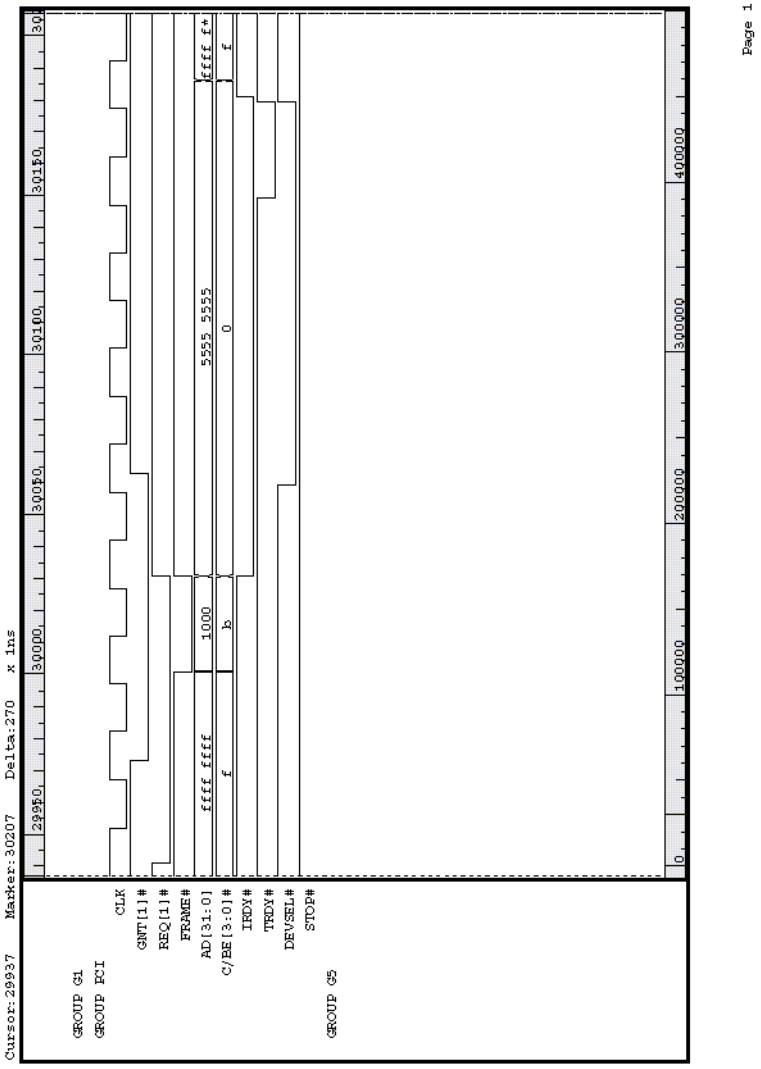
Note: *In the timing diagrams that follow, the "_" symbol at the end of the signal names represents the "#" symbol.*



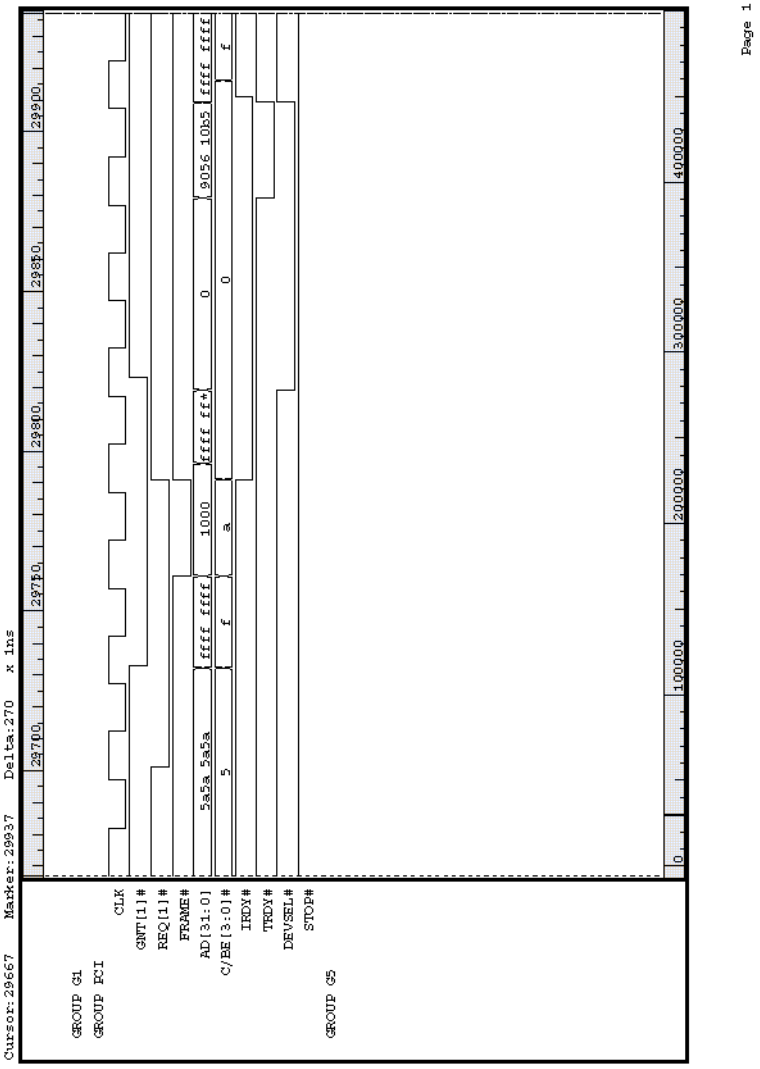
Timing Diagram 4-1. Initialization from Serial EEPROM (2K or 4K Bit)



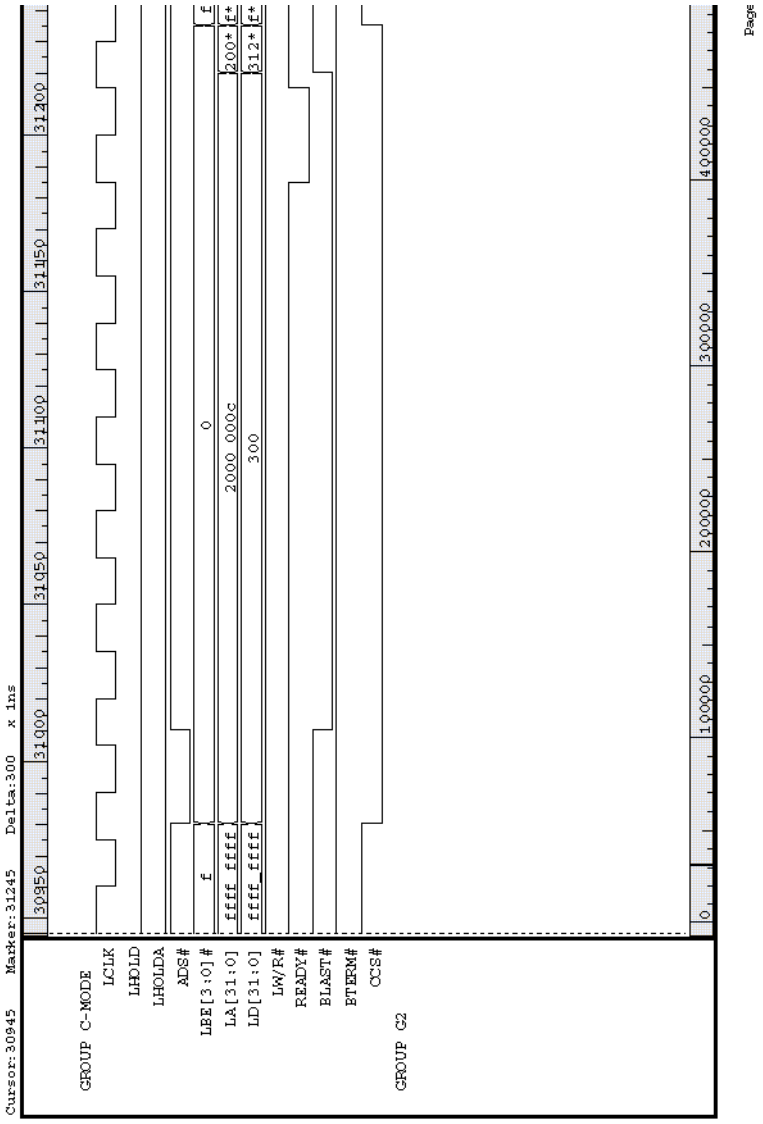
Timing Diagram 4-2. Local Interrupt Asserting PCI Interrupt



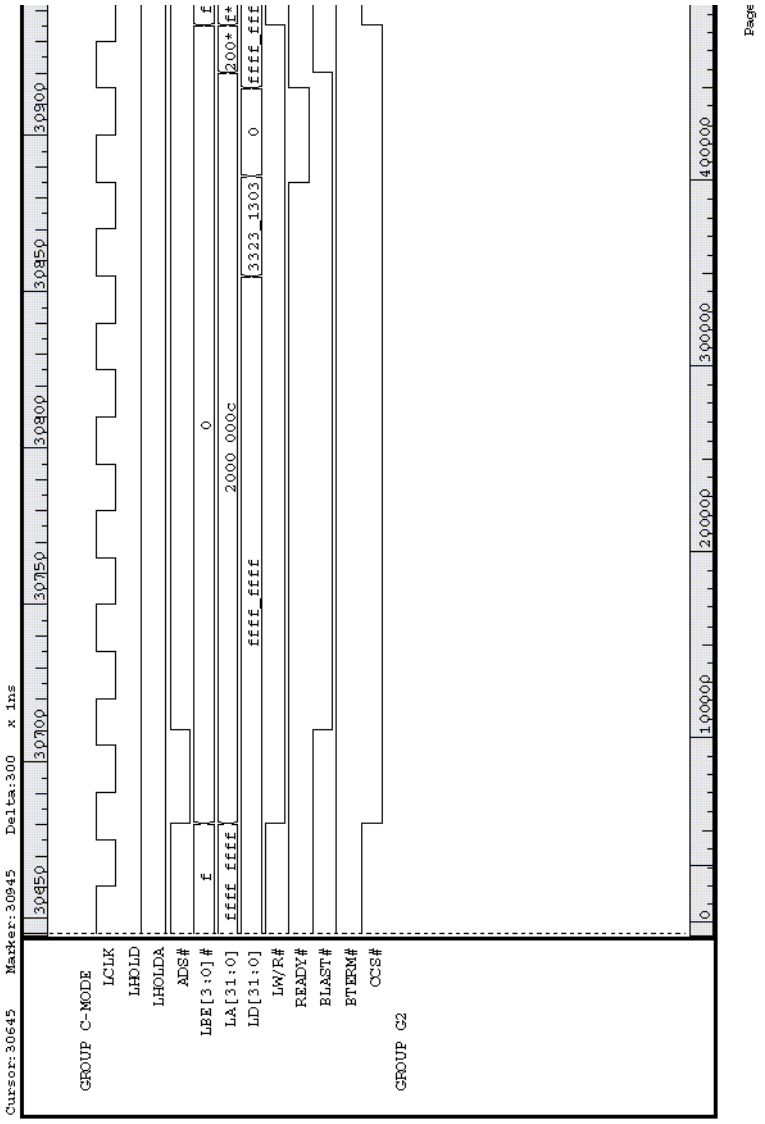
Timing Diagram 4-3. PCI Configuration Write to PCI Configuration Register



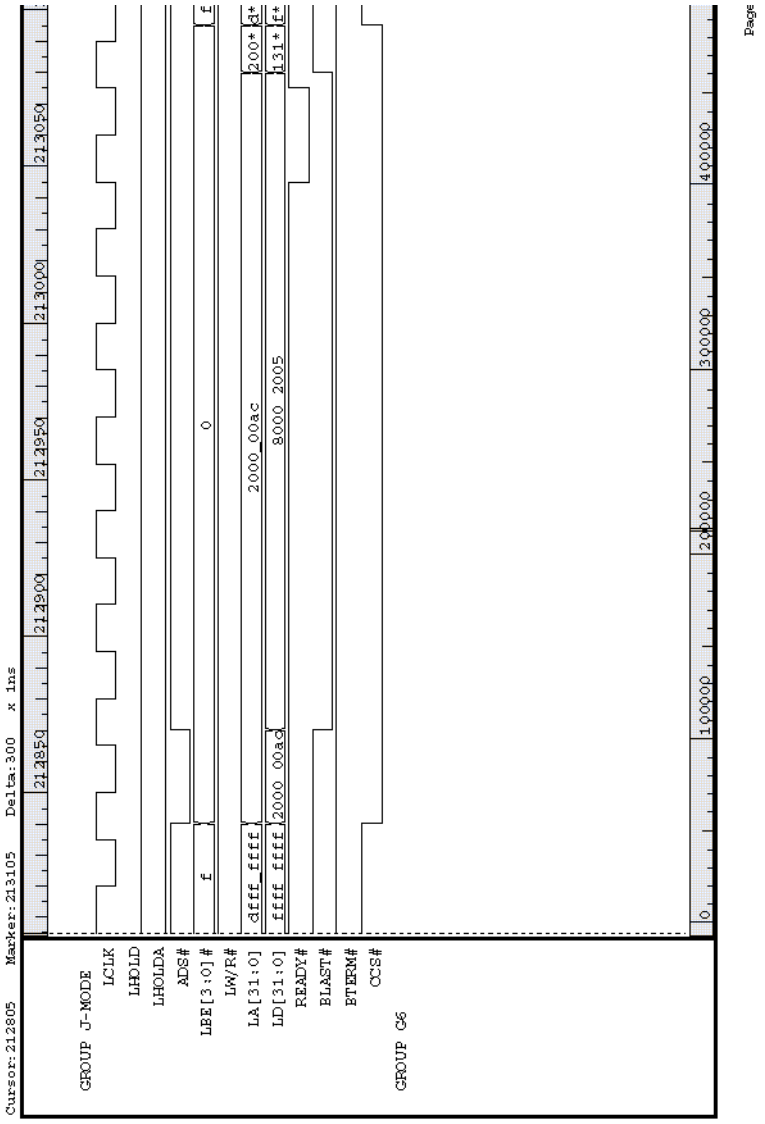
Timing Diagram 4-4. PCI Configuration Read to PCI Configuration Register



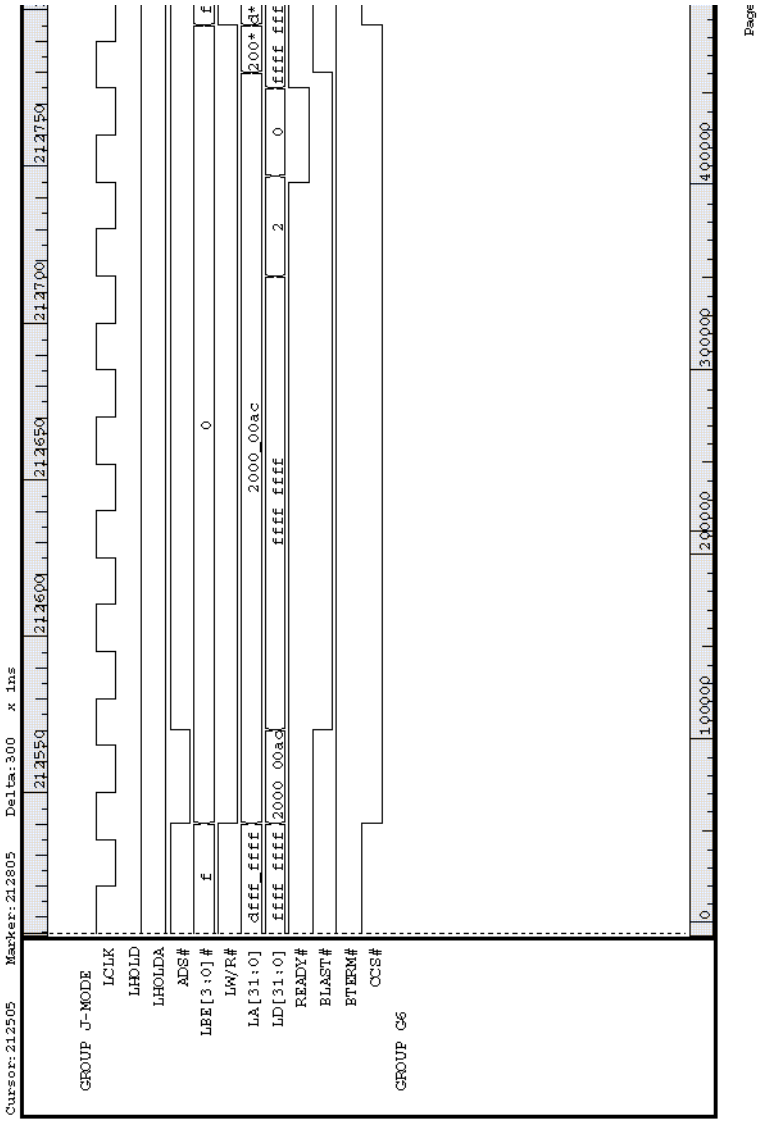
Timing Diagram 4-5. Local Configuration Write to Configuration Register (C Mode)



Timing Diagram 4-6. Local Configuration Read from Configuration Register (C Mode)



Timing Diagram 4-7. Local Configuration Write to Configuration Register (J Mode)



Timing Diagram 4-8. Local Configuration Read from Configuration Register (J Mode)

5 C AND J MODES FUNCTIONAL DESCRIPTION

The functional operation described can be modified through the PCI 9056 programmable internal registers.

5.1 RESET OPERATION

5.1.1 Adapter Mode

5.1.1.1 PCI Bus Input RST#

The PCI Bus RST# input pin is a PCI Host reset. It causes all PCI Bus outputs to float, resets the entire PCI 9056 and causes the Local LRESET# signal to be asserted.

5.1.1.2 Software Reset

A Host on the PCI Bus can set the PCI Adapter Software Reset bit (CNTRL[30]=1) to reset the PCI 9056 and assert LRESET# output. All Local Configuration registers are reset; however, the PCI Configuration DMA and Shared Runtime registers and the Local Init Status bit (LMISC1[2]) are not reset. When the Software Reset bit (CNTRL[30]) is set, the PCI 9056 responds to PCI accesses, but not to Local Bus accesses. The PCI 9056 remains in this reset condition until the PCI Host clears the bit. The serial EEPROM is reloaded, if the Reload Configuration Registers bit is set (CNTRL[29]=1).

Note: *The Local Bus cannot clear this reset bit because the Local Bus is in a reset state, even if the Local processor does not use LRESET# to reset.*

5.1.1.3 Power Management Reset

When the power management reset is asserted (transition from D_3 to any other state), the PCI 9056 resets as if a PCI reset was asserted. (Refer to Section 8, “PCI Power Management.”)

5.1.2 Host Mode

5.1.2.1 PCI Reset

The PCI Bus RST# output is driven when the Local LRESET# signal is asserted, the Software Reset bit is set (CNTRL[30]=1), or the PCI 9056 initiates an external reset.

5.1.2.2 Local LRESET#

When the Local LRESET# pin is asserted by an external source, the Local Bus interface circuitry, the configuration registers, and the PCI 9056 are reset. The PCI 9056 drives the Local LRESET# pin after it detects a reset for 62 clocks.

5.1.2.3 Software Reset

When the Software Reset bit is set (CNTRL[30]=1), the following occurs:

- PCI Master logic is held reset
- PCI 9056 PCI Configuration registers held in reset
- FIFOs are reset
- PCI RST# pin is asserted

Only the PCI Configuration registers are in reset. A software reset can only be cleared from another Host on the Local Bus, and the PCI 9056 remains in this reset condition until a Local Host clears the bit.

Note: *The PCI Bus cannot clear this reset bit because the PCI Bus is in a reset state.*

5.1.2.4 Power Management Reset

Power Management reset is not applicable for Host mode.

5.2 PCI 9056 INITIALIZATION

The PCI 9056 Configuration registers can be programmed by an optional serial EEPROM and/or by a Local processor, as listed in Table 4-18, “Serial EEPROM Guidelines,” on page 4-10. The serial EEPROM can be reloaded by setting the Reload Configuration Registers bit (CNTRL[29]).

The PCI 9056 retries all PCI cycles until the Local Init Status bit is set to “done” (LMISC1[2]=1).

Note: *The PCI Host processor can also access Internal Configuration registers after the Local Init Status bit is set.*

If a PCI Host is present, the Master Enable, Memory Space, and I/O Space bits (PCICR[2:0], respectively) are programmed by that Host after initialization completes (LMISC1[2]=1).

5.3 RESPONSE TO FIFO FULL OR EMPTY

Table 5-1 lists the response of the PCI 9056 to full and empty FIFOs.

5.4 DIRECT DATA TRANSFER MODES

The PCI 9056 supports three direct transfer modes:

- **Direct Master**—Local CPU accesses PCI memory or I/O
- **Direct Slave**—PCI Master accesses Local memory or I/O
- **DMA**—PCI 9056 DMA controller reads/writes PCI memory to/from Local memory

5.4.1 Direct Master Operation (Local Master-to-Direct Slave)

The PCI 9056 supports a direct access to the PCI Bus by the Local processor or an intelligent controller. Master mode must be enabled in the PCI Command register. The following registers define Local-to-PCI accesses:

- Direct Master Memory and I/O Range (DMRR)
- Local Base Address for Direct Master to PCI Memory (DMLBAM)
- Local Base Address for Direct Master to PCI I/O and Configuration (DMLBAI)
- PCI Base Address (DMPBAM)
- Direct Master Configuration (DMCFGA)
- Direct Master PCI Dual Address Cycles (DMDAC)
- Master Enable (PCICR)
- PCI Command Code (CNTRL)

Table 5-1. Response to FIFO Full or Empty

Mode	Direction	FIFO	PCI Bus	Local Bus
Direct Master Write	Local-to-PCI	Full	Normal	De-assert READY#
		Empty	De-assert REQ# (off PCI Bus)	Normal
Direct Master Read	PCI-to-Local	Full	De-assert REQ# or throttle IRDY#[1]	Normal
		Empty	Normal	De-assert READY#
Direct Slave Write	PCI-to-Local	Full	Disconnect or throttle TRDY#[2]	Normal
		Empty	Normal	De-assert LHOLD, assert BLAST#[3]
Direct Slave Read	Local-to-PCI	Full	Normal	De-assert LHOLD, assert BLAST#[3]
		Empty	Throttle TRDY#[2]	Normal
DMA	Local-to-PCI	Full	Normal	De-assert LHOLD, assert BLAST#[3]
		Empty	De-assert REQ#	Normal
	PCI-to-Local	Full	De-assert REQ#	Normal
		Empty	Normal	De-assert LHOLD, assert BLAST#[3]

1. Throttle IRDY# depends on the Direct Master PCI Read Mode bit (DMPBAM[4]).
2. Throttle TRDY# depends on the Direct Slave PCI Write Mode bit (LBRD0[27]).
3. LHOLD de-assert depends upon the Local Bus Direct Slave Release Bus Mode bit (MARBR[21]).

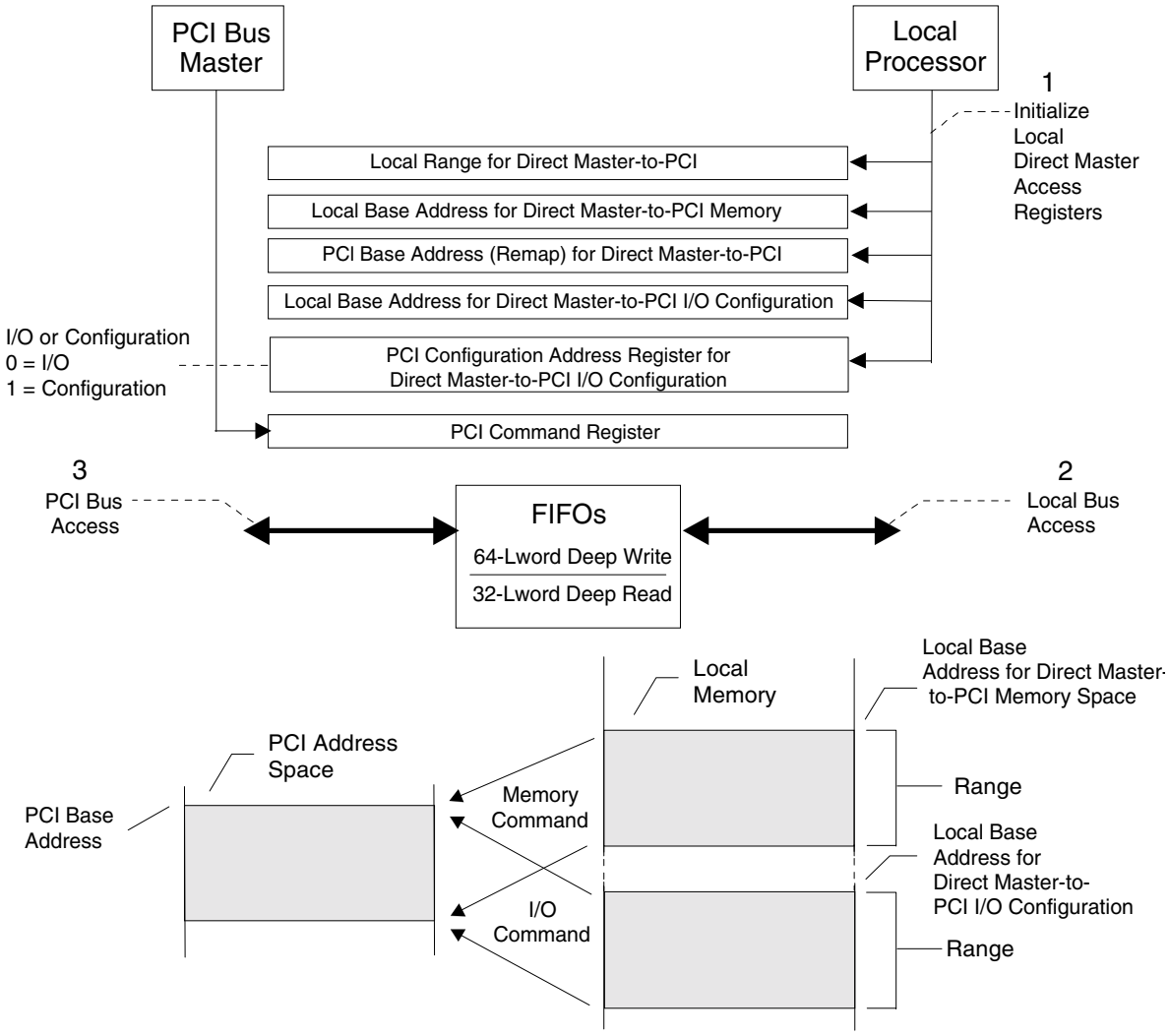


Figure 5-1. Direct Master Access of the PCI Bus

5.4.1.1 Direct Master Memory and I/O Decode

The Range register and the Local Base Address specifies the Local Address bits to use for decoding a Local-to-PCI access (Direct Master). The range of memory or I/O space must be a power of 2 and the Range register value must be the 2's complement of the Range value. In addition, the Local Base Address must be a multiple of the range value.

Any Local Master Address starting from the Direct Master Local Base Address (Memory or I/O) to the range value is recognized as a Direct Master access by the PCI 9056. All Direct Master cycles are then decoded as PCI Memory, I/O, or Configuration Type 0 or Type 1. Moreover, a Direct Master memory or I/O cycle is remapped according to the Remap register value. The Remap Register value must be a multiple of the Direct Master Range value (not the Range register value).

The PCI 9056 can only accept Memory cycles from the Local processor. The Local Base Address and/or the range determine whether PCI Memory or PCI I/O transactions occur.

5.4.1.2 Direct Master FIFOs

For Direct Master Memory access to the PCI Bus, the PCI 9056 has a 64-Lword (256-byte) Write FIFO and a 32-Lword (128-byte) Read FIFO. The FIFOs enable the Local Bus to operate independent of the PCI Bus and allows high-performance bursting on the PCI and Local Buses. In a Direct Master write, the Local processor (Master) writes data to the PCI Bus (Slave). In a Direct Master read, the Local processor (Master) reads data from the PCI Bus (Slave). The FIFOs that function during a Direct Master write and read are illustrated in Figure 5-2 and Figure 5-3.

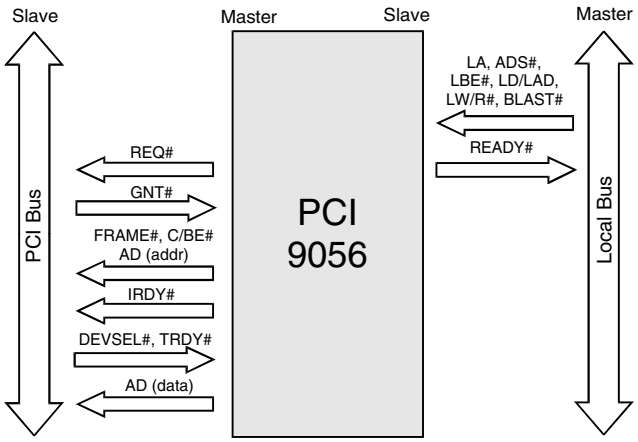


Figure 5-2. Direct Master Write

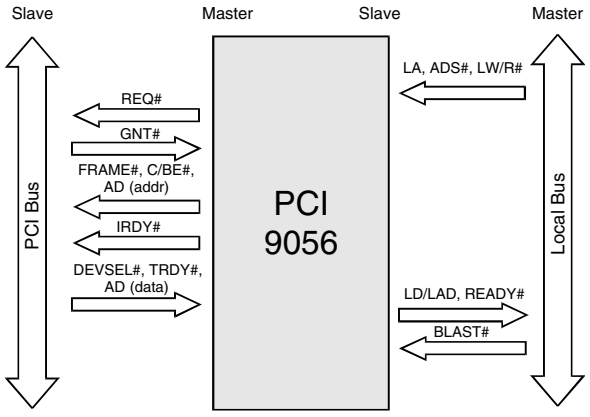


Figure 5-3. Direct Master Read

Note: *Figures 5-2 and 5-3 represent a sequence of Bus cycles.*

5.4.1.3 Direct Master Memory Access

The Local processor transfers data through a single or burst Read/Write Memory transaction to the PCI 9056 and PCI Bus.

Transactions are initiated by the Local Master (LCPU) when the Generic Local Bus memory address matches the Memory space decoded for Direct Master operations. Upon a Generic Local Bus Read, the PCI 9056 becomes a PCI Bus Master, arbitrates for the PCI Bus, and reads data from the PCI Slave device directly into the Direct Master Read FIFO. When sufficient data is placed into the FIFO, it asserts READY# signal onto the Generic Local Bus to indicate that the requested data is on the Generic Local Bus.

The Generic Local processor can read or write to PCI memory. The PCI 9056 converts the Local Read/Write access. The Local Address space starts from Direct Master Local Base Address up to the range. Remap (PCI Base Address) defines the PCI starting address.

The PCI 9056 supports both single and Burst cycles performed by the Generic Local processor.

A Generic Local Bus Processor single cycle causes a single cycle PCI transaction. A Generic Local Processor Burst cycle asserts a Burst cycle PCI transaction. The PCI 9056 supports infinite Burst transfers.

Writes—Upon a Local Bus Write, the Generic Local Bus Master writes data to the Direct Master Write FIFO. When the first data is in the FIFO, the PCI 9056 becomes the PCI Bus Master, arbitrates for the PCI Bus, and writes data to the PCI Slave device. The PCI 9056 continues to accept writes and returns READY# until the Write FIFO is full. It then holds off READY# until space becomes available in the Write FIFO. A programmable Direct Master “almost full” status output is provided (DMPAF).

A Generic Local Processor single cycle Write transaction results in PCI 9056 transfers of one Lword data onto a 32-bit PCI Bus. A Generic Local Processor Burst Cycle Write transaction of two Lwords results in PCI 9056 burst transfers of two Lwords to a 32-bit PCI Bus.

Any type of Burst Cycles of three Lwords or more results in the PCI 9056 bursting data onto the PCI Bus.

The PCI 9056 always starts Direct Master Burst Write transfers on the Lword-aligned PCI Data Addresses. This results in the PCI 9056 performing a dummy PCI cycle with PCI BE# "F" to a Qword-aligned part of the data, when a Qword-unaligned Direct Master Burst Write transfer is performed to a 32-bit PCI slave. Single cycle PCI writes result in a single 32-bit transfer.

Reads—The PCI 9056 holds off READY# while gathering an Lword from the PCI Bus. Programmable prefetch modes are available if prefetch is enabled—prefetch, 4, 8, 16, or continuous—until the Direct Master cycle ends. The Read cycle is terminated when Local BLAST# input is asserted. Unused Read data is flushed from the FIFO.

The PCI 9056 does not prefetch Read PCI data for single cycle Direct Master reads (Local BLAST# input asserted during the first Data phase). In this case, for the 32-bit PCI Bus, the PCI 9056 reads a single PCI Lword unless Direct Master Read Ahead mode is enabled. (Refer to Section 5.4.1.7.)

For single cycle Direct Master reads, the PCI 9056 passes corresponding PCI Bus byte enables from the Generic Local Bus byte enables (LBE#).

For Burst Cycle reads, the PCI 9056 reads entire Lwords (all PCI Bus byte enables are asserted).

If the Direct Master Prefetch Limit bit is enabled (DMPBAM[11]=1), the PCI 9056 terminates a read prefetch at 4-KB boundaries, and restarts it as a new PCI Read Prefetch cycle at the start of a new boundary. If the bit is disabled, the prefetch crosses the 4-KB boundaries.

5.4.1.4 Direct Master I/O Configuration Access

When a Local Direct Master I/O access to the PCI Bus occurs, the PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration Enable bit (DMCFGA[31]) determines whether an I/O or Configuration access is to be made to the PCI Bus.

Local Burst accesses are broken into single PCI I/O (address/data) cycles. The PCI 9056 does not prefetch Read data for I/O and Configuration reads.

For Direct Master I/O or Configuration cycles, the PCI 9056 asserts the same PCI Bus byte enables as set on the Local Bus.

5.4.1.5 Direct Master I/O

If the Configuration Enable bit is cleared (DMCFGA[31]=0), a single I/O access is made to the PCI Bus. The Local Address, Remapped Decode Address bits, and Local byte enables are encoded to provide the address and are output with an I/O Read or Write command during a PCI Address cycle.

When the I/O Remap Select bit is set (DMPBAM[13]=1), the PCI Address bits [31:16] are forced to 0 for the 64-KB I/O address limit.

For writes, data is loaded into the Write FIFO and READY# is returned to the Local Bus. For reads, the PCI 9056 holds off READY# while receiving an Lword from the PCI Bus.

5.4.1.6 Direct Master Delayed Write Mode

The PCI 9056 supports Direct Master Delayed Write mode transactions, where posted Write data accumulates in the Direct Master Write FIFO before the PCI 9056 requests the PCI Bus. Direct Master Delayed Write mode is programmable to delay REQ# assertion for the number of PCI clocks specified in DMPBAM[15:14]. This feature is useful for gaining higher throughput during Direct Master Write Burst transactions for conditions in which the Local clock frequency is slower than the PCI clock frequency.

The PCI 9056 only utilizes the delay counter and accumulates data in the Direct Master Write FIFO for burst transactions on the Local Bus. Otherwise, an immediate single cycle PCI transfer occurs.

5.4.1.7 Direct Master Read Ahead Mode

The PCI 9056 also supports Direct Master Read Ahead mode (DMPBAM[2]), where prefetched data can be read from the internal FIFO of the PCI 9056 instead of from the Local Bus. The address must be subsequent to the previous address and 32-bit aligned (next address = current address + 4) for 32-bit Direct Slave transfers.

A Local Bus single cycle Direct Master transaction, with Read Ahead mode (DMPBAM[2]) enabled results in the PCI 9056 processing continuous PCI Bus Read burst data with all bytes enabled (C/BE# = 0h).

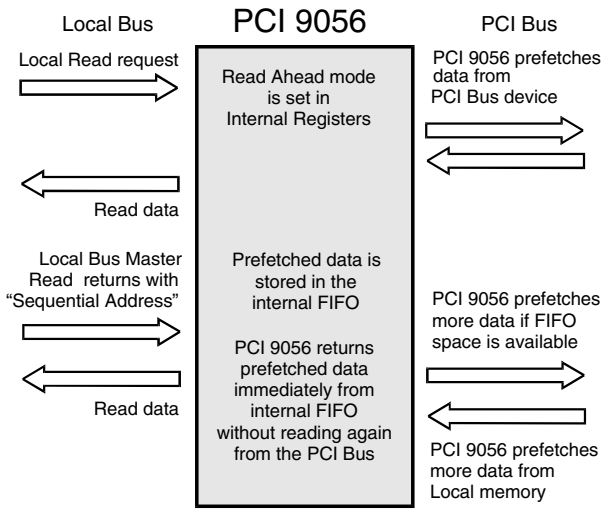


Figure 5-4. Direct Master Read Ahead Mode

Note: *Figure 5-4 represents a sequence of Bus cycles.*

5.4.1.8 Direct Master Configuration (PCI Configuration Type 0 or Type 1 Cycles)

If the Configuration Enable bit is set (DMCFGA[31]=1) is set, and if a Direct Master access is made to the Local Bus address programmed in DMLBAM, a Configuration access is made to the PCI Bus. In addition to enabling configuration of this bit, the user must provide all register information. The Register Number and Device Number bits (DMCFGA[7:2] and DMCFGA[15:11], respectively) must be modified and a new Configuration Read/Write cycle must be performed before accessing other registers or devices.

If the PCI Configuration Address register selects a Type 0 command, register bits [10:0] are copied to address bits [10:0]. Bits [15:11] (device number) are translated into a single bit being set in the PCI Address bits [31:11]. The PCI Address bits [31:11] can be used as a device select. For a Type 1 command, bits [23:0] are copied from the register to PCI address bits [23:0]. The PCI Address bits [31:24] are set to 0. A Configuration Read or Write command code is output with the address during the PCI Address cycle. (Refer to the DMCFGA register.)

For writes, Local data is loaded into the Write FIFO and READY# is returned. For reads, the PCI 9056 holds off READY# while gathering an Lword from the PCI Bus.

5.4.1.8.1 Direct Master Configuration Cycle Example

To perform a Configuration Type 0 cycle to PCI device on AD[21]:

1. The PCI 9056 must be configured to allow Direct Master access to the PCI Bus. The PCI 9056 must also be set to respond to I/O space accesses. These bits must be set (PCICR[2:0]=111b).

   In addition, Direct Master memory and I/O access must be enabled (DMPBAM[1:0]=11).

2. The Local memory map selects the Direct Master range. For this example, use a range of 1 MB:

   1 MB = 2^{20} = 00100000h

   The value to program into the Range register is the 2's complement of 00100000h (FFF00000h):

   DMRR = FFF00000h

3. The Local memory map determines the Local Base Address for the Direct Master-to-PCI I/O Configuration register. For this example, use 40000000h:

   DMLBAI = 40000000h

4. The PCI Address (Remap) for Direct Master-to-PCI Memory register must enable the Direct Master I/O access. The Direct Master I/O Access Enable bit must be set (DMPBAM[1]=1).

5. The user must know which PCI device and PCI Configuration register the PCI Configuration cycle is accessing. This example assumes the IDSEL signal of the Target PCI device is connected to AD[21] (logical device #10=0Ah). It also assumes access is to PCIBAR0 (the fourth register, counting from 0. Use Table 11-2 for reference). Set DMCFGA[31, 23:0] as follows:

Bit	Description	Value
1:0	Configuration Type 0.	00b
7:2	Register Number. Fourth register. Must program a "4" into this value, beginning with bit 2.	000100b
10:8	Function Number.	000b
15:11	Device Number n-11, where n is the value in AD[n]=21-11 = 10.	01010b
23:16	Bus Number.	00000000b
31	Configuration Enable.	1

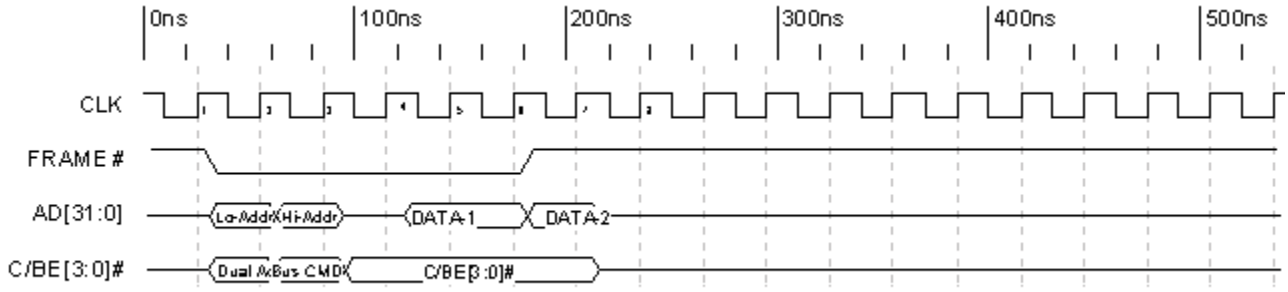


Figure 5-5. Dual Address Timing

After these registers are configured, a simple Local Master Memory cycle to the I/O Base Address is necessary to generate a PCI Configuration Read or Write cycle. An offset to the Base Address is not necessary because the register offset for the read or write is specified in the Configuration register. The PCI 9056 takes the Local Bus Master Memory cycle and checks for the Configuration Enable bit (DMCFGA[31]). If set, the PCI 9056 converts the current cycle to a PCI Configuration cycle, using the DMCFGA register and the Write/Read signal (LW/R#).

The Register Number and Device Number bits (DMCFGA[7:2] and DMCFGA[15:11], respectively) must be modified and a new Configuration Read/Write cycle must be performed before accessing other registers or devices.

5.4.1.9 Direct Master PCI Dual Address Cycle

The PCI 9056 supports PCI Dual Address Cycle (DAC) when it is a PCI Bus Master using the DMDAC register for Direct Master transactions. The DAC command is used to transfer a 32-bit address to devices that support 32-bit addressing when the address is not in the low 4-GB Address space. The PCI 9056 performs the address portion of a DAC in two PCI clock periods, where the first PCI address is a Lo-Addr with the command (C/BE[3:0]#) “D” and the second PCI address will be a Hi-Addr with the command (C/BE[3:0]#) “6” or “7”, depending upon it being a PCI Read or a PCI Write cycle. Whenever the DMDAC register contains a value of 0x00000000, the PCI 9056 performs a Single Address Cycle (SAC) on the PCI Bus. (Refer to Figure 5-5.)

5.4.1.10 PCI Master/Target Abort

The PCI 9056 PCI Master/Target Abort logic enables a Local Bus Master to perform a Direct Master Bus device poll to determine if devices exist (typically when the Local Bus performs Configuration cycles to the PCI Bus). When a PCI Master device attempts to access but does not receive DEVSEL# within six PCI clocks, it results in a Master Abort. The Local Bus Master must clear the Received Master Abort bit or Target Abort bit (PCISR[13 or 11]=0, respectively) and continue by processing the next task.

If a PCI Master/Target Abort, or Retry Timeout is encountered during a transfer, the PCI 9056 asserts LSERR# if enabled [INTCSR[1:0]=1, which can be used as a Non-Maskable Interrupt (NMI)]. If a Local Bus Master is waiting for READY#, it is asserted along with BTERM#. The Local Master’s interrupt handler can take the appropriate application-specific action It can then clear the Target Abort bit (PCISR[11]) to de-assert the LSERR# interrupt and re-enable Direct Master transfers.

If a Local Bus Master is attempting a Burst read from a nonresponding PCI device (Master/Target Abort), it receives READY# and BTERM# for the first cycle only. In addition, the PCI 9056 asserts LSERR# if the Enable Local Bus LSERR# bits are enabled (INTCSR[1:0], which can be used as an NMI). If the Local processor cannot terminate its Burst cycle, it may cause the Local processor to hang. The Local Bus must then be reset from the PCI Bus. If a Local Bus Master cannot terminate its cycle with BTERM# output, it should not perform Burst cycles when attempting to determine whether a PCI device exists.

If a PCI Master/Target Abort is encountered during during a Direct Master transfer, the PCI 9056 stores the PCI Abort Address into the PCI Abort Address register bits (PABTADR[31:0]).

5.4.1.11 Direct Master Memory Write and Invalidate

The PCI 9056 can be programmed to perform Memory Write and Invalidate cycles to the PCI Bus for Direct Master transfers, as well as for DMA transfers. (Refer to Section 5.5.4.) The PCI 9056 supports Memory Write and Invalidate transfers for cache line sizes of 8 or 16 Lwords. Size is specified in the System Cache Line Size bits (PCICLSR[7:0]). If a size other than 8 or 16 is specified, the PCI 9056 performs Write transfers rather than Memory Write and Invalidate transfers.

Direct Master Memory Write and Invalidate transfers are enabled when the Invalidate Enable and the Memory Write and Invalidate Enable bits are set (DMPBAM[9] and PCICR[4], respectively).

In Memory Write and Invalidate mode, if the start address of the Direct Master transfer is on a cache line boundary, the PCI 9056 waits until the number of Lwords required for the specified cache line size are written from the Local Bus before starting a PCI Memory Write and Invalidate access. This ensures a complete cache line write can complete in one PCI Bus ownership.

If the start address is not on a cache line boundary, the PCI 9056 starts a normal PCI Write access (PCI command code = 7h). The PCI 9056 does not terminate a normal PCI Write at an MWI cache boundary. The normal PCI Write transfer continues until the Data transfer is complete. If a Target disconnects before a cache line is completed, the PCI 9056 completes the remainder of that cache line, using normal writes.

5.4.2 Direct Slave Operation (PCI Master-to-Local Bus Access)

The PCI 9056 supports burst Memory-Mapped Transfer accesses and I/O-Mapped, PCI-to-Generic Local Bus single Transfer accesses through a 32-Lword (128-byte) Direct Slave Read FIFO and a 64-Lword (256-byte) Direct Slave Write FIFO. The PCI Base Address registers are provided to set up the location of the adapter in the PCI memory and the I/O space. In addition, Local mapping registers allow address translation from the PCI Address Space to the Local Address Space.

Three spaces are available:

- Space 0
- Space 1
- Expansion ROM

Expansion ROM is intended to support a bootable ROM device for the Host.

Writes—Upon a PCI Bus Write, the PCI Bus Master writes data to the Direct Slave Write FIFO. When the first data is in the FIFO, the PCI 9056 becomes the Generic Local Bus Master, arbitrates for the Generic Local Bus, and writes data to a Generic Local Slave device. The PCI 9056 continues to accept writes and returns TRDY# until the Write FIFO is full. It then holds off TRDY# until space becomes available in the Write FIFO or asserts STOP#, and Retries the PCI Bus Master, dependent upon the register bit setting (LBRD0[27]).

A 32-bit PCI Bus Master single cycle Write transaction results in a PCI 9056 transfer of one Lword of data onto a Generic Local Bus.

Reads—The PCI 9056 holds off TRDY# while gathering an Lword from the Local Bus, unless the Delayed Read Mode bit is enabled (MARBR[24]=1). (Refer to Section 5.4.2.2.) Programmable Prefetch modes are available, if prefetch is enabled—prefetch, 0-16, or continuous—until the Direct Slave read ends. The Read cycles are terminated on the following clock after FRAME# is de-asserted or the PCI 9056 issues a Retry or disconnect.

For the highest data transfer rate, the PCI 9056 supports posted writes and can be programmed to prefetch data during a PCI Burst read. The Prefetch size, when enabled, can be from one to 16 Lwords or until the PCI Bus stops requesting. When the PCI 9056 prefetches, if enabled, it drops the Generic Local Bus after reaching the prefetch counter limit. In Continuous Prefetch mode, the PCI 9056 prefetches as long as FIFO space is available and stops prefetching when the PCI Bus terminates the request. If Read prefetching is disabled, the PCI 9056 disconnects after one Read transfer.

In addition to Prefetch mode, the PCI 9056 supports Read Ahead mode. (Refer to Section 5.4.2.3.)

Only 32-bit PCI Bus single cycle Direct Slave Read transactions result in the PCI 9056 passing requested PCI bytes (C/BE#) to a Generic Local Bus Target device by way of LBE[3:0]# assertion back to a PCI Bus Master. This transaction results in the PCI 9056 reading one Lword or partial Lword data. For any other types of Read transactions (Burst transfers or Unaligned), the PCI 9056 reads Generic Local Bus data with all bytes asserted (LBE[3:0]# = 0h).

Each Local space can be programmed to operate in an 8-, 16-, or 32-bit Local Bus width. The PCI 9056 has an internal wait state generator and external wait state input, READY#. READY# can be disabled or enabled with the Internal Configuration registers.

With or without wait state(s), the Local Bus, independent of the PCI Bus, can perform the following:

- Burst as long as data is available (Continuous Burst mode)
- Burst four Lwords at a time (recommended)
- Perform continuous single cycles

5.4.2.1 Direct Slave Lock

The PCI 9056 supports direct PCI-to-Local-Bus Exclusive accesses (locked atomic operations). A PCI-locked operation to the Local Bus results in the entire address Space 0, Space 1, and Expansion ROM space being locked until they are released by the PCI Bus Master. Locked operations are enabled or disabled with the Direct Slave LOCK# Enable bit (MARBR[22]).

5.4.2.2 Direct Slave Delayed Read Mode

The PCI 9056 can be programmed through the Delayed Read Mode bit (MARBR[24]=1) to perform delayed reads.

PCI Bus single cycle aligned or unaligned 32-bit Direct Slave Delayed Read transactions always result in a 1-Lword single cycle transfer on the Local Bus, with corresponding Local byte enables LBE[3:0]# asserted to reflect the PCI byte enables (C/BE#), unless the PCI Read No Flush Mode bit is enabled (MARBR[28]=1). (Refer to Section 5.4.2.3.) This causes the PCI 9056 to Retry all PCI Bus Read requests that follow, until the original PCI byte enables (C/BE#) are matched.

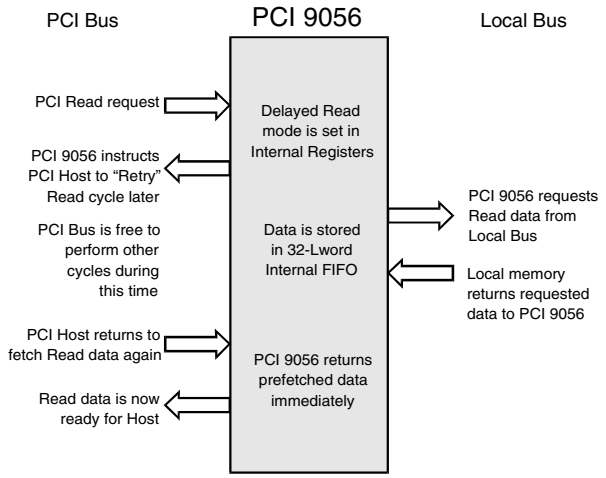


Figure 5-6. Direct Slave Delayed Read

Note: *Figure 5-6 represents a sequence of Bus cycles.*

In addition to delayed reads, the PCI 9056 supports the following Delayed Read mode functions:

- No writes while a read is pending (PCI Retry for writes)
- Write and flush pending read

5.4.2.3 Direct Slave Read Ahead Mode

The PCI 9056 also supports Direct Slave Read Ahead mode (MARBR[28]), where prefetched data can be read from the internal FIFO of the PCI 9056 instead of from the Local Bus. The address must be subsequent to the previous address and 32-bit aligned (next address = current address + 4) for 32-bit Direct Slave transfers.

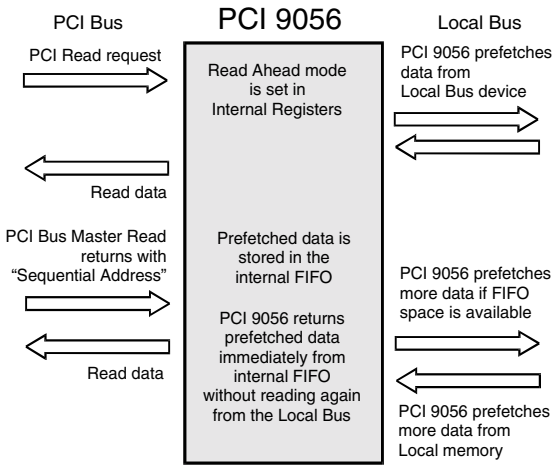


Figure 5-7. Direct Slave Read Ahead Mode

Note: *Figure 5-7 represents a sequence of Bus cycles.*

5.4.2.4 Direct Slave Delayed Write Mode

The PCI 9056 supports Direct Slave Delayed Write mode transactions, where posted Write data accumulates in the Direct Slave Write FIFO before the PCI 9056 requests a Write transaction (ADS# and/or ALE assertion) to be performed on the Local Bus. The Direct Slave Delayed Write mode is programmable to delay the ADS# and/or ALE assertion in the amount of Local clocks (LMISC2[4:2]). This feature is useful for gaining higher throughput during Direct Slave Write burst transactions for conditions in which the PCI clock frequency is slower than the Local clock frequency.

5.4.2.5 Direct Slave Local Bus READY# Timeout Mode

The PCI 9056 supports Direct Slave Local Bus READY# Timeout mode transactions, where the PCI 9056 asserts an internal READY# signal to recover from stalling the Local and PCI Buses. The Direct Slave Local Bus READY# Timeout mode transaction is programmable to select the amount of Local clocks before READY# times out (LMISC2[1:0]). If a Local Slave stalls with a READY# assertion during Direct Slave Write transactions, the PCI 9056 empties the Write FIFO by dumping the data into the Local Bus and does not pass an error condition to the PCI Bus Initiator. During Direct Slave Read transactions, the PCI 9056 issues a Direct Slave Abort to the PCI Bus Initiator every time the Direct Slave Local Bus READY# Timeout is detected.

5.4.2.6 Direct Slave Transfer

A PCI Bus Master addressing the Memory space decoded for the Local Bus initiates transactions. Upon a PCI Read/Write, the PCI 9056 becomes a Local Bus Master and arbitrates for the Local Bus.

The PCI 9056 then reads data into the Direct Slave Read FIFO or writes data to the Local Bus.

The Direct Slave or Direct Master preempts DMA; however, the Direct Slave does not preempt the Direct Master. (Refer to Section 5.4.3.1.)

The PCI 9056 can be programmed to retain the PCI Bus by generating a wait state(s) and de-asserting TRDY#, if the Write FIFO becomes full. The PCI 9056 can also be programmed to retain the Local Bus and continue asserting LHOLD, if the Direct Slave Write FIFO becomes empty or the Direct Slave Read FIFO becomes full. In either case, the Local Bus is dropped when the Local Bus Latency Timer is enabled and expires (MARBR[7:0]).

For Direct Slave writes, the PCI Bus writes data to the Local Bus. The Direct Slave is the "Command from the PCI Host," which has highest priority.

For Direct Slave reads, the PCI Bus Master reads data from the Local Bus Slave.

The PCI 9056 supports on-the-fly Endian conversion for Space 0, Space 1, and Expansion ROM space. The Local Bus can be Big/Little Endian by using the programmable internal register configuration.

Note: *The PCI Bus is always Little Endian.*

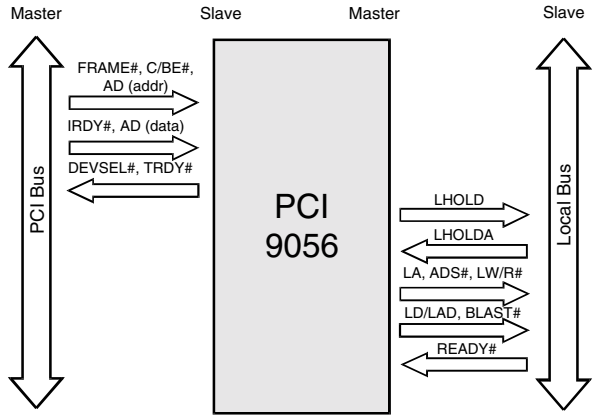


Figure 5-8. Direct Slave Write

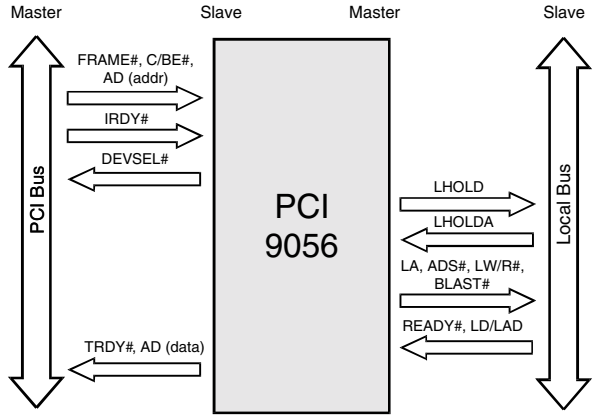


Figure 5-9. Direct Slave Read

Note: *Figures 5-8 and 5-9 represent a sequence of Bus cycles.*

5.4.2.7 Direct Slave PCI-to-Local Address Mapping

Note: *In I_2O mode (QSR[0]=1), Memory-Mapped Local Configuration registers and Space 1 share the PCIBAR0 Base Address. Refer to Section 7.1.10.*

Three Local Address spaces—Space 0, Space 1, and Expansion ROM—are accessible from the PCI Bus. Each is defined by a set of three registers:

- Local Address Range (LAS0RR, LAS1RR, and/or EROMRR)
- Local Base Address (LAS0BA, LAS1BA, and/or EROMBA)
- PCI Base Address (PCIBAR2, PCIBAR3, and/or PCIERBAR)

A fourth register, the Bus Region Descriptor register(s) for PCI-to-Local Accesses (LBRD0 and/or LBRD1), defines the Local Bus characteristics for the Direct Slave regions. (Refer to Figure 5-10.)

Each PCI-to-Local Address space is defined as part of reset initialization, as described in Section 5.4.2.7.1. These Local Bus characteristics can be modified at any time before actual data transactions.

5.4.2.7.1 Direct Slave Local Bus Initialization

Range—Specifies which PCI Address bits to use for decoding a PCI access to Local Bus space. Each bit corresponds to a PCI Address bit. Bit 31 corresponds to address bit 31. Write 1 to all bits that must be included in decode and 0 to all others.

Remap PCI-to-Local Addresses into a Local Address Space—Bits in this register remap (replace) the PCI Address bits used in decode as the Local Address bits.

Local Bus Region Descriptor—Specifies the Local Bus characteristics.

5.4.2.7.2 Direct Slave PCI Initialization

After a PCI reset, the software determines how much address space is required by writing all ones (1) to a PCI Base Address register and then reading back the value. The PCI 9056 returns zeroes (0) in the Don't Care Address bits, effectively specifying the address space required. The PCI software then maps the Local Address space into the PCI Address space by programming the PCI Base Address register. (Refer to Figure 5-10.)

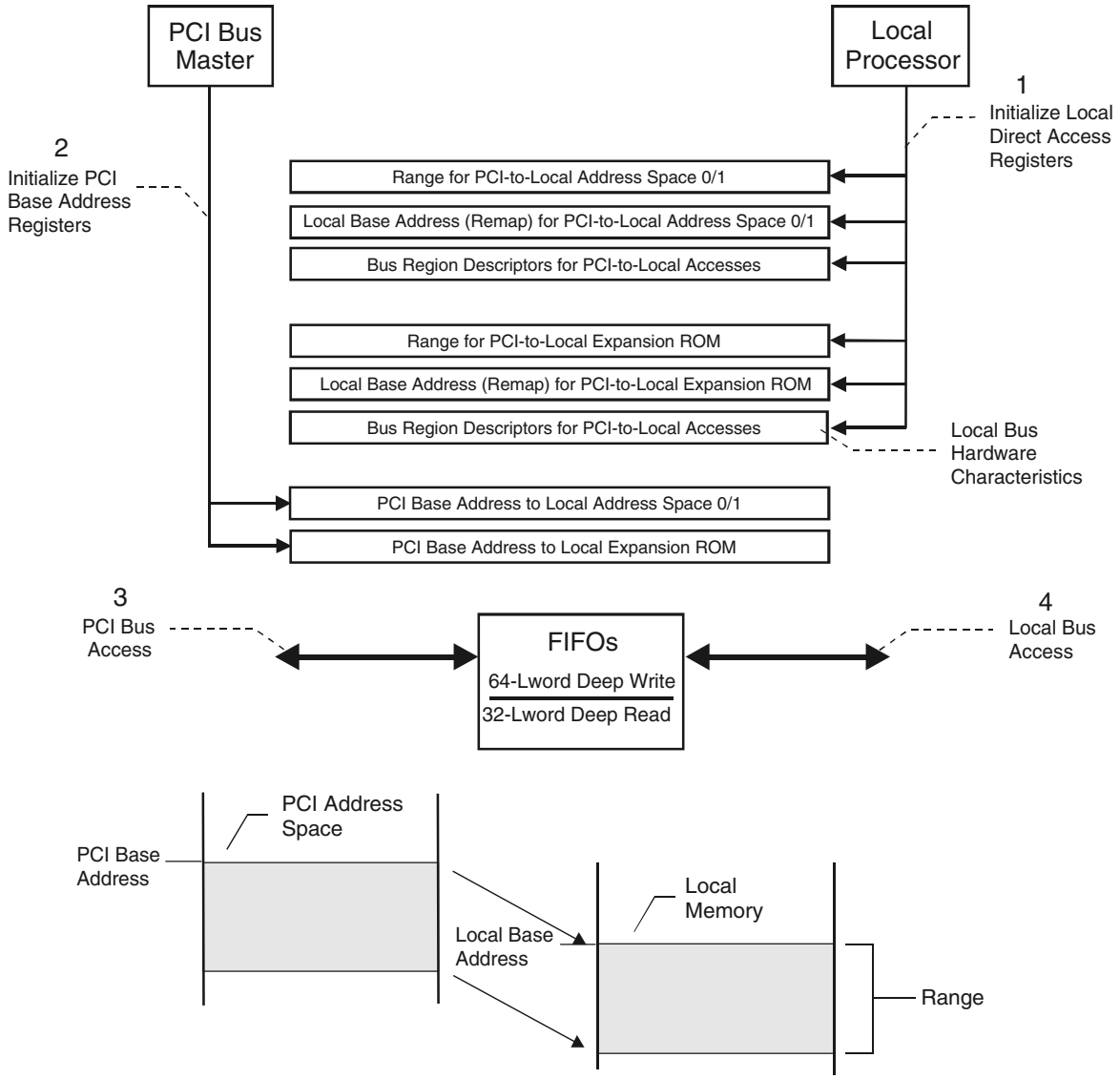


Figure 5-10. Local Bus Direct Slave Access

5.4.2.7.3 Direct Slave Byte Enables (C Mode)

During a Direct Slave transfer, each of three spaces (Space 0, Space 1, and Expansion ROM) can be programmed to operate in an 8-, 16-, or 32-bit Local Bus width by encoding the Local Byte Enables (LBE[3:0]#).

LBE[3:0]# are encoded, based on the configured bus width, as follows:

32-Bit Bus—The four-byte enables indicate which of the four bytes are active during a Data cycle:

- LBE3# Byte Enable 3—LD[31:24]
- LBE2# Byte Enable 2—LD[23:16]
- LBE1# Byte Enable 1—LD[15:8]
- LBE0# Byte Enable 0—LD[7:0]

16-Bit Bus—LBE[3, 1:0]# are encoded to provide BHE#, LA1, and BLE#, respectively:

- LBE3# Byte High Enable (BHE#)—LD[15:8]
- LBE2# not used
- LBE1# Address bit 1 (LA1)
- LBE0# Byte Low Enable (BLE#)—LD[7:0]

8-Bit Bus—LBE[1:0]# are encoded to provide LA[1:0], respectively:

- LBE3# not used
- LBE2# not used
- LBE1# Address bit 1 (LA1)
- LBE0# Address bit 0 (LA0)

5.4.2.7.4 Direct Slave Byte Enables (J Mode)

During a Direct Slave transfer, each of three spaces (Space 0, Space 1, and Expansion ROM) can be programmed to operate in an 8-, 16-, or 32-bit Local Bus width by encoding the Local Byte Enables (LBE[3:0]#).

LBE[3:0]# are encoded, based on the configured bus width, as follows:

32-Bit Bus—The four-byte enables indicate which of the four bytes are active during a Data cycle:

- LBE3# Byte Enable 3—LAD[31:24]
- LBE2# Byte Enable 2—LAD[23:16]
- LBE1# Byte Enable 1—LAD[15:8]
- LBE0# Byte Enable 0—LAD[7:0]

16-Bit Bus—LBE[3, 1:0]# are encoded to provide BHE#, LAD1, and BLE#, respectively:

- LBE3# Byte High Enable (BHE#)—LAD[15:8]
- LBE2# not used
- LBE1# Address bit 1 (LAD1)
- LBE0# Byte Low Enable (BLE#)—LAD[7:0]

8-Bit Bus—LBE[1:0]# are encoded to provide LAD[1:0], respectively:

- LBE3# not used
- LBE2# not used
- LBE1# Address bit 1 (LAD1)
- LBE0# Address bit 0 (LAD0)

5.4.2.7.4.1 Direct Slave Byte Enables Example

A 1 MB Local Address Space, 12300000h through 123FFFFFh, is accessible from the PCI Bus at PCI addresses 78900000h through 789FFFFFh.

a. Local initialization software sets the Range and Local Base Address registers as follows:
   - **Range**—FFF00000h (1 MB, decode the upper 12 PCI Address bits)
   - **Local Base Address (Remap)**—123XXXXXh (Local Base Address for PCI-to-Local accesses) [Space Enable bit(s) must be set to be recognized by the PCI Host (LAS0BA[0]=1 and/or LAS1BA[0]=1)]

b. PCI Initialization software writes all ones (1) to the PCI Base Address, then reads it back again.
   - The PCI 9056 returns a value of FFF00000h. The PCI software then writes to the PCI Base Address register(s).
   - **PCI Base Address**—789XXXXXh (PCI Base Address for Access to the Local Address Space registers, PCIBAR2 and PCIBAR3).

For a PCI Direct access to the Local Bus, the PCI 9056 has a 64-Lword (256-byte) Write FIFO and a 32-Lword (128-byte) Read FIFO. The FIFOs enable the Local Bus to operate independent of the PCI Bus. The PCI 9056 can be programmed to return a Retry response or to throttle TRDY# for any PCI Bus transaction attempting to write to the PCI 9056 Local Bus when the FIFO is full.

For PCI Read transactions from the Local Bus, the PCI 9056 holds off TRDY# while gathering data from the Local Bus. For Read accesses mapped to PCI Memory space, the PCI 9056 prefetches up to 16 Lwords (has Continuous Prefetch mode) from the Local Bus. Unused Read data is flushed from the FIFO. For Read accesses mapped to PCI I/O space, the PCI 9056 does not prefetch Read data. Rather, it breaks each read of a Burst cycle into a single Address/Data cycle on the Local Bus.

The Direct Slave Retry Delay Clocks bits (LBRD0[31:28]) can be used to program the period of time in which the PCI 9056 holds off TRDY#. The PCI 9056 issues a Retry to the PCI Bus Transaction Master when the programmed time period expires. This occurs when the PCI 9056 cannot gain control of the Local Bus and return TRDY# within the programmed time period.

5.4.2.8 Direct Slave Priority

Direct Slave accesses have a higher priority than DMA accesses, thereby preempting DMA transfers. During a DMA transfer, if the PCI 9056 detects a pending Direct Slave access, it releases the Local Bus within two Data transfers. The PCI 9056 resumes operation after the Direct Slave access completes.

When the PCI 9056 DMA controller owns the Local Bus, its LHOLD output and LHOLDA input are asserted. When a Direct Slave access occurs, the PCI 9056 releases the Local Bus within two Lword transfers by de-asserting LHOLD and floating the Local Bus outputs. After the PCI 9056 acknowledges that LHOLDA is de-asserted, it requests the Local Bus for a Direct Slave transfer by asserting LHOLD. When the PCI 9056 receives LHOLDA, it drives the bus and performs the Direct Slave transfer. Upon completing a Direct Slave transfer, the PCI 9056 releases the Local Bus by de-asserting LHOLD and floating the Local Bus outputs. After the PCI 9056 samples LHOLDA is de-asserted and the Local Bus Pause Timer is set to zero (0), it requests a DMA transfer from the Local Bus by re-asserting LHOLD. When it receives LHOLDA, it drives the bus and continues the DMA transfer.

5.4.3 Deadlock Conditions

Deadlock can occur when a PCI Bus Master must access the PCI 9056 Local Bus at the same time a Master on the PCI 9056 Local Bus must access the PCI Bus.

There are two types of deadlock:

- **Partial Deadlock**—A Local Bus Master is performing a Direct Bus Master access to a PCI Bus device other than the PCI Bus device concurrently trying to access the Local Bus
- **Full Deadlock**—A Local Bus Master is performing a Direct Bus Master access to the same PCI Bus device concurrently trying to access the Local Bus

This applies only to Direct Master and Direct Slave accesses through the PCI 9056. Deadlock does not occur in transfers through the PCI 9056 DMA channels or the PCI 9056 internal registers (such as mailboxes).

For partial deadlock, the PCI access to the Local Bus times out [the Direct Slave Retry Delay Clock (LBRD0[31:28]), which is programmable through the Local Bus Region Descriptor register] and the PCI 9056 responds with a PCI Retry. *PCI r2.2* requires that a PCI Master release its request for the PCI Bus (de-assert REQ#) for a minimum of two PCI clocks after receiving a Retry. This allows the PCI Bus arbiter to grant the PCI Bus to the PCI 9056 so that it can complete its Direct Master access and free up the Local Bus. Possible solutions are described in the following sections for cases in which the PCI Bus arbiter does not function as described (PCI Bus architecture dependent), waiting for a time out is undesirable, or a full deadlock condition exists.

When a full deadlock occurs, the only solution is to back off the Local Bus Master.

5.4.3.1 Backoff

The PCI 9056 BREQo signal indicates whether a possible deadlock condition exists. The PCI 9056 starts the Backoff Timer (programmable through registers) when it detects the following conditions:

- A PCI Bus Master is attempting to access memory or an I/O device on the Local Bus and is not gaining access (*for example*, LHOLDA is not received).
- A Local Bus Master is performing a Direct Bus Master Read access to the PCI Bus. Or, a Local Bus Master is performing a Direct Bus Master Write access to the PCI Bus and the PCI 9056 Direct Master Write FIFO cannot accept another Write cycle.

If the Local Bus Backoff Enable bit is enabled (EROMBA[4]=1), the Backoff Timer expires, and the PCI 9056 has not received LHOLDA, the PCI 9056 asserts BREQo. External bus logic can use this signal to perform backoff.

The Backoff cycle is device/bus architecture dependent. External logic (an arbiter) can assert the necessary signals necessary to cause a Local Bus Master to release a Local Bus (backoff). After the Local Bus Master backs off, it can grant the bus to the PCI 9056 by asserting LHOLDA.

Once BREQo is asserted, READY# for the current Data cycle is never asserted (the Local Bus Master must perform backoff). When the PCI 9056 detects LHOLDA, it proceeds with the PCI Master-to-Local-Bus access. When this access completes and the PCI 9056 releases the Local Bus, external logic can release the backoff and the Local Bus Master can resume the cycle interrupted by the Backoff cycle. The PCI 9056 Write FIFO retains all data it acknowledged (*that is*, the last data for which READY# was asserted).

After the backoff condition ends, the Local Bus Master restarts the last cycle with ADS#. For writes, data following ADS# should be the data the PCI 9056 did not acknowledge prior to the Backoff cycle (*for example*, the last data for which READY# is not asserted).

If a PCI Read cycle completes when the Local Bus is backed off, the Local Bus Master receives that data if the Local Master restarts the same last cycle (data is not read twice). A new read is performed, if the resumed Local Bus cycle is not the same as the Backed Off cycle.

5.4.3.1.1 Software/Hardware Solution for Systems without Backoff Capability

For adapters that do not support backoff, a possible deadlock solution is as follows.

The PCI Host software can use PCI Host software, external Local Bus hardware, general purpose output USERo and general purpose input USERi to prevent deadlock. USERo can be asserted to request that the external arbiter not grant the bus to any Local Bus Master except the PCI 9056. Status output from the Local arbiter can be connected to the general purpose input USERi to indicate that no Local Bus Master owns the Local Bus, or the PCI Host to determine that no Local Bus Master that currently owns the Local Bus can read input. The PCI Host can then perform Direct Slave access. When the Host finishes, it de-asserts USERo.

5.4.3.1.2 Preempt Solution

For devices that support preempt, USERo can be used to preempt the current Bus Master device. When USERo is asserted, the current Local Bus Master device completes its current cycle and releases the Local Bus, de-asserting LHOLD.

5.4.3.2 Software Solutions to Deadlock

Both PCI Host and Local Bus software can use a combination of mailbox registers, doorbell registers, interrupts, direct Local-to-PCI accesses and direct PCI-to-Local accesses to avoid deadlock.

5.5 DMA OPERATION

The PCI 9056 supports two independent DMA channels capable of transferring data from the:

- Local-to-PCI Bus
- PCI-to-Local Bus

Each channel consists of a DMA controller and a dedicated bidirectional FIFO. Both channels support Block transfers, Scatter/Gather transfers, with or without End of Transfer (EOT#). Master mode must be enabled with the Master Enable bit (PCICR[2]) before the PCI 9056 can become a PCI Bus Master. In addition, both DMA channels can be programmed to:

- Operate in 8-, 16-, or 32-bit Local Bus width
- Use zero to 15 internal wait states (Local Bus)

- Enable/disable internal wait states (Local Bus)
- Enable/disable Local Bus Burst capability
- Limit Local Bus bursts to four (BTERM# enable/ disable)
- Hold Local address constant (Local Slave is FIFO) or increment
- Perform PCI Memory Write and Invalidate (command code = Fh) or normal PCI Memory Write (command code = 7h)
- Pause Local transfer with/without BLAST# (DMA Fast/Slow termination)
- Assert PCI interrupt (INTA#) or Local interrupt (LINTo#) when DMA transfer is complete or Terminal Count is reached during Scatter/Gather DMA mode transfers
- Operate in DMA Clear Count mode (only if the descriptor is in Local memory)

The PCI 9056 also supports PCI Dual Address with the upper 32-bit register(s) (DMADAC0 and/or DMADAC1).

The Local Bus Latency Timer determines the number of Local clocks the PCI 9056 can burst data before relinquishing the Local Bus. The Local Bus Pause Timer sets how soon the DMA channel can request the Local Bus.

5.5.1 DMA PCI Dual Address Cycle

The PCI 9056 supports PCI Dual Address Cycles (DAC) when it is a PCI Bus Master, using the DMADAC0 and/or DMADAC1 register(s) for Block DMA transactions. Scatter/Gather DMA can utilize the DAC function by way of the DMADAC0 and/or DMADAC1 register(s) or DMAMODE0[18] and/or DMAMODE1[18]. The DAC command is used to transfer a 32-bit address to devices that support 32-bit addressing when the address is above the 4-GB Address space. The PCI 9056 performs a DAC within two PCI clock periods, where the first PCI address is a Lo-Addr, with the command (C/BE[3:0]#) “D”, and the second PCI address is a Hi-Addr, with the command (C/BE[3:0]#) “6” or “7”, depending upon whether it is a PCI Read or PCI Write cycle.

5.5.2 Block DMA Mode

The Host processor or the Local processor sets the Local and PCI starting addresses, transfer byte count, and transfer direction. The Host or Local processor then sets the DMA Start bit (DMACSR0[1] and/or DMACSR1[1]) to initiate a transfer. The PCI 9056 requests the PCI and Local Buses and transfers data. Once the transfer completes, the PCI 9056 sets the Channel Done bit(s) (DMACSR0[4]=1 and/or DMACSR1[4]=1) and, if enabled, asserts an interrupt(s) (DMAMODE0[10] and/or DMAMODE1[10]) to the Local processor or the PCI Host (programmable). The Channel Done bit(s) can be polled, instead of interrupt generation, to indicate the DMA transfer status.

DMA registers are accessible from the PCI and Local Buses. (Refer to Figure 5-11.)

During DMA transfers, the PCI 9056 is a Master on both the PCI and Local Buses. For simultaneous access, Direct Slave or Direct Master has a higher priority than DMA.

The PCI 9056 releases the PCI Bus, if one of the following conditions occur. (Refer to Figure 5-12 and Figure 5-13):

- FIFO is full (PCI-to-Local Bus)
- FIFO is empty (Local-to-PCI Bus)
- Terminal count is reached
- PCI Bus Latency Timer expires (PCILTR[7:0])—normally programmed by the Host PCI BIOS—and PCI GNT# de-asserts
- PCI Host asserts STOP#

The PCI 9056 releases the Local Bus, if one of the following conditions occurs:

- FIFO is empty (PCI-to-Local Bus)
- FIFO is full (Local-to-PCI Bus)
- Terminal count is reached
- Local Bus Latency Timer is enabled and expires (MARBR[7:0])
- Special cycle BREQi# is asserted
- Direct Slave request is pending

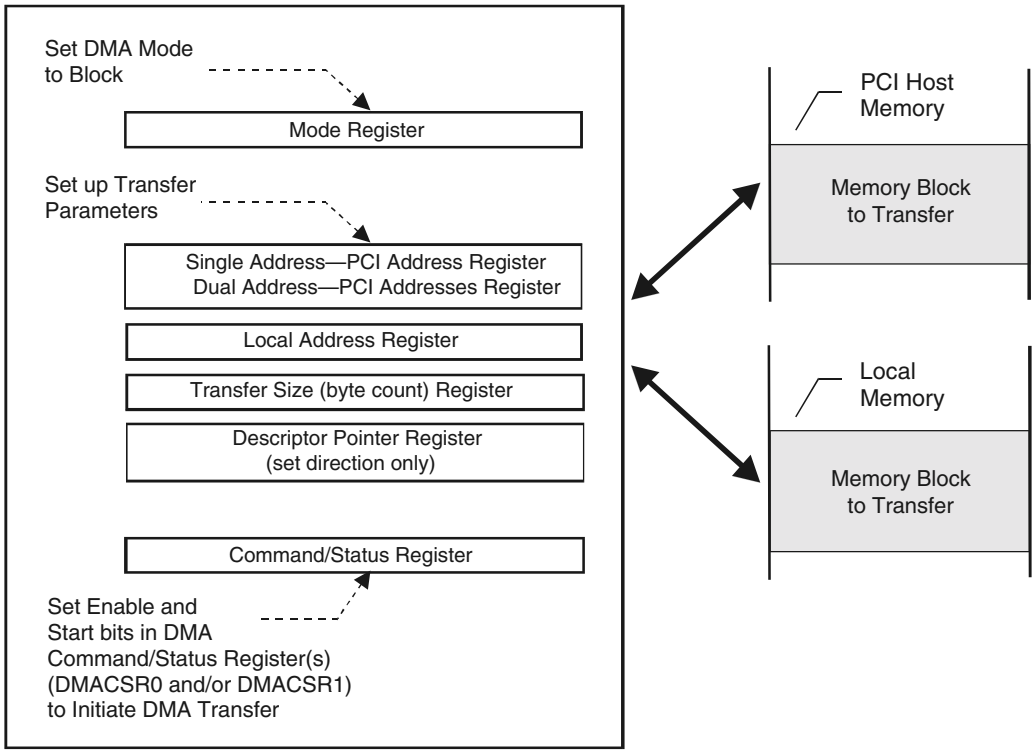


Figure 5-11. Block DMA Mode Initialization (Single Address or Dual Address PCI)

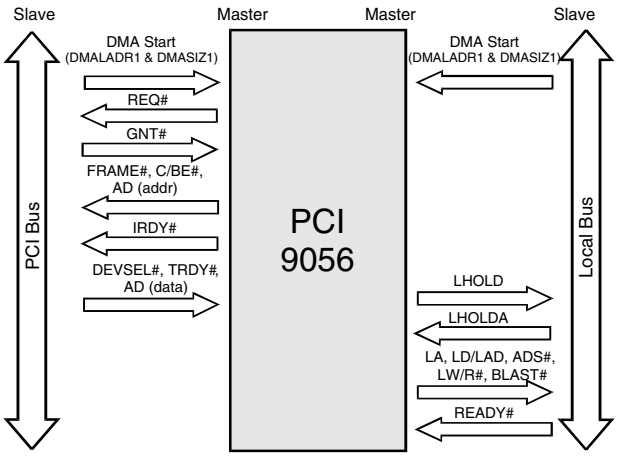


Figure 5-12. DMA, PCI-to-Local Bus

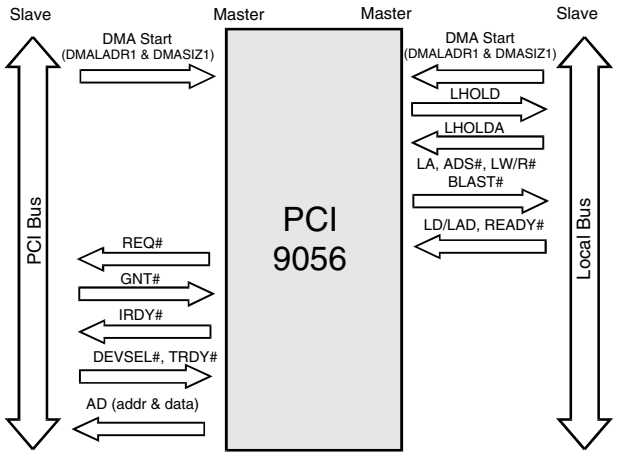


Figure 5-13. DMA, Local-to-PCI Bus

Note: *Figures 5-12 and 5-13 represent a sequence of Bus cycles.*

Table 5-2. DMA Local Burst Mode

Burst Enable Bit	BTERM# Enable Bit	Result
0	X	Single cycle
1	0	Burst up to four Data cycles
1	1	Burst forever (terminate when BTERM# is asserted or transfer is completed)

Note: *"X" is "Don't Care."*

5.5.2.1 Block DMA PCI Dual Address Cycle

The PCI 9056 supports the DAC feature in Block DMA mode. Whenever the DMADAC0 and/or DMADAC1 register(s) contain a value of 0x00000000, the PCI 9056 performs a Single Address Cycle (SAC) on the PCI Bus. Any other value causes a Dual Address to appear on the PCI Bus. (Refer to Figure 5-14.)

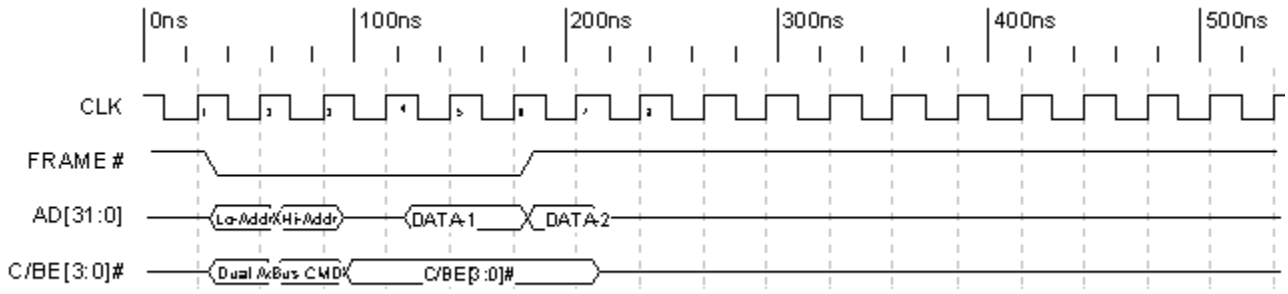


Figure 5-14. Dual Address Timing

5.5.3 Scatter/Gather DMA Mode

In Scatter/Gather DMA mode, the Host processor or Local processor sets up descriptor blocks in Local or Host memory composed of PCI and Local addresses, transfer count, transfer direction, and address of next descriptor block. (Refer to Figure 5-15 and Figure 5-16.) The Host or Local processor then:

- Enables the Scatter/Gather mode bit(s) (DMAMODE0[9]=1 and/or DMAMODE1[9]=1)
- Sets up the address of initial descriptor block in the PCI 9056 Descriptor Pointer register(s) (DMADPR0 and/or DMADPR1)
- Initiates the transfer by setting a control bit(s) (DMACSR0[1:0] and/or DMACSR1[1:0])

The PCI 9056 supports zero wait state Descriptor Block bursts from the Local and PCI Bus when the Local Burst Enable bit(s) is enabled (DMAMODE0[8]=1 and/or DMAMODE1[8]=1).

The PCI 9056 loads the first descriptor block and initiates the Data transfer. The PCI 9056 continues to load descriptor blocks and transfer data until it detects the End of Chain bit(s) is set (DMADPR0[1]=1 and/or DMADPR1[1]=1) (these bits are part of each descriptor). When the End of Chain bit(s) is detected, the PCI 9056 completes the current descriptor block and sets the DMA Done bit(s) (DMACSR0[4] and/or DMACSR1[4]). If the End of Chain bit(s) is detected, the PCI 9056 asserts a PCI interrupt (INTA#) and/or Local interrupt (LINTo#).

The PCI 9056 can also be programmed to assert PCI or Local interrupts after each descriptor is loaded, then finish transferring.

If Scatter/Gather descriptors are in Local memory, the DMA controller can be programmed to clear the transfer size at completion of each DMA, using the DMA Clear Count Mode bit(s) (DMAMODE0[16] and/or DMAMODE1[16]).

Notes: *In Scatter/Gather DMA mode, the descriptor includes the PCI and Local Address Space, transfer size, and next descriptor pointer. It also includes a DAC value, if the DAC Chain Load bit(s) is enabled (DMAMODE0[18]=1 and/or DMAMODE1[18]=1). Otherwise, the register (DMADAC0 and/or DMADAC1) values are used. The Descriptor Pointer register(s) (DMADPR0 and/or DMADPR1) contains end of chain (bit 1), direction of transfer (bit 3), next descriptor address (bits [31:4]), interrupt after terminal count (bit 2), and next descriptor location (bit 0) bits.*

The Local Bus width must be the same as Local Memory Bus width.

A DMA descriptor can be on the Local memory or the PCI memory, or both (for example, one descriptor on Local memory, another descriptor on PCI memory and vice-versa).

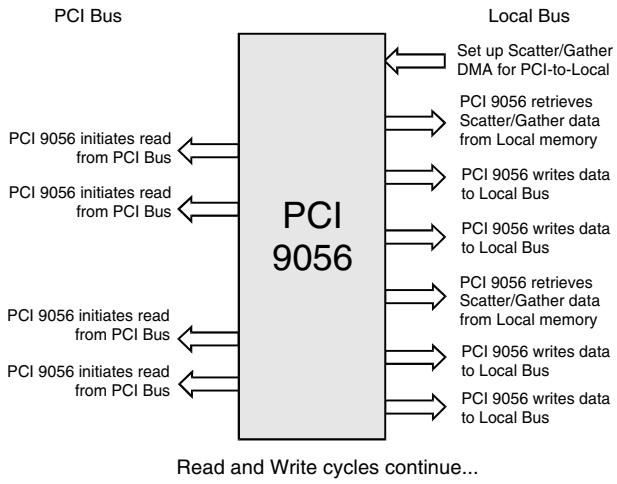


Figure 5-15. Scatter/Gather DMA Mode from PCI-to-Local Bus (Control Access from the Local Bus)

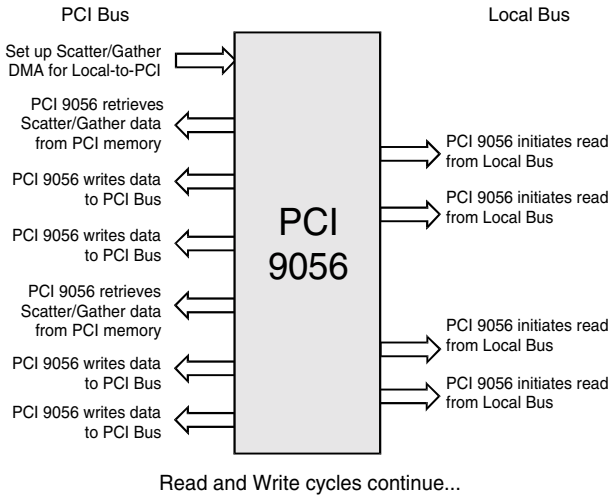


Figure 5-16. Scatter/Gather DMA Mode from Local-to-PCI Bus (Control Access from the PCI Bus)

Note: *Figures 5-15 and 5-16 represent a sequence of Bus cycles.*

5.5.3.1 Scatter/Gather DMA PCI Dual Address Cycle

The PCI 9056 supports the PCI DAC feature in Scatter/Gather DMA mode for Data transfers only. The descriptor blocks should reside below the 4-GB Address space. The PCI 9056 offers three different options of how PCI DAC Scatter/Gather DMA is utilized. Assuming the descriptor blocks are located on the PCI Bus:

- DMADAC0 and/or DMADAC1 contain(s) a non-zero value. DMAMODE0[18] and/or DMAMODE1[18] is set to 0. The PCI 9056 performs a Single Address Cycle (SAC) four-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. (Refer to Figure 5-17.)
- DMADAC0 and/or DMADAC1 contain(s) an 0x00000000 value. DMAMODE0[18] and/or DMAMODE1[18] is set to 1. The PCI 9056 performs a SAC five-Lword descriptor block load from PCI memory and DMA transfer with PCI DAC on the PCI Bus. (Refer to Figure 5-18.)
- DMADAC0 and/or DMADAC1 contain(s) a non-zero value. DMAMODE0[18] and/or DMAMODE1[18] is set to 1. The PCI 9056 performs a SAC five-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. The fifth descriptor overwrites the value of the DMADAC0 and/or DMADAC1 register(s). (Refer to Figure 5-18.)

5.5.3.2 DMA Clear Count Mode

The PCI 9056 supports DMA Clear Count mode (Write-Back feature, DMAMODE0[16] and/or DMAMODE1[16]). This feature allows users to control the Data transfer blocks during Scatter/Gather DMA operations. The PCI 9056 clears the Transfer Size descriptor to zero (0) by writing to a descriptor memory location at the end of each transfer chain. This feature is available for DMA descriptors located on the Local and PCI Buses.

5.5.3.3 DMA Descriptor Ring Management (Valid Mode)

In Scatter/Gather DMA mode, when the Valid Mode Enable bit(s) is set to 0 (DMAMODE0[20]=0 and/or DMAMODE1[20]=0), the Valid bit (bit 31 of transfer count) is ignored. When the Valid Mode Enable bit(s) is set to 1 (DMAMODE0[20]=1 and/or DMAMODE1 [20]=1), the DMA descriptor proceeds only when the Valid bit is set. If the Valid bit is set, the transfer count is 0, and the descriptor is not the last descriptor, then the DMA controller moves on to the next descriptor in the chain.

When the Valid Stop Control bit(s) is set to 0 (DMAMODE0[21]=0 and/or DMAMODE1[21]=0), the DMA Scatter/Gather controller continuously polls the descriptor with the Valid bit set to 0 (invalid descriptor) until the Valid bit is read to be a 1. When the Valid Stop Control bit(s) is set to 1 (DMAMODE0[21]=1 and/or DMAMODE1[21]=1), the DMA Scatter/Gather controller pauses if a Valid bit with a value of 0 is detected. In this case, the PCI 9056 must restart the DMA controller by setting bit 1 of the DMA Control/Status register(s) (DMACSR0[1] and/or DMACSR1[1]). The DMA Clear Count mode bit(s) (DMAMODE0[16] and/or DMAMODE1[16]) must be enabled for the Ring Management Valid bit to be cleared at the completion of each descriptor.

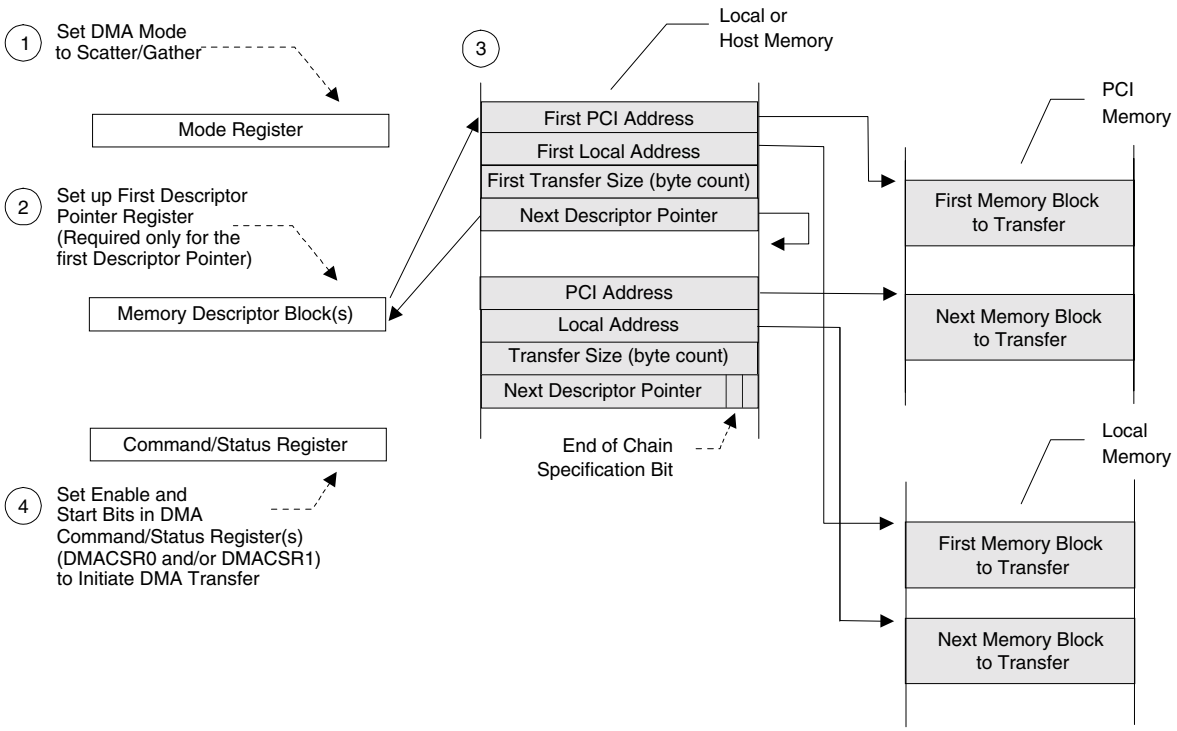


Figure 5-17. Scatter/Gather DMA Mode Descriptor Initialization [PCI SAC/DAC PCI Address (DMADAC0 and/or DMADAC1) Register Dependent]

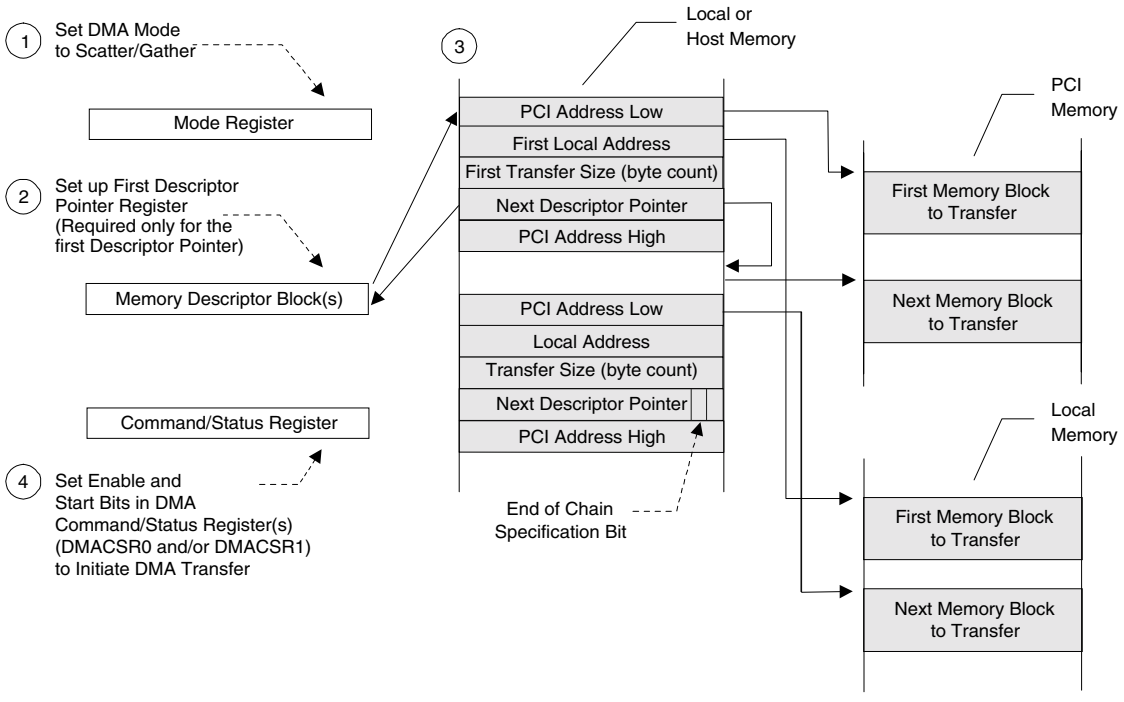


Figure 5-18. Scatter/Gather DMA Mode Descriptor Initialization [DAC PCI Address (DMAMODE0[18] and/or DMAMODE1[18]) Descriptor Dependent] (PCI Address High Added)

5.5.4 DMA Memory Write and Invalidate

The PCI 9056 can be programmed to perform Memory Write and Invalidate cycles to the PCI Bus for DMA transfers, as well as Direct Master transfers. (Refer to Section 5.4.1.11.) The PCI 9056 supports Memory Write and Invalidate transfers for cache line sizes of 8 or 16 Lwords. Size is specified in the System Cache Line Size bits (PCICLSR[7:0]). If a size other than 8 or 16 is specified, the PCI 9056 performs Write transfers rather than Memory Write and Invalidate transfers.

DMA Memory Write and Invalidate transfers are enabled when the DMA controller Memory Write and Invalidate Enable bit(s) (DMAMODE0[13] and/or DMAMODE1[13]) and the Memory Write and Invalidate Enable bit (PCICR[4]) are set.

In Memory Write and Invalidate mode, the PCI 9056 waits until the number of Lwords required for specified cache line size are read from the Local Bus before starting the PCI access. This ensures a complete cache line write can complete in one PCI Bus ownership. If a Target disconnects before a cache line completes, the PCI 9056 completes the remainder of that cache line, using normal writes before resuming Memory Write and Invalidate transfers. If a Memory Write and Invalidate cycle is in progress, the PCI 9056 continues to burst if another cache line is read from the Local Bus before the cycle completes. Otherwise, the PCI 9056 terminates the burst and waits for the next cache line to be read from the Local Bus. If the final transfer is not a complete cache line, the PCI 9056 completes the DMA transfer, using normal writes.

EOT# signal assertion, in any DMA transfer type, or DREQ0# and/or DREQ1# signal de-assertion in Demand Mode before the cache line is read from the Local Bus, results in the PCI 9056 performing a normal PCI Memory Write to data read into a DMA FIFO.

5.5.4.1 DMA Abort

DMA transfers can be aborted, in addition to the EOT# signal, as follows:

1. Clear the DMA Channel Enable bit(s) (DMACSR0[0]=0 and/or DMACSR1[0]=0).
2. Abort DMA by setting the Channel Abort bit(s) (DMACSR0[2]=1 and/or DMACSR1[2]=1).
3. Wait until the Channel Done bit(s) is set (DMACSR0[4]=1 and/or DMACSR1[4]=1).

Note: *One to two Data transfers occur after the Abort bit is set. Aborting when no DMA cycles are in progress causes the next DMA to abort.*

5.5.5 DMA Priority

The DMA Channel Priority bits (MARBR[20:19]) can be used to specify the following priorities:

- Rotating (MARBR[20:19]=00)
- DMA Channel 0 (MARBR[20:19]=01)
- DMA Channel 1 (MARBR[20:19]=10)

5.5.6 DMA Channel 0 and Channel 1 Interrupts

A DMA channel can assert a PCI Bus or Local Bus interrupt when done (transfer complete) or after a transfer is complete for the current descriptor in Scatter/Gather DMA mode. The DMA Channel Interrupt Select bit(s) determine whether to assert a PCI (DMAMODE0[17]=1 and/or DMAMODE1[17]=1) or Local (DMAMODE0[17]=0 and/or DMAMODE1[17]=0) interrupt. The PCI or Local processor can read the DMA Channel 0 Interrupt Active bits to determine whether a DMA Channel 0 (INTCSR[21]) or DMA Channel 1 (INTCSR[22]) interrupt is pending.

The Channel Done bit(s) (DMACSR0[4] and/or DMACSR1[4]) can be used to determine whether an interrupt is:

- DMA Done interrupt
- Transfer complete for current descriptor interrupt

The Done Interrupt Enable bit(s) (DMAMODE0[10] and/or DMAMODE1[10]) enable a Done interrupt. In Scatter/Gather DMA mode, a bit in the Next Descriptor Pointer register of the channel (loaded from Local memory) specifies whether to assert an interrupt at the end of the transfer for the current descriptor.

A DMA Channel interrupt is cleared by the Channel Clear Interrupt bit(s) (DMACSR0[3]=1 and/or DMACSR1[3]=1).

5.5.7 DMA Data Transfers

The PCI 9056 DMA controller can be programmed to transfer data from the Local-to-PCI Bus or from the PCI-to-Local Bus.

5.5.7.1 Local-to-PCI Bus DMA Transfer

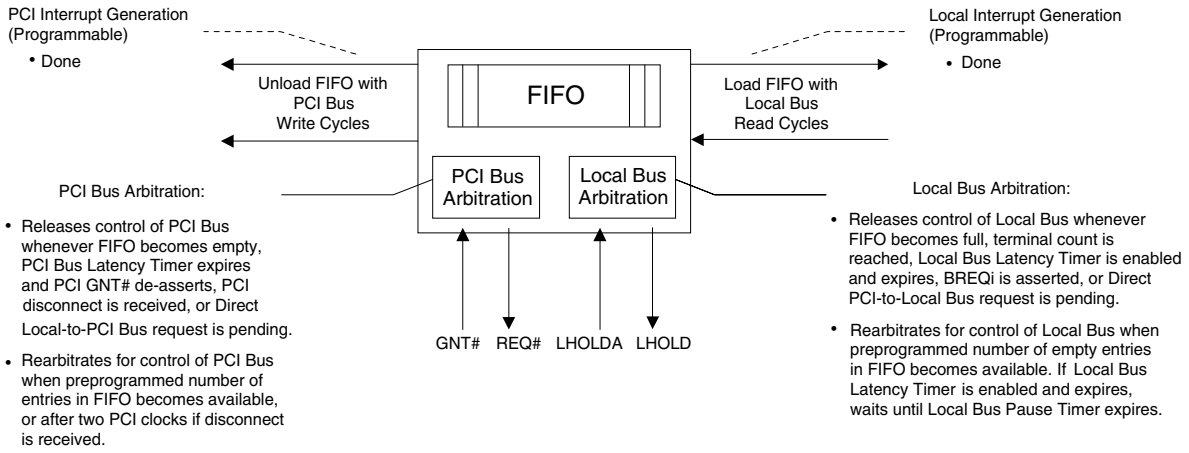


Figure 5-19. Local-to-PCI Bus DMA Data Transfer Operation

5.5.7.2 PCI-to-Local Bus DMA Transfer

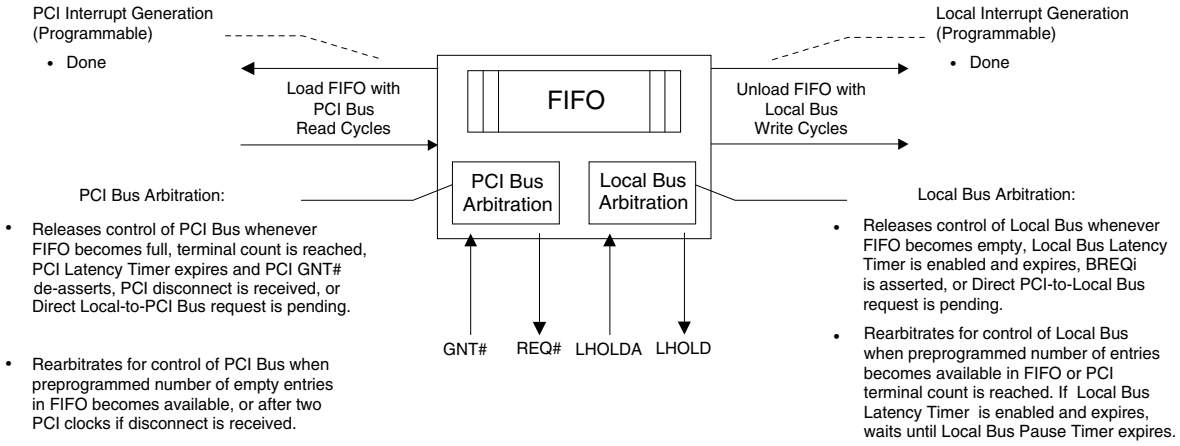


Figure 5-20. PCI-to-Local Bus DMA Data Transfer Operation

5.5.7.3 DMA Unaligned Transfers

For unaligned Local-to-PCI transfers, the PCI 9056 reads a partial Lword from the Local Bus. It continues to read Lwords from the Local Bus. Lwords are assembled, aligned to the PCI Bus address, and loaded into the FIFO.

For PCI-to-Local transfers, Lwords are read from the PCI Bus and loaded into the FIFO. On the Local Bus, Lwords are assembled from the FIFO, aligned to the Local Bus address and written to the Local Bus.

On both the PCI and Local Buses, the byte enables for writes determine LA[1:0] for the start of a transfer. For the last transfer, byte enables specify the bytes to be written. All reads are Lwords.

5.5.8 Demand Mode DMA, Channel 0 and Channel 1

The Fast/Slow Terminate Mode Select bit(s) (DMAMODE0[15] and/or DMAMODE1[15]) determines the number of Lwords to transfer after the DMA controller DREQ0# and/or DREQ1# input is de-asserted.

If BLAST# output is not required for the last Lword of a DMA transfer (bit [15]=1), the DMA controller releases the data bus after it receives an external READY# or the internal wait state counter decrements to 0 for the current Lword. If the DMA controller is currently bursting data, which is not the last Data phase for the Burst, BLAST# is not asserted.

When the PCI 9056 is in Demand Mode DMA Local-to-PCI Slow Terminate mode (DMAMODE0[15] and/or DMAMODE1[15]), it monitors unaligned DMA transfers PCI address increments to guarantee a Qword PCI data, 32-bit data completion when DREQ0# and/or DREQ1# is de-asserted in the middle of the Data-Pocket transfer, Demand Mode DMA pause. Due to the nature of unaligned transfers, the PCI 9056 retains partial Lword data, three or fewer bytes remain in the DMA FIFO and are not transferred when DREQ0# and/or DREQ1# is de-asserted in the middle of the Data-Pocket transfer. When DREQ0# and/or DREQ1# resumes, the data is transferred to the PCI Bus. If DREQ0# and/or DREQ1# assertion is never resumed for ongoing transfers, the EOT# signal assertion (along with DREQ0# and/or DREQ1# de-assertion) should be used to ensure the partial data successfully transfers to the PCI Bus.

These same conditions for DMA PCI-to-Local cause the PCI 9056 to pause the DMA transfer on the Local Bus at the Lword boundary with BLAST# asserted at the last Data transfer. EOT# assertion (along with DREQ0# and/or DREQ1# de-assertion) causes the PCI 9056 to terminate the ongoing Data transfer and flush the DMA FIFO with BLAST# asserted at the last Data transfer.

If BLAST# output is required for the last Lword of the DMA transfer (bit [15]=0), the DMA controller transfers one or two Lwords. If DREQ0# and/or DREQ1# is de-asserted during the Address phase of the first transfer in the PCI 9056 Local Bus ownership (ADS#, LHOLDA asserted), the DMA controller completes current Lword. If DREQ0# and/or DREQ1# is de-asserted during any phase other than the Address phase of the first transfer in the PCI 9056 Local Bus ownership, the DMA controller completes the current Lword, and one additional Lword (this allows BLAST# output to be asserted during the final Lword). If the DMA FIFO is full or empty after the Data phase in which DREQ0# and/or DREQ1# is de-asserted, the second Lword is not transferred.

DREQ0# and/or DREQ1# controls only the number of Lword transfers. For an 8-bit bus, the PCI 9056 releases the bus after transferring the last byte for the Lword. For a 16-bit bus, the PCI 9056 releases the bus after transferring the last word for the Lword. (Refer to the timing diagrams in Section 5.6.)

When the PCI 9056 is in Demand Mode DMA Local-to-PCI Fast Terminate mode (DMAMODE0[15] and/or DMAMODE1[15]) unaligned DMA transfers, it monitors PCI address increments to guarantee a Qword PCI data, 32-bit data completion when DREQ0# and/or DREQ1# is de-asserted in the middle of the Data-Pocket transfer, Demand Mode DMA pause. Due to the nature of unaligned transfers, the PCI 9056 retains partial Lword data, three or fewer bytes remain in the DMA FIFO and are not transferred when DREQ0# and/or DREQ1# is de-asserted in the middle of the Data-Pocket transfer. When DREQ0# and/or DREQ1# resumes, the data is transferred to the PCI Bus. If DREQ0# and/or DREQ1# assertion is never resumed for ongoing transfers, the EOT# signal assertion (along with DREQ0# and/or DREQ1# de-assertion) should be used to ensure the partial data successfully transfers to the PCI Bus.

These same conditions for DMA PCI-to-Local cause the PCI 9056 to immediately pause the DMA transfer on the Local Bus at Lword boundary without BLAST# being asserted. EOT# assertion (along with DREQ0# and/or DREQ1# de-assertion) causes the PCI 9056 to immediately terminate the ongoing Data transfer and flush the DMA FIFO without BLAST# being asserted.

5.5.9 End of Transfer (EOT#) Input

The DMA EOT# Enable bit(s) (DMAMODE0[14] and/or DMAMODE1[14]) determines the number of Lwords to transfer after a DMA controller asserts EOT# input. EOT# input should be asserted only when the PCI 9056 owns a bus.

If BLAST# output is not required for the last Lword of the DMA transfer (DMAMODE0[15]=1 and/or DMAMODE1[15]=1), the DMA controller releases the data bus and terminates DMA after it receives an external READY#. Or, the internal wait state counter decrements to 0 for the current Lword. If the DMA controller is currently bursting data that is not the last Data phase for the burst, BLAST# output is not asserted.

If BLAST# output is required for last Lword of the DMA transfer (DMAMODE0[15]=0 and/or DMAMODE1 [15]=0), the DMA controller transfers one or two Lwords, depending on the Local Bus width. If EOT# is asserted, the DMA controller completes the current Lword and one additional Lword (this allows BLAST# output to be asserted during the final Lword). If the DMA FIFO is full or empty after the Data phase in which EOT# is asserted, the second Lword is not transferred.

The DMA controller terminates a transfer on an Lword boundary after EOT# is asserted. For an 8-bit bus, the PCI 9056 terminates after transferring the last byte for the Lword. For a 16-bit bus, the PCI 9056 terminates after transferring the last word for the Lword.

During the descriptor loading on the Local Bus, assertion of EOT# causes a complete descriptor load and no subsequent Data transfer; however, this is not recommended. This has no effect when the descriptor is loaded from the PCI Bus.

5.5.10 DMA Arbitration

The PCI 9056 DMA controller releases control of the Local Bus (de-asserts LHOLD) when one of the following conditions occur:

- Local Bus Latency Timer is enabled and expires (MARBR[7:0])
- BREQi is asserted (BREQi can be enabled or disabled, or gated with a Local Bus Latency Timer before the PCI 9056 releases the Local Bus)
- Direct Slave access is pending
- EOT# input is received (if enabled)

The DMA controller releases control of the PCI Bus when one of the following conditions occurs:

- FIFOs are full or empty
- PCI Bus Latency Timer expires (PCILTR[7:0])—and loses the PCI GNT# signal
- Target disconnect response is received

The DMA controller de-asserts PCI REQ# for a minimum of two PCI clocks.

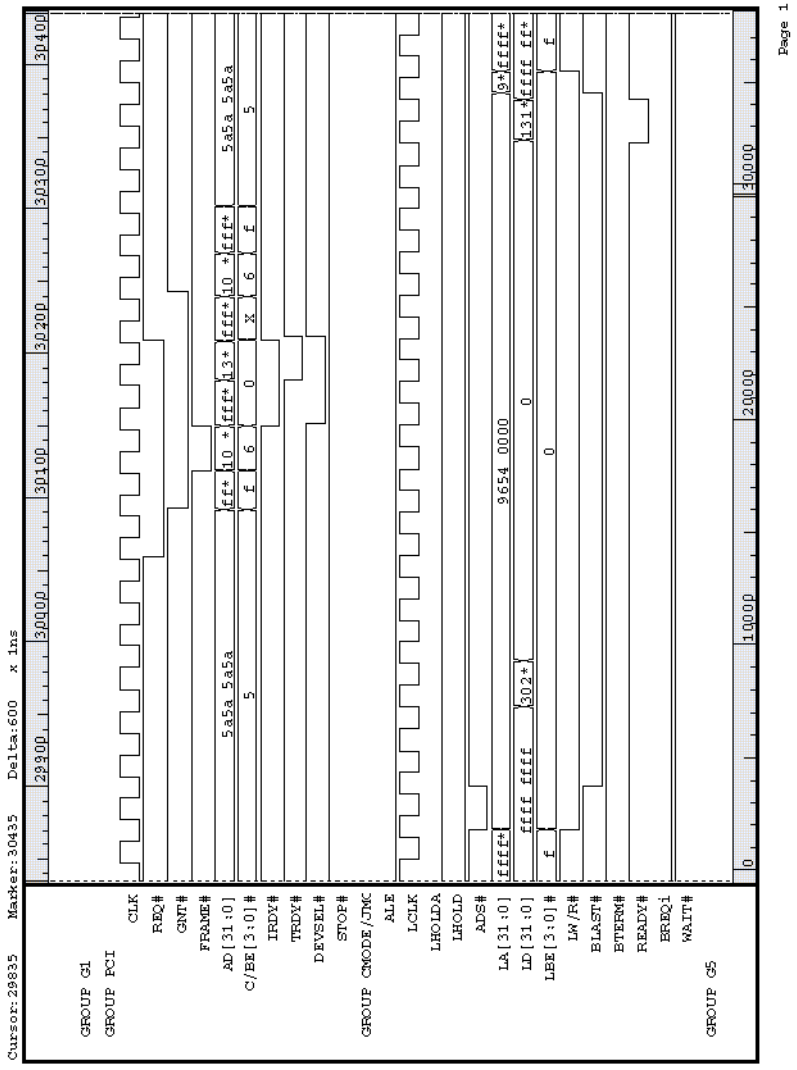
5.5.11 Local Bus Latency and Pause Timers

The Local Bus Latency and Pause Timers are programmable with the Mode/DMA Arbitration register (MARBR[7:0, 15:8], respectively). If the Local Bus Latency Timer is enabled and expires, the PCI 9056 completes the current Lword transfer and releases LHOLD. After its programmable Pause Timer expires, it reasserts LHOLD. It continues to transfer when it receives LHOLDA. The PCI Bus transfer continues until the FIFO is empty for a Local-to-PCI transfer or full for a PCI-to-Local transfer.

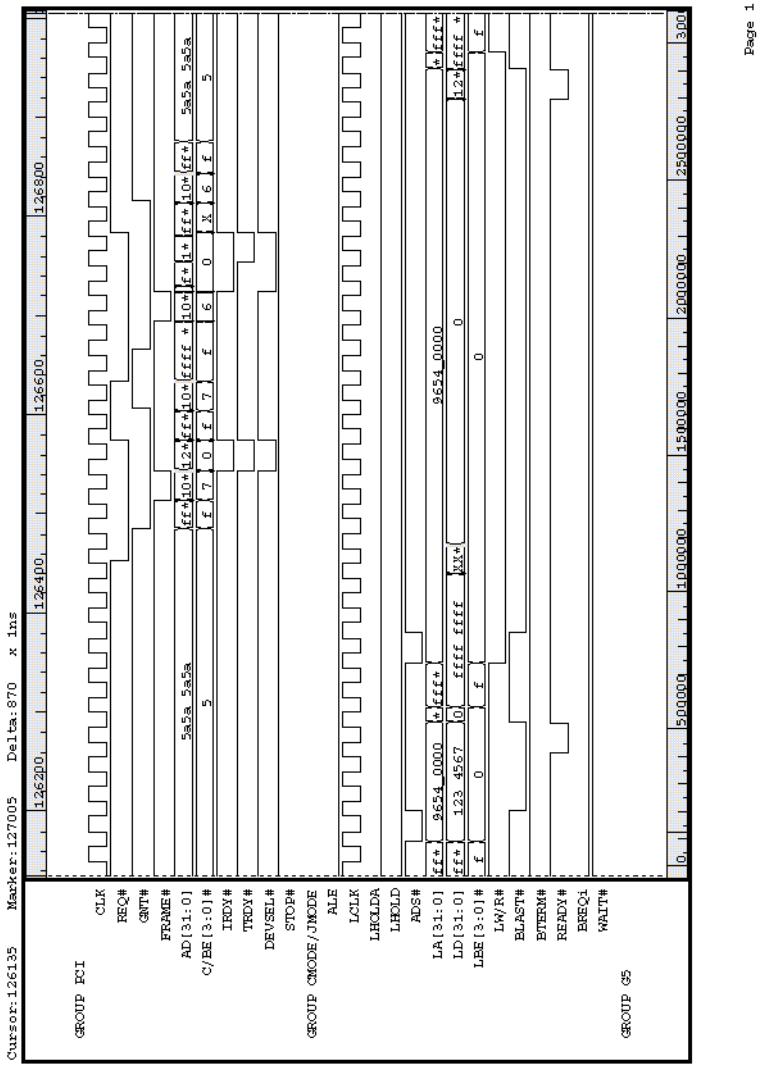
The DMA transfer can be paused by writing a 0 to the Channel Enable bit. To acknowledge the disable, the PCI 9056 gets at least one data from the bus before it stops. However, this is not recommended during a burst.

The DMA Local Bus Timer starts after the Local Bus is granted to the PCI 9056 and the Local Bus Pause Timer starts after LHOLDA is de-asserted.

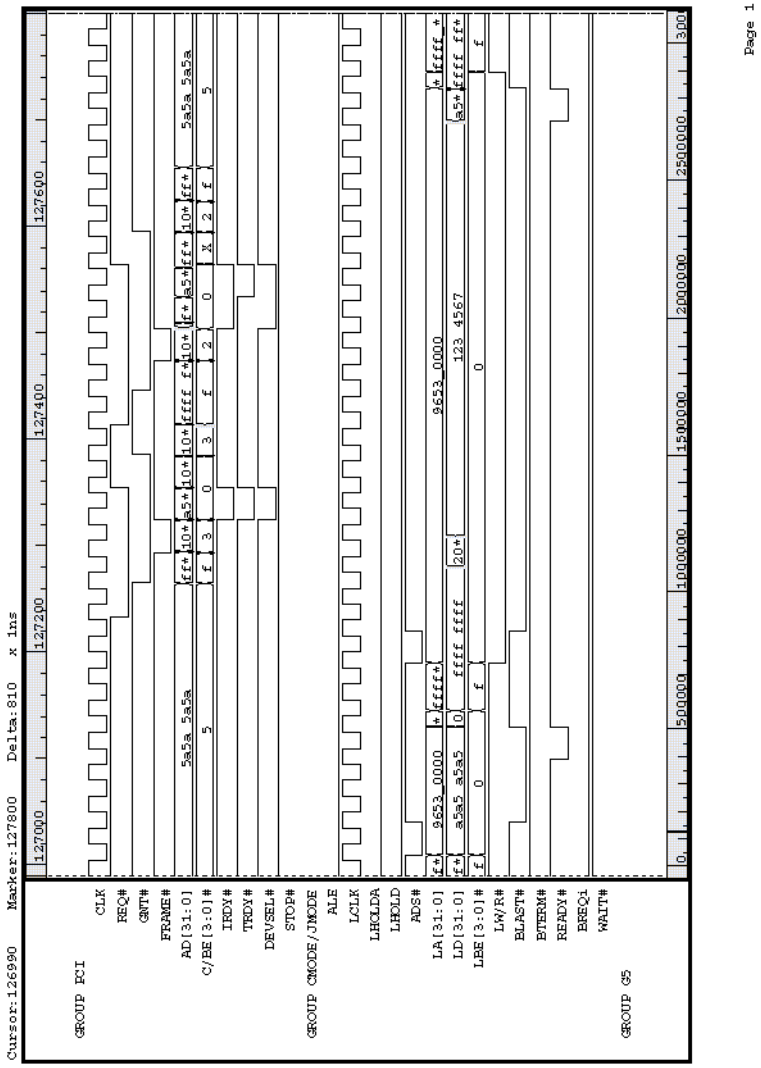
5.6 C AND J MODES TIMING DIAGRAMS



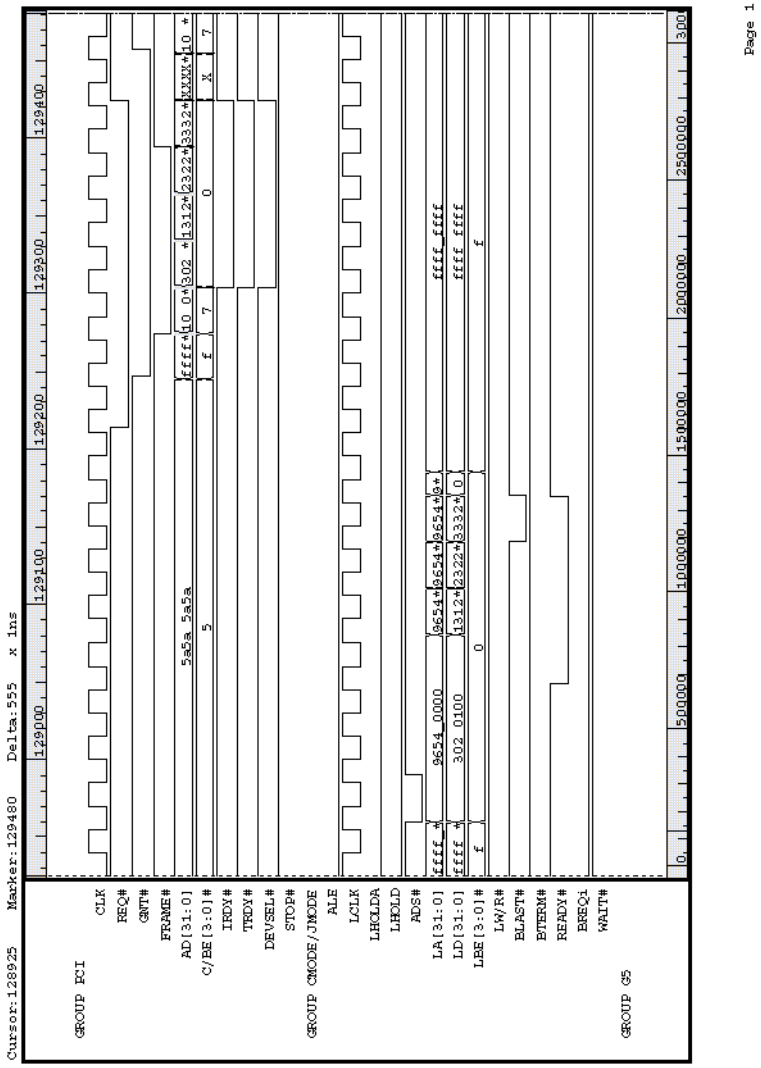
Timing Diagram 5-1. Direct Master Single Read



Timing Diagram 5-2. Direct Master Single Write and Single Read to PCI Memory and I/O Space

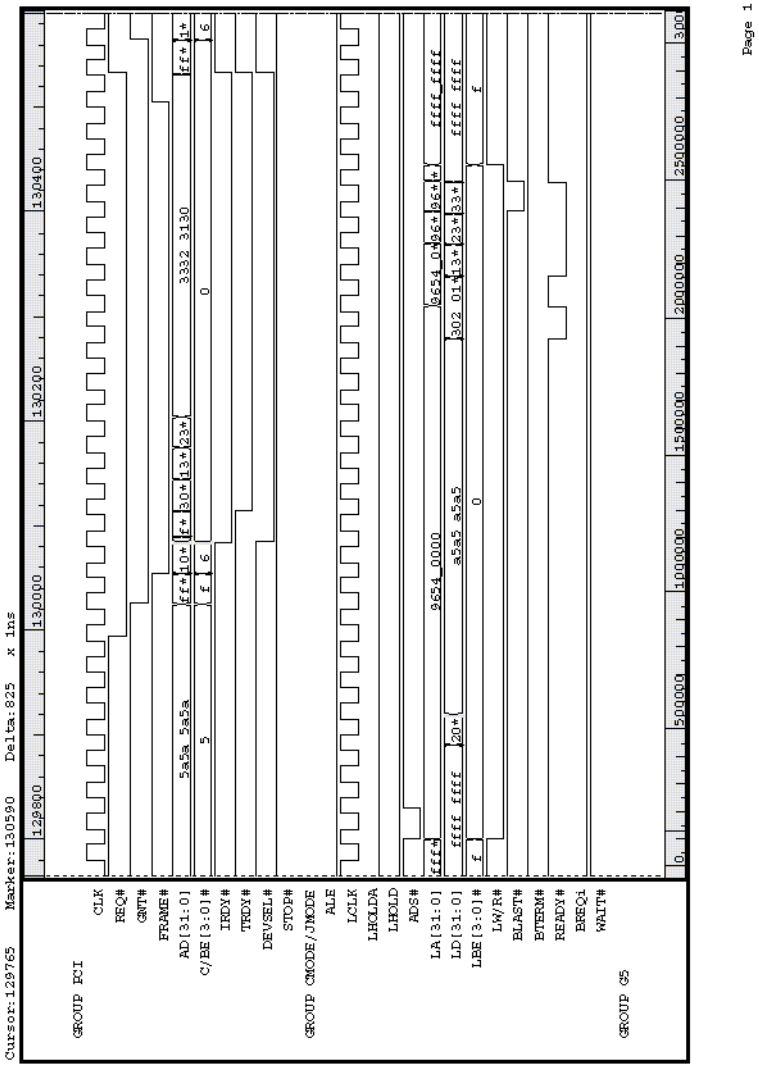


Timing Diagram 5-3. Direct Master Single Write and Single Read to and from PCI I/O Space



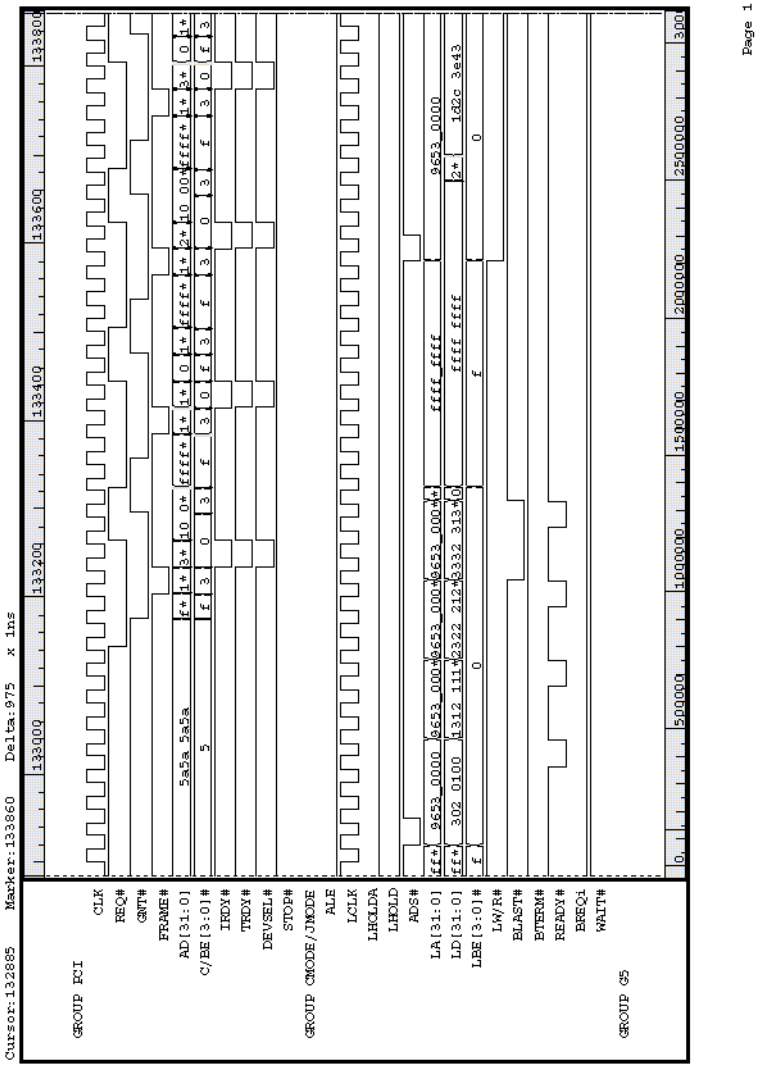
Timing Diagram 5-4. Direct Master Burst Write to PCI Memory Space

Timing Diagram 5-5. TD-092.tif

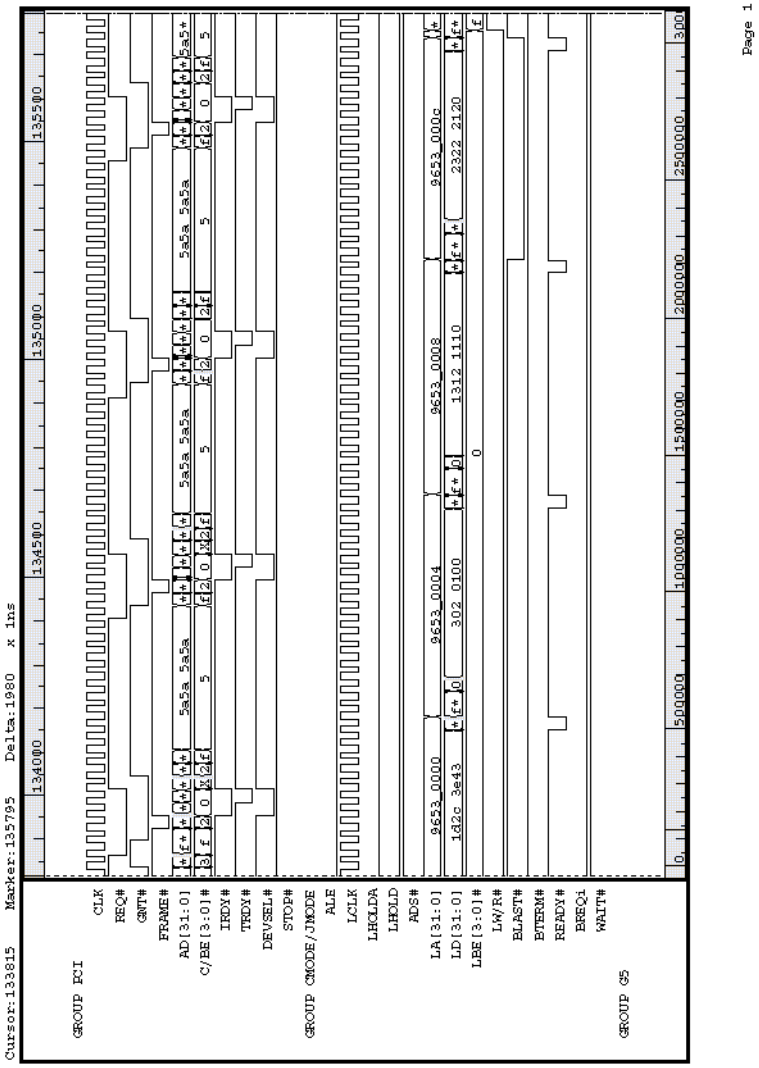


Timing Diagram 5-6. Direct Master Burst Read from PCI Memory Space

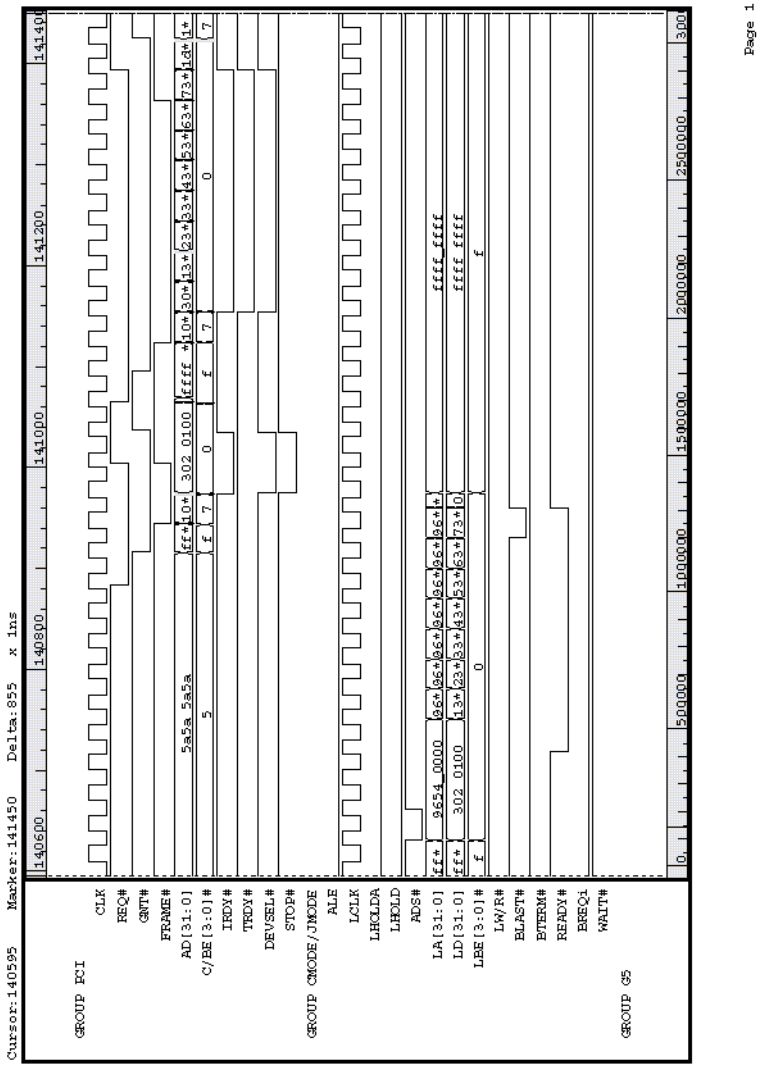
Timing Diagram 5-7. TD-096.tif



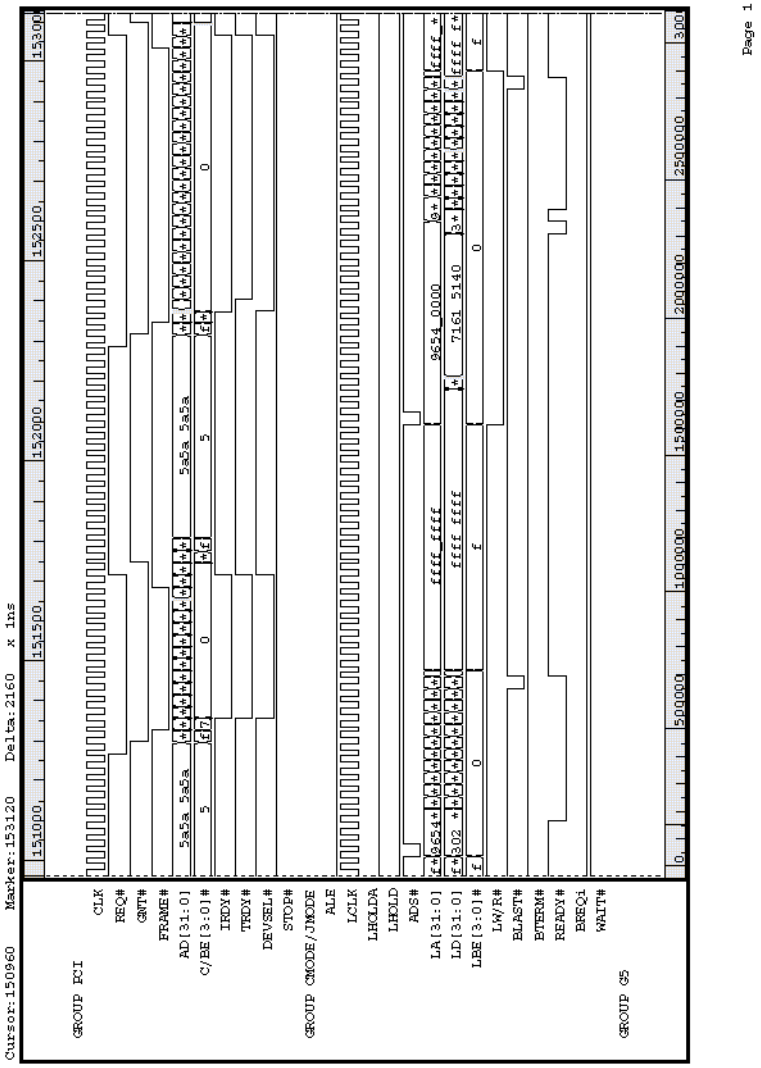
Timing Diagram 5-8. Direct Master Burst Write to PCI I/O Space



Timing Diagram 5-9. Direct Master Burst Read from PCI I/O Space

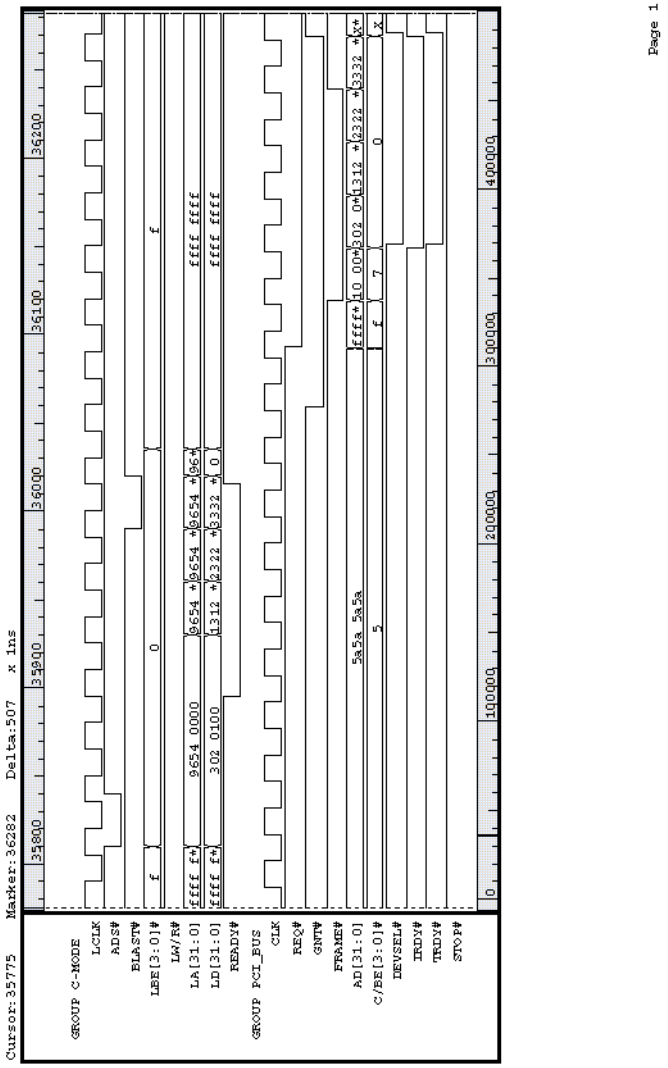


Timing Diagram 5-10. Direct Master Burst Write with a Retry on PCI Bus

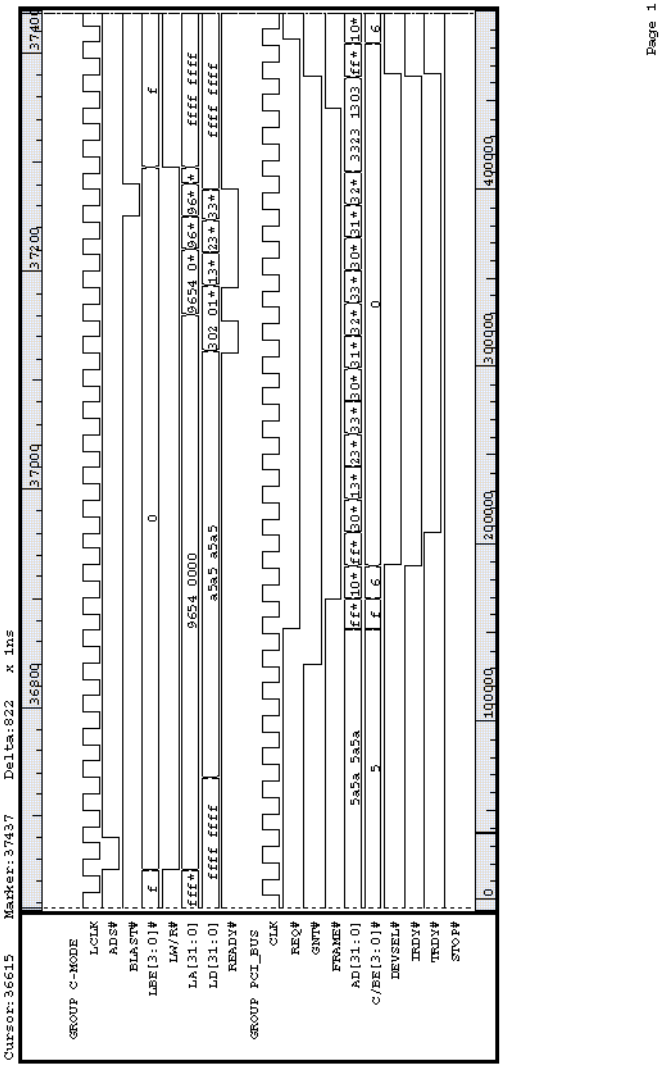


Timing Diagram 5-11. Direct Master Burst Write followed by Direct Master Burst Read

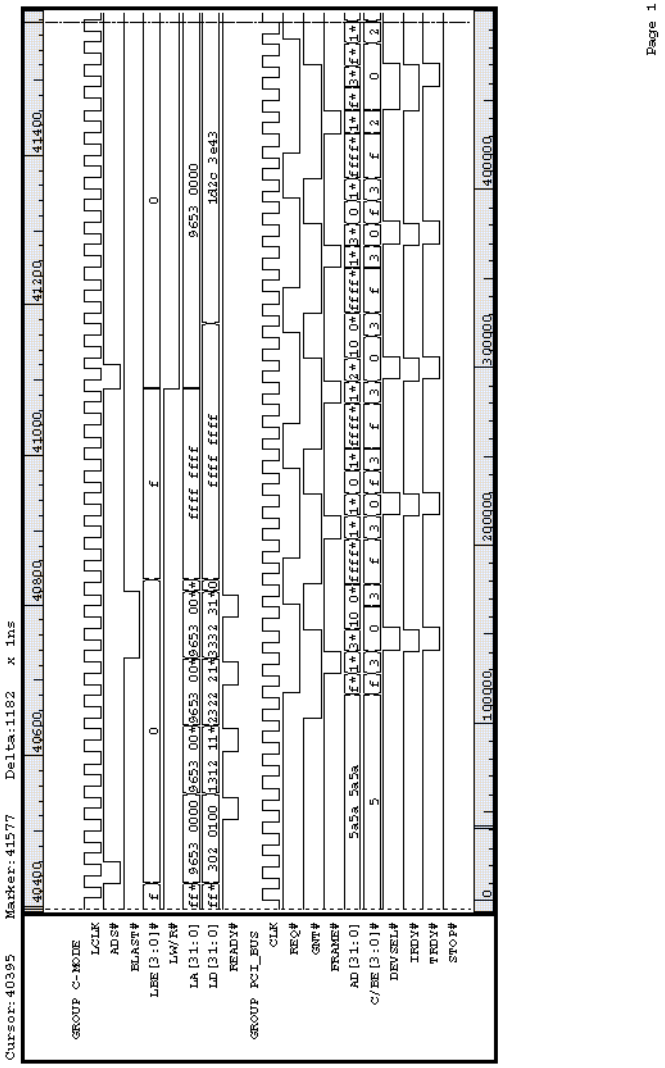
5.6.1 C Mode Only Direct Master Timing Diagrams



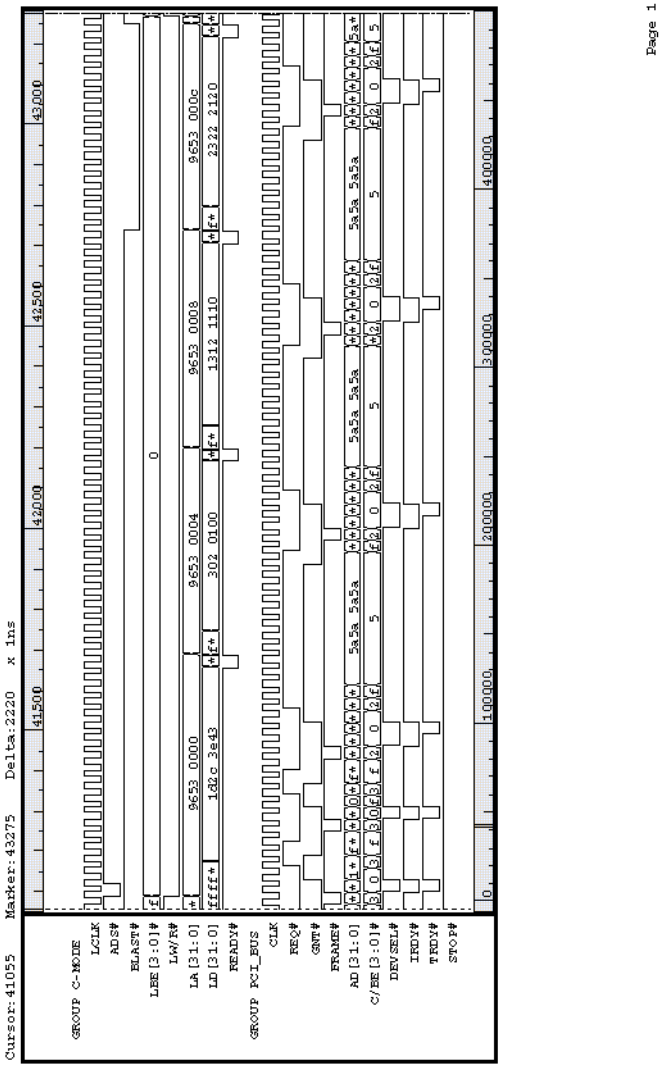
Timing Diagram 5-12. Direct Master Memory Write of Four Lwords



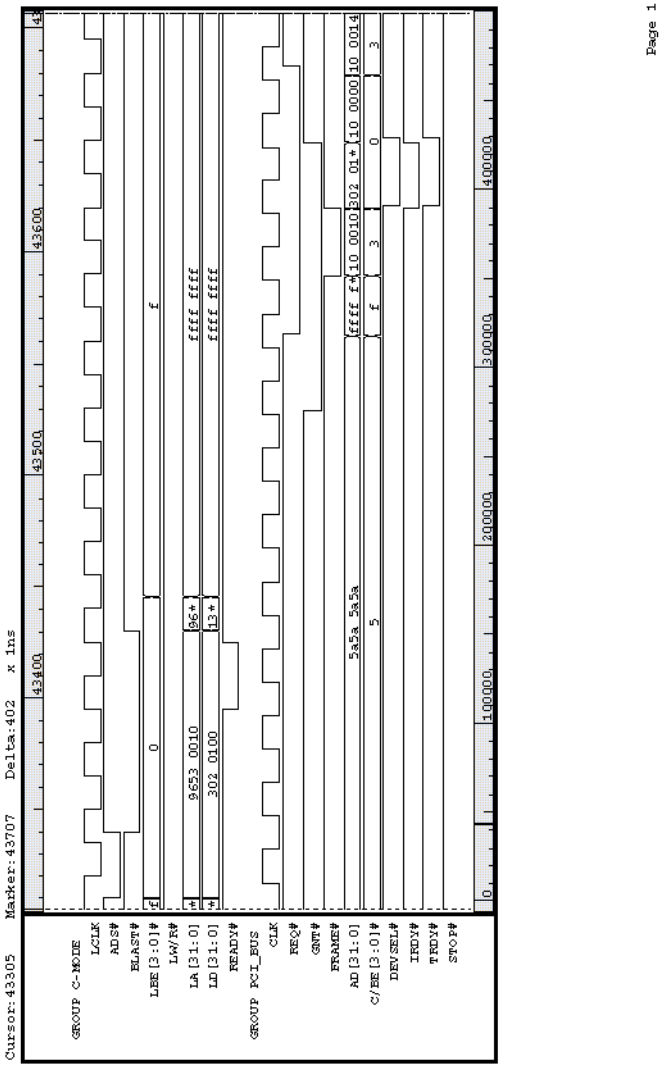
Timing Diagram 5-13. Direct Master Memory Read of Four Lwords



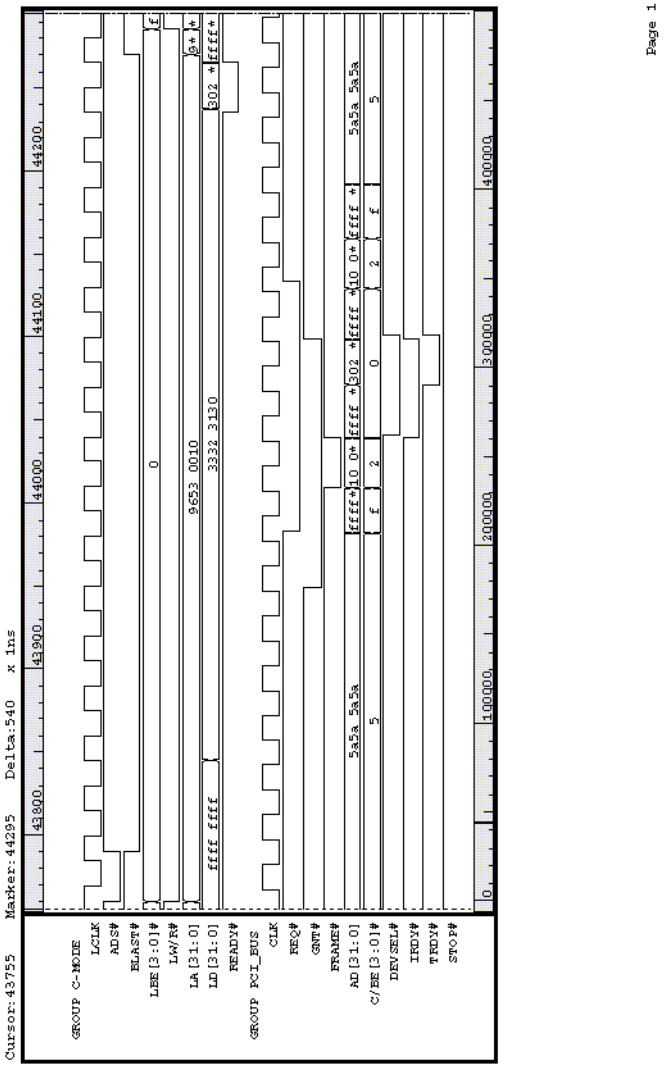
Timing Diagram 5-14. Direct Master I/O Write of Four Lwords



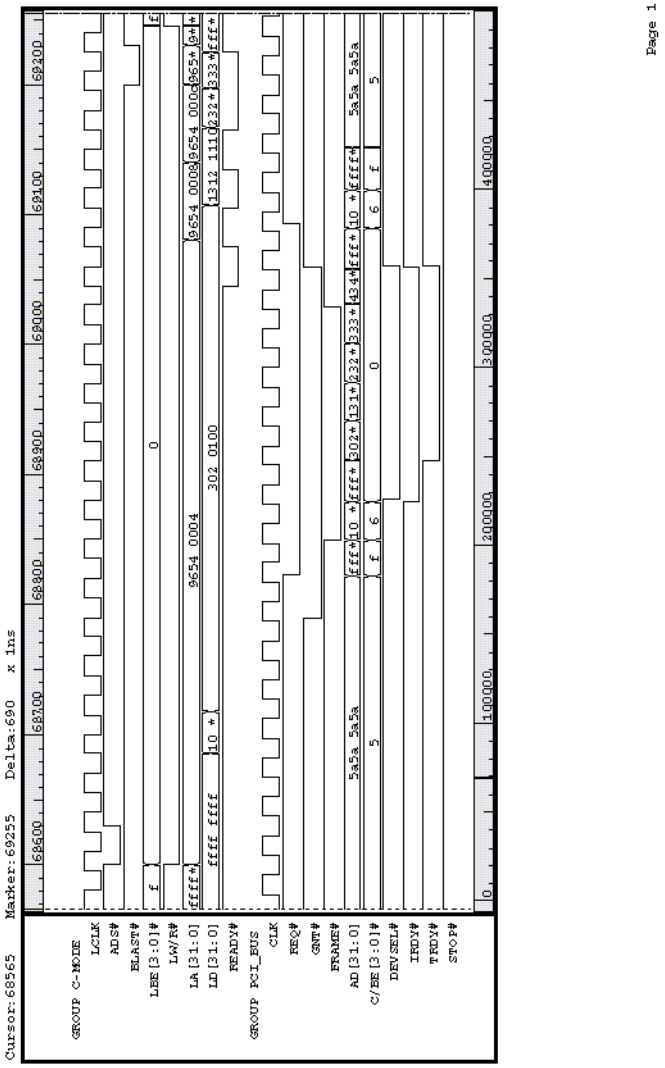
Timing Diagram 5-15. Direct Master I/O Read of Four Lwords



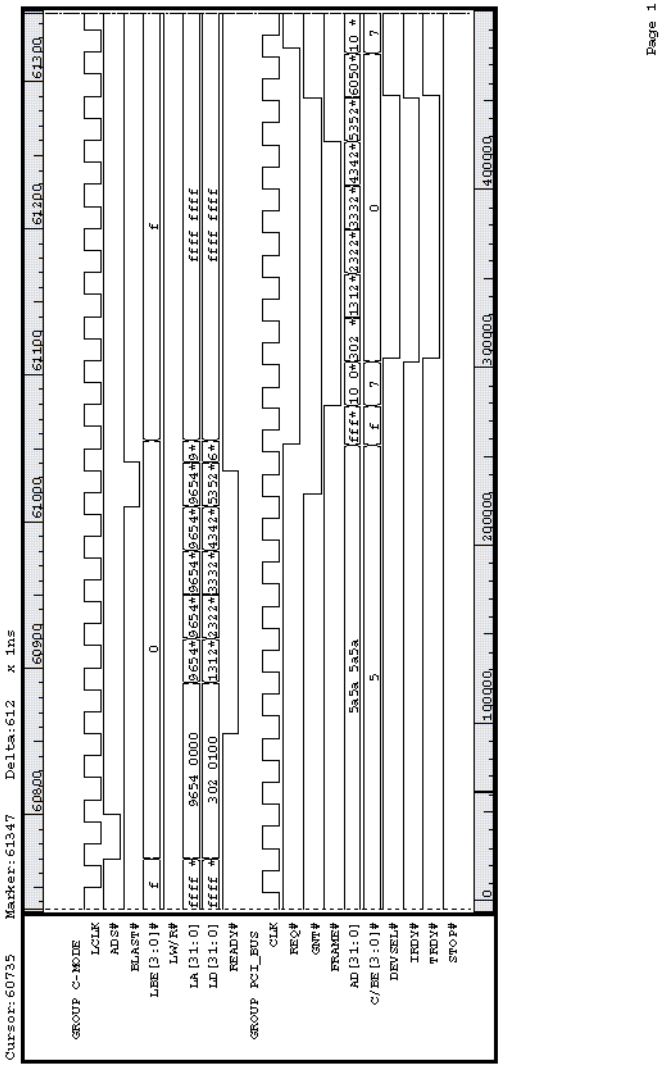
Timing Diagram 5-16. Direct Master Single I/O Write



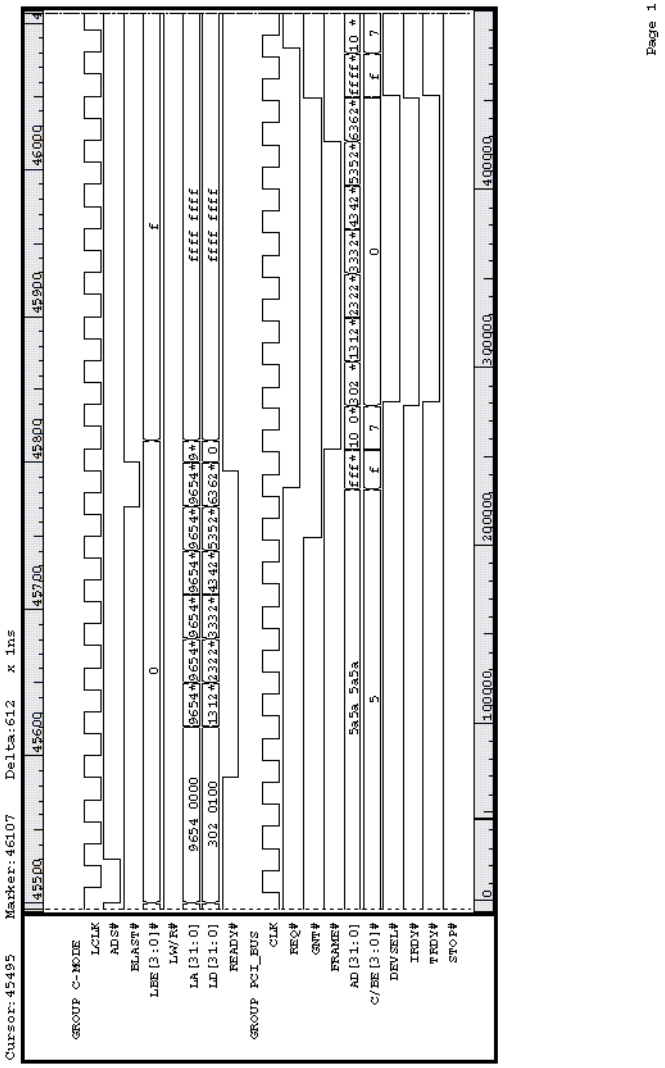
Timing Diagram 5-17. Direct Master Single I/O Read



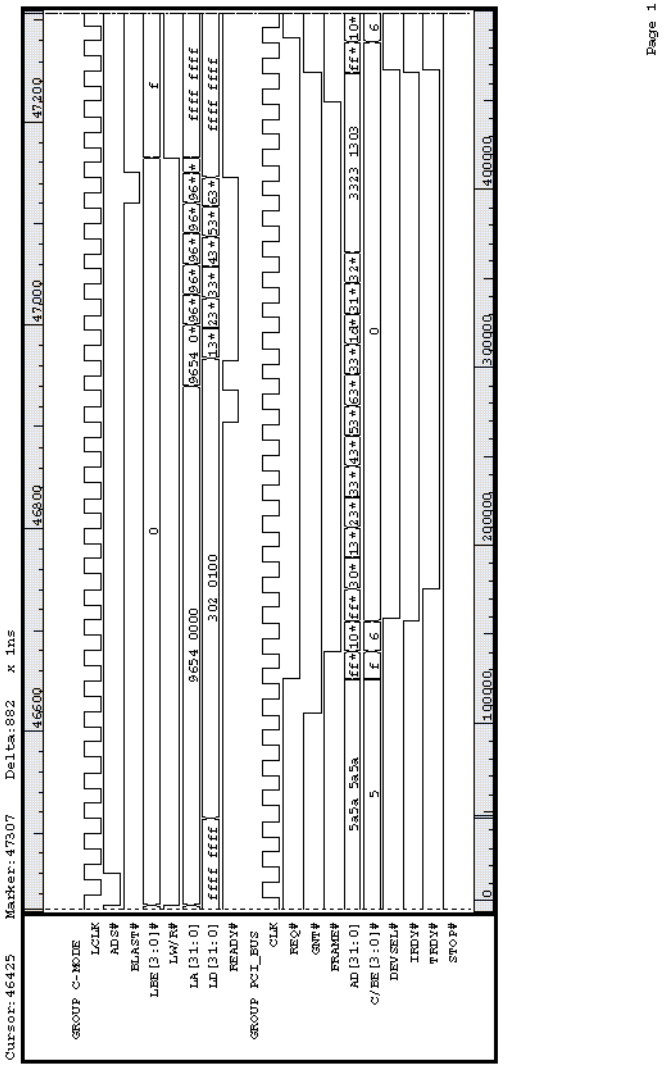
Timing Diagram 5-18. Direct Master Memory Read of Four Lwords



Timing Diagram 5-19. Direct Master Memory Write of Six Lwords

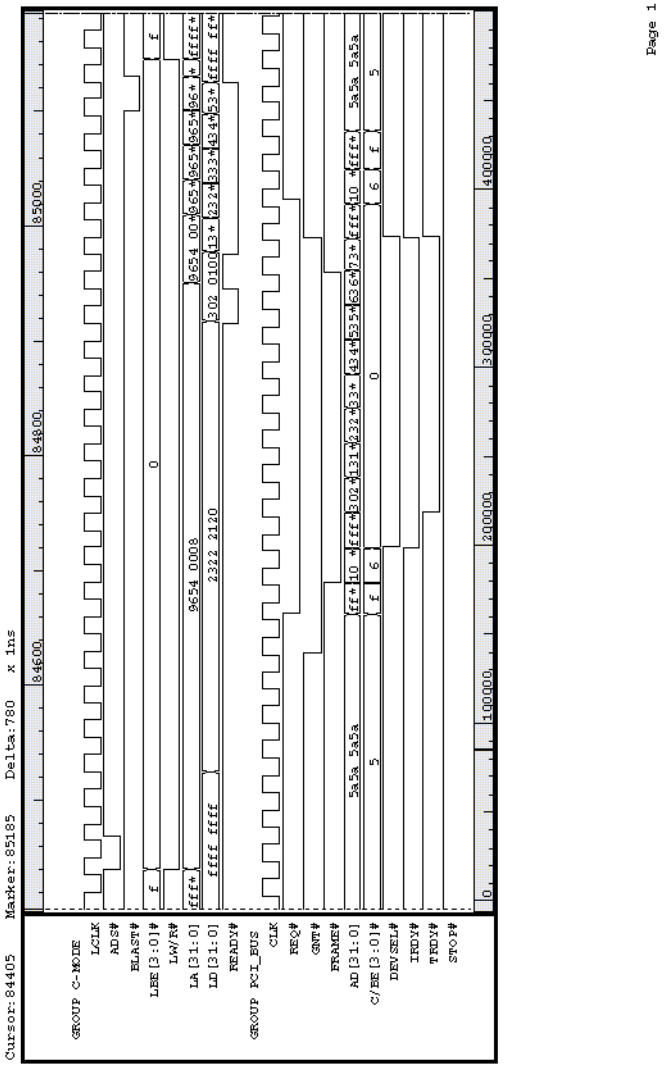


Timing Diagram 5-20. Direct Master Memory Write of Seven Lwords

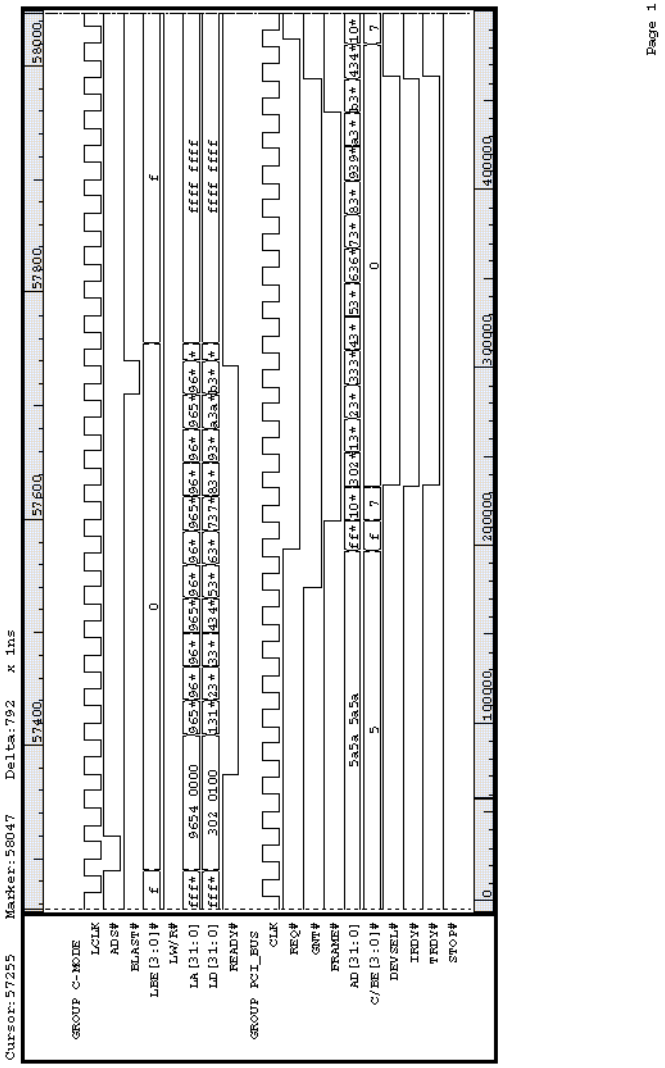


Timing Diagram 5-21. Direct Master Memory Read of Seven Lwords

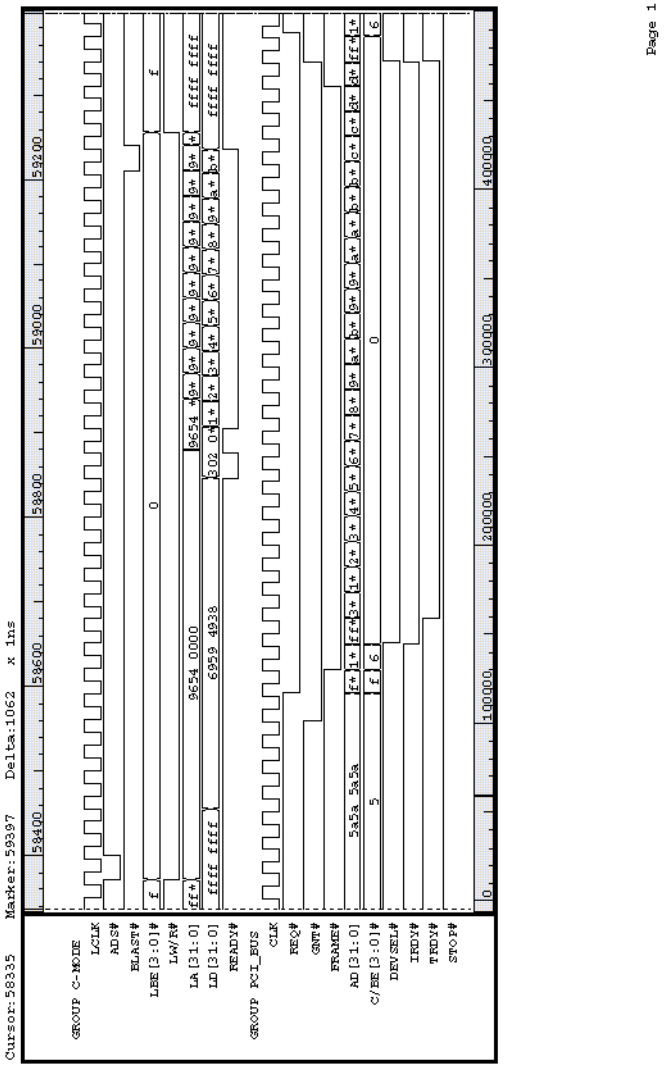
Timing Diagram 5-22. Direct Master Memory Write of Eight Lwords



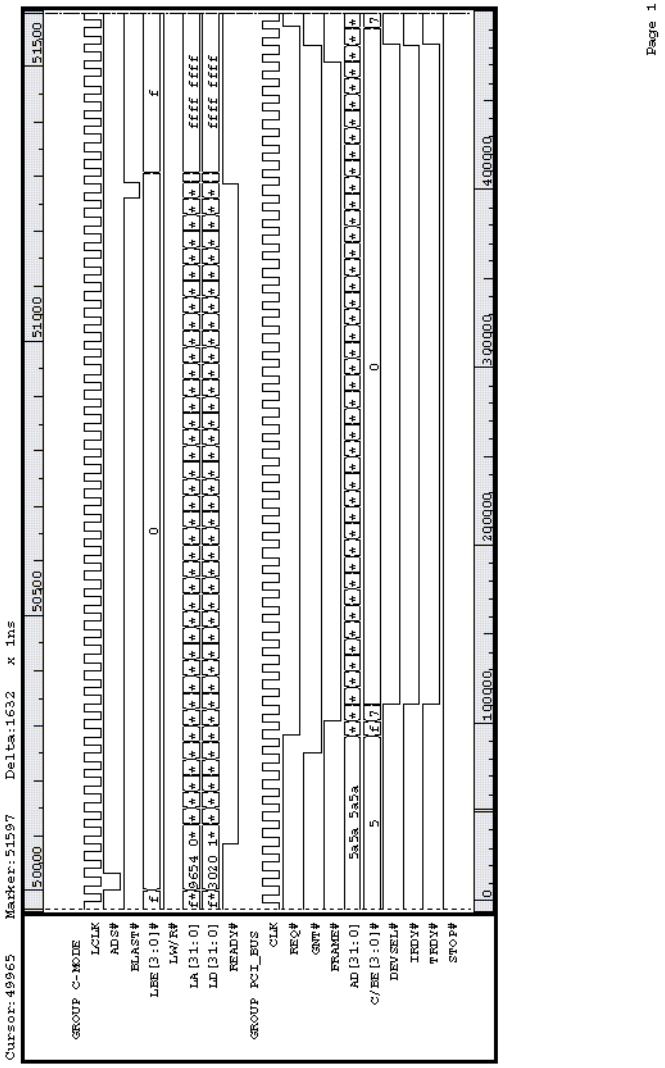
Timing Diagram 5-23. Direct Master Memory Read of Eight Lwords



Timing Diagram 5-24. Direct Master Memory Write of 12 Lwords

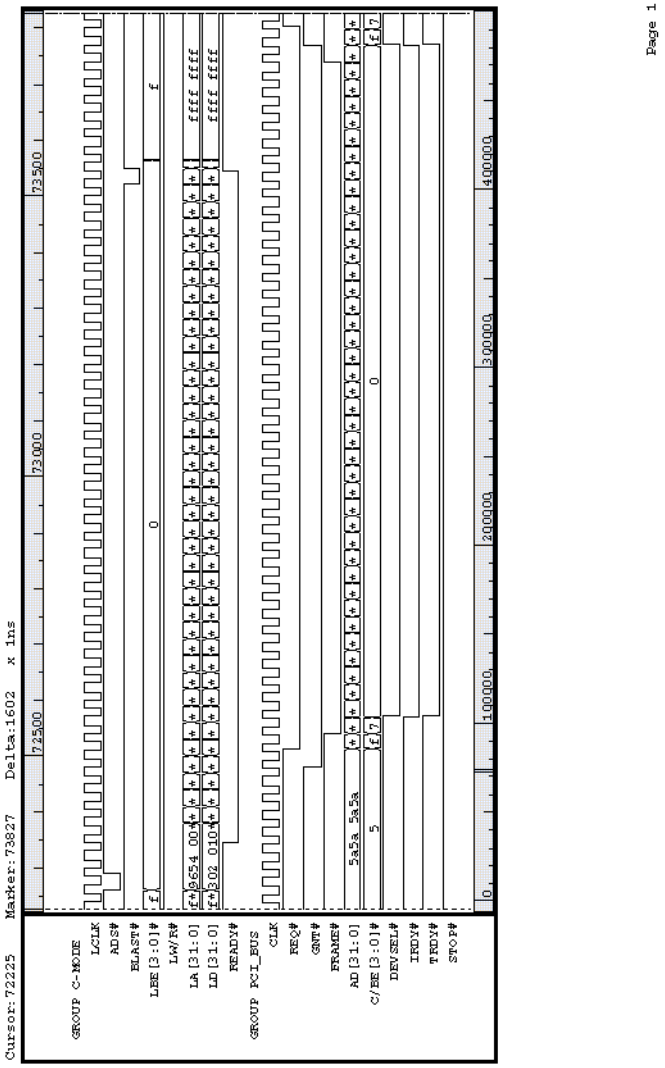


Timing Diagram 5-25. Direct Master Memory Read of 12 Lwords

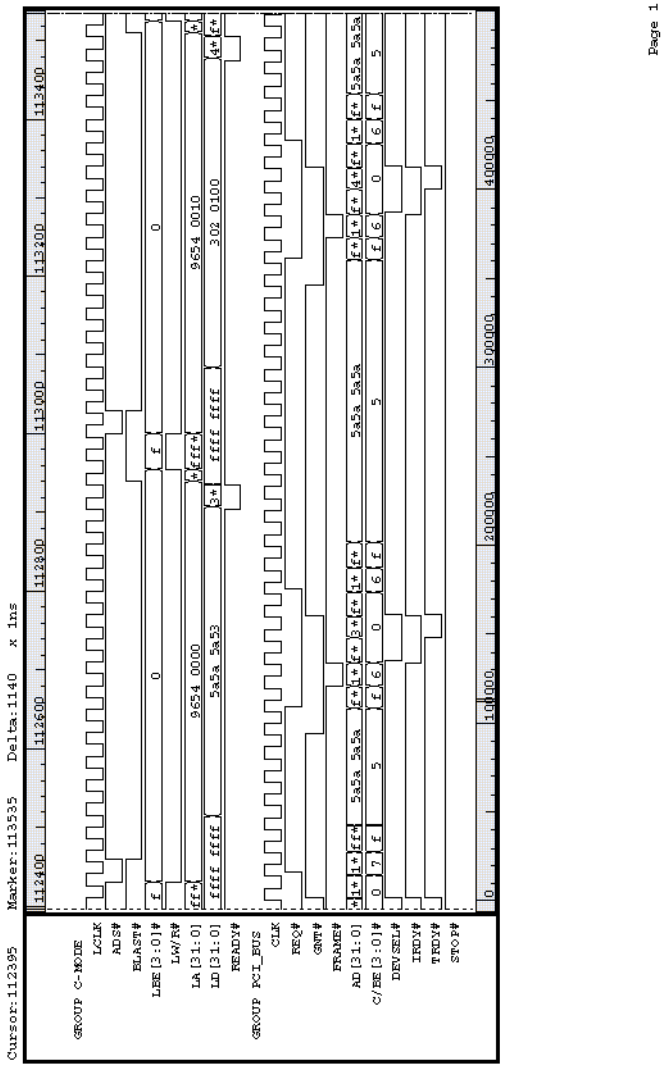


Timing Diagram 5-26. Direct Master Memory Write of 32 Lwords

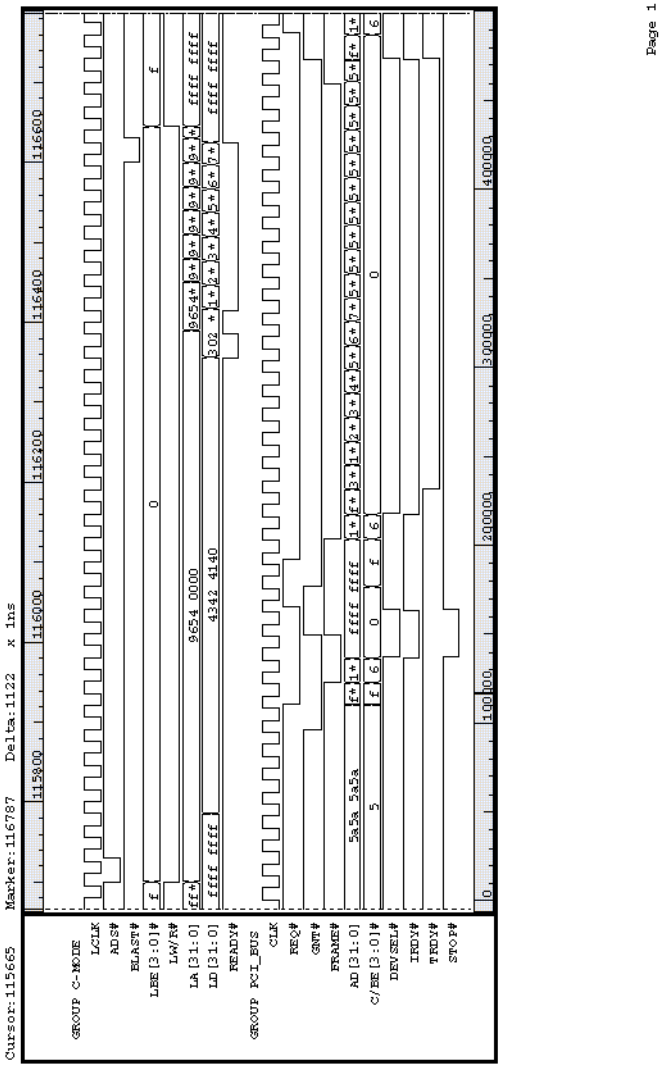
Timing Diagram 5-27. Direct Master Memory Read of 32 Lwords



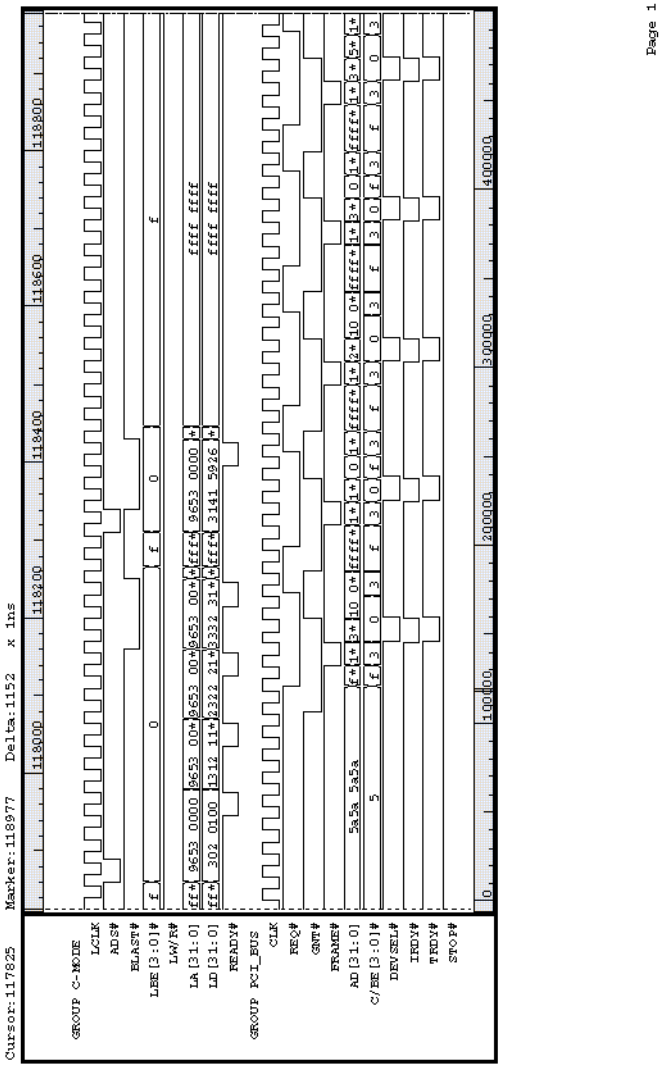
Timing Diagram 5-28. Direct Master Memory Write of 40 Lwords



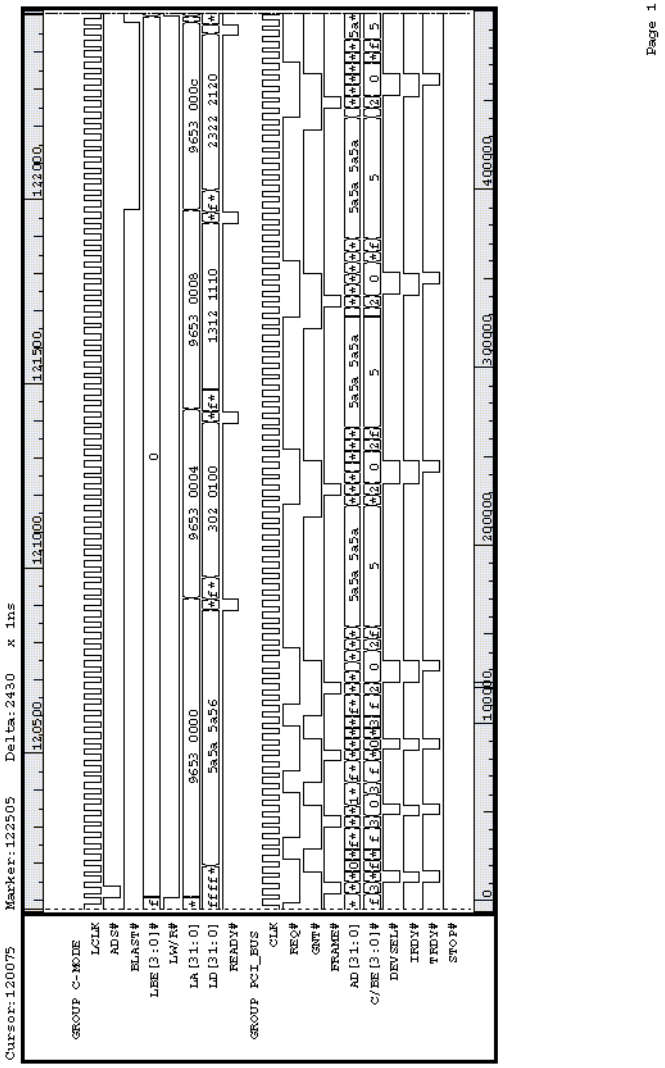
Timing Diagram 5-29. Direct Master Single Read by Direct Master Single Read



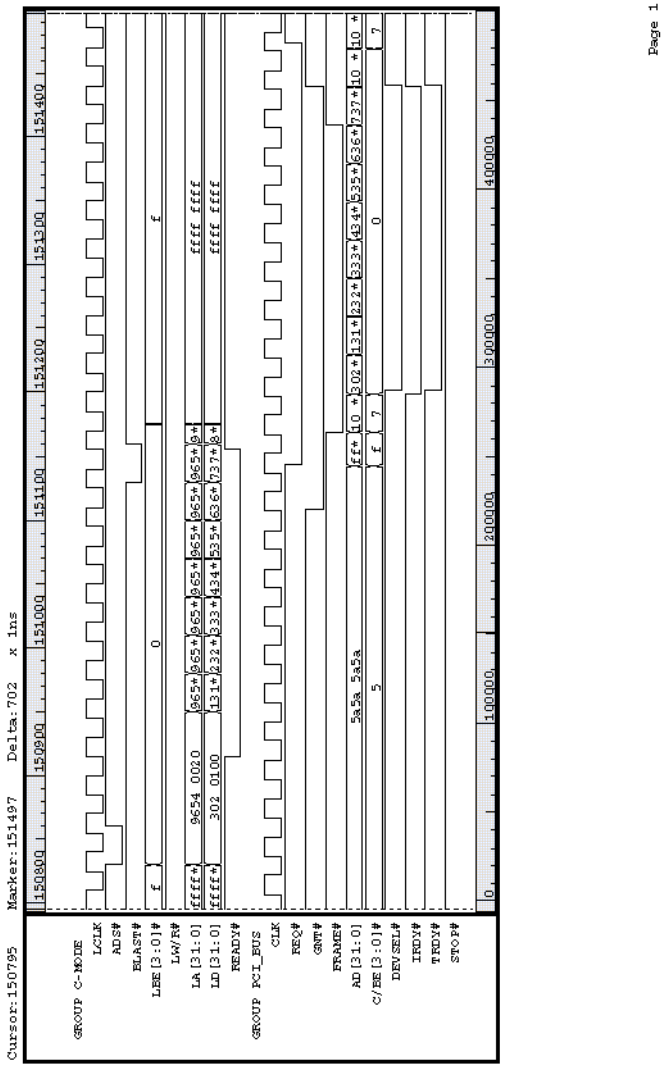
Timing Diagram 5-30. Direct Master Burst Read with a PCI Retry



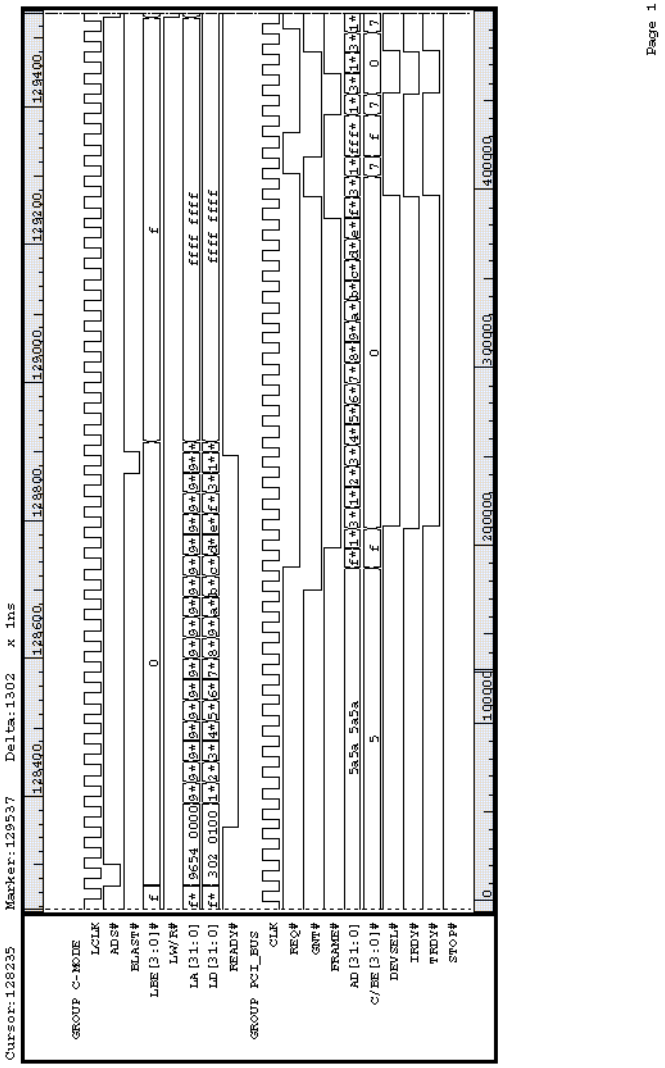
Timing Diagram 5-31. Direct Master Burst Write Four Lwords



Timing Diagram 5-32. Direct Master Burst Read Four Lwords from I/O Space

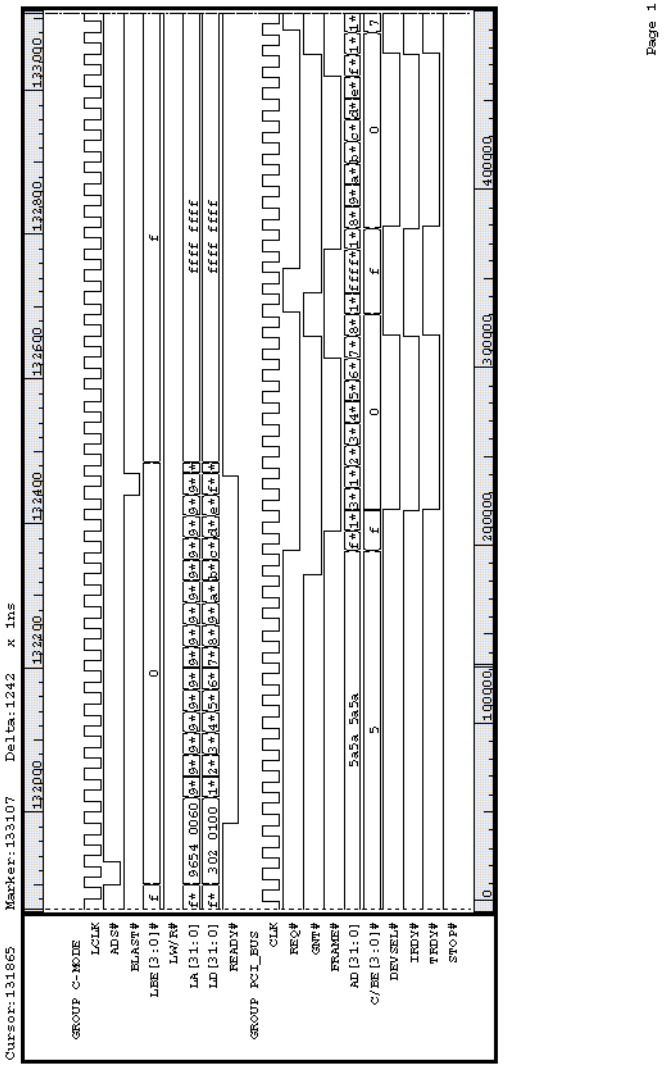


Timing Diagram 5-33. Direct Master MWI 7, Transfer Eight Lwords

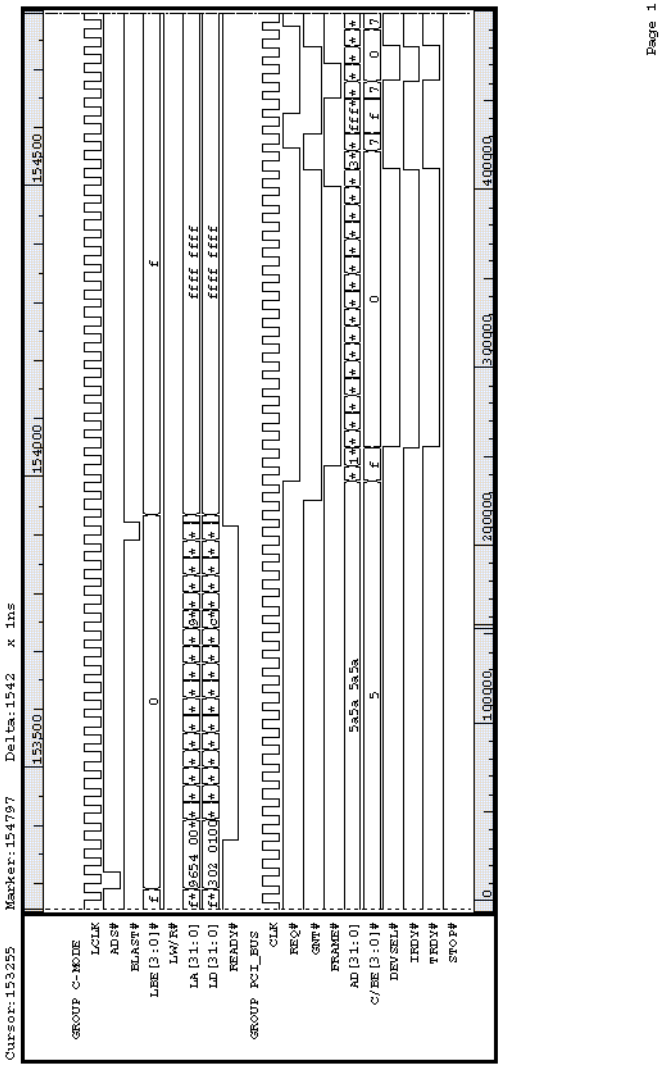


Timing Diagram 5-34. Direct Master MWI 8

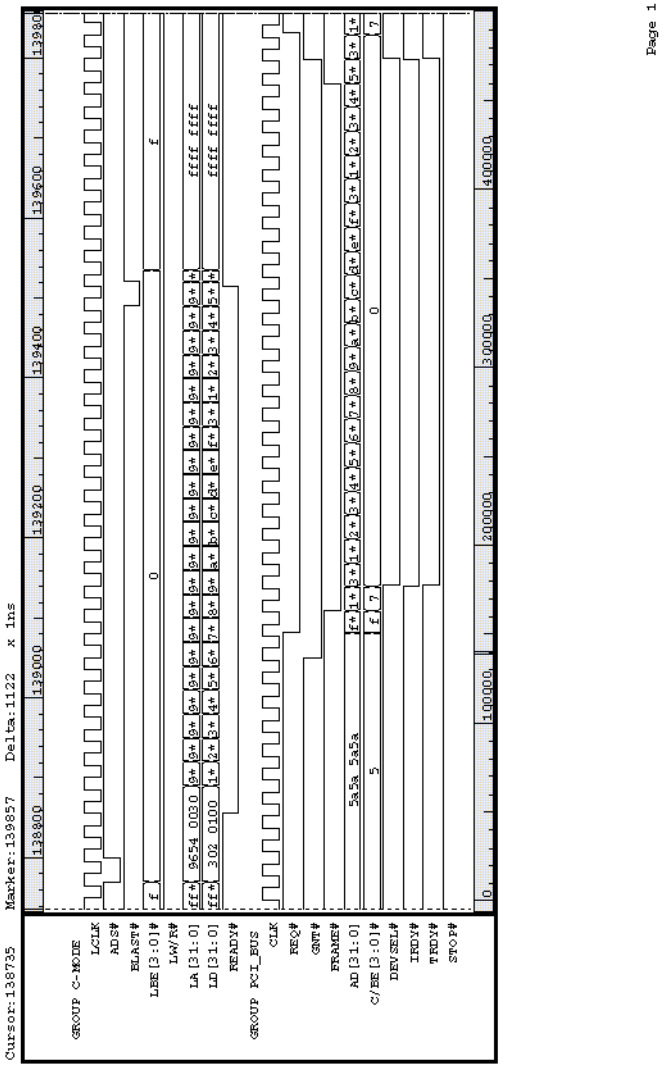
Timing Diagram 5-35. TD-023.tif



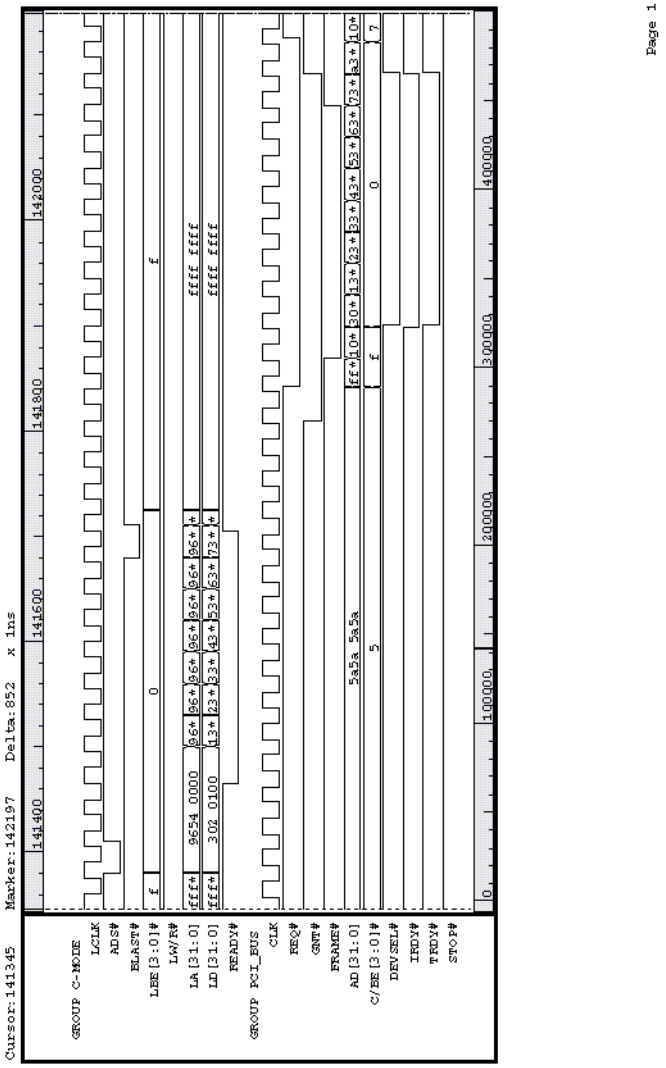
Timing Diagram 5-36. Direct Master MWI 8, Transfer 16 Lwords



Timing Diagram 5-37. Direct Master MWI 16, Transfer Eight Lwords

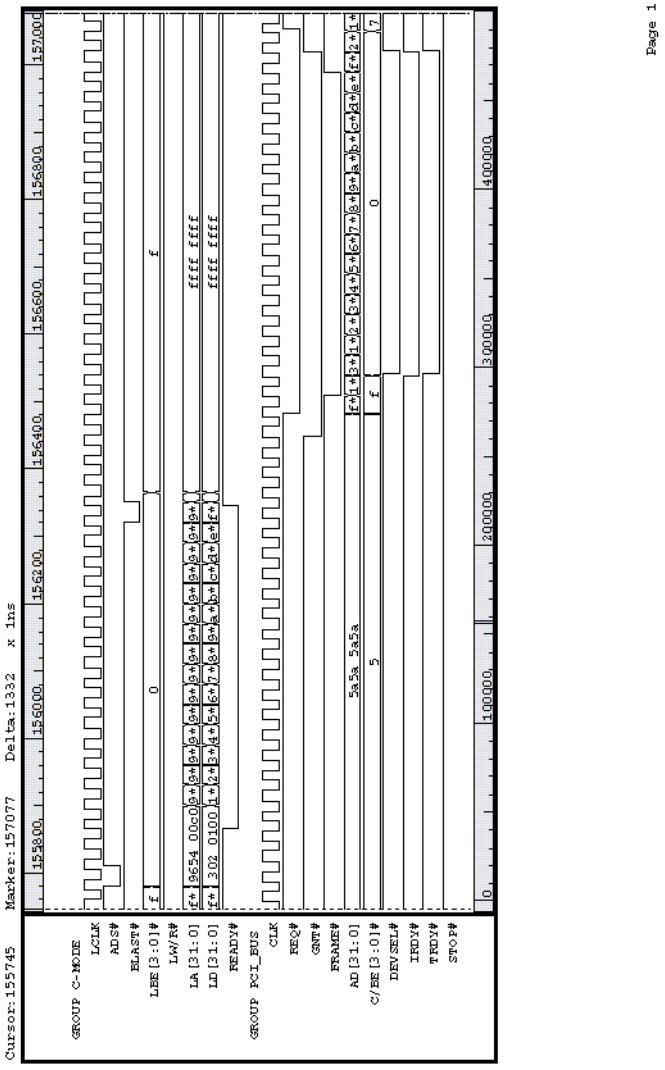


Timing Diagram 5-38. Set Direct Master Write Mode to 8 for Write and Invalidate



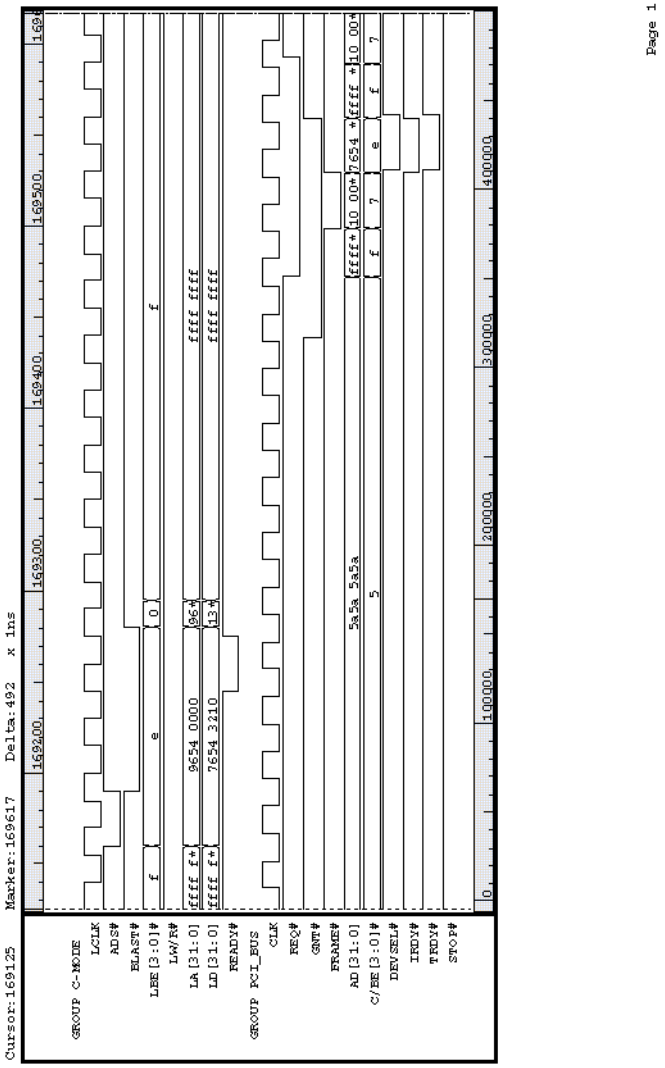
Timing Diagram 5-39. Set Direct Master Write Mode to 8 for Write and Invalidate (Direct Master MWI 8)

Timing Diagram 5-40. TD-028.tif

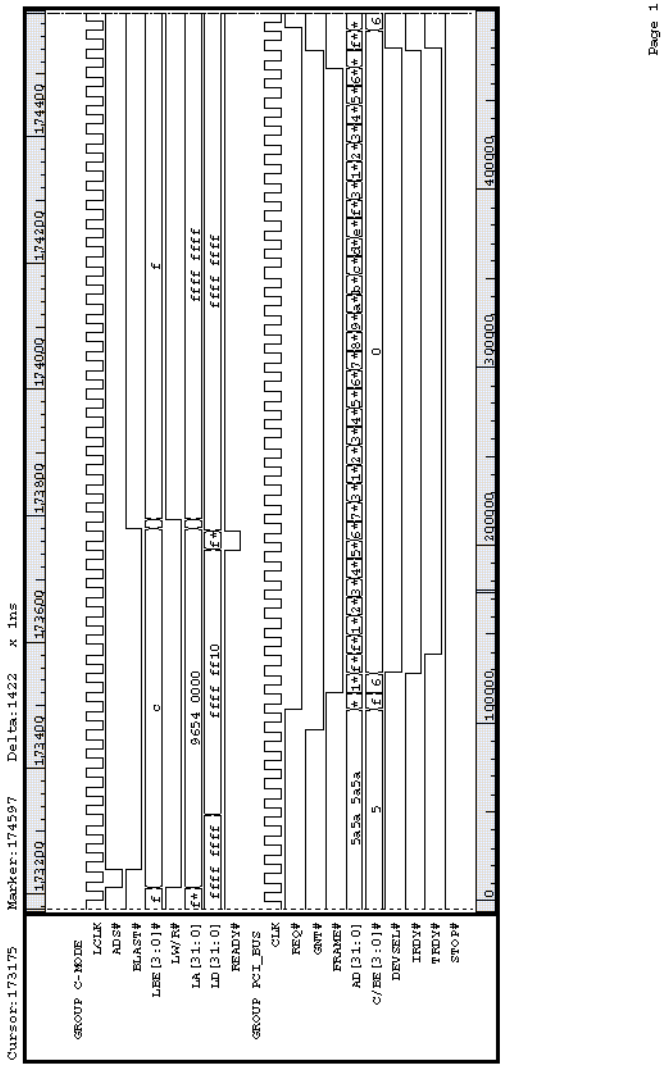


Timing Diagram 5-41. Set Direct Master Write Mode to 16 for Write and Invalidate

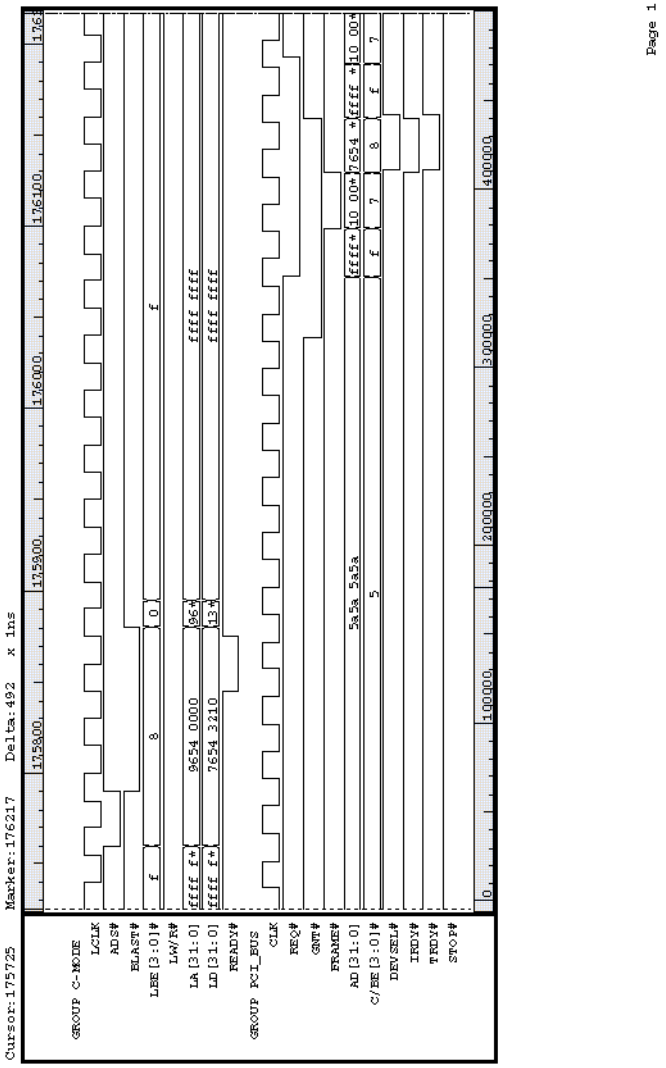
Timing Diagram 5-42. TD-034.tif



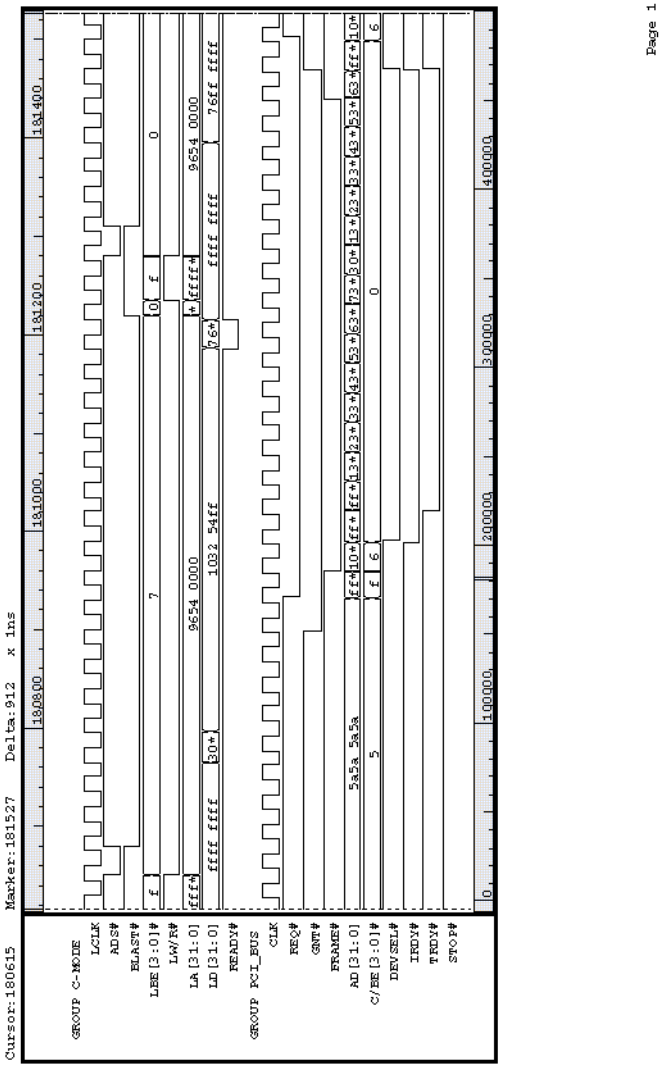
Timing Diagram 5-43. Direct Master Single Write Read Memory, LBE = 1110b



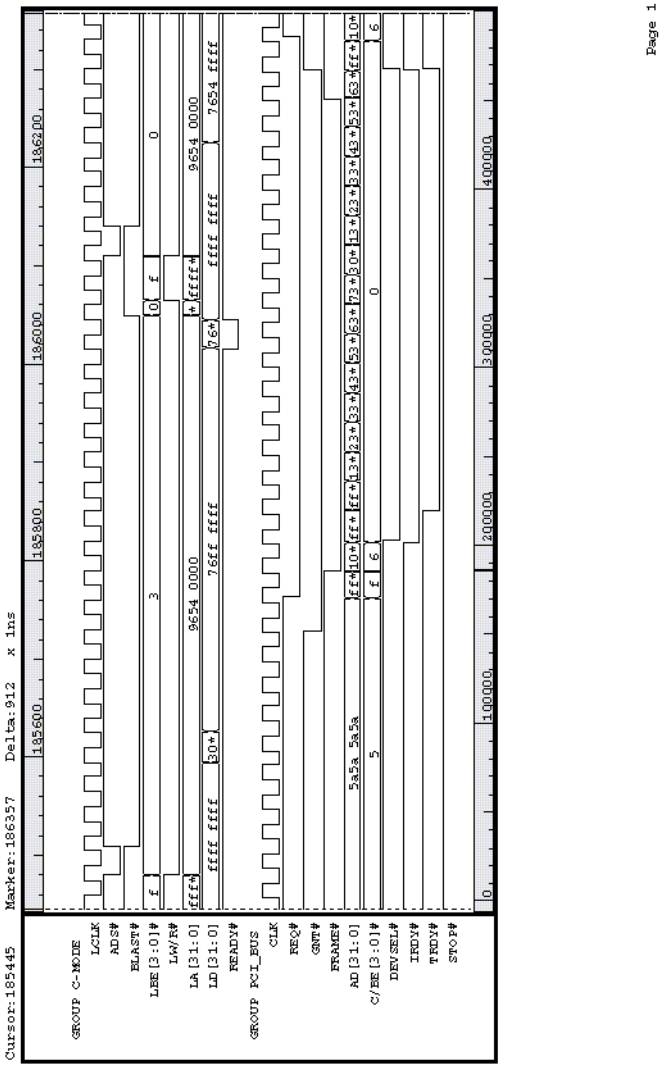
Timing Diagram 5-44. Direct Master Single Write Read Memory, LBE = 1100b



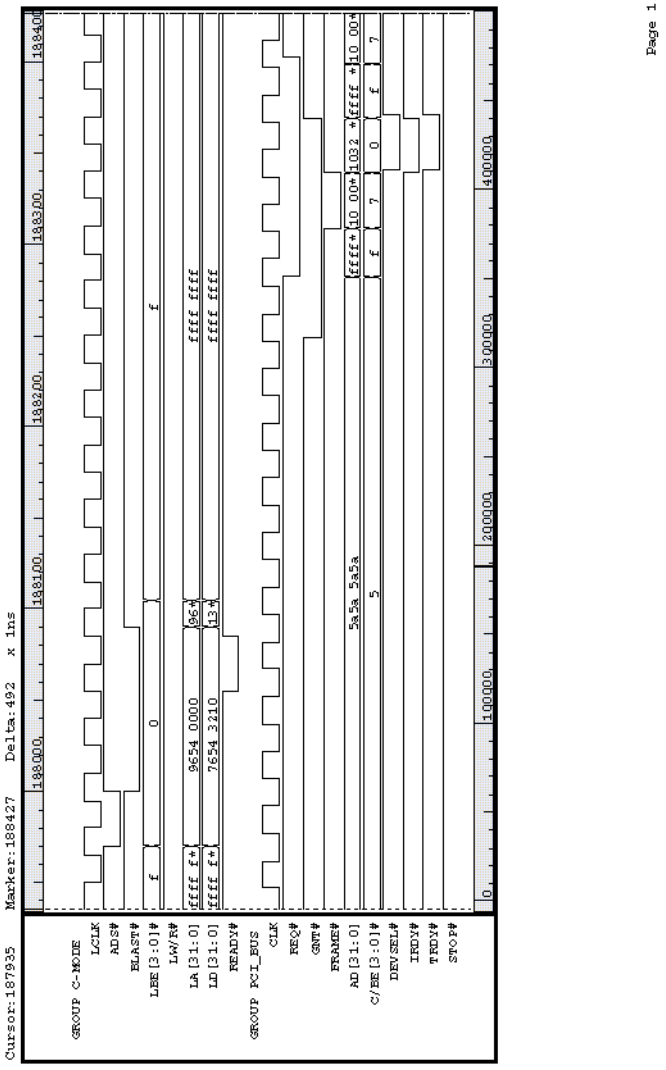
Timing Diagram 5-45. Direct Master Single Write Read Memory, LBE = 1000b



Timing Diagram 5-46. Direct Master Memory Single Write Read Big Endian, LBE = 0111b

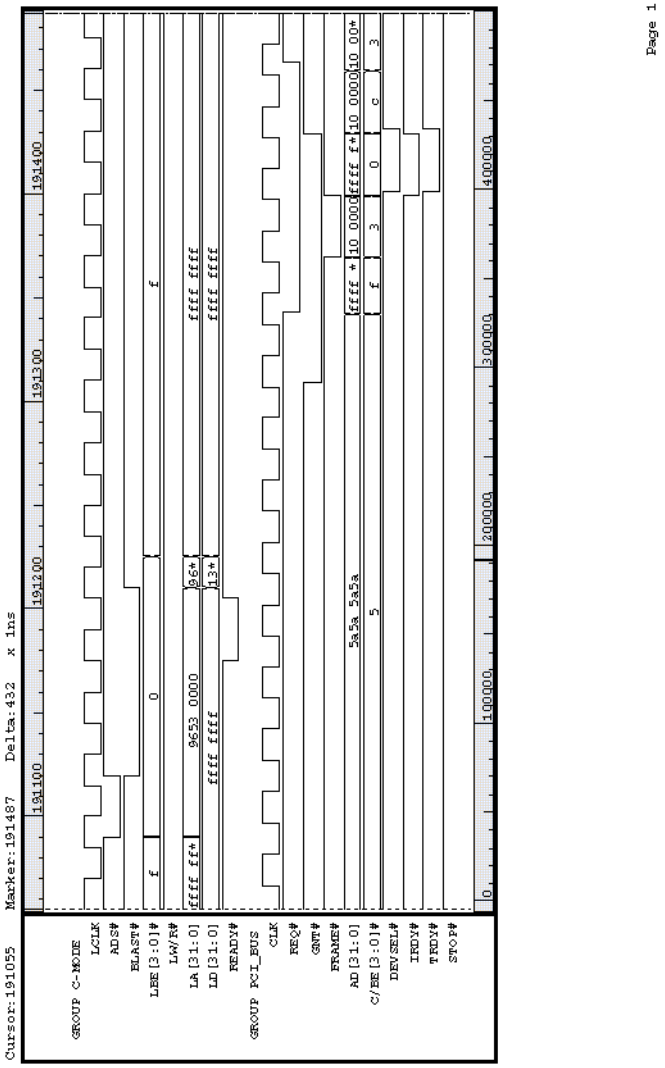


Timing Diagram 5-47. Direct Master Memory Single Write Read Big Endian, LBE = 0011b



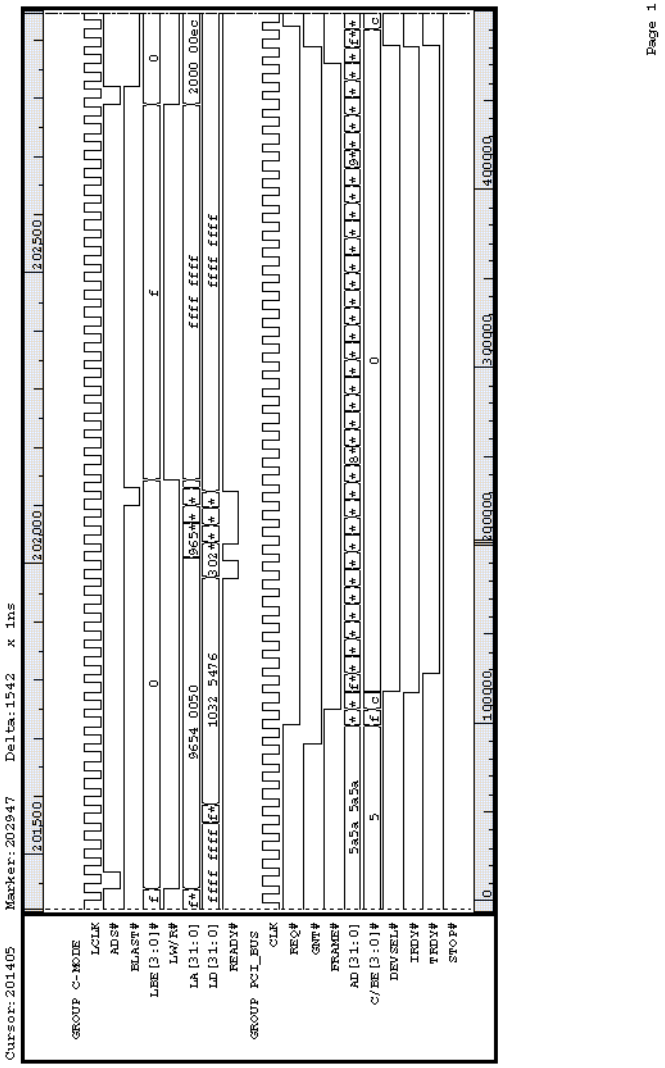
Timing Diagram 5-48. Direct Master Single Write Read Memory Big Endian Input

Timing Diagram 5-49. TD-041.tif



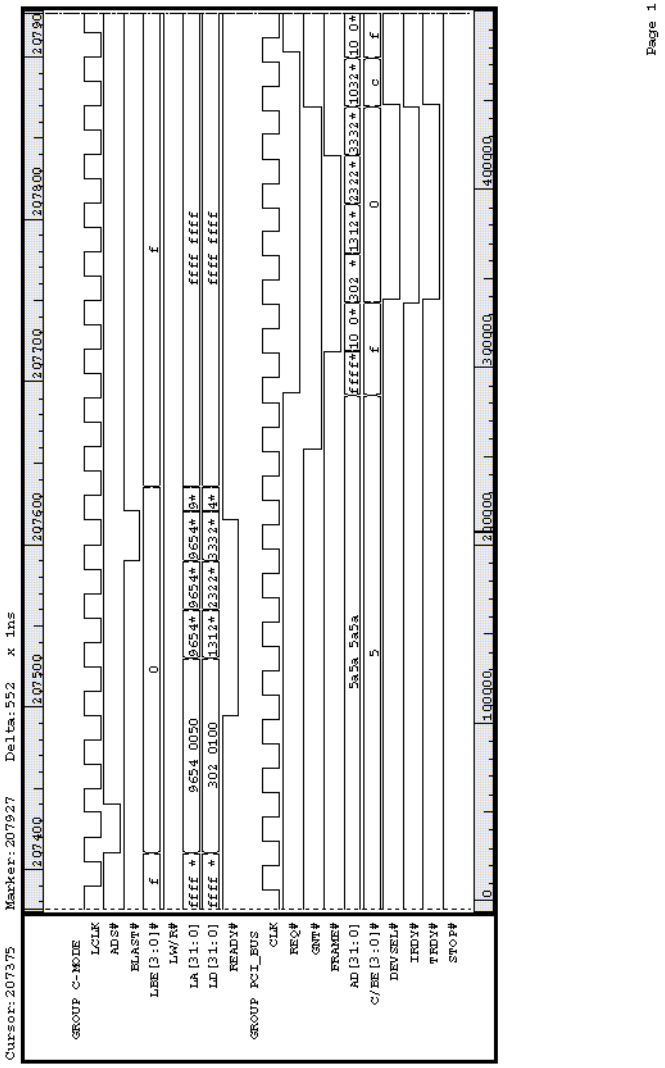
Timing Diagram 5-50. Direct Master I/O Single Write Read Big Endian, LBE = 1110b

Timing Diagram 5-51. TD-045.tif

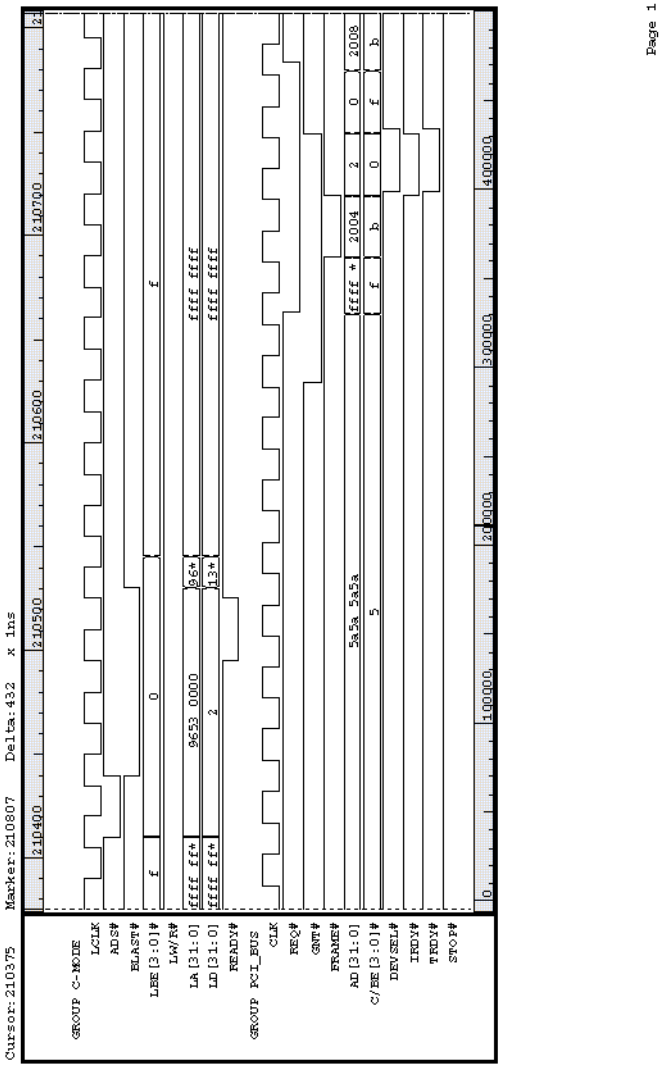


Timing Diagram 5-52. Direct Master Memory Read Programmable Command Code, CBE = 1100, 1110

Timing Diagram 5-53. TD-047.tif

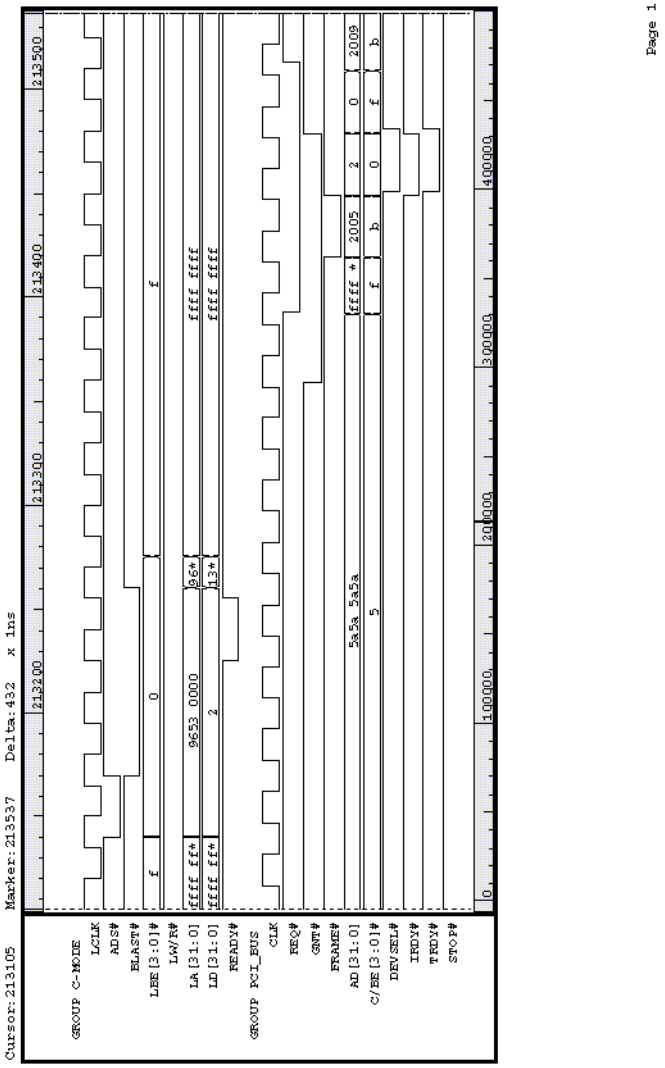


Timing Diagram 5-54. Direct Master Memory Write Programmable Command Code, CBE = 1111



Timing Diagram 5-55. Direct Master Type 0, Configuration Device 2, Address 4

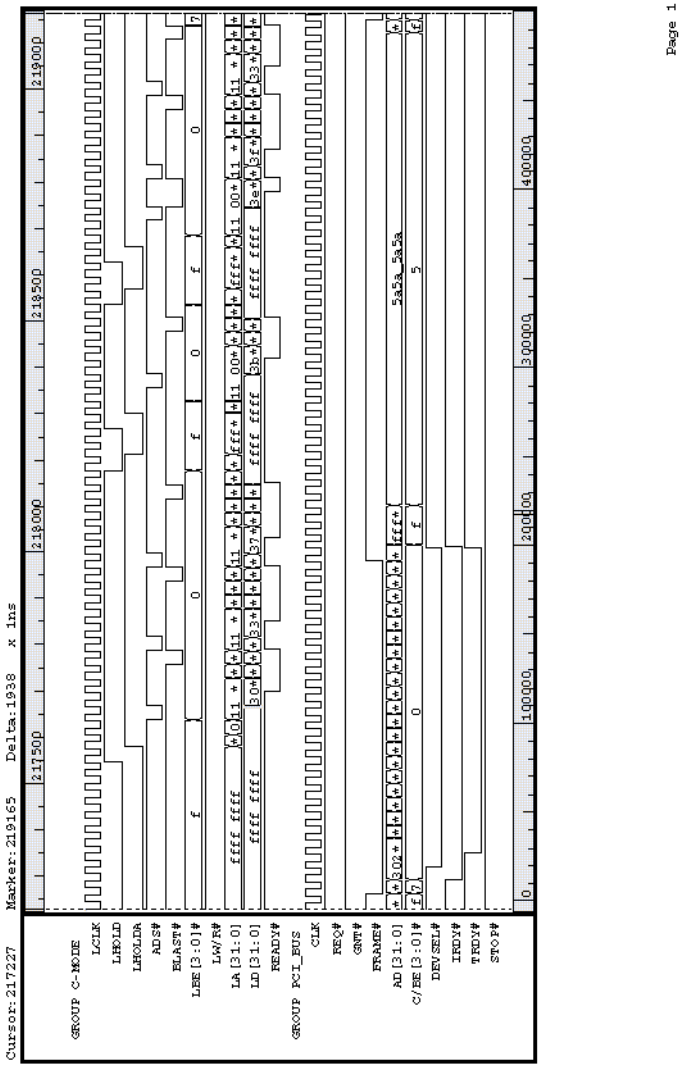
Timing Diagram 5-56. TD-050tif



Timing Diagram 5-57. Direct Master Type 1, Configuration Device 2, Address 5

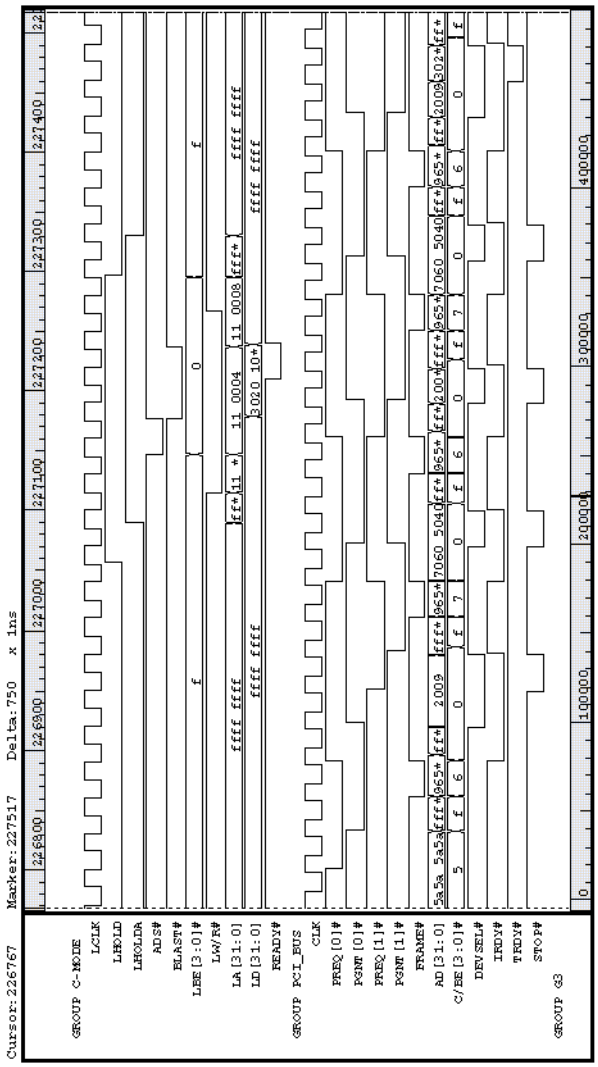
Timing Diagram 5-58. TD-052.tif

5.6.2 C Mode Only Direct Slave Timing Diagrams



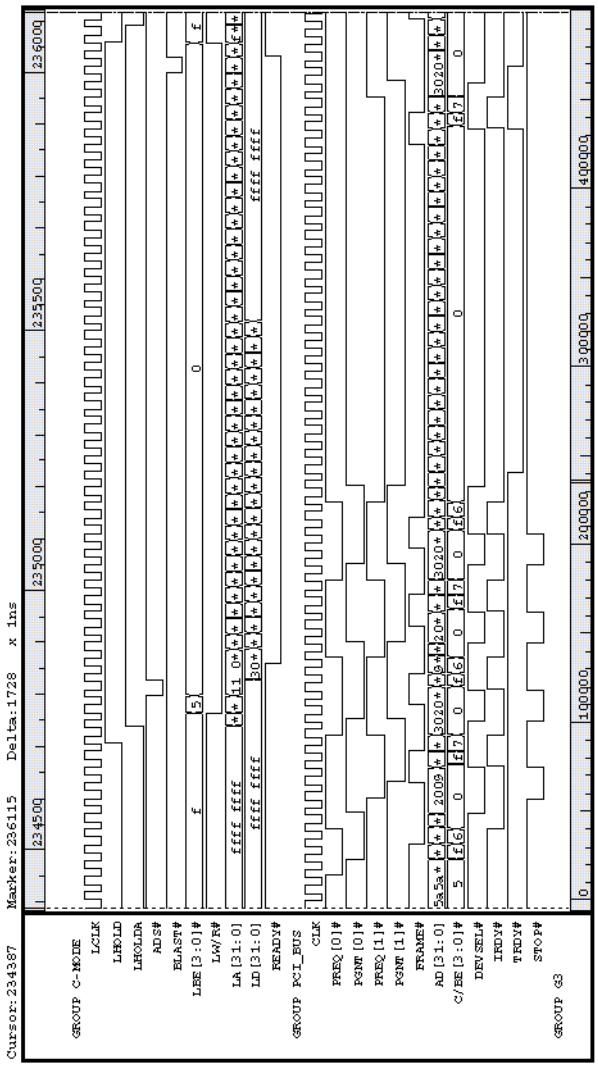
Timing Diagram 5-59. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, BREQ Enabled

Timing Diagram 5-60. TD-054.tif

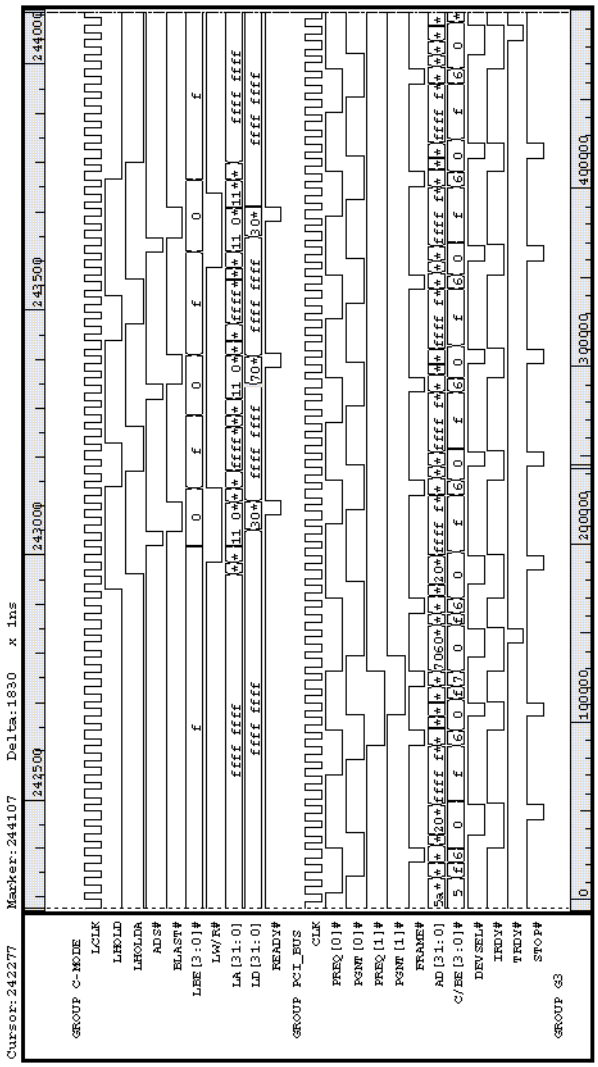


Timing Diagram 5-61. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Single Read No Write Enabled

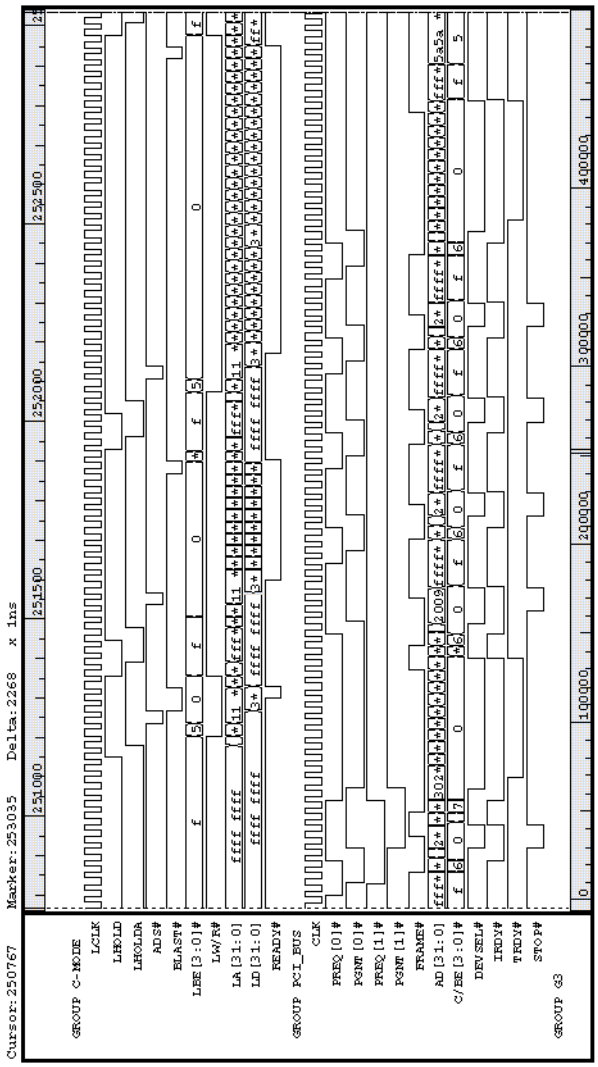
Timing Diagram 5-62. TD-056.tif



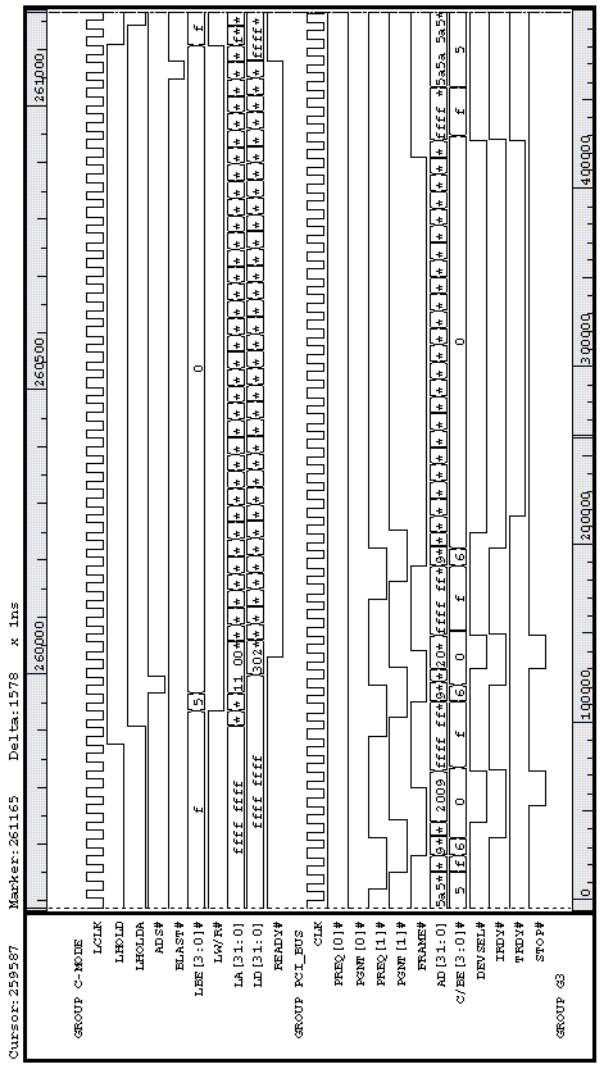
Timing Diagram 5-63. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Burst Read No Write Enabled



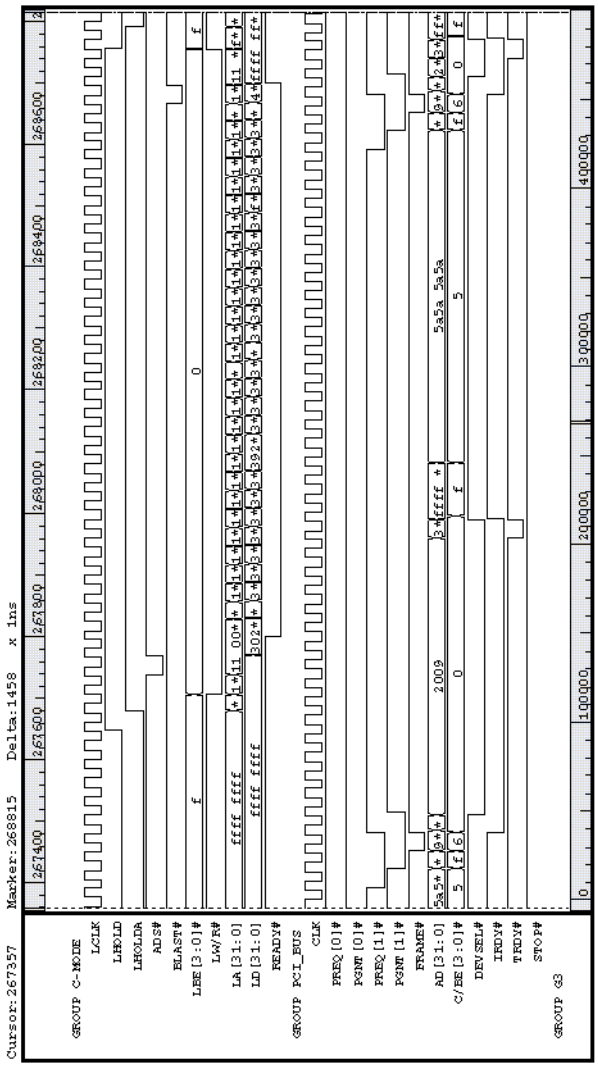
Timing Diagram 5-64. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Single Read Write Flush Enabled



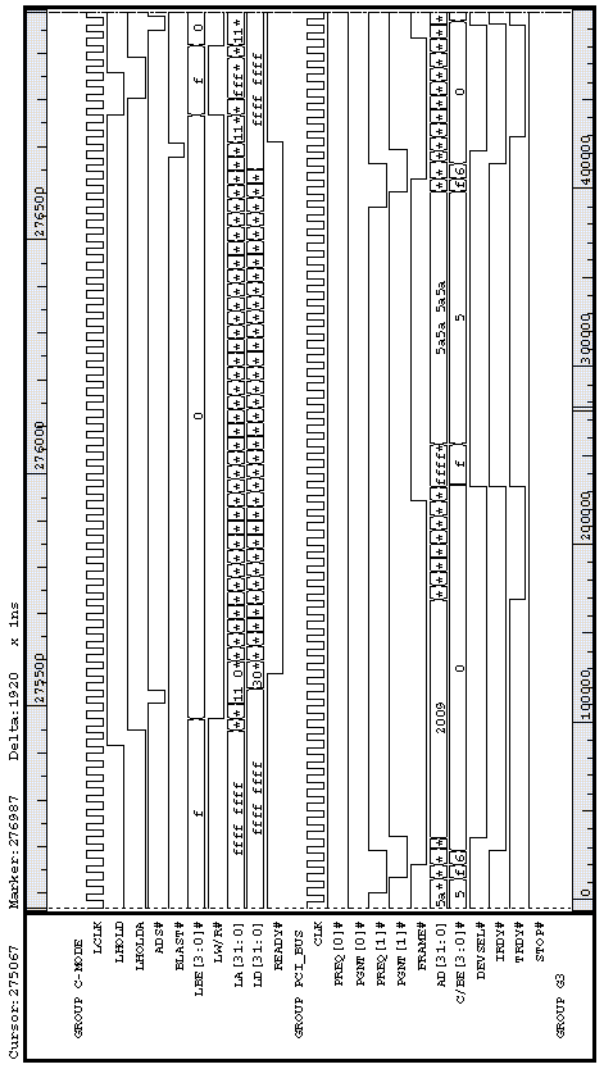
Timing Diagram 5-65. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Burst Read Write Flush Enabled



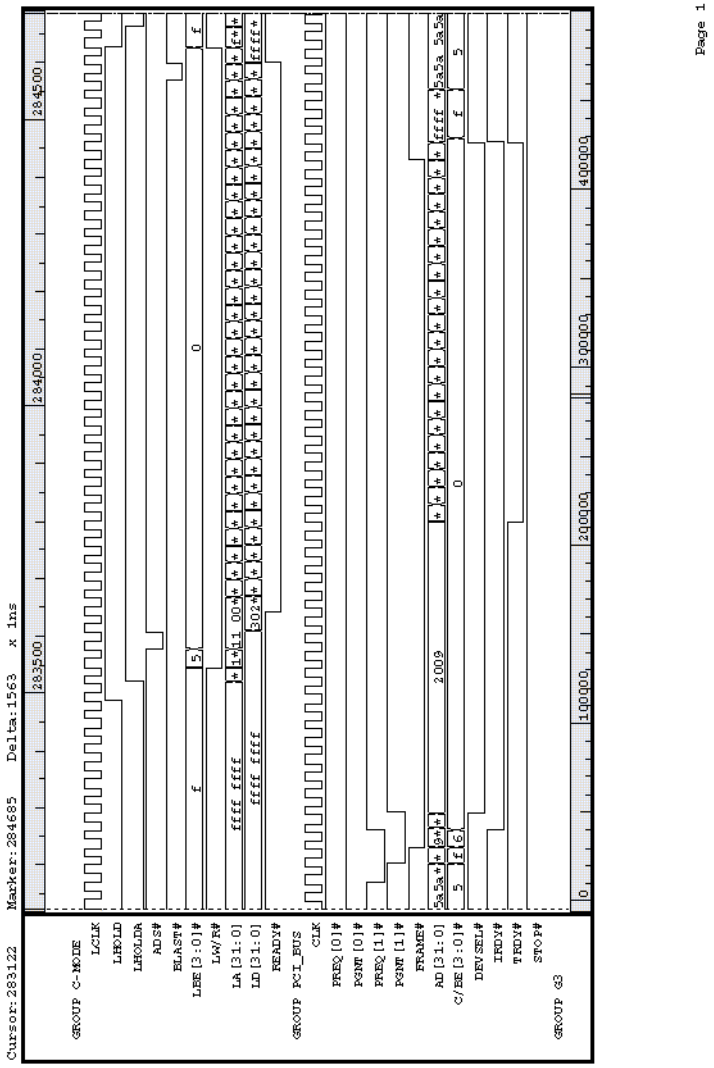
Timing Diagram 5-66. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Delay Read Enabled



Timing Diagram 5-67. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Single Read Ahead Enabled

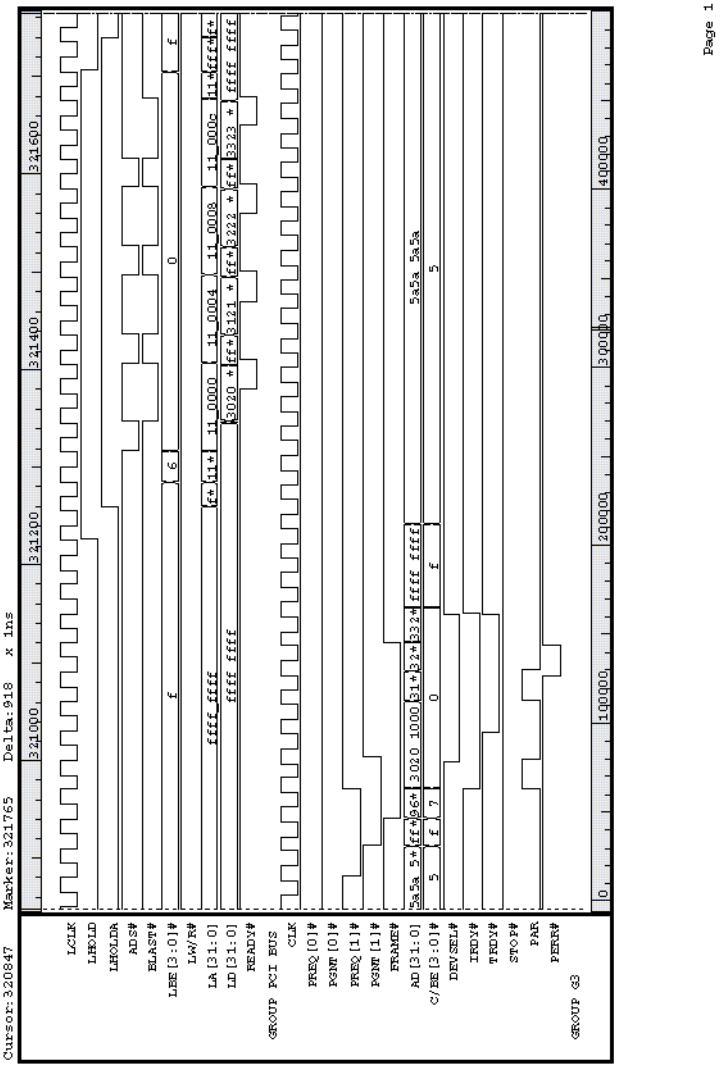


Timing Diagram 5-68. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Burst Read Ahead Enabled

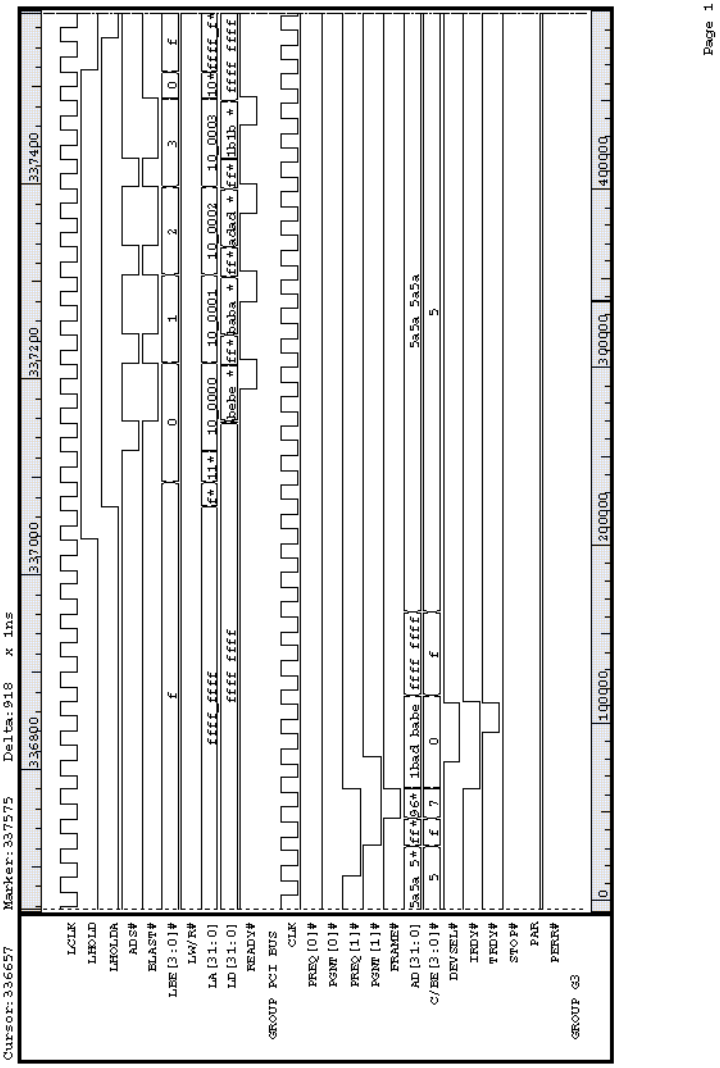


Timing Diagram 5-69. Direct Slave Burst Read with Prefetch Data

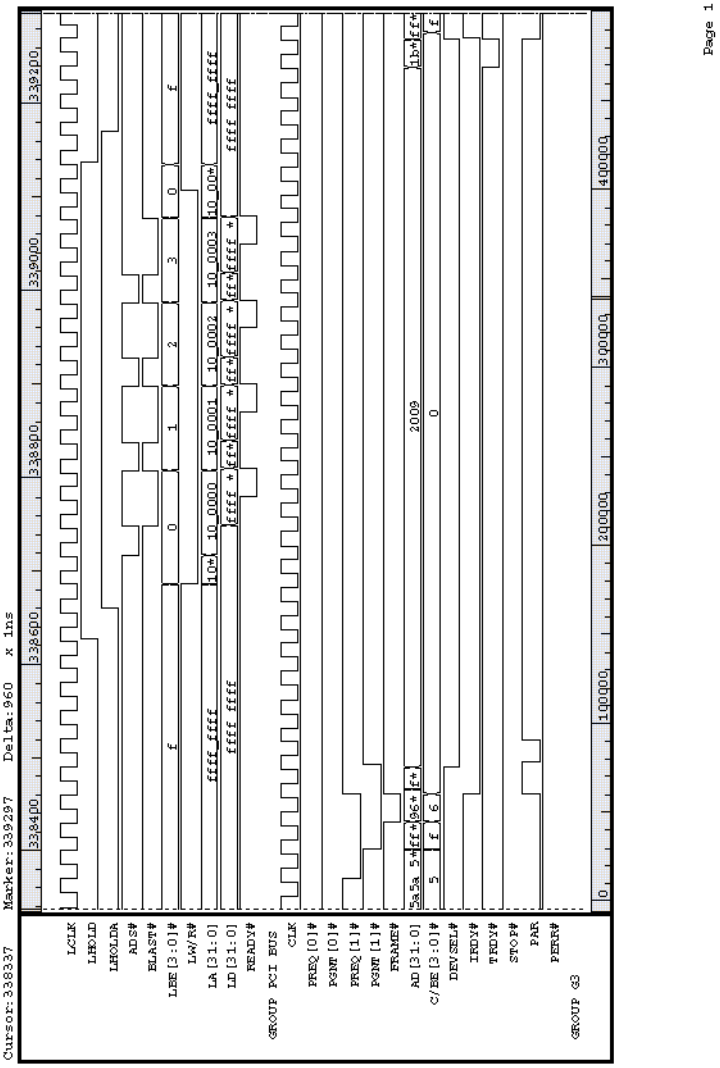
Timing Diagram 5-70. Direct Slave from PCI Bus to 32-Bit Device on Local Bus, Local Timer 8 Expired



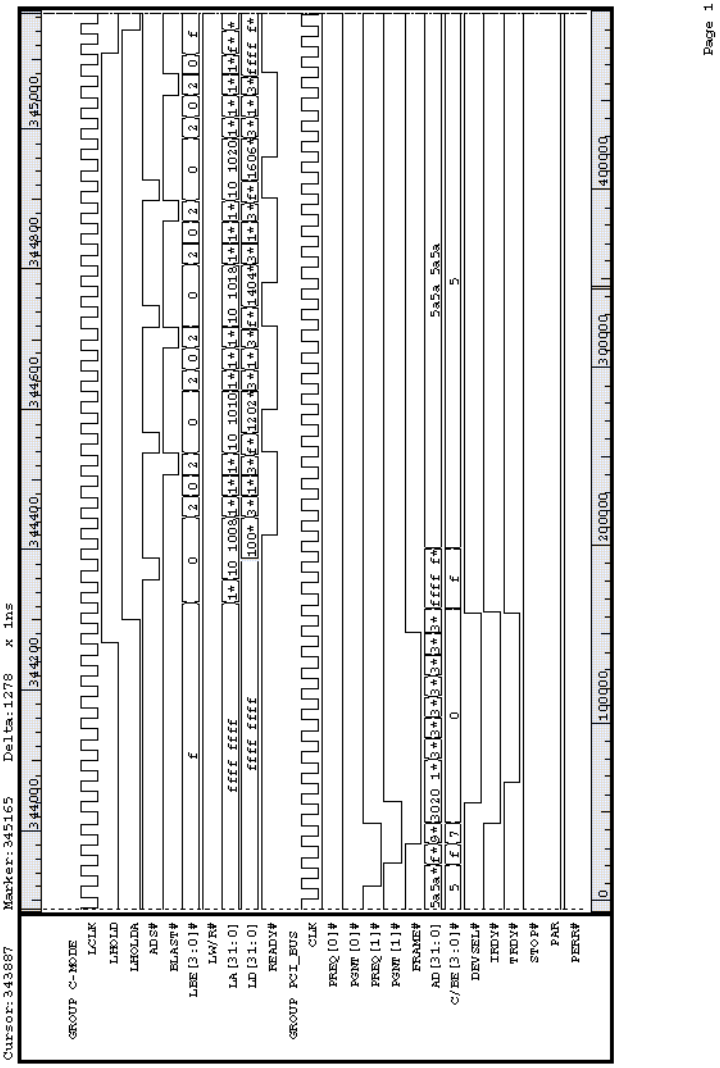
Timing Diagram 5-71. PCI Memory Write with Parity Error



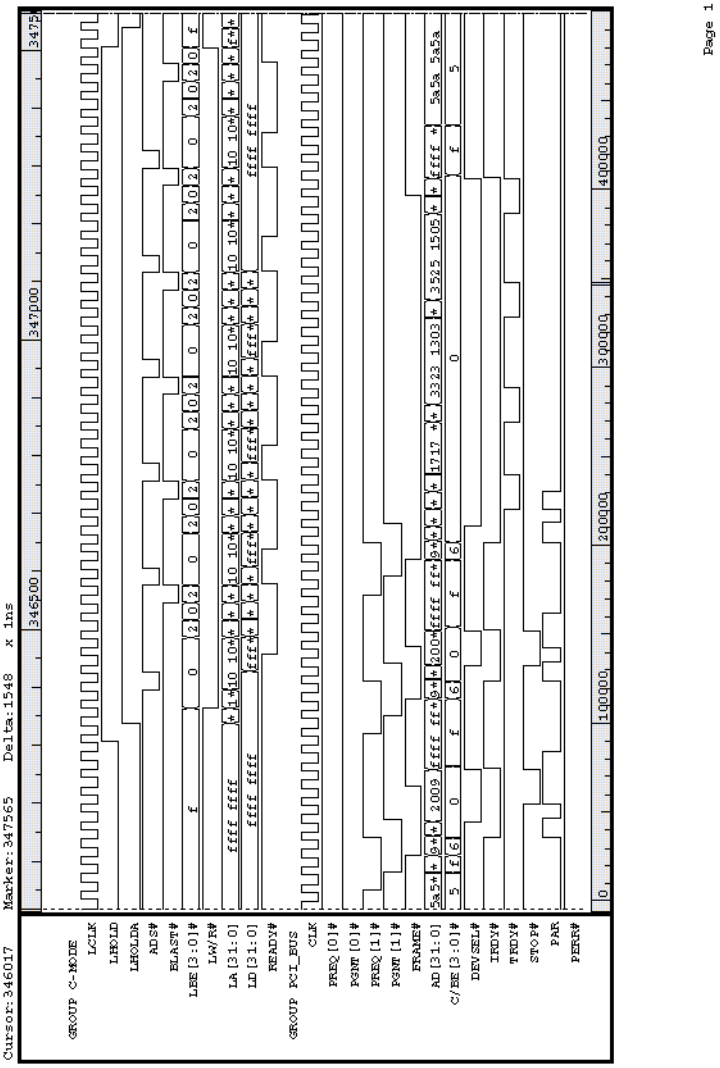
Timing Diagram 5-72. Direct Slave Single Write



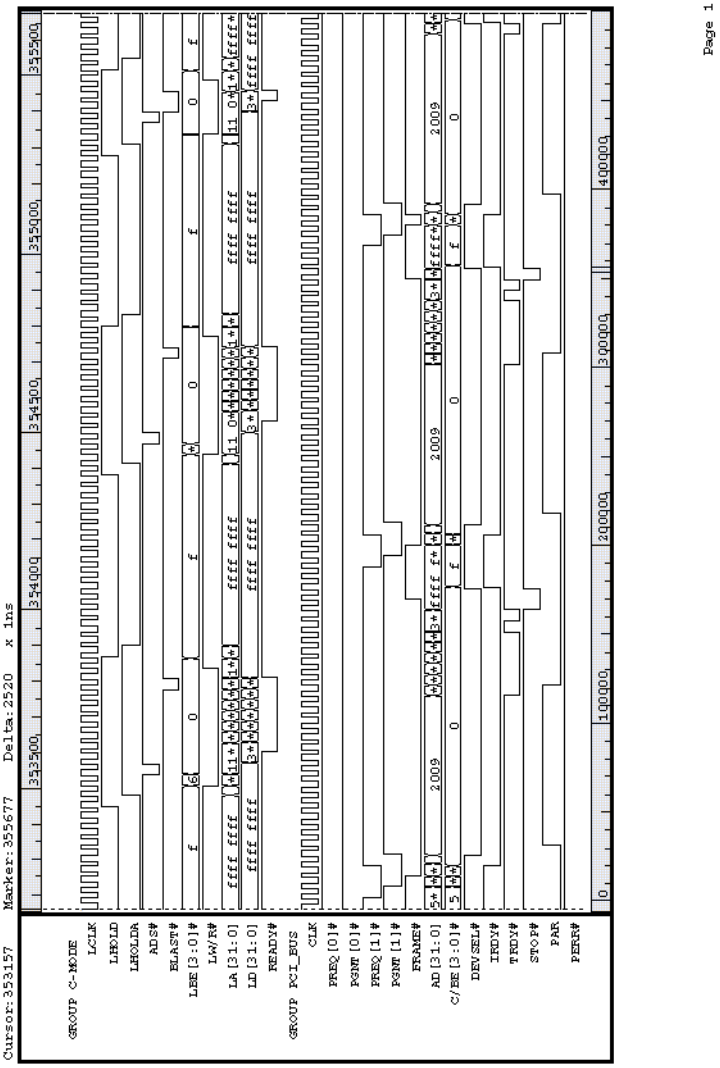
Timing Diagram 5-73. Direct Slave Single Read



Timing Diagram 5-74. Direct Slave Burst 20 Write

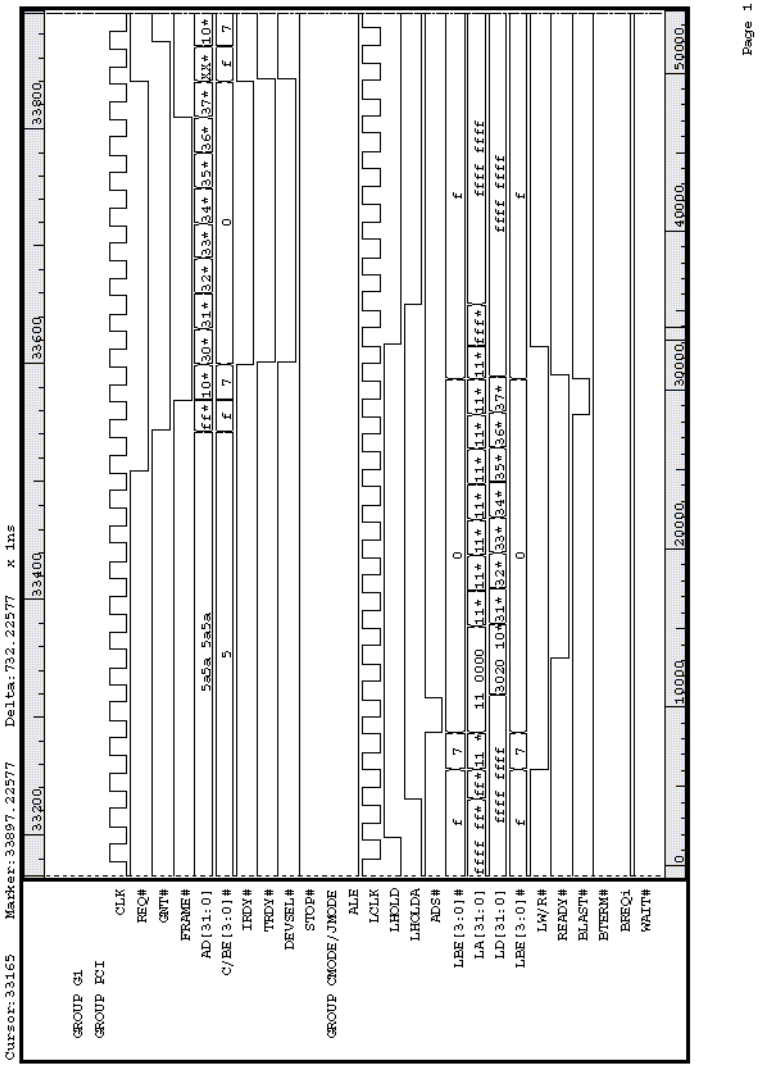


Timing Diagram 5-75. Direct Slave Burst 20 Read

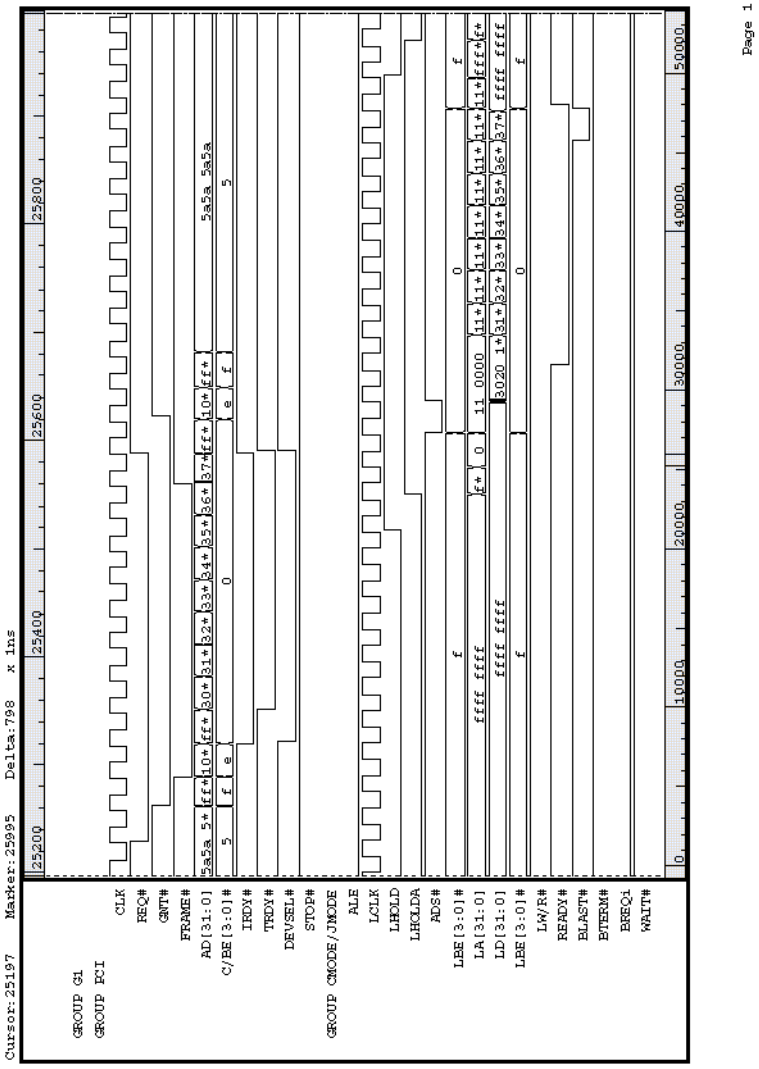


Timing Diagram 5-76. Direct Slave Burst Read

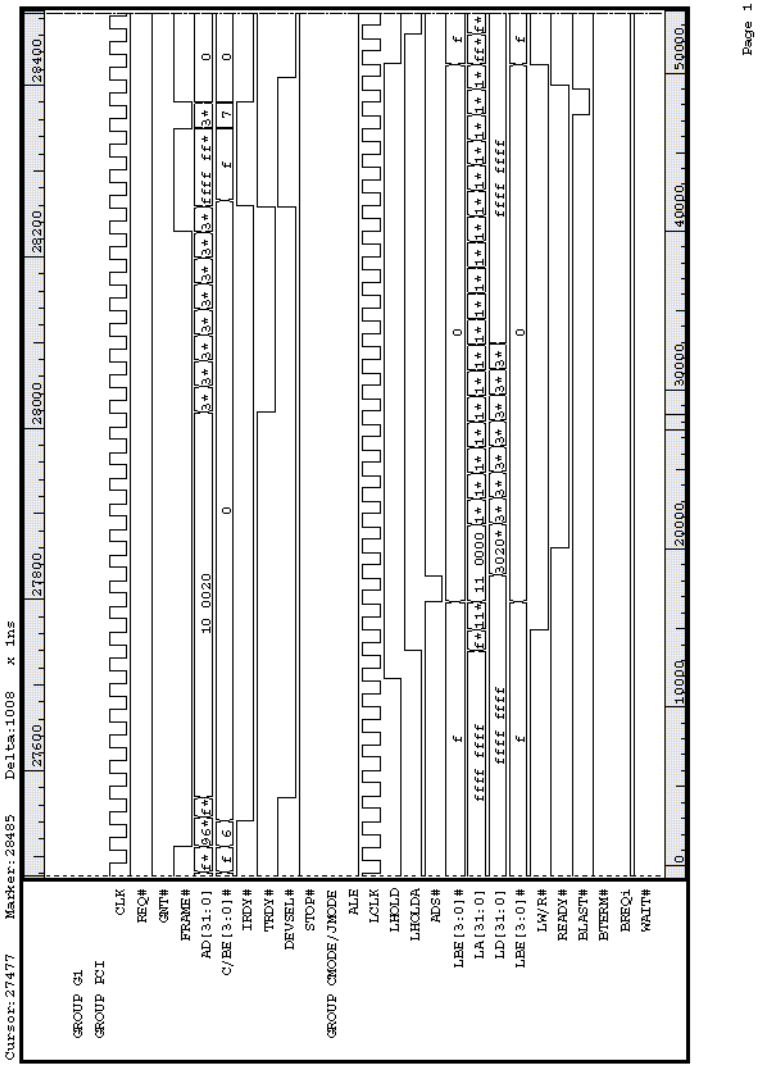
5.6.3 C Mode Only DMA Timing Diagrams



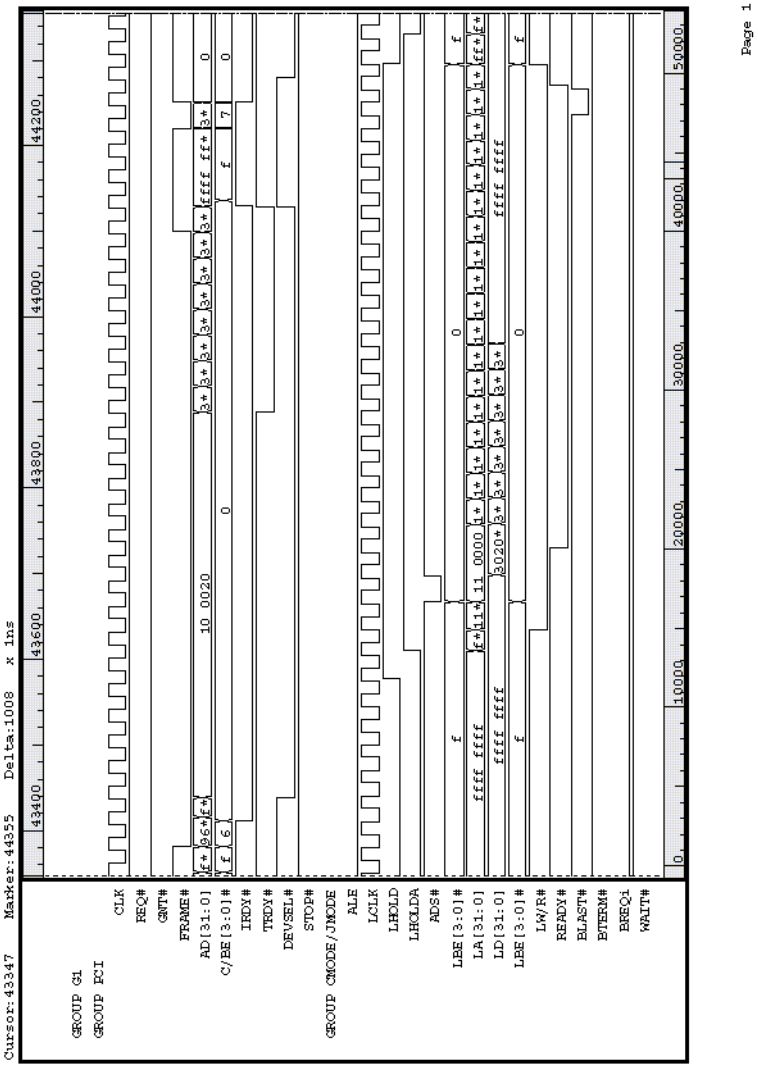
Timing Diagram 5-77. DMA Channel 0 Local-to-PCI (Memory Write Command)



Timing Diagram 5-78. DMA Channel 0 PCI-to-Local (Memory Read Line Command)

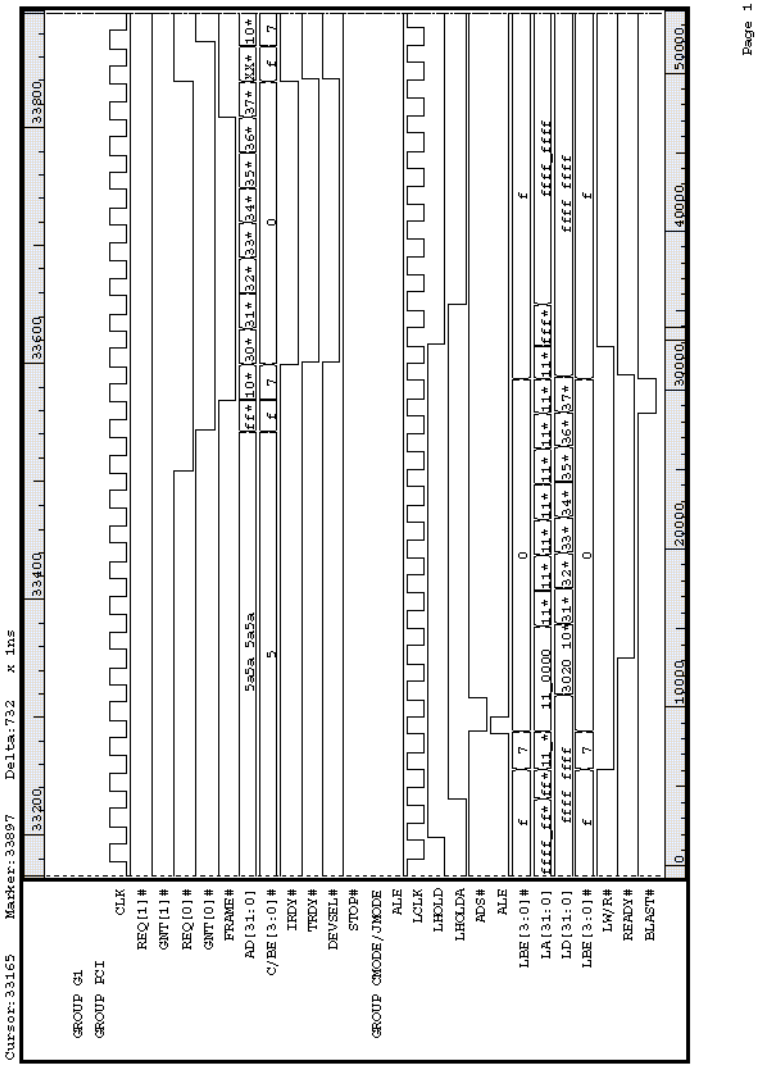


Timing Diagram 5-79. DMA Channel 0 PCI-to-Local (Memory Read Command)

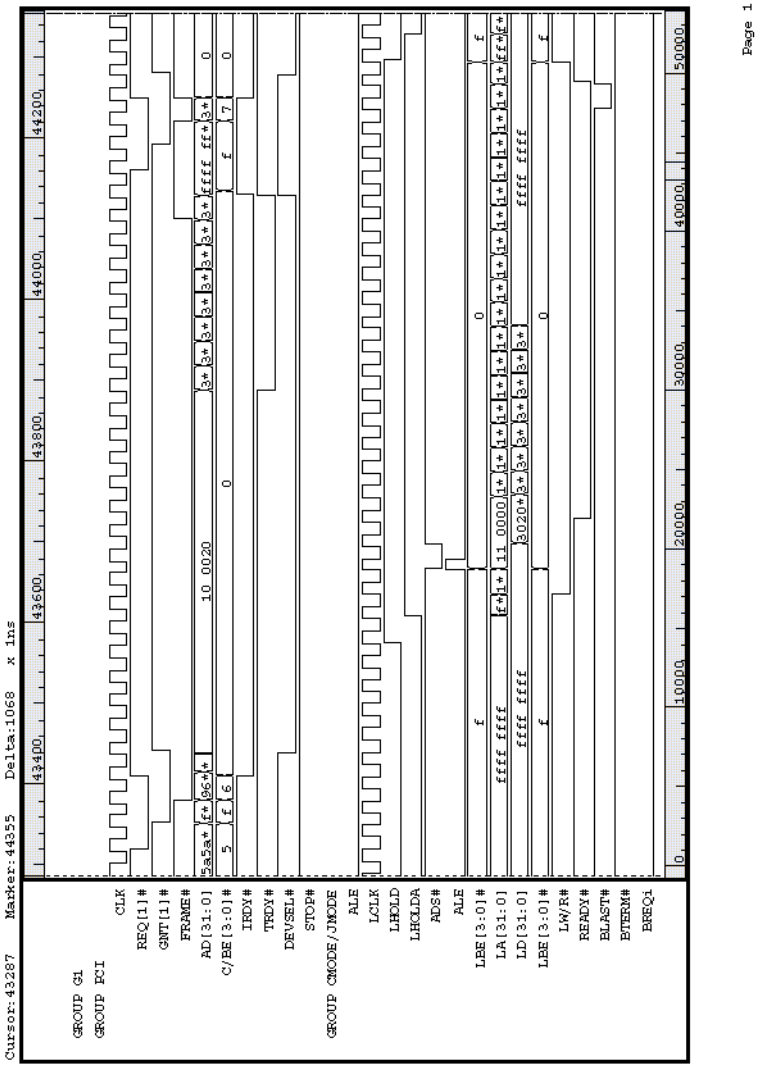


Timing Diagram 5-80. DMA Channel 1 PCI-to-Local (Memory Read Command)

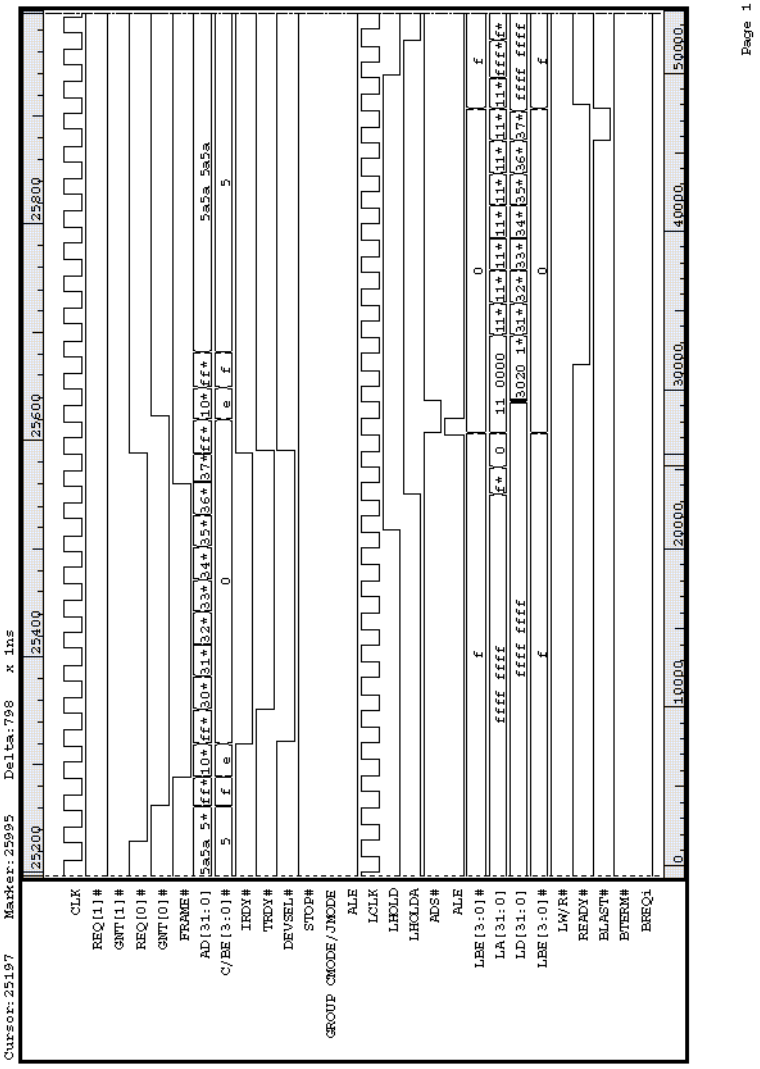
5.6.4 J Mode Only DMA Timing Diagrams



Timing Diagram 5-81. DMA Channel 0 Local-to-PCI (Memory Write Command)



Timing Diagram 5-82. DMA Channel 0 PCI-to-Local (Memory Read Command)



Timing Diagram 5-83. DMA Channel 0 PCI-to-Local (Memory Read Line Command)

6 PCI/LOCAL INTERRUPTS AND USER I/O

6.1 INTERRUPTS

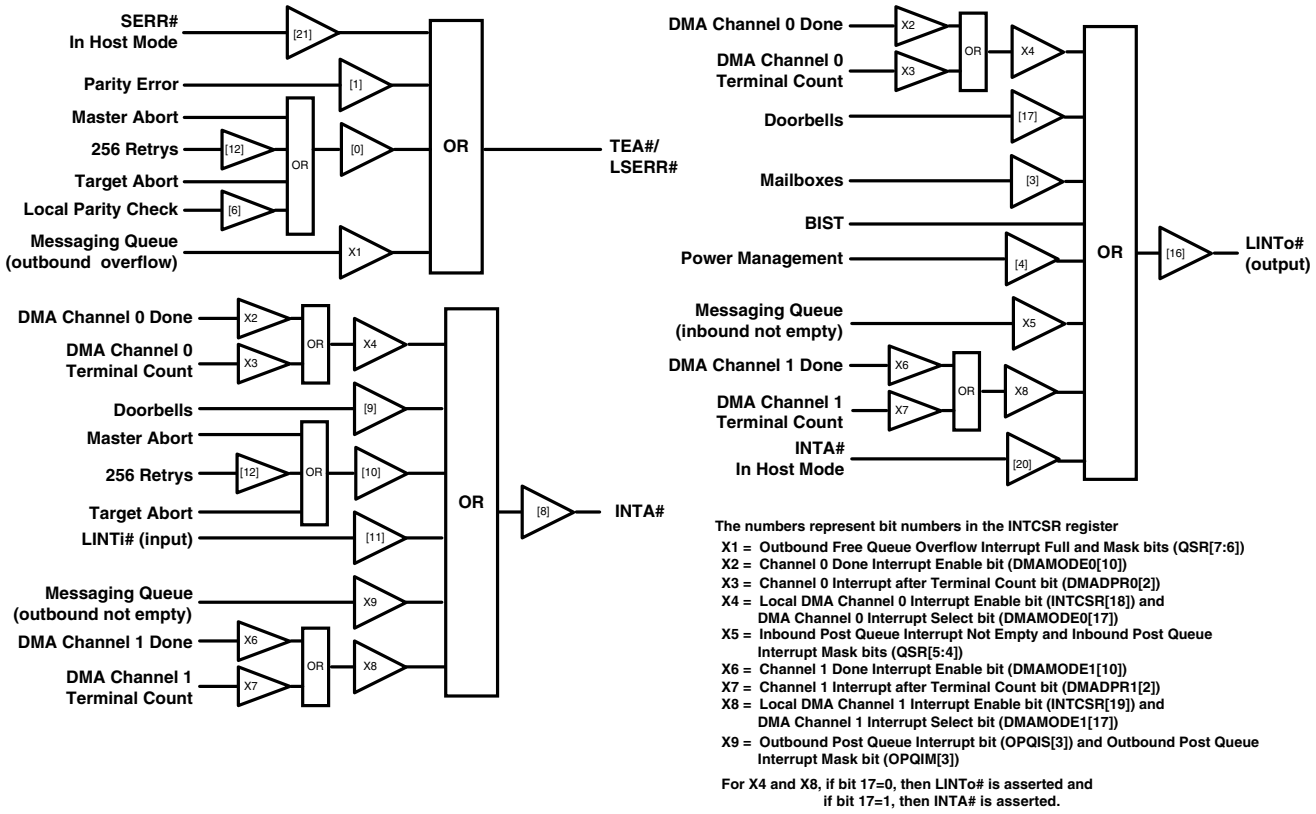


Figure 6-1. Interrupt and Error Sources

6.1.1 PCI Interrupts (INTA#)

In Adapter mode, a PCI 9056 PCI Interrupt (INTA#) can be asserted by one of the following:

- Local-to-PCI Doorbell register
- Local Interrupt input
- Master/Target Abort Status condition
- DMA Channel 0 and Channel 1 Done
- DMA Channel 0 and Channel 1 Terminal Count is reached
- Messaging Outbound Post Queue not empty
- 256 consecutive PCI Retries

INTA#, or individual sources of an interrupt, can be enabled or disabled with the PCI 9056 Interrupt Control/Status register (INTCSR). This register also provides the interrupt status of each interrupt source.

The PCI 9056 PCI Bus interrupt is a level output. Disabling an interrupt enable bit or clearing the cause of the interrupt can clear an interrupt.

6.1.2 Local Interrupt Input (LINTi#)

The Local Interrupt Input Enable bit must be enabled (INTCSR[11]=1) for interrupts to be acknowledged by the PCI 9056.

Asserting the Local Bus input LINTi# can assert a PCI Bus interrupt. The PCI Host processor can read the PCI 9056 Interrupt Control/Status register (INTCSR) to determine whether an interrupt is pending as a result of LINTi# being asserted (INTCSR[15]).

The interrupt remains asserted as long as LINTi# is asserted and the Local Interrupt input is enabled. The PCI Host processor can take adapter-specific action to cause the Local Bus to release LINTi#.

If the PCI Interrupt Enable bit is cleared (INTCSR[8]=0), the PCI interrupt (INTA#) is de-asserted; however, the Local interrupts (LINTi#) and the status bit remain active.

6.1.3 Local Interrupt Output (LINTo#)

The PCI 9056 Local Interrupt output (LINTo#) can be asserted by one of the following:

- PCI-to-Local Doorbell/Mailbox register access.
- PCI BIST interrupt.
- DMA Channel 0 and Channel 1 Done interrupt.
- DMA Channel 0 and Channel 1 Terminal Count is reached.
- DMA Abort Interrupt or Messaging Outbound Post Queue is not empty.
- PCI INTA# when the HOSTEN# signal is asserted. The PCI 9056 is a PCI Host.

LINTo#, or individual sources of an interrupt, can be enabled or disabled with the PCI 9056 Interrupt Control/Status register (INTCSR). This register also provides interrupt status for each interrupt source.

The PCI 9056 Local interrupt is a level output. Interrupts can be cleared by disabling the Interrupt Enable bit of a source or by clearing the cause of an interrupt.

6.1.4 Master/Target Abort Interrupt

The PCI 9056 sets the Received Master Abort bit or Target Abort bit (PCISR[13 or 11]=1, respectively) when it detects a Master or Target Abort. These status bits cause the PCI INTA# to be asserted if interrupts are enabled.

The interrupt remains set as long as the Receive Master Abort or Target Abort bits remain set and the Master/Target Abort interrupt is enabled. Use PCI Type 0 Configuration or Local accesses to clear the Received Master Abort and Target Abort interrupt bits (PCISR[13, 11]=0, respectively).

The Interrupt Control/Status Register bits (INTCSR[26:24]) are latched at the time of a Master or Target Abort interrupt. These bits provide information when an abort occurs, such as which device was the Master when the abort occurred.

The PCI Abort Address is stored in the PCI Abort Address register bits (PABTADR[31:0]).

6.1.5 Mailbox Registers

The PCI 9056 has eight 32-bit Mailbox registers that can be written to and read from both the PCI and Local Buses. These registers can be used to pass command and status information directly between the PCI and Local Bus devices.

A Local interrupt can be asserted, if enabled (INTCSR[3] and INTCSR[16]), when the PCI Host writes to one of the first four Mailbox registers (MBOX0, MBOX1, MBOX2, or MBOX3).

To clear the Mailbox registry, the destination bus should read the values currently in the Mailbox registers.

6.1.6 Doorbell Registers

The PCI 9056 has two 32-bit Doorbell Interrupt/Status registers. One is assigned to the PCI Bus interface. The other is assigned to the Local Bus interface.

A Local processor can assert a PCI Bus interrupt by writing any number other than all zeroes (0) to the Local-to-PCI Doorbell register bits (L2PDBELL[31:0]).

A PCI Host can assert a Local Bus interrupt by writing any number other than all zeroes (0) to the PCI-to-Local Doorbell register bits (P2LDBELL[31:0]). The PCI Interrupt and Local Interrupt remain asserted until all bits are cleared to zero (0).

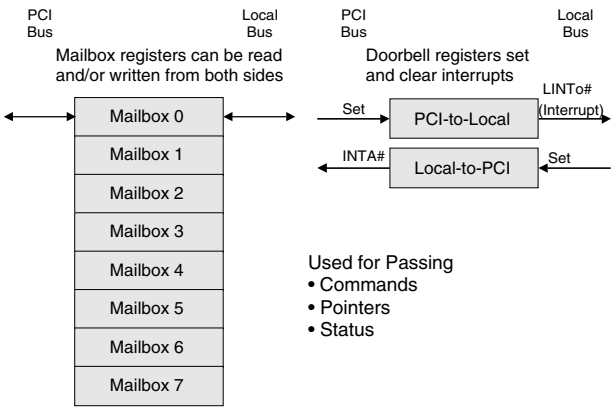


Figure 6-2. Mailbox and Doorbell Message Passing

6.1.6.1 Local-to-PCI Doorbell Interrupt

A Local Bus Master can assert a PCI Bus interrupt by writing to the Local-to-PCI Doorbell Register bit(s) (L2PDBELL[31:0]). The PCI Host processor can read the PCI Doorbell Interrupt Active bit to determine whether a PCI Doorbell interrupt is pending (INTCSR[13]), and if so, read the PCI 9056 Local-to-PCI Doorbell register.

Each bit in the Local-to-PCI Doorbell register is individually controlled. The Local Bus can only set bits in the Local-to-PCI Doorbell register. From Local Bus, writing 1 to any bit position sets that bit and writing 0 has no effect. Bits in the Local-to-PCI Doorbell register can only be cleared from the PCI Bus. From the PCI Bus, writing 1 to any bit position clears that bit and writing 0 has no effect.

Interrupts remain set as long as any Local-to-PCI Doorbell register bits are set and the PCI Doorbell Interrupt Enable bit (INTCSR[9]) is set.

6.1.6.1.1 M Mode Local-to-PCI Doorbell Interrupt

To prevent race conditions from occurring when the PCI Bus is accessing the Local-to-PCI Doorbell register (or any Configuration register), the PCI 9056 automatically de-asserts TA# output to prevent Local Bus configuration accesses.

6.1.6.1.2 C and J Modes Local-to-PCI Doorbell Interrupt

To prevent race conditions from occurring when the PCI Bus is accessing the Local-to-PCI Doorbell register (or any Configuration register), the PCI 9056 automatically de-asserts READY# output to prevent Local Bus configuration accesses.

6.1.6.2 PCI-to-Local Doorbell Interrupt

A PCI Bus Master can assert a Local Bus interrupt by writing to the PCI-to-Local Doorbell Register bits (P2LDBELL[31:0]). The Local processor can read the Local Doorbell Interrupt Active bit to determine whether a Local doorbell interrupt is pending (P2LDBELL[20]), and if so, read the PCI 9056 PCI-to-Local Doorbell register.

Each bit in the PCI-to-Local Doorbell register is individually controlled. The PCI Bus only sets bits in the PCI-to-Local Doorbell register. From the PCI Bus, writing 1 to any bit position sets that bit and writing 0 to a bit position has no effect. Bits in the PCI-to-Local Doorbell register can only be cleared from the Local Bus. From the Local Bus, writing 1 to any bit position clears that bit and writing 0 has no effect.

Note: *If the Local Bus cannot clear a Doorbell Interrupt, do not use the PCI-to-Local Doorbell register.*

Interrupts remain set as long as any PCI-to-Local Doorbell register bits are set and the Local Doorbell Interrupt Enable bit is set (INTCSR[17]=1).

To prevent race conditions when the Local Bus is accessing the PCI-to-Local Doorbell register (or any Configuration register), the PCI 9056 automatically issues a Retry to the PCI Bus.

6.1.7 Built-In Self Test Interrupt (BIST)

A PCI Bus Master can assert a Local Bus interrupt by performing a PCI Configuration write, which sets the PCI BIST Interrupt Enable bit (PCIBISTR[6]=1). A Local processor can read the BIST Interrupt Active bit (INTCSR[23]) to determine whether a BIST interrupt is pending.

Interrupts remain set as long as the bit is set and the PCI BIST Interrupt Enable bit is set (PCIBISTR[6]=1). The Local Bus then resets the bit when BIST completes. The PCI Host software may fail the device if the bit is not reset after two seconds.

Note: *The PCI 9056 does not have an internal BIST.*

6.1.8 DMA Channel 0 and Channel 1 Interrupts

A DMA channel can assert a PCI Bus or Local Bus interrupt when done (transfer complete) or after a transfer is complete for the current descriptor in Scatter/Gather DMA mode. The DMA Channel Interrupt Select bit(s) determine whether to assert a PCI (DMAMODE0[17]=1 and/or DMAMODE1[17]=1) or Local (DMAMODE0[17]=0 and/or DMAMODE1[17]=0) interrupt. The Local or PCI processor can read the DMA Channel Interrupt Active bit(s) to determine whether a DMA Channel 0 (INTCSR[21]) or DMA Channel 1 (INTCSR[22]) interrupt is pending.

The Channel Done bit(s) (DMACSR0[4] and/or DMACSR1[4]) can be used to determine whether an interrupt is one of the following:

- DMA Done interrupt
- Transfer complete for current descriptor interrupt

The Done Interrupt Enable bit(s) (DMAMODE0[10] and/or DMAMODE1[10]) enable a Done interrupt. In Scatter/Gather DMA mode, a bit in the Next Descriptor Pointer register of the channel (loaded from Local memory) specifies whether to assert an interrupt at the end of the transfer for the current descriptor.

A DMA Channel interrupt is cleared by the Channel Clear Interrupt bit(s) (DMACSR0[3]=1 and/or DMACSR1[3]=1).

6.1.9 All Modes PCI SERR# (PCI NMI)

The PCI 9056 asserts an SERR# pulse if parity checking is enabled (PCICR[6]=1) and it detects an address or 1 is written to the Generate PCI Bus SERR# Interrupt bit (INTCSR[2]) with a current value of 0.

SERR# output can be enabled or disabled with the SERR# Enable bit (PCICR[8]).

6.1.10 M Mode PCI SERR#

The PCI 9056 also asserts SERR# if the Local Bus responds with TEA# to the PCI 9056. The TEA# Input Interrupt Mask bit (LMISC1[5]) masks out the SERR# interrupt assertion process.

6.1.11 Local NMI

If the Parity Error Response bit is set (PCICR[6]=1), the PCI 9056 sets the Master Data Parity Error Detected bit (PCISR[8]=1) when the following three conditions are met:

- The PCI 9056 asserted PERR# or acknowledged PERR# was asserted
- The PCI 9056 was the Bus Master for the operation in which the error occurred
- The Parity Error Response bit is set (PCICR[6]=1)

The PCI 9056 sets the Detected Parity Error bit (PCISR[15]=1) if it detects one of the following conditions:

- The PCI 9056 detected a parity error during a PCI Address phase
- The PCI 9056 detected a data parity error when it is the target of a write
- The PCI 9056 detected a data parity error when performing Master Read operation

6.1.12 M Mode Local TEA# (Local NMI)

A TEA# interrupt is asserted if the following occurs:

- PCI Bus Target Abort bit is set (PCISR[11]=1) or Received Master Abort bit is set (PCISR[13]=1).
- Detected Parity Error bit is set (PCISR[15]=1).
- Direct Master Local Data Parity Check Error Status bit is set (INTCSR[7]=1).
- Messaging Outbound Free queue overflows.
- PCI SERR# when the HOSTEN# signal is asserted. The PCI 9056 is a PCI Host.

The Enable Local Bus TEA# bit (INTCSR[0]) can be used to enable or disable TEA# for an abort or parity error. TEA# is a level output that remains asserted as long as the Abort or Parity Error Status bits are set.

The PCI 9056 tolerates TEA# input assertion only during Direct Slave or DMA transactions. The PCI 9056 does not sample TEA# assertion during Direct Master transactions.

6.1.13 C and J Modes Local LSERR# (Local NMI)

An LSERR# interrupt is asserted if the following conditions occur:

- PCI Bus Target Abort bit is set (PCISR[11]=1) or Received Master Abort bit is set (PCISR[13]=1).
- Detected Parity Error bit is set (PCISR[15]=1).
- Direct Master Local Data Parity Check Error Status bit is set (INTCSR[7]=1).
- Messaging Outbound Free queue overflows.
- PCI SERR# when the HOSTEN# signal is asserted. The PCI 9056 is a PCI Host.

The Enable Local Bus LSERR# bit (INTCSR[0]) can be used to enable or disable LSERR# for an abort or parity error. LSERR# is a level output that remains asserted as long as the Abort or Parity Error Status bits are set.

6.2 USER INPUT AND OUTPUT

The PCI 9056 supports user input and output pins, USERi and USERo (B14 and C14, respectively). Both are multiplexed with other functional pins. The default PCI 9056 condition are the USERi and USERo functions. USERi is selected when CNTRL[18]=1, and USERo is selected when CNTRL[19]=1. User output data can be logged by writing to the General Purpose Output bit (CNTRL[16]). User input data can be read from the General Purpose Input bit (CNTRL[17]).

7 INTELLIGENT I/O (I_2O)

7.1 I_2O-COMPATIBLE MESSAGE UNIT

The I_2O-compatible Messaging Unit supplies two paths for messages, two inbound FIFOs to receive messages from the primary PCI Bus, and two outbound FIFOs to pass messages to the primary PCI Bus. Refer to *I_2O r1.5* for details.

Figure 7-1 and Figure 7-2 illustrate I_2O architecture.

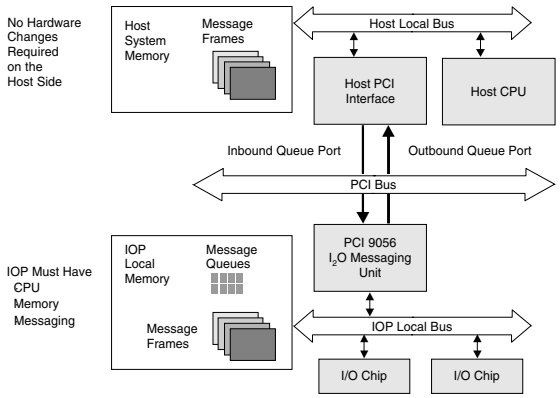


Figure 7-1. Typical I_2O Server/Adapter Card Design

Note: *IOP = I/O Processor.*

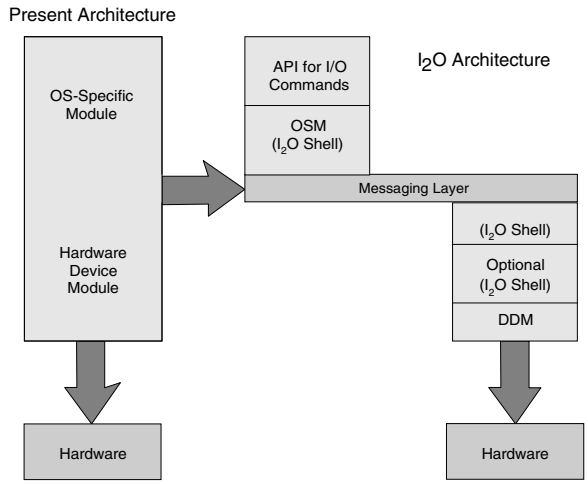


Figure 7-2. Driver Architecture Compared

7.1.1 Inbound Messages

Inbound messages reside in a pool of message frames (minimum 64-byte frames) allocated in the shared Local Bus (IOP) memory. The inbound message queue is comprised of a pair of rotating FIFOs implemented in Local memory. The Inbound Free List FIFO holds the message frame addresses (MFA) of available message frames in Local memory. The Inbound Post Queue FIFO holds the MFA of all currently posted messages.

External PCI agents, through the Inbound Queue Port location in PCI Address space, access inbound circular FIFOs. (Refer to Table 7-2 on page 7-6.) The Inbound Queue Port, when read by an external PCI agent, returns the Inbound Free List FIFO MFA. The external PCI agent places a message frame into the Inbound Post Queue FIFO by writing its MFA to the Inbound Queue Port location.

7.1.2 Outbound Messages

Outbound messages reside in a pool of message frames (minimum 64-byte frames) allocated in the shared PCI Bus (Host System) memory. The Outbound message queue is comprised of a pair of rotating FIFOs implemented in Local memory. The Outbound Free List FIFO holds the message frame addresses (MFA) of available message frames in system memory. The Outbound Post Queue FIFO holds the MFA of all currently posted messages.

External PCI agents, through the Outbound Queue Port location in PCI Address space access outbound circular FIFOs. (Refer to Table 7-2 on page 7-6.) The Outbound Queue Port, when read by an external PCI agent, returns the Outbound Post Queue FIFO MFA. The External PCI agent places free message frames into the Outbound Free List FIFO by writing the free MFA into the Outbound Queue Port location.

Memory for the circular FIFOs must be allocated in Local (IOP) memory. The base address of the queue is contained in the Queue Base Address bits (QBAR[31:20]). Each FIFO entry is a 32-bit data value. Each read and write of the queue must be a single 32-bit access.

Circular FIFOs range in size from 4- to 64-KB entries. All four FIFOs must be the same size and contiguous. Therefore, the total amount of Local memory needed for circular FIFOs ranges from 64 KB to 1 MB. A FIFO

size is specified in the Circular Queue Size bits (MQCR[5:1]).

The starting address of each FIFO is based on the Queue Base Address and the FIFO Size, as listed in Table 7-1.

Table 7-1. Queue Starting Address

FIFO	Starting Address
Inbound Free List	QBAR
Inbound Post List	QBAR + (1 * FIFO Size)
Outbound Post List	QBAR + (2 * FIFO Size)
Outbound Free List	QBAR + (3 * FIFO Size)

7.1.3 I_2O Pointer Management

The FIFOs always reside in shared Local (IOP) memory and are allocated and initialized by the IOP. Before setting the Queue Enable bit (MQCR[0]=1), the Local processor must initialize the following registers, with the initial offset according to the configured FIFO size:

- Inbound Post and Free Head Pointer registers (IPHPR and IFHPR)
- Inbound Post and Free Tail Pointer registers (IPTPR and IFTPR)
- Outbound Post and Free Head Pointer registers (OPHPR and OFHPR)
- Outbound Post and Free Tail Pointer registers (OPTPR and OFTPR)

The Messaging Unit automatically adds the Queue Base Address to offset in each head and tail pointer register. The software can then enable I_2O. After initialization, the Local software should not write to the pointers managed by the MU hardware.

Empty flags are set if the queues are disabled (MQCR[0]=0) or head and tail pointers are equal. This occurs independent of how the head and tail pointers are set.

An empty flag is cleared, signifying not empty, only if the queues are enabled and pointers become not equal.

If an empty flag is cleared and the queues are enabled, the empty flag is set only if the tail pointer is incremented and the head and tail pointers become equal.

Full flags are always cleared when the queues are disabled or the head and tail pointers are not equal.

A full flag is set when the queues are enabled, the head pointer is incremented, and the head and tail pointers become equal.

Each circular FIFO has a head pointer and a tail pointer, which are offsets from the Queue Base Address. (Refer to Table 7-2 on page 7-6.) Writes to a FIFO occur at the head of the FIFO and reads occur from the tail. Head and tail pointers are incremented by either the Local processor or the MU hardware. The unit that writes to the FIFO also maintains the pointer. Pointers are incremented after a FIFO access. Both pointers wrap around to the first address of the circular FIFO when they reach the FIFO size, so that the head and tail pointers continuously “chase” each other around in the circular FIFO. The MU wraps the pointers automatically for the pointers that it maintains. IOP software must wrap the pointers that it maintains. Whenever they are equal, the FIFO is empty. To prevent overflow conditions, I_2O specifies that the number of message frames allocated should be less than or equal to the number of entries in a FIFO. (Refer to Figure 7-3.)

Each inbound MFA is specified by I_2O as the offset from the start of shared Local (IOP) memory region 0 to the start of the message frame. Each outbound MFA is specified as the offset from Host memory location 0x00000000h to the start of the message frame in shared Host memory. Because the MFA is an actual address, the message frames need not be contiguous. IOP allocates and initializes inbound message frames in shared IOP memory using any suitable memory allocation technique. Host allocates and initializes outbound message frames in shared Host memory using any suitable memory allocation technique. Message frames are a minimum of 64 bytes in length.

I_2O uses a “push” (write-preferred) memory model. That means the IOP writes messages and data to the shared Host memory, and the Host writes messages and data to shared IOP memory. Software should make use of Burst and DMA transfers whenever possible to ensure efficient use of the PCI Bus for message passing.

Additional information on message passing implementation may be found in *I_2O r1.5*.

7.1.4 Inbound Free List FIFO

The Local processor allocates inbound message frames in its shared memory and can place the address of a free (available) message frame into the Inbound Free List FIFO by writing its MFA into the FIFO location pointed to by the Queue Base register + Inbound Free Head Pointer register. The Local processor must then increment the Inbound Free Head Pointer register.

A PCI Master (Host or other IOP) can obtain the MFA of a free message frame by reading the Inbound Queue Port Address (40h of the first PCI Memory Base Address register). If the FIFO is empty (no free inbound message frames are currently available, head and tail pointers are equal), the MU returns -1 (FFFFFFFFh). If the FIFO is not empty (head and tail pointers are not equal), the MU reads the MFA pointed to by the Queue Base register + Inbound Free Tail Pointer register, returns its value and increments the Inbound Free Tail Pointer register. If the Inbound Free Queue is not empty, and the Inbound Free Queue Prefetch Enable bit is set (QSR[3]=1), the next entry in the FIFO is read from the Local Bus into a prefetch register. The prefetch register then provides the data for the next PCI read from this queue, thus reducing the number of PCI wait states. (Refer to Figure 7-3.)

7.1.5 Inbound Post Queue FIFO

A PCI Master (Host or other IOP) can write a message into an available message frame in the shared Local (IOP) memory. It can then post that message by writing the Message Frame Address (MFA) to the Inbound Queue Port Address, IQP (40h of the first PCI Memory Base Address register). When the port is written, the MU writes the MFA to the Inbound Post Queue FIFO location pointed to by the Queue Base register + FIFO Size + Inbound Post Head Pointer register. After the MU writes the MFA to the Inbound Post Queue FIFO, it increments the Inbound Post Head Pointer register.

The Inbound Post Tail Pointer register points to the Inbound Post Queue FIFO location, which holds the MFA of the oldest posted message. The Local processor maintains the tail pointer. After a Local processor reads the oldest MFA, it can remove the MFA from the Inbound Post Queue FIFO by incrementing the Inbound Post Tail Pointer register.

The PCI 9056 asserts a Local Interrupt when the Inbound Post Queue FIFO is not empty. The Inbound Post Queue FIFO Interrupt bit in the Queue Status/ Control register (QSR[5]) indicates the interrupt status. The interrupt clears when the Inbound Post Queue FIFO is empty. The Inbound Post Queue FIFO Interrupt Mask bit (QSR[4]) can mask the interrupt.

To prevent racing between the time the PCI Write transaction is received until the data is written in Local memory and the Inbound Post Head Pointer register is incremented, any Direct Slave access to the PCI 9056 is issued a Retry.

7.1.6 Outbound Post Queue FIFO

A Local Master (IOP) can write a message into an available message frame in shared Host memory. It can then post that message by writing the Message Frame Address (MFA) to the Outbound Post Queue FIFO location pointed to by the Queue Base register + Outbound Post Head Pointer register + (2 * FIFO Size). The Local processor should then increment the Outbound Post Head Pointer register.

A PCI Master can obtain the MFA of the oldest posted message by reading the Outbound Queue Port Address (44h of the first PCI Memory Base Address register). If the FIFO is empty (no further outbound messages are posted, head and tail pointers are equal), the MU returns -1 (FFFFFFFFh). If the Outbound Post Queue FIFO is not empty (head and tail pointers are not equal), the MU reads the MFA pointed to by the Queue Base register + (2 * FIFO Size) + outbound Post Tail Pointer register, returns its value and increments the Outbound Post Tail Pointer register.

The PCI 9056 asserts a PCI Interrupt when the Outbound Post Head Pointer register is not equal to the Outbound Post Tail Pointer register. The Outbound Post Queue FIFO Interrupt bit of the Outbound Post Queue Interrupt Status register (OPQIS) indicates the interrupt status. When the pointers become equal, both the interrupt and the Outbound Post Queue FIFO interrupt bit are automatically cleared. Pointers become equal when a PCI Master (Host or other IOP) reads sufficient FIFO entries to empty the FIFO. The Outbound Post Queue FIFO Interrupt Mask register (OPLFIM) can mask the Interrupt.

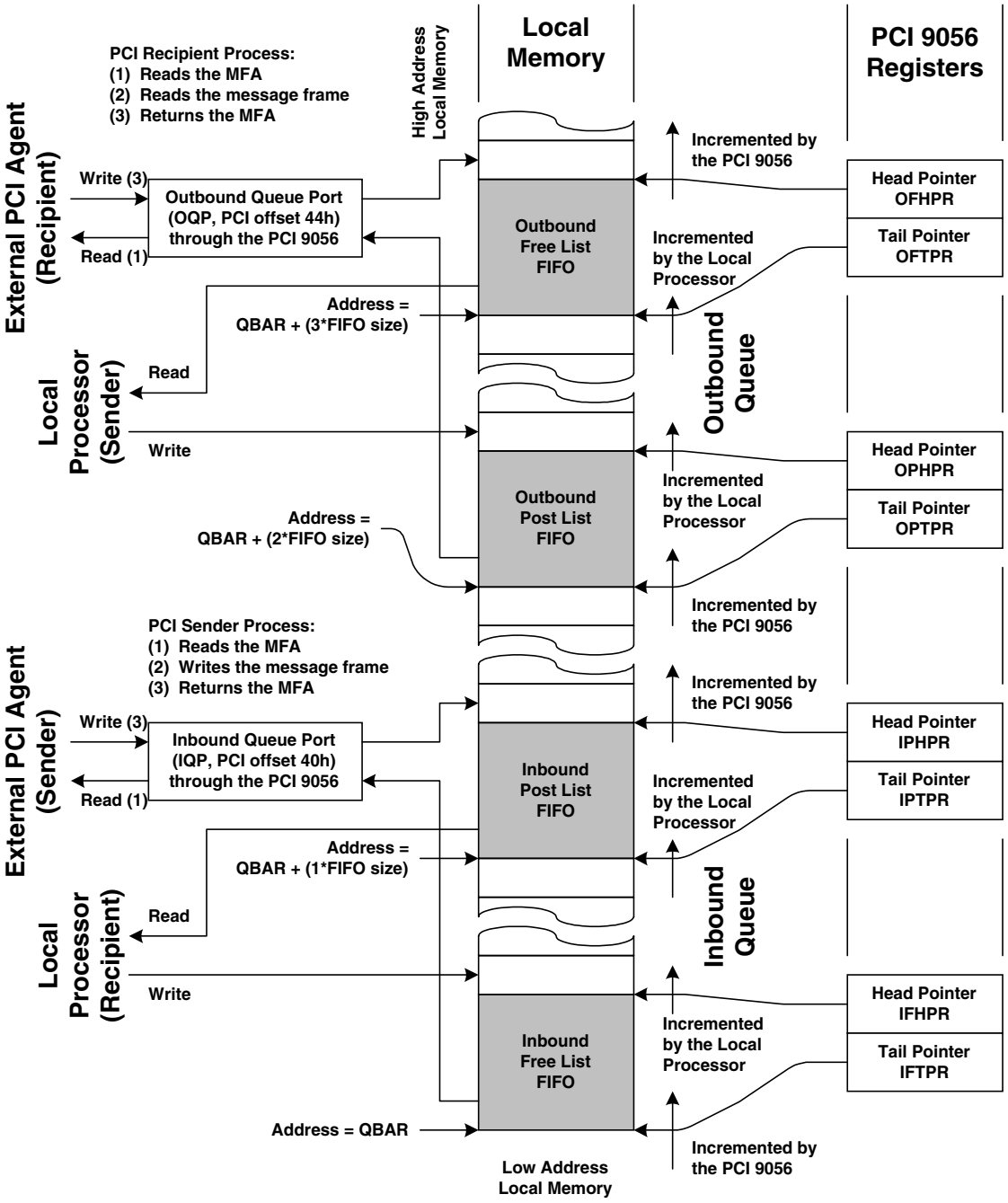


Figure 7-3. I_2O Circular FIFO Operation

7.1.7 Outbound Post Queue

To reduce read latency, prefetching from the tail of the queue occurs whenever the queue is not empty and the tail pointer is incremented (queue has been read from), or when the queue is empty and the head pointer is incremented (queue has been written to). When the Host CPU reads the Outbound Post Queue, the data is immediately available.

7.1.8 Inbound Free Queue

To reduce read latency, prefetching from the tail of the queue occurs whenever the queue is not empty and the tail pointer is incremented (queue has been read

from), or when the queue is empty and the head pointer is incremented (queue has been written to). When the Host CPU reads the Inbound Free Queue, the data is immediately available.

7.1.9 Outbound Free List FIFO

The PCI Bus Master (Host or other IOP) allocates outbound message frames in its shared memory. The PCI Bus Master can place the address of a free (available) message frame into the Outbound Free List FIFO by writing a Message Frame Address (MFA) to the Outbound Queue Port Address (44h of the first PCI Memory Base Address register). When the port is written, the MU writes the MFA to the Outbound Free List FIFO location pointed to by the Queue Base register + (3 * FIFO Size) + Outbound Free Head Pointer register. After the MU writes the MFA to the Outbound Free List FIFO, it increments the Outbound Free Head Pointer register.

When the IOP needs a free outbound message frame, it must first check whether any free frames are available. If the Outbound Free List FIFO is empty (outbound free head and tail pointers are equal), the IOP must wait for the Host to place additional outbound free message frames in the Outbound Free List FIFO. If the Outbound Free List FIFO is not empty (head and tail pointers are not equal), the IOP can obtain the MFA of the oldest free outbound message frame by reading the location pointed to by the Queue Base register + (3 * FIFO Size) + Outbound Free Tail Pointer register. After the IOP reads the MFA, it must increment the Outbound Free Tail Pointer register. To prevent overflow conditions, I_2O specifies the number of message frames allocated should be less than or equal to the number of entries in a FIFO. The MU also checks for overflows of the Outbound Free List FIFO. When the head pointer is incremented and becomes equal to the tail pointer, the Outbound Free List FIFO is full, and the MU asserts a LINTo# interrupt. The interrupt is recorded in the Queue Status/Control register (QSR).

From the time the PCI Write transaction is received until the data is written into Local memory and the Outbound Free Head Pointer register is incremented, any Direct Slave access to the PCI 9056 is issued a Retry.

7.1.10 I_2O Enable Sequence

To enable I_2O, the Local processor performs the following:

- Initialize Space 1 address and range
- Initialize all FIFOs and Message Frame memory
- Set the PCI Base Class Code bits (PCICCR[23:16]) to be an I_2O device with Programming Interface 01h
- Set the I_2O Decode Enable bit (QSR[0])
- Set Local Init Status bit to “done” (LMISC1[2]=1)

Note: *The serial EEPROM must not set the Local Init Status bit so that the PCI 9056 issues Retrys to all PCI accesses until the Local Init Status bit is set to “done” by the Local processor.*

The I_2O Decode Enable bit (QSR[0]) causes remapping of resources for use in I_2O mode. When set, all Memory-Mapped Configuration registers (*for example*, queue ports 40h and 44h) and Space 1 share the PCIBAR0 register. PCI accesses to offset 00h–FFh of PCIBAR0 result in accesses to the PCI 9056 Internal Configuration registers.

Accesses above offset FFh of PCIBAR0 result in Local Space accesses, beginning at offset 100h from the Remap PCI Address to Local Address Space 1 into the Local Address Space bits (LAS1BA[31:4]). Therefore, space located at offset 00h-FFh from LAS1BA is not addressable from the PCI Bus using PCIBAR0.

Note: *Because PCI accesses to offset 00h-FFh of PCIBAR0 result in internal configuration accesses, the Inbound Free MFAs must be greater than FFh.*

Table 7-2. Circular FIFO Summary

FIFO Name	PCI Port	Generate PCI Interrupt	Generate Local Interrupt	Head Pointer Maintained By	Tail Pointer Maintained By
Inbound Free List FIFO	Inbound Queue Port (Host read)	No	No	Local processor	MU hardware
Inbound Post List FIFO	Inbound Queue Port (Host write)	No	Yes, when Port is written	MU hardware	Local processor
Outbound Post List FIFO	Outbound Queue Port (Host read)	Yes, when FIFO is not empty	No	Local processor	MU hardware
Outbound Free List FIFO	Outbound Queue Port (Host write)	No	Yes, (LINTo#) when FIFO is full	MU hardware	Local processor

8 PCI POWER MANAGEMENT

8.1 OVERVIEW

The *PCI Power Mgmt. r1.1*, provides a standard mechanism for operating systems to control add-in cards for power management. *PCI Power Mgmt. r1.1* defines four PCI functional power states—D_0, D_1, D_2, and D_3. States D_0 and D_3 are required, while states D_1 and D_2 are optional. State D_0 represents the highest power consumption and state D_3 the least.

- **D_0 (Uninitialized)**—Enters this state from Power-On Reset or from state D_{3hot}. Supports Direct Slave PCI transactions only.
- **D_0 (Active)**—All functions active.
- **D_1**—Uses less power than State D_0, and more than state D_2. Light sleep state.
- **D_2**—Uses very little power.

  The functional states are defined by the allowed activities of the add-in card with the PCI 9056.

  The function supports PCI Configuration cycles to function if clock is running (Memory, I/O, Bus Mastering, and Interrupts are disabled). It also supports the Wakeup Event from function, but not standard PCI interrupts.
- **D_{3hot}**—Uses lower power than any other state. Supports PCI Configuration cycles to function if clock is running. Supports Wakeup Event from function, but not standard PCI interrupts. When programmed for state D_0, an internal soft reset occurs. The PCI Bus drivers must be disabled. PME# context must be retained during this soft reset.
- **D_{3cold}**—No power. Supports Bus reset only. All context is lost in this state.

From a power management perspective, the PCI Bus can be characterized at any point in time by one of four power management states—B_0, B_1, B_2, and B_3:

- **B_0 (Fully On)**—Bus is fully usable with full power and clock frequency, *PCI r2.2*-compliant. Fully operational bus activity. This is the only Power Management state in which data transactions can occur.
- **B_1**—Intermediate power management state. Full power with clock frequency. PME Event driven bus activity. V_{CC} is applied to all devices on the bus, and no transactions are allowed to occur on the bus.
- **B_2**—Intermediate power management state. Full power clock frequency stopped (in the low state). PME Event-driven bus activity. V_{CC} is applied to all devices on the bus; however, the clock is stopped and held in the Low state.
- **B_3 (Off)**—Power to the bus is switched off. PME Event-driven bus activity. V_{CC} is removed from all devices on the PCI Bus.

All system PCI Buses have an originating device, which can support one or more power states. In most cases, this creates a bridge (such as a Host-to-PCI Bus or a PCI-to-PCI bridge).

Function States must be at the same or lower energy state than the bus on which they reside.

8.1.1 PCI Power Management Functional Description

The PCI 9056 passes power management information and has no inherent power-saving feature.

The PCI Status register (PCISR) and the New Capability Pointer register (CAP_PTR) indicate whether a new capability (the Power Management function) is available. The New Capability Functions Support bit (PCISR[4]) enables a PCI BIOS to identify a New Capability function support. This bit is executable for writes from the Local Bus, and reads from both the Local and PCI Buses. CAP_PTR provides an offset into PCI Configuration Space, the start location of the first item in a New Capabilities Linked List.

The Power Management Capability ID register (PMCAPID) specifies the Power Management Capability ID, 01h, assigned by the PCI SIG. The Power Management Next Capability Pointer register (PMNEXT) points to the first location of the next item in the capabilities linked list. If Power Management is the last item in the list, then this register should be set to 0. The default value for the PCI 9056 is 48h (Hot Swap).

For the PCI 9056 to change the power state and assert PME#, a Local or PCI Host should set the PME_En bit (PMCSR[8]=1). The Local Host then determines to which power state the backplane should change by reading the Power State bits (PMCSR[1:0]).

The Local Host sets up the following:

- D_2_Support and D_1_Support bits (PMC[10:9]) are used by the Local Host to identify power state support
- PME_Support bits (PMC[14:11]) are used by the PCI 9056 to identify the PME# Support correspondent to a specific power state (PMCSR[1:0])

The Local Host then sets the PME_Status bit (PMCSR[15]=1) and the PCI 9056 asserts PME#. To clear the PME_Status bit, the PCI Host must write a 1 to the PME# Status bit (PMCSR[15]=1). To disable the PME# Interrupt signal, either Host can write a 0 to the PME_En bit (PMCSR[8]=0).

LINTo# is asserted every time the power state in the PMCSR register changes. Transmission from state 11 (D_{3hot}) to state 00 (D_0) causes a soft reset. A soft reset should only be initiated from the PCI Bus because the Local Bus interface is reset during a soft reset. The PCI 9056 issues LRESET# and resets all its internal registers to their default values. In state D_{3hot}, PCI Memory and I/O accesses are disabled, as well as PCI interrupts, and only configuration is allowed. Before making LINTo# work, set the Power Management Interrupt Enable bit (INTCSR[4]=1), and clear the interrupt by setting the Power Management Interrupt bit (INTCSR[5]=1).

The Data_Scale bits (PMCSR[14:13]) indicate the scaling factor to use when interpreting the value of the Power Management Data bits (PMDATA[7:0]). The value and meaning of the bits depend upon the data value specified in the Data_Select bits (PMCSR[12:9]). The Data_Scale bit value is unique for each Data_Select bit. For Data_Select values from 8 to 15, the Data_Scale bits always return a zero (PMCSR[14:13]=0).

PMDATA provides operating data, such as power consumed or heat dissipation.

8.1.2 66 MHz PCI Clock Power Management D_2 Support

The PCI 9056 provides full support for the D_2 Power Management state at a 33 MHz PCI clock frequency. The *PCI r2.2*-compliant 66 MHz PCI clock frequency prohibits any change to the clock without the system reset (RST#) being asserted. (Refer to *PCI r2.2* and *PCI Power Mgmt. r1.1*.) Therefore, the PCI 9056 cannot support the D_2 Power Management state at 66 MHz. To do that, the PCI 9056 requires an external control to avoid enabling the D_2 Power Management feature at a 66 MHz clock frequency. Default booting of the PCI 9056 sets D_2 support to a disabled state. All 66 MHz add-in cards, capable of running at a 66 MHz PCI clock frequency, must monitor the M66EN# PCI connector pin. When this pin is present on a card, the Local Processor can monitor the pin, and enable D_2 Power Management support (PMC[10]) by way of the register access whenever the M66EN# PCI connector pin is sampled false.

8.1.3 Power Management D_{3cold} Support

The PCI 9056 provides full support for the D_{3cold} Power Management state with PME# assertion and register contents storage. The PCI 9056 has all pins required by the *PCI Power Mgmt. r1.1*. Special attention is necessary for the following pins:

- **2.5V$_{AUX}$**—Power input pin routed to the D_{3cold} support core logic. Due to unavailable power from the PCI slot, 2.5V power must be supplied by an external power source, voltage regulator.
- **Card_V$_{AUX}$**—3.3V$_{AUX}$ power input pin driven by the PCI backplane through add-in card Auxiliary Power Routing. (Refer to *PCI Power Mgmt. r1.1*, Figure 12.)
- **PRESENT_DET**—Present Detect pin provided by add-in card Auxiliary Power Routing (refer to *PCI Power Mgmt. r1.1,* Figure 12) to enable the D_{3cold} PME# assertion feature within the PCI 9056 silicon.

- **PME#**—Optional open drain, active low signal intended to be driven low by the PCI 9056 to request a change in its current power management state and/or to indicate that a PME# has occurred. The PCI 9056 requires external logic to avoid unexpected Wake-Up events to occur whenever an add-in card is plugged into the *PCI r2.2*-compliant PCI backplane. (Refer to *PCI Power Mgmt. r1.1,* Chapter 7.)
- **PMEREQ#**—Input signal used to request a wake-up event only when the add-in card is in the D_{3cold} Power Management state.
- **IDDQEN#**—Input signal providing main power status to the PCI 9056 D_{3cold} Power Management logic.

Note: *All signal I/Os used for D_{3cold} Power Management support are powered by Card_V_{AUX} power.*

8.1.4 System Changes Power Mode Example

1. The Host writes to the PCI 9056 PMCSR register to change the power states.
2. The PCI 9056 sends a local interrupt (LINTo#) to a Local CPU (LCPU).
3. The LCPU has 200 µs to read the power management information from the PCI 9056 PMCSR register to implement the power-saving function.
4. After the LCPU implements the power saving function, the PCI 9056 disables all Direct Slave accesses and PCI Interrupt output (INTA#). In addition, the BIOS disables the PCI 9056 Master Access Enable bit (PCICR[2]).

Notes: *In Power-Saving mode, all PCI and Local Configuration cycles are granted.*

The PCI 9056 automatically performs a soft reset to a Local Bus on D_3-to-D_0 transitions.

8.1.5 Non-D_{3cold} Wake-Up Request Example

1. The add-in card (with a PCI 9056 chip installed) is in a powered-down state.
2. The Local CPU performs a write to the PCI 9056 PMCSR register to request a wake-up procedure.
3. As soon as the request is detected, the PCI 9056 drives PME# out to the PCI Bus.
4. The PCI Host accesses the PCI 9056 PMCSR register to disable the PME# output signal and restores the PCI 9056 to the D_0 power state.
5. The PCI 9056 completes the power management task by issuing the Local interrupt (LINTo#) to the Local CPU, indicating that the power mode has changed.

9 COMPACTPCI HOT SWAP

The PCI 9056 is compliant with the *PICMG 2.1, R2.0* requirements for Hot Swap Silicon, including support for Programming Interface (PI = 0), and BIAS Voltage, Early Power, and Initially Not Respond Support.

9.1 OVERVIEW

Hot Swap is used for many CompactPCI applications. Hot Swap functionality allows the orderly insertion and removal of boards without adversely affecting system operation. This is done for repair of faulty boards or system reconfiguration. Additionally, Hot Swap provides access to Hot Swap services, allowing system reconfiguration and fault recovery to occur with no system down time and minimum operator interaction. Adapter insertion/removal logic control resides on the individual adapters. The PCI 9056 uses four pins—BD_SEL#, CPCISW, ENUM# and LEDon#—to implement the hardware aspects of Hot Swap functionality. The PCI 9056 uses the Hot Swap Capabilities register to implement the software aspects of Hot Swap.

The PCI 9056 supports the following features specified in the *PICMG 2.1, R2.0* requirements for Hot Swap Silicon:

- *PICMG 2.1, R2.0* compliance
- Tolerate V_{CC} from early power
- Tolerate asynchronous reset
- Tolerate precharge voltage
- I/O Buffers must meet modified V/I requirements
- Limited I/O pin leakage at precharge voltage
- **Incorporates Hot Swap Control/Status register (HS_CSR)**—Contained within the configuration space.
- **Incorporates an Extended Capability Pointer (ECP) mechanism**—It is required that Software retain a standard method of determining whether a specific function is designed in accordance with *PICMG 2.1, R2.0*. The Capabilities Pointer is located within standard CSR space, using a bit in the PCI Status register (offset 04h).
- Incorporates remaining software connection control resources. Provides ENUM#, Hot Swap switch, and the blue LED.
- Early Power Support.
- **Incorporates a 1V BIAS precharge voltage to the PCI I/O pins**—All PCI Bus signals are required to be precharged to a 1V BIAS through a 10K ohm resistor during the Hot Swap process. The PCI 9056 provides an internal voltage regulator to supply 1V, with a built-in 10K ohm resistor, to all required PCI I/O buffers. Other PCI signals can be precharged to V_{IO}.

9.1.1 Silicon Behavior during Initialization on PCI Bus

The PCI 9056 supports an Initialization-time PCI option, which may be utilized by CompactPCI peripheral adapter cards designed for live insertion. Section 3.1.10 of the *PICMG 2.1, R2.0* states, "it is far preferable for boards that are not ready for PCI accesses to Initially Not Respond." The PCI 9056 Initialization-time PCI option allows configuration of the PCI 9056 to provide this preferable Initially Not Respond behavior. This option is detailed in Section 2.4.1.2, "Local Initialization," and Section 4.4.1.2, "Local Initialization."

The PCI 9056 supports the *PICMG 2.1, R2.0* Programming Interface 0 (PI = 0), as detailed in Figures 29 and 30 of that document. All required register bits and supporting control functionality are included in the CompactPCI Hot Swap Control and Status register (HS_CSR).

9.2 CONTROLLING CONNECTION PROCESSES

The following sections are excerpted from *PICMG 2.1, R2.0*, and modified, as appropriate, for the PCI 9056. (Refer to *PICMG 2.1, R2.0* for more details.)

9.2.1 Connection Control

Hardware Control provides a means for the platform to control the hardware connection process. The signals listed in the following sections must be supported on all Hot Swap boards for interoperability. Implementations on different platforms may vary.

9.2.1.1 Board Slot Control

BD_SEL#, one of the shortest pins from the CompactPCI backplane, is driven low to enable power-on. For systems not implementing hardware control, it is grounded on the backplane.

Systems implementing hardware control radially connect BD_SEL# to a Hot Swap Controller (HSC). The controller terminates the signal with a weak pull-down resistor, and can detect board present when the board pull-up resistor overrides the pull-down resistor. HSC can then control the power-on process by driving BD_SEL# low.

The PCI 9056 uses the BD_SEL# signal to high-impedance all local output buffers during the insertion and extraction process. In addition, the PCI 9056 uses BD_SEL# as a qualifier to dynamically connect 1V and V_{IO} BIAS precharge resistors to all required PCI I/O buffers. A pull-up resistor must be provided to the BD_SEL# pin or add-in card, where the pull-up resistor is connected to an early power Power Supply, which provides for proper PCI 9056 operation. (Refer to Section 12, “Pin Description,” for precharge connections.)

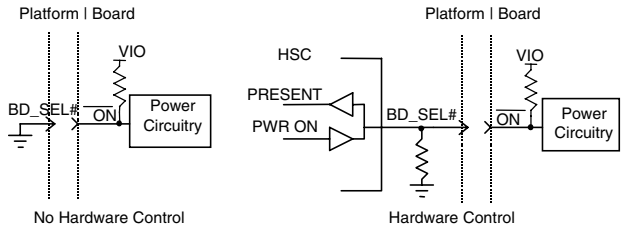


Figure 9-1. Redirection of BD_SEL#

9.2.1.2 Board Healthy

A second radial signal is used to acknowledge board health. It signals that a board is suitable to be released from reset and allowed onto the PCI Bus.

Minimally, this signal must be connected to the board’s power controller “power good” status line. Use of HEALTHY# can be expanded for applications requiring additional conditions to be met for the board to be considered healthy.

On platforms that do not use Hardware Connection Control, this line is not monitored. Platforms implementing this signaling, route these signals radially to a Hot Swap controller.

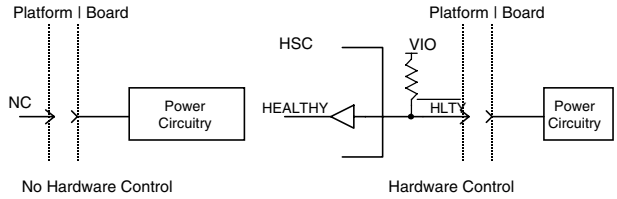


Figure 9-2. Board Healthy

9.2.1.3 Platform Reset

PCI Reset (RST#), as defined by *PICMG 2.1, R2.0*, is a bus signal on the backplane, driven by the Host. Platforms may implement this signal as a radial signal from the Hot Swap Controller to further control the electrical connection process. Platforms that maintain function of the bus signal, must *OR* the Host reset signal with the slot-specific signal.

Locally, boards must not exit reset until the H1 State is reached (healthy), and they must honor the backplane reset. The Local board reset (LRESET#) must be the logical *OR* of these two conditions. LRESET# is connected to the PCI 9056 RST# input pin.

During a BIAS voltage precharge and platform reset, in insertion and extraction procedures, all PCI I/O buffers must be in a high-impedance state. The PCI 9056 supports this condition when the Host RST# is asserted. To protect the Local board components from early power, the PCI 9056 floats the Local Bus I/Os. The BD_SEL# pin is used to perform the high-impedance condition on the Local Bus. With full contact of the add-in card to the backplane, BD_SEL# is asserted which ensures that the PCI 9056 asserts

the LRESET# signal to complete a Local board reset task.

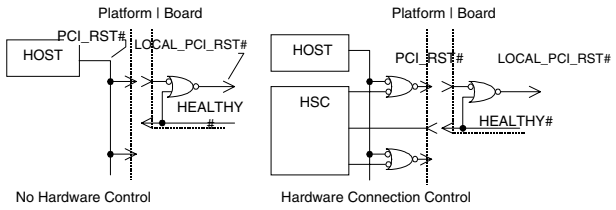


Figure 9-3. PCI Reset

9.2.2 Software Connection Control

Software Connection Control provides a means to control the Software Connection Process. Hot Swap board resources facilitate software Connection Control. Access to these resources occurs by way of the bus, using PCI protocol transfers (in-band).

These resources consist of four elements:

- ENUM# driven active indicates the need to change the Hot Swap Board state
- A switch, tied to the ejector, indicates the intent to remove a board
- LED indicates the software connection process status
- Control/Status register allows the software to interact with these resources

9.2.2.1 Ejector Switch and Blue LED

A microswitch (switch), located in the Hot Swap CompactPCI board card-ejector mechanism, is used to signal impending board removal. This signal asserts ENUM#. When the switch is activated, it is necessary to wait for the LED to light, indicating it is now okay to remove the board. The PCI 9056 implements separate control logic for the microswitch and Blue LED in two different pins (CPCISW and LEDon#, respectively).

When the ejector is opened or closed, the switch bounces for a time. The PCI 9056 uses internal debounce circuitry to clean the signal before the remainder of Hot Swap logic acknowledges it. The switch state is sampled six times, at 1 ms intervals, before it is determined to be closed or open.

The Blue “Status” LED, located on the front of the Hot Swap CompactPCI board, is illuminated when it is permissible to remove a board. The hardware connection layer provides protection for the system during all insertions and extractions. This LED indicates the system software is in a state that tolerates board extraction.

Upon insertion, the LED is automatically illuminated by the hardware until the hardware connection process completes. The LED remains *OFF* until the software uses it to indicate extraction is once again permitted.

The PCI 9056 uses an open-drain output pin to sink the external LED. The LED state is driven from the LED Software On/Off Switch bit (HS_CSR[3]).

The CPCISW input signal acknowledges the state ejector handle change to identify when a board is inserted or removed. The appropriate status bits are set (HS_CSR[7:6]).

9.2.2.2 ENUM#

ENUM# is provided to notify the Host CPU that a board was recently inserted or is about to be removed. This signal informs the CPU that system configuration changed, at which time the CPU performs necessary maintenance such as installing a device driver upon board insertion, or quiescing a device driver prior to board extraction.

ENUM# is an open collector bused signal with a pull-up resistor on the Host bus. It may drive an interrupt (preferred) or be polled by the system software at regular intervals. The CompactPCI Hot-Plug System Driver on the system Host manages the ENUM# sensing. Full Hot Swap Boards assert ENUM# until serviced by the Hot-Plug system driver.

When a board is inserted into the system and comes out of reset, the PCI 9056 acknowledges the ejector switch state. If this switch is open (ejector handle closed), the PCI 9056 asserts the ENUM# interrupt and sets the ENUM# Status Indicator for Board Insertion bit (HS_CSR[7]). Once the Host CPU installs the proper drivers, it can logically include this board by clearing the interrupt.

When a board is about to be removed, the PCI 9056 acknowledges the ejector handle is open, asserts the ENUM# interrupt, and sets the ENUM# Status Indicator for the Board Removal bit (HS_CSR[6]). The Host then logically removes the board and turns on the LED, at which time the board can be removed from the system.

9.2.2.3 Hot Swap Control/Status Register (HS_CSR)

The PCI 9056 supports Hot Swap directly, as a control/status register is provided in Configuration space. This register is accessed through the PCI Extended Capabilities Pointer (ECP) mechanism.

The Hot Swap Control/Status register (HS_CSR) provides status read-back for the Hot-Plug System Driver to determine which board is driving ENUM#. This register is also used to control the Hot Swap Status LED on the board front panel, and to de-assert ENUM#.

9.2.2.4 Hot Swap Capabilities Register

Register 9-1. Hot Swap Capabilities Register

31 24	23 16	15 8	7 0
Reserved	Control	Next_Cap Pointer	Hot Swap ID (06h)

Hot Swap ID. Bits [7:0] (HS_CNTL[7:0]; PCI:48h, LOC:188h). These bits are set to a default value of 0x06.

Next_Cap Pointer. Bits [15:8] (HS_NEXT[7:0]; PCI:49h, LOC:189h). These bits either point to the next New Capability structure, or are set to 0 if this is the last capability in the structure. Bits [9:8] are reserved and should be set to 0.

Control. Bits [23:16] (HS_CSR[7:0]; PCI:4Ah, LOC:18Ah). This 8-bit control register is defined in the following table.

Table 9-1. Hot Swap Control

Bit	Description
23	ENUM# status—Insertion (1 = board is inserted).
22	ENUM# status—Removal (1 = board is being removed).
21:20	Programming Interface (PI = 0).
19	LED state (1 = LED on, 0 = LED off).
18	Not used.
17	ENUM# interrupt enable (1 = de-assert, 0 = enable interrupt).
16	Not used.

10 PCI VITAL PRODUCT DATA (VPD)

10.1 OVERVIEW

The Vital Product Data (VPD) function in the *PCI r2.2* defines a new location and access method. It also defines the Read Only and Read/Write bits. Currently Device ID, Vendor ID, Revision ID, Class Code, Subsystem ID, and Subsystem Vendor ID are required in the Configuration Space Header and are required for basic identification of the device and device configuration. Though this information allows a device to be configured, it is not sufficient to allow a device to be uniquely identified. With the addition of VPD, optional information is provided that allows a device to be uniquely identified and tracked. These additional bits enable current and/or future support tools and reduces the total cost of ownership of PCs and systems.

This provides an alternate access method other than Expansion ROM for VPD. VPD is stored in an external serial EEPROM, which is accessed using the Configuration Space New Capabilities function.

The VPD registers—PVPDCNTL, PVPD_NEXT, PVPDAD, and PVPDATA—are not accessible for reads from the Local Bus. It is recommended that the VPD function be exercised only from the PCI Bus.

10.1.1 VPD Capabilities Register

VPD ID. Bits [7:0] (PVPDCNTL[7:0]; PCI:4Ch, LOC:18Ch). The PCI SIG assigns these bits a value of 03h by. The VPD ID is hardwired.

Next_Cap Pointer. Bits [15:8] (PVPD_NEXT[7:0]; PCI:4Dh, LOC:18Dh). These bits either point to the next New Capability structure, or are set to 0 if this is the last capability in the structure. Bits [9:8] are reserved and should be set to 0. The PCI 9056 defaults to 0x00. This value can be overwritten from the Local Bus.

VPD Address. Bits [24:16] (PVPDAD[14:0]; PCI:4Eh, LOC:18Eh). These bits specify the byte address of the VPD to be accessed. All accesses are 32-bits wide; bits [17:16] must be set to 0, with the maximum serial EEPROM size being 4K bits. Bits [30:25] are ignored.

F. Bit 31 (PVPDAD[15]; PCI:4Eh, LOC:18Eh). This bit sets a flag to indicate when a serial EEPROM data operation is complete. For Write cycles, the four bytes of data are first written into the VPD Data bits, after which the VPD Address is written at the same time the F flag is set to 1. The F flag clears when the serial EEPROM Data transfer completes. For Read cycles, the VPD Address is written at the same time the F flag is cleared to 0. The F flag is set when four bytes of data are read from the serial EEPROM.

VPD Data. Bits [31:0] (PVPDATA[31:0]; PCI:50h, LOC:190h). The PVPDATA register is not a pure read/write register. Data read into the register depends upon the last Read operation performed in PVPDAD[15]. VPD data is written or read through this register. The least-significant byte corresponds to the VPD byte at the address specified by the VPD Address register. Four bytes are always transferred between the register and the serial EEPROM.

Register 10-1. VPD Capabilities Register

31	30 16	15 8	7 0
F	VPD Address	Next_Cap Pointer (0h)	VPD ID (03h)
VPD Data			

10.1.2 VPD Serial EEPROM Partitioning

To support VPD, the serial EEPROM is partitioned into read only and read/write sections.

10.1.3 Sequential Read Only

The first 736 bits, 92 bytes of the serial EEPROM contain read-only information. The read-only portion of the serial EEPROM is loaded into the PCI 9056, using a sequential Read protocol to the serial EEPROM and occurs after PCI reset. Sequential words are read by holding EECS asserted, following issuance of a serial EEPROM Read command.

10.1.4 Random Read and Write

The PCI 9056 can read and write the read/write portion of serial EEPROM. The Serial EEPROM Starting at Lword Boundary for VPD Accesses bits (PROT_AREA[6:0]) designates this portion. This register is loaded upon power-on and can be written with a desired value starting at location 0. This provides the capability of writing the entire serial EEPROM. Writes to serial EEPROM are comprised of the following three commands:

- Write Enable
- Write Enable, followed by Write data
- Write Disable

This is done to ensure against accidental writes to the serial EEPROM. Random cycles allow VPD information to be written and read at any time.

To perform a simple VPD write to the serial EEPROM, the following steps are necessary:

1. Change the write-protected serial EEPROM address in PROT_AREA[6:0] to the desired address. 0x0000000 makes the serial EEPROM removable from the beginning.
2. Write desired data into the PVPDATA register.
3. Write destination serial EEPROM address and flag of operation, value of 1.
4. Probe a flag of operation until it changes to a 0 to ensure the write completes.

To perform a simple VPD read from serial EEPROM, the following steps are necessary:

1. Write a destination serial EEPROM address and flag of operation, value of 0.
2. Probe a flag of operation until it changes to a 1 to ensure the Read data is available.
3. Read back the PVPDATA register to see the requested data.

11 REGISTERS

11.1 SUMMARY OF NEW REGISTERS

This section summarizes the new registers, as compared to the PCI 9054.
Refer to the following sections for a full explanation of each register.

Table 11-1. Summary of New Registers (as Compared to the PCI 9054)

PCI Offset	Local Offset	Register	Bits	Description
06h	06h	PCI Status	5	66 MHz-Capable.
4Ah	18Ah	Hot Swap Control	7	Board Insertion ENUM# Status Indicator.
0Dh	8Dh	Local Miscellaneous Control 1	3	Direct Master (PCI Initiator) Write FIFO Flush during PCI Master Abort.
			7	Disconnect with Flush Read FIFO.
0Fh	8Fh	Local Miscellaneous Control 2	0	READY# Timeout Enable.
			1	READY# Timeout Select.
			4:2	Direct Slave Write Delay.
			5	Direct Slave Write FIFO Full Condition.
			7:6	***Reserved***.
28h	A8h	PCI Base Address (Remap) Register for Direct Master-to-PCI Memory	12, 3	Direct Master Read Prefetch Size Control.
100h	1A0h	PCI Arbiter Control	0	PCI Arbiter Enable.
			1	PCI 9056 High Priority.
			2	Early Grant Release.
			3	PCI Arbiter Parking on PCI 9056.
			31:4	***Reserved.***
104h	1A4h	PCI Abort Address	31:0	PCI Abort Address.
68h	E8h	Interrupt Control/Status	0	Enable Local Bus TEA#/LSERR#.
6Ch	ECh	Serial EEPROM Control, PCI Command Codes, User I/O Control, and Init Control	20	LINTo# Interrupt Status.
			21	TEA#/LSERR# Interrupt Status.
			23:22	***Reserved.***
			30	PCI Adapter Software Reset when HOSTEN#=1 and PCI Host Software Reset when HOSTEN#=0.
			31	EEDO Input Enable.
80h	100h	DMA Channel 0 Mode	19	EOT# End Link.
			20	Valid Mode Enable.
			21	Valid Stop Control.
			31:22	***Reserved.***

Table 11-1. Summary of New Registers (as Compared to the PCI 9054) (Continued)

PCI Offset	Local Offset	Register	Bits	Description
84h (when DMAMODE0[20]=0) 88h (when DMAMODE0[20]=1)	104h (when DMAMODE0[20]=0) 108h (when DMAMODE0[20]=1)	DMA Channel 0 PCI Address	31:0	PCI Address Register.
88h (when DMAMODE0[20]=0) 8Ch (when DMAMODE0[20]=1)	108h (when DMAMODE0[20]=0) 10Ch (when DMAMODE0[20]=1)	DMA Channel 0 Local Address	31:0	Local Address Register.
8Ch (when DMAMODE0[20]=0) 84h (when DMAMODE0[20]=1)	10Ch (when DMAMODE0[20]=0) 104h (when DMAMODE0[20]=1)	DMA Channel 0 Transfer Size (Bytes)	30:23	***Reserved.***
			31	Valid.
94h	114h	DMA Channel 1 Mode	12	Demand Mode.
			19	EOT# End Link.
			20	Valid Mode Enable.
			21	Valid Stop Control.
			31:22	***Reserved.***
98h (when DMAMODE1[20]=0) 9Ch (when DMAMODE1[20]=1)	118h (when DMAMODE1[20]=0) 11Ch (when DMAMODE1[20]=1)	DMA Channel 1 PCI Address	31:0	PCI Address Register.
9Ch (when DMAMODE1[20]=0) A0h (when DMAMODE1[20]=1)	11Ch (when DMAMODE1[20]=0) 120h (when DMAMODE1[20]=1)	DMA Channel 1 Local Address	31:0	Local Address Register.
A0h (when DMAMODE1[20]=0) 98h (when DMAMODE1[20]=1)	120h (when DMAMODE1[20]=0) 118h (when DMAMODE1[20]=1)	DMA Channel 1 Transfer Size (Bytes)	30:23	***Reserved.***
			31	Valid.
B0h	130h	DMA Threshold	19:16	DMA Channel 1 PCI-to-Local Almost Full (C1PLAF).
			23:20	DMA Channel 1 Local-to-PCI Almost Empty (C1LPAE).
			27:24	DMA Channel 1 Local-to-PCI Almost Full (C1LPAF).
			31:28	DMA Channel 1 PCI-to-Local Almost Empty (C1PLAE).

11.2 REGISTER ADDRESS MAPPING

11.2.1 PCI Configuration Registers

Table 11-2. PCI Configuration Registers

PCI Configuration Register Address	Local Access (Offset from Chip Select Address	To ensure software compatibility with other versions of the PCI 9056 family and to ensure compatibility with future enhancements, write 0 to all unused bits. 31 30 24 23 16 15 8 7 0	PCI/ Local Writable	Serial EEPROM Writable
00h	00h	Device ID | Vendor ID	N	Y
04h	04h	Status | Command	Y	N
08h	08h	Class Code | Revision ID	N	Y [31:8]
0Ch	0Ch	BIST | Header Type | PCI Bus Latency Timer | Cache Line Size	Y [7:0]	N
10h	10h	PCI Base Address 0; used for Memory-Mapped Configuration Registers (PCIBAR0)	Y	N
14h	14h	PCI Base Address 1; used for I/O-Mapped Configuration Registers (PCIBAR1)	Y	N
18h	18h	PCI Base Address 2; used for Local Address Space 0 (PCIBAR2)	Y	N
1Ch	1Ch	PCI Base Address 3; used for Local Address Space 1 (PCIBAR3)	Y	N
20h	20h	**Refer to the document, *PCI 9056 Blue Book Revision 0.91***		
24h	24h	***Correction*, for the corrected version of these table entries.**		
28h	28h	Cardbus CIS Pointer ***(Not supported)***	N	N
2Ch	2Ch	Subsystem ID | Subsystem Vendor ID	N	Y
30h	30h	PCI Base Address for Local Expansion ROM	Y	N
34h	34h	***Reserved*** | New Capability Pointer	N, Local [7:0]	N
38h	38h	***Reserved***	N	N
3Ch	3Ch	Max_Lat | Min_Gnt | Interrupt Pin | Interrupt Line	Y [7:0], Local [31:0]	Y [31:8]
40h	180h	Power Management Capabilities | Next_Cap Pointer | Capability ID	N, Local [31:8]	N
44h	184h	Data | PMCSR Bridge Support Extensions | Power Management Control/Status Register	Y [15, 12:8, 1:0]	N
48h	188h	***Reserved*** | Control/Status Register | Next_Cap Pointer | Capability ID	Y [23:16], Local [15:0]	Y [15:0]
4Ch	18Ch	F | VPD Address | Next_Cap Pointer | Capability ID	Y [31:16], Local [15:8]	N
50h	190h	VPD Data	Y	N

Note: *Refer to PCI r2.2 for definitions of these registers.*

11.2.2 Local Configuration Registers

Table 11-3. Local Configuration Registers

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9056 family and to ensure compatibility with future enhancements, write 0 to all unused bits. 31 24 23 16 15 8 7 0	PCI/Local Writable	Serial EEPROM Writable
00h	80h	Range for PCI-to-Local Address Space 0	Y	Y
04h	84h	Local Base Address (Remap) for PCI-to-Local Address Space 0	Y	Y
08h	88h	Mode/DMA Arbitration	Y	Y
0Ch	8Ch	Local Miscellaneous Control 2 | Serial EEPROM Write-Protected Address Boundary | Local Miscellaneous Control 1 | Big/Little Endian Descriptor	Y	Y
10h	90h	Range for PCI-to-Local Expansion ROM	Y	Y
14h	94h	Local Base Address (Remap) for PCI-to-Local Expansion ROM and BREQo Control	Y	Y
18h	98h	Local Bus Region Descriptors (Space 0 and Expansion ROM) for PCI-to-Local Accesses	Y	Y
1Ch	9Ch	Range for Direct Master-to-PCI	Y	Y
20h	A0h	Local Base Address for Direct Master-to-PCI Memory	Y	Y
24h	A4h	Local Base Address for Direct Master-to-PCI I/O Configuration	Y	Y
28h	A8h	PCI Base Address (Remap) for Direct Master-to-PCI	Y	Y
2Ch	ACh	PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration	Y	Y
F0h	170h	Range for PCI-to-Local Address Space 1	Y	Y
F4h	174h	Local Base Address (Remap) for PCI-to-Local Address Space 1	Y	Y
F8h	178h	Local Bus Region Descriptor (Space 1) for PCI-to-Local Accesses	Y	Y
FCh	17Ch	PCI Base Dual Address Cycle (Remap) for Direct Master-to-PCI (Upper 32 bits)	Y	N
100h	1A0h	Reserved | PCI Arbiter Control	Y	Y
104h	1A4h	PCI Abort Address	N	N

11.2.3 Runtime Registers

Table 11-4. Runtime Registers

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9056 family and to ensure compatibility with future enhancements, write 0 to all unused bits. 31 16 15 0		PCI/Local Writable	Serial EEPROM Writable
40h	C0h	Mailbox Register 0 (refer to Note)		Y	Y
44h	C4h	Mailbox Register 1 (refer to Note)		Y	Y
48h	C8h	Mailbox Register 2		Y	N
4Ch	CCh	Mailbox Register 3		Y	N
50h	D0h	Mailbox Register 4		Y	N
54h	D4h	Mailbox Register 5		Y	N
58h	D8h	Mailbox Register 6		Y	N
5Ch	DCh	Mailbox Register 7		Y	N
60h	E0h	PCI-to-Local Doorbell Register		Y	N
64h	E4h	Local-to-PCI Doorbell Register		Y	N
68h	E8h	Interrupt Control/Status		Y	N
6Ch	ECh	Serial EEPROM Control, PCI Command Codes, User I/O Control, and Init Control		Y	N
70h	F0h	Device ID	Vendor ID	N	N
74h	F4h	***Unused***	Revision ID	N	N
78h	C0h	Mailbox Register (refer to Note)		Y	N
7Ch	C4h	Mailbox Register (refer to Note)		Y	N

Note: *MBOX0 and MBOX1 are always accessible at addresses 78h/C0h and 7Ch/C4h, respectively. When the I_2O function is disabled (QSR[0]=0), MBOX0 and MBOX1 are also accessible at PCI addresses 40h and 44h for PCI 9060 compatibility. When the I_2O function is enabled, the Inbound and Outbound Queue pointers are accessed at addresses 40h and 44h, replacing MBOX0 and MBOX1 in PCI Address space.*

11.2.4 DMA Registers

Table 11-5. DMA Registers

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9056 family and to ensure compatibility with future enhancements, write 0 to all unused bits. 31 16 15 8 7 0			PCI/Local Writable	Serial EEPROM Writable
80h	100h	DMA Channel 0 Mode			Y	N
84h	104h	DMA Channel 0 PCI Address			Y	N
88h	108h	DMA Channel 0 Local Address			Y	N
8Ch	10Ch	DMA Channel 0 Transfer Byte Count			Y	N
90h	110h	DMA Channel 0 Descriptor Pointer			Y	N
94h	114h	DMA Channel 1 Mode			Y	N
98h	118h	DMA Channel 1 PCI Address			Y	N
9Ch	11Ch	DMA Channel 1 Local Address			Y	N
A0h	120h	DMA Channel 1 Transfer Byte Count			Y	N
A4h	124h	DMA Channel 1 Descriptor Pointer			Y	N
A8h	128h	***Reserved***	DMA Channel 1 Command/ Status	DMA Channel 0 Command/ Status	Y	N
ACh	12Ch	Mode/DMA Arbitration			Y	N
B0h	130h	DMA Threshold			Y	N
B4h	134h	DMA Channel 0 PCI Dual Address Cycle (Upper 32 bits)			Y	N
B8h	138h	DMA Channel 1 PCI Dual Address Cycle (Upper 32 bits)			Y	N

11.2.5 Messaging Queue Registers

Table 11-6. Messaging Queue Registers

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9056 family and to ensure compatibility with future enhancements, write 0 to all unused bits. 31 — 0	PCI/Local Writable	Serial EEPROM Writable
30h	B0h	Outbound Post Queue Interrupt Status	N	N
34h	B4h	Outbound Post Queue Interrupt Mask	Y	N
40h	—	Inbound Queue Port	PCI	N
44h	—	Outbound Queue Port	PCI	N
C0h	140h	Messaging Unit Configuration	Y	N
C4h	144h	Queue Base Address	Y	N
C8h	148h	Inbound Free Head Pointer	Y	N
CCh	14Ch	Inbound Free Tail Pointer	Y	N
D0h	150h	Inbound Post Head Pointer	Y	N
D4h	154h	Inbound Post Tail Pointer	Y	N
D8h	158h	Outbound Free Head Pointer	Y	N
DCh	15Ch	Outbound Free Tail Pointer	Y	N
E0h	160h	Outbound Post Head Pointer	Y	N
E4h	164h	Outbound Post Tail Pointer	Y	N
E8h	168h	Queue Status/Control	Y	N

Notes: *When I_2O messaging is enabled (QSR[0]= 1), the PCI Master (Host or another IOP) uses the Inbound Queue Port to read Message Frame Addresses (MFAs) from the Inbound Free List FIFO and to write MFAs to the Inbound Post Queue FIFO. The Outbound Queue Port reads MFAs from the Outbound Post Queue FIFO and writes MFAs to the Outbound Free List FIFO.*

Each Inbound MFA is specified by I_2O as an offset from the PCI Base Address 0 (programmed in PCIBAR0) to the start of the message frame. This means that all inbound message frames should reside in PCI Base Address 0 Memory space.

Each Outbound MFA is specified by I_2O as an offset from system address 0x00000000h. Outbound MFA is a physical 32-bit address of the frame in shared PCI system memory.

The Inbound and Outbound Queues may reside in Local Address Space 0 or Space 1 by programming QSR. The queues need not be in shared memory.

11.3 PCI CONFIGURATION REGISTERS

All registers may be written to or read from in Byte, Word, or Lword accesses.

Register 11-1. (PCIIDR; PCI:00h, LOC:00h) PCI Configuration ID

Bit	Description	Read	Write	Value after Reset
15:0	**Vendor ID.** Identifies manufacturer of device. Defaults to the PCI SIG-issued Vendor ID of PLX (10B5h) if blank or if no serial EEPROM is present.	Yes	Local/ Serial EEPROM	10B5h or 0
31:16	**Device ID.** Identifies particular device. Defaults to PLX part number for PCI interface chip (9056h) if blank or no serial EEPROM is present.	Yes	Local/ Serial EEPROM	9056h or 0

Register 11-2. (PCICR; PCI:04h, LOC:04h) PCI Command

Bit	Description	Read	Write	Value after Reset
0	**I/O Space.** Writing a 1 allows the device to respond to I/O space accesses. Writing a 0 disables the device from responding to I/O space accesses.	Yes	Yes	0
1	**Memory Space.** Writing a 1 allows the device to respond to Memory Space accesses. Writing a 0 disables the device from responding to Memory Space accesses.	Yes	Yes	0
2	**Master Enable.** Writing a 1 allows device to behave as a Bus Master. Writing a 0 disables device from generating Bus Master accesses.	Yes	Yes	0
3	**Special Cycle.** ***Not supported.***	Yes	No	0
4	**Memory Write and Invalidate Enable.** Writing a 1 enables the Memory Write and Invalidate mode for Direct Master and DMA. (Refer to the DMA Mode register(s), DMAMODE0[13] and/or DMAMODE1[13].)	Yes	Yes	0
5	**VGA Palette Snoop.** ***Not supported.***	Yes	No	0
6	**Parity Error Response.** Writing a 0 indicates parity error is ignored and the operation continues. Writing a 1 indicates parity checking is enabled.	Yes	Yes	0
7	**Wait Cycle Control.** Controls whether a device does address/data stepping. Writing a 0 indicates the device never does stepping. Writing a 1 indicates the device always does stepping. ***Note:*** *Hardwired to 0.*	Yes	No	0
8	**SERR# Enable.** Writing a 1 enables SERR# driver. Writing a 0 disables SERR# driver.	Yes	Yes	0
9	**Fast Back-to-Back Enable.** Indicates what type of fast back-to-back transfers a Master can perform on the bus. Writing a 1 indicates fast back-to-back transfers can occur to any agent on the bus. Writing a 0 indicates fast back-to-back transfers can only occur to the same agent as in the previous cycle. ***Note:*** *Hardwired to 0.*	Yes	No	0
15:10	***Reserved.***	Yes	No	0h

Register 11-3. (PCISR; PCI:06h, LOC:06h) PCI Status

Bit	Description	Read	Write	Value after Reset
3:0	***Reserved.***	Yes	No	0h
4	**New Capability Functions Support.** Writing a 1 supports New Capabilities Functions. If enabled, the first New Capability Function ID is located at PCI Configuration offset [40h]. Can be written only from the Local Bus. Read-only from the PCI Bus.	Yes	Local	1
5	**66 MHz-Capable.** If set to 1, this device supports a 66 MHz PCI clock environment.	Yes	Local	1
6	**User Definable Functions.** If set to 1, this device supports user definable functions. Can be written only from the Local Bus. Read-only from the PCI Bus.	Yes	Local	0
7	**Fast Back-to-Back Capable.** Writing a 1 indicates an adapter can accept fast back-to-back transactions. ***Note:*** *Hardwired to 1.*	Yes	No	1
8	**Master Data Parity Error Detected.** Set to 1 when three conditions are met: 1) PCI 9056 asserted PERR# or acknowledged PERR# asserted; 2) PCI 9056 was Bus Master for operation in which error occurred; 3) Parity Error Response bit is set (PCICR[6]=1). Writing a 1 clears this bit to 0.	Yes	Yes/Clr	0
10:9	**DEVSEL# Timing.** Indicates timing for DEVSEL# assertion. Writing a 01 sets this bit to medium. ***Note:*** *Hardwired to 01.*	Yes	No	01
11	**Target Abort.** When set to 1, indicates the PCI 9056 has signaled a Target Abort. Writing a 1 clears this bit to 0.	Yes	Yes/Clr	0
12	**Received Target Abort.** When set to 1, indicates the PCI 9056 has received a Target Abort signal. Writing a 1 clears this bit to 0.	Yes	Yes/Clr	0
13	**Received Master Abort.** When set to 1, indicates the PCI 9056 has received a Master Abort signal. Writing a 1 clears this bit to 0.	Yes	Yes/Clr	0
14	**Signaled System Error.** When set to 1, indicates the PCI 9056 has reported a system error on SERR#. Writing a 1 clears this bit to 0.	Yes	Yes/Clr	0
15	**Detected Parity Error.** When set to 1, indicates the PCI 9056 has detected a PCI Bus parity error, even if parity error handling is disabled (the Parity Error Response bit in the Command register is clear). One of three conditions can cause this bit to be set: 1) PCI 9056 detected parity error during PCI Address phase; 2) PCI 9056 detected data parity error when it is the target of a write; 3) PCI 9056 detected data parity error when performing Master Read operation. Writing a 1 clears this bit to 0.	Yes	Yes/Clr	0

Register 11-4. (PCIREV; PCI:08h, LOC:08h) PCI Revision ID

Bit	Description	Read	Write	Value after Reset
7:0	**Revision ID.** Silicon revision of the PCI 9056.	Yes	Local/ Serial EEPROM	Current Rev # (AA)

Register 11-5. (PCICCR; PCI:09-0Bh, LOC:09-0Bh) PCI Class Code

Bit	Description	Read	Write	Value after Reset
7:0	**Register Level Programming Interface.** None defined.	Yes	Local/ Serial EEPROM	0h
15:8	**Subclass Code (Other Bridge Device).**	Yes	Local/ Serial EEPROM	80h
23:16	**Base Class Code (Bridge Device).**	Yes	Local/ Serial EEPROM	06h

Register 11-6. (PCICLSR; PCI:0Ch, LOC:0Ch) PCI Cache Line Size

Bit	Description	Read	Write	Value after Reset
7:0	**System Cache Line Size.** Specified in units of 32-bit words (8 or 16 Lwords). If a size other than 8 or 16 is specified, the PCI 9056 performs Write transfers rather than Memory Write and Invalidate transfers.	Yes	Yes	0h

Register 11-7. (PCILTR; PCI:0Dh, LOC:0Dh) PCI Bus Latency Timer

Bit	Description	Read	Write	Value after Reset
7:0	**PCI Bus Latency Timer.** Specifies amount of time (in units of PCI Bus clocks) the PCI 9056, as a Bus Master, can burst data on the PCI Bus.	Yes	Yes	0h

Register 11-8. (PCIHTR; PCI:0Eh, LOC:0Eh) PCI Header Type

Bit	Description	Read	Write	Value after Reset
6:0	**Configuration Layout Type.** Specifies layout of bits 10h through 3Fh in configuration space. Only one encoding, 0h, is defined. All other encodings are reserved.	Yes	Local	0h
7	**Header Type.** Writing a 1 indicates multiple functions. Writing a 0 indicates single function.	Yes	Local	0

Register 11-9. (PCIBISTR; PCI:0Fh, LOC:0Fh) PCI Built-In Self Test (BIST)

Bit	Description	Read	Write	Value after Reset
3:0	**BIST Pass/Fail.** Writing 0h indicates a device passed its test. Non-0h values indicate a device failed its test. Device-specific failure codes can be encoded in a non-0h value.	Yes	Local	0h
5:4	***Reserved.***	Yes	No	00
6	**PCI BIST Interrupt Enable.** The PCI Bus writes 1 to enable BIST. Generates an interrupt to the Local Bus. The Local Bus resets this bit when BIST is complete. The software should fail device if BIST is not complete after two seconds. Refer to the Runtime registers for Interrupt Control/Status.	Yes	Yes	0
7	**BIST Support.** Returns 1 if device supports BIST. Returns 0 if device is not BIST-compatible.	Yes	Local	0

Register 11-10. (PCIBAR0; PCI:10h, LOC:10h) PCI Base Address Register for Memory Accesses to Local, Runtime, and DMA

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing a 0 indicates the register maps into Memory space. Writing a 1 indicates the register maps into I/O space. ***Note:*** *Hardwired to 0.*	Yes	No	0
2:1	**Register Location.** Values: 00—Locate anywhere in 32-bit Memory Address space 01—*PCI r2.1*: Locate below 1-MB Memory Address space *PCI r2.2*: ***Reserved*** 10—Locate anywhere in 32-bit Memory Address space 11—***Reserved*** ***Note:*** *Hardwired to 00.*	Yes	No	00
3	**Prefetchable.** Writing a 1 indicates there are no side effects on reads. Does not affect operation of the PCI 9056. ***Note:*** *Hardwired to 0.*	Yes	No	0
7:4	**Memory Base Address.** Memory base address for access to Local, Runtime, and DMA registers (requires 256 bytes). ***Note:*** *Hardwired to 0h.*	Yes	No	0h
31:8	**Memory Base Address.** Memory base address for access to Local, Runtime, and DMA registers.	Yes	Yes	0h

Note: *For I_2O, Inbound message frame pool must reside in address space pointed to by PCIBAR0. Message Frame Address (MFA) is defined by I_2O as offset from this base address to start of message frame.*

Register 11-11. (PCIBAR1; PCI:14h, LOC:14h) PCI Base Address Register for I/O Accesses to Local, Runtime, and DMA

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing a 0 indicates the register maps into Memory space. Writing a 1 indicates the register maps into I/O space. ***Note:*** *Hardwired to 1.*	Yes	No	1
1	***Reserved.***	Yes	No	0
7:2	**I/O Base Address.** Base Address for I/O access to Local, Runtime, and DMA registers (requires 256 bytes). ***Note:*** *Hardwired to 0h.*	Yes	No	0h
31:8	**I/O Base Address.** Base Address for I/O access to Local, Runtime, and DMA registers. PCIBAR1 can be enabled or disabled by setting or clearing the Base Address Register 1 Enable bit (LMISC1[0]).	Yes	Yes	0h

Register 11-12. (PCIBAR2; PCI:18h, LOC:18h) PCI Base Address Register for Memory Accesses to Local Address Space 0

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing a 0 indicates the register maps into Memory space. Writing a 1 indicates the register maps into I/O space. (Specified in LAS0RR register.)	Yes	No	0
2:1	**Register Location (If Memory Space).** Values: 00—Locate anywhere in 32-bit Memory Address space 01—*PCI r2.1*: Locate below 1-MB Memory Address space *PCI r2.2*: ***Reserved*** 10—Locate anywhere in 32-bit Memory Address space 11—***Reserved*** (Specified in LAS0RR register.) If I/O Space, bit 1 is always 0 and bit 2 is included in the base address.	Yes	Mem: No I/O: bit 1 no, bit 2 yes	00
3	**Prefetchable (If Memory Space).** Writing a 1 indicates there are no side effects on reads. Reflects value of LAS0RR[3] and provides only status to the system. Does not affect operation of the PCI 9056. The associated Bus Region Descriptor register controls prefetching functions of this address space. (Specified in LAS0RR register.) If I/O Space, bit 3 is included in the base address.	Yes	Mem: No I/O: Yes	0
31:4	**Memory Base Address.** Memory base address for access to Local Address Space 0. PCIBAR2 can be enabled or disabled by setting or clearing the Space 0 Enable bit (LAS0BA[0]).	Yes	Yes	0h

Register 11-13. (PCIBAR3; PCI:1Ch, LOC:1Ch) PCI Base Address Register for Memory Accesses to Local Address Space 1

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing a 0 indicates the register maps into Memory space. Writing a 1 indicates the register maps into I/O space. (Specified in LAS1RR register.)	Yes	No	0
2:1	**Register Location.** Values: 00—Locate anywhere in 32-bit Memory Address space 01—*PCI r2.1*: Locate below 1-MB Memory Address space *PCI r2.2*: ***Reserved*** 10—Locate anywhere in 32-bit Memory Address space 11—***Reserved*** (Specified in LAS1RR register.) If I/O Space, bit 1 is always 0 and bit 2 is included in the base address.	Yes	Mem: No I/O: Bit 1 No, Bit 2 Yes	00
3	**Prefetchable (If Memory Space).** Writing a 1 indicates there are no side effects on reads. Reflects value of LAS1RR[3] and only provides status to the system. Does not affect operation of the PCI 9056. The associated Bus Region Descriptor register controls prefetching functions of this address space. (Specified in LAS1RR register.) If I/O Space, bit 3 is included in base address.	Yes	Mem: No I/O: Yes	0
31:4	**Memory Base Address.** Memory base address for access to Local Address Space 1. PCIBAR3 can be enabled or disabled by setting or clearing the Space 1 Enable bit (LAS1BA[0]). If QSR[0]=1, PCIBAR3 returns 0h.	Yes	Yes	0h

Register 11-14. (PCIBAR4; PCI:20h, LOC:20h) PCI Base Address

Bit	Description	Read	Write	Value after Reset
31:0	***Reserved.***	Yes	No	0h

Register 11-15. (PCIBAR5; PCI:24h, LOC:24h) PCI Base Address

Bit	Description	Read	Write	Value after Reset
31:0	***Reserved.***	Yes	No	0h

Register 11-16. (PCICIS; PCI:28h, LOC:28h) PCI Cardbus CIS Pointer

Bit	Description	Read	Write	Value after Reset
31:0	**Cardbus Information Structure Pointer for PCMCIA. *Not supported.***	Yes	No	0h

Register 11-17. (PCISVID; PCI:2Ch, LOC:2Ch) PCI Subsystem Vendor ID

Bit	Description	Read	Write	Value after Reset
15:0	**Subsystem Vendor ID (unique add-in board Vendor ID).**	Yes	Local/ Serial EEPROM	10B5h

Register 11-18. (PCISID; PCI:2Eh, LOC:2Eh) PCI Subsystem ID

Bit	Description	Read	Write	Value after Reset
15:0	**Subsystem ID (unique add-in board Device ID).**	Yes	Local/ Serial EEPROM	9056h

Register 11-19. (PCIERBAR; PCI:30h, LOC:30h) PCI Expansion ROM Base

Bit	Description	Read	Write	Value after Reset
0	**Address Decode Enable.** Writing a 1 indicates a device accepts accesses to the Expansion ROM address. Writing a 0 indicates a device does not accept accesses to Expansion ROM space. Should be set to 0 if there is no Expansion ROM. Works in conjunction with EROMRR[0].	Yes	Yes	0
10:1	***Reserved.***	Yes	No	0h
31:11	**Expansion ROM Base Address (upper 21 bits).**	Yes	Yes	0h

Register 11-20. (CAP_PTR; PCI:34h, LOC:34h) New Capability Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.*** Must be set to 0.	Yes	No	00
7:2	**New Capability Pointer.** Offset into PCI Configuration Space for the location of the first item in the New Capabilities Linked List.	Yes	Local	40h
31:8	***Reserved.***	Yes	No	0h

Register 11-21. (PCIILR; PCI:3Ch, LOC:3Ch) PCI Interrupt Line

Bit	Description	Read	Write	Value after Reset
7:0	**Interrupt Line Routing Value.** Value indicates which input of the system interrupt controller(s) is connected to each interrupt line of the device.	Yes	Yes	0h

Register 11-22. (PCIIPR; PCI:3Dh, LOC:3Dh) PCI Interrupt Pin

Bit	Description	Read	Write	Value after Reset
7:0	**Interrupt Pin Register.** Indicates which interrupt pin the device uses. The following values are decoded (the PCI 9056 supports only INTA#): 0 = No Interrupt pin 1 = INTA# 2 = INTB# 3 = INTC# 4 = INTD#	Yes	Local/ Serial EEPROM	1h

Register 11-23. (PCIMGR; PCI:3Eh, LOC:3Eh) PCI Min_Gnt

Bit	Description	Read	Write	Value after Reset
7:0	**Min_Gnt.** Specifies how long a burst period device needs, assuming a clock rate of 33 MHz. Value is a multiple of 1/4 µs increments.	Yes	Local/ Serial EEPROM	0h

Register 11-24. (PCIMLR; PCI:3Fh, LOC:3Fh) PCI Max_Lat

Bit	Description	Read	Write	Value after Reset
7:0	**Max_Lat.** Specifies how often the device must gain access to the PCI Bus. Value is a multiple of 1/4 µs increments.	Yes	Local/ Serial EEPROM	0h

Register 11-25. (PMCAPID; PCI:40h, LOC:180h) Power Management Capability ID

Bit	Description	Read	Write	Value after Reset
7:0	**Power Management Capability ID.**	Yes	No	1h

Register 11-26. (PMNEXT; PCI:41h, LOC:181h) Power Management Next Capability Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.*** Must be set to 0.	Yes	No	00
7:2	**Next_Cap Pointer.** Points to the first location of the next item in the capabilities linked list. If Power Management is the last item in the list, then this register should be set to 0.	Yes	Local	48h

Register 11-27. (PMC; PCI:42h, LOC:182h) Power Management Capabilities

Bit	Description	Read	Write	Value after Reset
2:0	**Version.** Writing a 1h indicates this function complies with *PCI Power Mgmt. r1.1*.	Yes	Local	001
3	**PCI Clock Required for PME# Signal.** When set to 1, indicates a function relies on the presence of the PCI clock for PME# operation. The PCI 9056 does not require the PCI clock for PME#, so this bit should be set to 0.	Yes	Local	0
4	***Reserved.***	Yes	No	0
5	**DSI.** When set to 1, the PCI 9056 requires special initialization following a transition to a D_0 uninitialized state before a generic class device driver is able to use it.	Yes	Local	0
8:6	**AUX_Current.** Supported by way of the PMDATA register per *PCI Power Mgmt. r1.1*.	Yes	No	000
9	**D_1_Support.** When set to 1, the PCI 9056 supports the D_1 power state.	Yes	Local	0
10	**D_2_Support.** When set to 1, the PCI 9056 supports the D_2 power state.	Yes	Local	0
15:11	**PME_Support.** Indicates power states in which the PCI 9056 may assert PME#. Value — Description XXXX1 — PME# can be asserted from D_0 XXX1X — PME# can be asserted from D_1 XX1XX — PME# can be asserted from D_2 X1XXX — PME# can be asserted from D_{3hot} 1XXXX — PME# can be asserted from D_{3cold}	Yes	Local	0h

Register 11-28. (PMCSR; PCI:44h, LOC:184h) Power Management Control/Status

Bit	Description	Read	Write	Value after Reset
1:0	**Power State.** Determines or changes the current power state. Value State 00 D_0 01 D_1 10 D_2 11 D_{3hot} Transition from a D_{3hot} state to a D_0 state causes a soft reset. Should only be initiated from the PCI Bus because the Local Bus interface is reset during a soft reset. In a D_{3hot} state, PCI Memory and I/O accesses are disabled, as well as PCI interrupts, and only configuration access is allowed. The same is true for the D_2 state if the corresponding D_2_Support pin is set.	Yes	Yes	00
7:2	***Reserved.***	Yes	No	0h
8	**PME_En.** Writing a 1 enables PME# to be asserted.	Yes	Yes	0
12:9	**Data_Select.** Selects which data to report through the Data register and Data_Scale bits.	Yes	Yes	0h
14:13	**Data_Scale.** Indicates the scaling factor to use when interpreting the value of the Data register. Value and meaning of these bits depends on the data value selected by the Data_Select bit. When the Local CPU initializes the Data_Scale values, must use the Data_Select bit to determine which Data_Scale value it is writing. For Power Consumed and Power Dissipated data, the following scale factors are used. Unit values are in watts. Value Scale 0 Unknown 1 0.1x 2 0.01x 3 0.001x	Yes	Local	00
15	**PME_Status.** Indicates PME# is being driven if the PME_En bit is set (PMCSR[8]=1). Writing a 1 from the Local Bus sets this bit; writing a 1 from the PCI Bus clears this bit to 0. Depending on the current power state, set only if the appropriate PME_Support bit(s) is set (PMC[15:11]=1).	Yes	Local/ Set, PCI/Clr	0

Register 11-29. (PMCSR_BSE; PCI:46h, LOC:186h) PMCSR Bridge Support Extensions

Bit	Description	Read	Write	Value after Reset
7:0	***Reserved.***	Yes	No	0h

Register 11-30. (PMDATA; PCI:47h, LOC:187h) Power Management Data

Bit	Description	Read	Write	Value after Reset
7:0	**Power Management Data.** Provides operating data, such as power consumed or heat dissipation. Data returned is selected by the Data_Select bit(s) (PMCSR[12:9]) and scaled by the Data_Scale bit(s) (PMCSR[14:13]). Data_Select — Description 0 — D_0 Power Consumed 1 — D_1 Power Consumed 2 — D_2 Power Consumed 3 — D_3 Power Consumed 4 — D_0 Power Dissipated 5 — D_1 Power Dissipated 6 — D_2 Power Dissipated 7 — D_3 Power Dissipated	Yes	Local	0h

Register 11-31. (HS_CNTL; PCI:48h, LOC:188h) Hot Swap Control

Bit	Description	Read	Write	Value after Reset
7:0	**Hot Swap ID.**	Yes	Local/ Serial EEPROM	06h

Register 11-32. (HS_NEXT; PCI:49h, LOC:189h) Hot Swap Next Capability Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.*** Must be set to 0.	Yes	No	00
7:2	**Next_Cap Pointer.** Points to the first location of the next item in the capabilities linked list. If Hot Swap is the last item in the list, then this register should be set to 0.	Yes	Local/ Serial EEPROM	4Ch

Register 11-33. (HS_CSR; PCI:4Ah, LOC:18Ah) Hot Swap Control/Status

Bit	Description	Read	Write	Value after Reset
0	***Reserved.***	Yes	No	0
1	**ENUM# Interrupt Clear.** Writing a 0 enables the interrupt. Writing a 1 clears the interrupt.	Yes	Yes/Clr	0
2	***Reserved.***	Yes	No	0
3	**LED Software On/Off Switch.** Writing a 1 turns on the LED. Writing a 0 turns off the LED.	Yes	PCI	0
5:4	**Programming Interface.**	Yes	No	00
6	**Board Removal ENUM# Status Indicator.** Writing a 1 reports the ENUM# assertion for removal process.	Yes	Yes	0
7	**Board Insertion ENUM# Status Indicator.** Writing a 1 reports the ENUM# assertion for insertion process.	Yes	Yes	0
15:8	***Reserved.***	Yes	No	0h

Register 11-34. (PVPDCNTL; PCI:4Ch, LOC:18Ch) PCI Vital Product Data Control

Bit	Description	Read	Write	Value after Reset
7:0	**VPD ID.** Capability ID = 03h for VPD.	PCI	No	03h

Register 11-35. (PVPD_NEXT; PCI:4Dh, LOC:18Dh) PCI Vital Product Data Next Capability Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.*** Must be set to 0.	Yes	No	00
7:2	**Next_Cap Pointer.** Points to first location of next item in the capabilities linked list. VPD is the last item in the capabilities linked list. This register is set to 0h.	PCI	Local	0h

Register 11-36. (PVPDAD; PCI:4Eh, LOC:18Eh) PCI Vital Product Data Address

Bit	Description	Read	Write	Value after Reset
14:0	**VPD Address.** Byte address of the VPD address to be accessed. Supports a 2K or 4K bit serial EEPROM. Bits [1:0] must be set to 0.	PCI	Yes	0h
15	**F.** Flag used to indicate when the transfer of data between PVPDATA and the storage component is complete. Writing a 0 along with the VPD address causes a read of VPD information into PVPDATA. The hardware sets this bit to 1 when the VPD Data transfer is complete. Writing a 1 along with the VPD address causes a write of VPD information from PVPDATA into a storage component. The hardware sets this bit to 0 after the Write operation is complete.	PCI	Yes	0

Register 11-37. (PVPDATA; PCI:50h, LOC:190h) PCI VPD Data

Bit	Description	Read	Write	Value after Reset
31:0	**VPD Data.**	PCI	Yes	0h

11.4 LOCAL CONFIGURATION REGISTERS

Register 11-38. (LAS0RR; PCI:00h, LOC:80h) Local Address Space 0 Range Register for PCI-to-Local Bus

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing a 0 indicates Local Address Space 0 maps into PCI Memory space. Writing a 1 indicates Local Address Space 0 maps into PCI I/O space.	Yes	Yes	0
2:1	When mapped into Memory space, encoding is as follows: 2/1 Meaning 0 0 Locate anywhere in 32-bit PCI Address space 0 1 *PCI r2.1*: Locate below 1-MB Memory Address space *PCI r2.2*: ***Reserved*** 1 0 Locate anywhere in 32-bit PCI Address space 1 1 ***Reserved*** When mapped into I/O space, bit 1 must be set to 0.Bit 2 is included with bits [31:3] to indicate the decoding range.	Yes	Yes	00
3	When mapped into Memory space, writing a 1 indicates reads are prefetchable (does not affect operation of the PCI 9056, but is used for system status). When mapped into I/O space, it is included with bits [31:2] to indicate the decoding range.	Yes	Yes	0
31:4	Specifies which PCI Address bits to use for decoding a PCI access to Local Bus Space 0. Each bit corresponds to a PCI Address bit. Bit 31 corresponds to address bit 31. Write 1 to all bits that must be included in decode and 0 to all others (used in conjunction with PCIBAR2). Default is 1 MB. ***Notes:*** *Range (**not** Range register) must be power of 2. "Range register value" is 2's complement of range.* *User should limit all I/O spaces to 256 bytes per PCI r2.2.*	Yes	Yes	FFF0000h

Register 11-39. (LAS0BA; PCI:04h, LOC:84h) Local Address Space 0 Local Base Address (Remap)

Bit	Description	Read	Write	Value after Reset
0	**Space 0 Enable.** Writing a 1 enables decoding of PCI addresses for Direct Slave access to Local Bus Space 0. Writing a 0 disables decoding.	Yes	Yes	0
1	***Reserved.***	Yes	No	0
3:2	If Local Bus Space 0 is mapped into Memory space, bits are not used. When mapped into I/O space, included with bits [31:4] for remapping.	Yes	Yes	00
31:4	**Remap PCI Address to Local Address Space 0 into Local Address Space.** Bits in this register remap (replace) PCI Address bits used in decode as Local Address bits. ***Note:*** *Remap Address value must be a multiple of the Range (**not** the Range register).*	Yes	Yes	0h

Register 11-40. (MARBR; PCI:08h or ACh, LOC:88h or 12Ch) Mode/DMA Arbitration

Bit	Description	Read	Write	Value after Reset
7:0	**Local Bus Latency Timer.** Number of Local Bus clock cycles to occur before de-asserting HOLD and releasing the Local Bus.	Yes	Yes	0h
15:8	**Local Bus Pause Timer.** Number of Local Bus Clock cycles to occur before reasserting HOLD after releasing the Local Bus. The pause timer is valid only during DMA.	Yes	Yes	0h
16	**Local Bus Latency Timer Enable.** Writing a 1 enables the latency timer. Writing a 0 disables the latency timer.	Yes	Yes	0
17	**Local Bus Pause Timer Enable.** Writing a 1 enables the pause timer. Writing a 0 disables the pause timer.	Yes	Yes	0
18	**Local Bus BREQ Enable**. Writing a 1 enables the Local Bus BR#/BREQi. When BR#/BREQi is active, the PCI 9056 de-asserts HOLD and releases the Local Bus.	Yes	Yes	0
20:19	**DMA Channel Priority.** Writing a 00 indicates a rotational priority scheme. Writing a 01 indicates Channel 0 has priority. Writing a 10 indicates Channel 1 has priority. Writing an 11 indicates ***reserved***.	Yes	Yes	00
21	**Local Bus Direct Slave Release Bus Mode.** When set to 1, the PCI 9056 de-asserts HOLD and releases the Local Bus when the Direct Slave Write FIFO becomes empty during a Direct Slave Write or when the Direct Slave Read FIFO becomes full during a Direct Slave Read.	Yes	Yes	1
22	**Direct Slave LOCK# Enable.** Writing a 1 enables Direct Slave locked sequences. Writing a 0 disables Direct Slave locked sequences.	Yes	Yes	0
23	**PCI Request Mode.** Writing a 1 causes the PCI 9056 to de-assert REQ# when it asserts FRAME during a Master cycle. Writing a 0 causes the PCI 9056 to leave REQ# asserted for the entire Bus Master cycle.	Yes	Yes	0
24	**Delayed Read Mode.** When set to 1, the PCI 9056 operates in Delayed Transaction mode for Direct Slave reads. The PCI 9056 issues a Retry to the PCI Host and prefetches Read data.	Yes	Yes	0
25	**PCI Read No Write Mode.** Writing a 1 forces a Retry on writes if a read is pending. Writing a 0 allows writes to occur while a read is pending.	Yes	Yes	0
26	**PCI Read with Write Flush Mode.** Writing a 1 submits a request to flush a pending Read cycle if a Write cycle is detected. Writing a 0 submits a request to not effect pending reads when a Write cycle occurs (*PCI r2.2*-compatible).	Yes	Yes	0
27	**Gate Local Bus Latency Timer with BREQi (C and J modes only).**	Yes	Yes	0
28	**PCI Read No Flush Mode.** Writing a 1 submits a request to not flush the Read FIFO if the PCI Read cycle completes (Read Ahead mode). Writing a 0 submits a request to flush the Read FIFO if a PCI Read cycle completes.	Yes	Yes	0
29	When set to 0, reads from the PCI Configuration Register address 00h returns Device ID and Vendor ID. When set to 1, reads from the PCI Configuration register address 00h returns Subsystem ID and Subsystem Vendor ID.	Yes	Yes	0
30	**FIFO Full Status Flag.** When set to 1, the Direct Master Write FIFO is almost full. Reflects the value of the DMPAF pin.	Yes	No	0
31	**BIGEND#/WAIT# Input/Output Select (M mode only).** Writing a 1 selects the wait functionality of the signal. Writing a 0 selects Big Endian input functionality.	Yes	Yes	0

Register 11-41. (BIGEND; PCI:0Ch, LOC:8Ch) Big/Little Endian Descriptor

Bit	Description	Read	Write	Value after Reset
0	**Configuration Register Big Endian Mode (Address Invariance).** Writing a 1 specifies use of Big Endian data ordering for Local accesses to the Configuration registers. Writing a 0 specifies Little Endian ordering. Big Endian mode can be specified for Configuration register accesses by asserting BIGEND# during the Address phase of the access.	Yes	Yes	0
1	**Direct Master Big Endian Mode (Address Invariance).** Writing a 1 specifies use of Big Endian data ordering for Direct Master accesses. Writing a 0 specifies Little Endian ordering. Big Endian mode can be specified for Direct Master accesses by asserting BIGEND# input pin during the Address phase of the access.	Yes	Yes	0
2	**Direct Slave Address Space 0 Big Endian Mode (Address Invariance).** Writing a 1 specifies use of Big Endian data ordering for Direct Slave accesses to Local Address Space 0. Writing a 0 specifies Little Endian ordering.	Yes	Yes	0
3	**Direct Slave Address Expansion ROM 0 Big Endian Mode (Address Invariance).** Writing a 1 specifies use of Big Endian data ordering for Direct Slave accesses to Expansion ROM. Writing a 0 specifies Little Endian ordering.	Yes	Yes	0
4	**Big Endian Byte Lane Mode.** Writing a 1 specifies that in any Endian mode, use the following byte lanes for the modes listed: **M Mode** [0:15] for a 16-bit Local Bus [0:7] for an 8-bit Local Bus **C and J Modes** [31:16] for a 16-bit Local Bus [31:24] for an 8-bit Local Bus Writing a 0 specifies that in any Endian mode, use the following byte lanes for the modes listed: **M Mode** [16:31] for a 16-bit Local Bus [24:31] for an 8-bit Local Bus **C and J Modes** [15:0] for a 16-bit Local Bus [7:0] for an 8-bit Local Bus	Yes	Yes	0
5	**Direct Slave Address Space 1 Big Endian Mode (Address Invariance).** Writing a 1 specifies use of Big Endian data ordering for Direct Slave accesses to Local Address Space 1. Writing a 0 specifies Little Endian ordering.	Yes	Yes	0
6	**DMA Channel 1 Big Endian Mode (Address Invariance).** Writing a 1 specifies use of Big Endian data ordering for DMA Channel 1 accesses to the Local Address space. Writing a 0 specifies Little Endian ordering.	Yes	Yes	0
7	**DMA Channel 0 Big Endian Mode (Address Invariance).** Writing a 1 specifies use of Big Endian data ordering for DMA Channel 0 accesses to the Local Address space. Writing a 0 specifies Little Endian ordering.	Yes	Yes	0

Register 11-42. (LMISC1; PCI:0Dh, LOC:8Dh) Local Miscellaneous Control1

Bit	Description	Read	Write	Value after Reset
0	**Base Address Register 1 Enable.** If set to 1, the Base Address 1 Register for I/O accesses to Configuration registers is enabled. If set to 0, the Base Address 1 Register for I/O accesses to Configuration registers is disabled. This option is intended for embedded systems only. This bit should be set to 1 for PC platforms.	Yes	Yes	1
1	**Base Address Register 1 Shift.** If Base Address Register 1 Enable is low, and this bit is set to 0, then PCIBAR2 and PCIBAR3 remain at PCI Configuration addresses 18h and 1Ch. If Base Address Register 1 Enable is low, and this bit is set to 1, then PCIBAR2 (Local Address Space 0) and PCIBAR3 (Local Address Space 1) are shifted to become PCIBAR1 and PCIBAR2 at PCI Configuration addresses 14h and 18h. Set if a blank region in Base Address Register Space cannot be accepted by system BIOS.	Yes	Yes	0
2	**Local Init Status.** Writing a 1 indicates Local Init done. Responses to PCI accesses prior to this bit being set are determined by the USERi state at PCI RST# de-assertion, as described in Sections 2.4.1.2 and 4.4.1.2.	Yes	Local/ Serial EEPROM	0
3	**Direct Master (PCI Initiator) Write FIFO Flush during PCI** Master Abort**.** When set to 1, the PCI 9056 flushes the Direct Master Write FIFO each time the Direct Slave or Master Abort occurs. When set to 0, the PCI 9056 keeps data in the Direct Master Write FIFO.	Yes	Yes	0
4	**M Mode Direct Master Delayed Read Enable.** Writing a 1 enables the PCI 9056 to operate in Delayed Transaction mode for Direct Master reads. The PCI 9056 issues a RETRY# to the M mode Master and prefetches Read data from the PCI Bus.	Yes	Yes	0
5	**M Mode TEA# Input Interrupt Mask.** When set to 1, TEA# input causes SERR# output on the PCI Bus if enabled (PCICR[8]=1) and the Signaled System Error bit is set (PCISR[14]=1). Writing 0 masks the TEA# input to create SERR#. The SERR# Status bit is set in both cases.	Yes	Yes	0
6	**Direct Master Write FIFO Almost Full RETRY# Output Enable.** When set to 1, the PCI 9056 issues a RETRY# to the MPC850 or MPC860.	Yes	Yes	0
7	**Disconnect with Flush Read FIFO.** Value of 1 causes a disconnect with flushing of the Read FIFO in Delayed Read mode (MARBR[24]). Value of 0 causes a disconnect without flushing the Read FIFO (as a Retry).	Yes	Yes	0

Register 11-43. (PROT_AREA; PCI:0Eh, LOC:8Eh) Serial EEPROM Write-Protected Address Boundary

Bit	Description	Read	Write	Value after Reset
6:0	**Serial EEPROM Starting at Lword Boundary (48 Lwords = 192 bytes) for VPD Accesses.** Any serial EEPROM address below this boundary is read-only. ***Note:*** *Anything below the programmed address may contain the PCI 9056 Configuration data.*	Yes	Yes	0110000
7	***Reserved.***	Yes	No	0

Register 11-44. (LMISC2; PCI:0Fh, LOC:8Fh) Local Miscellaneous Control 2

Bit	Description	Read	Write	Value after Reset
0	**READY# Timeout Enable.** Value of 1 enables READY# timeout enable.	Yes	Yes	0
1	**READY# Timeout Select.** Values: 1 = 64 clocks 0 = 32 clocks	Yes	Yes	0
4:2	**Direct Slave Write Delay.** Delay in LCLK of TS#/ADS# from a valid address. Values: 0 = 0 LCLK 2 = 8 LCLK 4 =20 LCLK 6 = 28 LCLK 1 = 4 LCLK 3 = 16 LCLK 5 = 24 LCLK 7 = 32 LCLK	Yes	Yes	000
5	**Direct Slave Write FIFO Full Condition.** Value of 1 guarantees that when the Direct Slave Write FIFO is full with Direct Slave Write data, there is always one location remaining empty for the Direct Slave Read address to be accepted by the PCI 9056. Value of 0 Retries all Direct Slave Read accesses when the Direct Slave Write FIFO is full with Direct Slave Write data.	Yes	Yes	0
7:6	***Reserved.***	Yes	No	00

Register 11-45. (EROMRR; PCI:10h, LOC:90h) Expansion ROM Range

Bit	Description	Read	Write	Value after Reset
0	**Address Decode Enable.** Bit 0 can only be enabled from the serial EEPROM. To disable, set the PCI Expansion ROM Address Decode Enable bit to 0 (PCIERBAR[0]=0).	Yes	Serial EEPROM Only	0
10:1	***Reserved.***	Yes	No	0h
31:11	Specifies which PCI Address bits to use for decoding a PCI-to-Local Bus Expansion ROM. Each bit corresponds to a PCI Address bit. Bit 31 corresponds to address bit 31. Write 1 to all bits that must be included in decode and 0 to all others (used in conjunction with PCIERBAR). ***Note:*** *Range (**not** Range register) must be power of 2. "Range register value" is 2's complement of range.*	Yes	Yes	0h

Register 11-46. (EROMBA; PCI:14h, LOC:94h) Expansion ROM Local Base Address (Remap) and BREQo Control

Bit	Description	Read	Write	Value after Reset
3:0	**M Mode: RETRY# Signal Assertion Delay Clocks.** Number of Local Bus clocks in which a Direct Slave BR# request is pending and a Local Direct Master access is in progress and not being granted the bus BG# before asserting RETRY#. Once asserted, RETRY# remains asserted until PCI 9056 samples de-assertion of BB# by the Local Arbiter (LSB is 8 or 64 clocks). **C and J Modes: Backoff Request Delay Clocks.** Number of Local Bus clocks in which a Direct Slave HOLD request is pending and a Local Direct Master access is in progress and not being granted the bus (LHOLDA) before asserting BREQo (Backoff Request Out). BREQo remains asserted until the PCI 9056 receives LHOLDA (LSB is 8 or 64 clocks).	Yes	Yes	0h
4	**Local Bus Backoff Enable (M, C, and J modes).** Writing a 1 enables the PCI 9056 to assert RETRY#/BREQo.	Yes	Yes	0
5	**Backoff Timer Resolution.** Writing a 1 changes the LSB of the Backoff Timer from 8 to 64 clocks.	Yes	Yes	0
10:6	***Reserved.***	Yes	No	0h
31:11	**Remap PCI Expansion ROM Space into Local Address Space.** Bits in this register remap (replace) the PCI Address bits used in decode as Local Address bits. ***Note:*** *Remap Address value must be a multiple of the Range (**not** the Range register).*	Yes	Yes	0h

Register 11-47. (LBRD0; PCI:18h, LOC:98h) Local Address Space 0/Expansion ROM Bus Region Descriptor

Bit	Description	Read	Write	Value after Reset
1:0	**Memory Space 0 Local Bus Width.** Writing a 00 indicates an 8-bit bus width. Writing a 01 indicates a 16-bit bus width. Writing a 10 or 11 indicates a 32-bit bus width.	Yes	Yes	M = 11 J = 11 C = 11
5:2	**Memory Space 0 Internal Wait States (data-to-data; 0-15 wait states).**	Yes	Yes	0h
6	**Memory Space 0 TA#/READY# Input Enable.** Writing a 1 enables TA#/READY# input. Writing a 0 disables TA#/READY# input.	Yes	Yes	1
7	**Memory Space 0 BTERM# Input Enable.** Writing a 1 enables BTERM# input. Writing a 0 disables BTERM# input. For more information, refer to Section 2.2.5 for M mode or Section 4.2.5 for C and J modes.	Yes	Yes	0
8	**Memory Space 0 Prefetch Disable.** When mapped into Memory space, writing a 0 enables Read prefetching. Writing a 1 disables prefetching. If prefetching is disabled, the PCI 9056 disconnects after each Memory read.	Yes	Yes	0
9	**Expansion ROM Space Prefetch Disable.** Writing a 0 enables Read prefetching. Writing a 1 disables prefetching. If prefetching is disabled, the PCI 9056 disconnects after each Memory read.	Yes	Yes	0
10	**Prefetch Counter Enable.** When set to 1 and Memory prefetching is enabled, the PCI 9056 prefetches up to the number of Lwords specified in prefetch count. When set to 0, the PCI 9056 ignores the count and continues prefetching until it is terminated by the PCI Bus.	Yes	Yes	0
14:11	**Prefetch Counter.** Number of Lwords to prefetch during Memory Read cycles (0-15). A count of zero selects a prefetch of 16 Lwords.	Yes	Yes	0h
15	***Reserved.***	Yes	No	0
17:16	**Expansion ROM Space Local Bus Width.** Writing a 00 indicates an 8-bit bus width. Writing a 01 indicates a 16-bit bus width. Writing a 10 or 11 indicates a 32-bit bus width.	Yes	Yes	M = 11 J = 11 C = 11

Register 11-47. (LBRD0; PCI:18h, LOC:98h) Local Address Space 0/Expansion ROM Bus Region Descriptor (Continued)

Bit	Description	Read	Write	Value after Reset
21:18	**Expansion ROM Space Internal Wait States (data-to-data; 0-15 wait states).**	Yes	Yes	0h
22	**Expansion ROM Space TA#/READY# Input Enable.** Writing a 1 enables TA#/READY# input. Writing a 0 disables TA#/READY# input.	Yes	Yes	1
23	**Expansion ROM Space BTERM# Input Enable.** Writing a 1 enables BTERM# input. Writing a 0 disables BTERM# input. For more information, refer to Section 2.2.5 for M mode or to Section 4.2.5 for C and J modes.	Yes	Yes	0
24	**Memory Space 0 Burst Enable.** Writing a 1 enables bursting. Writing a 0 disables bursting.	Yes	Yes	0
25	**Extra Long Load from Serial EEPROM.** Writing a 1 loads the Subsystem ID and Local Address Space 1 registers. Writing a 0 indicates not to load them.	Yes	Serial EEPROM Only	0
26	**Expansion ROM Space Burst Enable.** Writing a 1 enables bursting. Writing a 0 disables bursting.	Yes	Yes	0
27	**Direct Slave PCI Write Mode.** Writing a 0 indicates the PCI 9056 should disconnect when the Direct Slave Write FIFO is full. Writing a 1 indicates the PCI 9056 should de-assert TRDY# when the Direct Slave Write FIFO is full.	Yes	Yes	0
31:28	**Direct Slave Retry Delay Clocks.** Contains the value (multiplied by 8) of the number of PCI Bus clocks after receiving a PCI-to-Local Read or Write access and not successfully completing a transfer. Pertains to Direct Slave writes only when the Direct Slave PCI Write Mode bit is set (bit [27]=1).	Yes	Yes	4h (32 clocks)

Register 11-48. (DMRR; PCI:1Ch, LOC:9Ch) Local Range Register for Direct Master-to-PCI

Bit	Description	Read	Write	Value after Reset
15:0	***Reserved*** (64-KB increments).	Yes	No	0h
31:16	Specifies which Local Address bits to use for decoding a Local-to-PCI Bus access. Each bit corresponds to a PCI Address bit. Bit 31 corresponds to address bit 31. Write 1 to all bits that must be included in decode and 0h to all others. ***Note:*** *Range (**not** Range register) must be power of 2. "Range register value" is 2's complement of range.*	Yes	Yes	0h

Register 11-49. (DMLBAM; PCI:20h, LOC:A0h) Local Bus Base Address Register for Direct Master-to-PCI Memory

Bit	Description	Read	Write	Value after Reset
15:0	***Reserved.***	Yes	No	0h
31:16	Assigns a value to bits to use for decoding Local-to-PCI Memory accesses. ***Note:*** *Local Base Address value must be a multiple of the Range (**not** the Range register).*	Yes	Yes	0h

Register 11-50. (DMLBAI; PCI:24h, LOC:A4h) Local Base Address Register for Direct Master-to-PCI I/O Configuration

Bit	Description	Read	Write	Value after Reset
15:0	***Reserved.***	Yes	No	0h
31:16	Assigns a value to bits to use for decoding Local-to-PCI I/O or Configuration accesses. ***Notes:*** *Local Base Address value must be a multiple of the Range (**not** the Range register).* *Refer to DMPBAM[13] for the I/O Remap Address option.*	Yes	Yes	0h

Register 11-51. (DMPBAM; PCI:28h, LOC:A8h) PCI Base Address (Remap) Register for Direct Master-to-PCI Memory

Bit	Description	Read	Write	Value after Reset
0	**Direct Master Memory Access Enable.** Writing a 1 enables decode of Direct Master Memory accesses. Writing a 0 disables decode of Direct Master Memory accesses.	Yes	Yes	0
1	**Direct Master I/O Access Enable.** Writing a 1 enables decode of Direct Master I/O accesses. Writing a 0 disables decode of Direct Master I/O accesses.	Yes	Yes	0
2	**Direct Master Cache Enable.** Writing a 1 causes prefetch to occur infinitely.	Yes	Yes	0
12, 3	**Direct Master Read Prefetch Size Control.** Values: 00 = The PCI 9056 continues to prefetch Read data from the PCI Bus until the Direct Master access is finished. This may result in an additional four unneeded Lwords being prefetched from the 32-bit PCI Bus. 01 = Prefetch up to four Lwords from the 32-bit PCI Bus. 10 = Prefetch up to eight Lwords from the 32-bit PCI Bus. 11 = Prefetch up to 16 Lwords from the 32-bit PCI Bus. Direct Master Burst reads must not exceed programmed limit.	Yes	Yes	00
4	**Direct Master PCI Read Mode.** Writing a 0 indicates the PCI 9056 should release the PCI Bus when the Read FIFO becomes full. Writing a 1 indicates the PCI 9056 should retain the PCI Bus and de-assert IRDY# when the Read FIFO becomes full.	Yes	Yes	0
10, 8:5	**Programmable Almost Full Flag.** When the number of entries in the 32-word Direct Master Write FIFO exceeds a (programmed value +1), the MDREQ#/DMPAF signal is asserted high.	Yes	Yes	00000
9	**Memory Write and Invalidate Mode.** When set to 1, the PCI 9056 waits for 8 or 16 Lwords to be written from the Local Bus before starting a PCI access. In addition, all Memory Write and Invalidate cycles to the PCI Bus must be 8 or 16 Lword bursts.	Yes	Yes	0
11	**Direct Master Prefetch Limit.** Writing a 1 causes the PCI 9056 to terminate a prefetch at 4-KB boundaries and restart when the boundary is crossed. Writing an 0 results in continuous prefetch over the boundary space.	Yes	Yes	0
13	**I/O Remap Select.** Writing a 1 forces PCI Address bits [31:16] to all zeros. Writing a 0 uses bits [31:16] of this register as PCI Address bits [31:16].	Yes	Yes	0

Register 11-51. (DMPBAM; PCI:28h, LOC:A8h) PCI Base Address (Remap) Register for Direct Master-to-PCI Memory (Continued)

Bit	Description	Read	Write	Value after Reset
15:14	**Direct Master Write Delay.** Delays PCI Bus request after Direct Master Burst Write cycle has started. Values: 00 = No delay; start cycle immediately 01 = Delay 4 PCI clocks 10 = Delay 8 PCI clocks 11 = Delay 16 PCI clocks	Yes	Yes	00
31:16	**Remap Local-to-PCI Space into PCI Address Space.** Bits in this register remap (replace) Local Address bits used in decode as the PCI Address bits. ***Note:*** *Remap Address value must be a multiple of the Range (**not** the Range register).*	Yes	Yes	0h

Register 11-52. (DMCFGA; PCI:2Ch, LOC:ACh) PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration

Bit	Description	Read	Write	Value after Reset
1:0	**Configuration Type.** Values: 00 = Type 0 01 = Type 1	Yes	Yes	00
7:2	**Register Number.**	Yes	Yes	0
10:8	**Function Number.**	Yes	Yes	0
15:11	**Device Number.**	Yes	Yes	0
23:16	**Bus Number.**	Yes	Yes	0h
30:24	***Reserved.***	Yes	No	0h
31	**Configuration Enable.** Writing a 1 allows Local-to-PCI I/O accesses to be converted to a PCI Configuration cycle. Parameters in this table are used to assert the PCI Configuration address. ***Note:*** *For more information, refer to the Direct Master Configuration cycle example in Section 3.4.1.9 for M mode or Section 5.4.1.8.1 for C and J modes.*	Yes	Yes	0

Register 11-53. (LAS1RR; PCI:F0h, LOC:170h) Local Address Space 1 Range Register for PCI-to-Local Bus

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing a 0 indicates Local Address Space 1 maps into PCI Memory space. Writing a 1 indicates Address Space 1 maps into PCI I/O space.	Yes	Yes	0
2:1	When mapped into Memory space, encoding is as follows: 2/1 Meaning 0 0 Locate anywhere in 32-bit PCI Address space 0 1 *PCI r2.1*: Locate below 1-MB Memory Address space *PCI r2.2*: ***Reserved*** 1 0 Locate anywhere in 32-bit PCI Address space 1 1 ***Reserved*** When mapped into I/O space, bit 1 must be set to 0. Bit 2 is included with bits [31:3] to indicate the decoding range.	Yes	Yes	00
3	When mapped into Memory space, writing a 1 indicates reads are prefetchable (does not affect operation of the PCI 9056, but is used for system status). When mapped into I/O space, included with bits [31:2] to indicate the decoding range.	Yes	Yes	0
31:4	Specifies which PCI Address bits to use for decoding a PCI access to Local Bus Space 1. Each bit corresponds to a PCI Address bit. Bit 31 corresponds to address bit 31. Write 1 to all bits that must be included in decode and 0 to all others. (Used in conjunction with PCIBAR3.) Default is 1 MB. If QSR[0]=1, defines PCI Base Address 0. ***Notes:*** *Range (**not** Range register) must be power of 2. "Range register value" is 2's complement of range.* *User should limit all I/O spaces to 256 bytes.*	Yes	Yes	FFF0000h

Register 11-54. (LAS1BA; PCI:F4h, LOC:174h) Local Address Space 1 Local Base Address (Remap)

Bit	Description	Read	Write	Value after Reset
0	**Space 1 Enable.** Writing a 1 enables decoding of PCI addresses for Direct Slave access to Local Bus Space 1. Writing a 0 disables decoding.	Yes	Yes	0
1	***Reserved.***	Yes	No	0
3:2	Not used if Local Bus Space 1 is mapped into Memory space. Included with bits [31:4] for remapping when mapped into I/O space.	Yes	Yes	00
31:4	**Remap PCI Address to Local Address Space 1 into Local Address Space.** Bits in this register remap (replace) the PCI Address bits used in decode as Local Address bits. ***Note:*** *Remap Address value must be a multiple of the Range (**not** the Range register).*	Yes	Yes	0h

Register 11-55. (LBRD1; PCI:F8h, LOC:178h) Local Address Space 1 Bus Region Descriptor

Bit	Description	Read	Write	Value after Reset
1:0	**Memory Space 1 Local Bus Width.** Writing a 00 indicates an 8-bit bus width. Writing a 01 indicates a 16-bit bus width. Writing a 10 or 11 indicates a 32-bit bus width.	Yes	Yes	M = 11 J = 11 C = 11
5:2	**Memory Space 1 Internal Wait States (data-to-data; 0-15 wait states).**	Yes	Yes	0h
6	**Memory Space 1 TA#/READY# Input Enable.** Writing a 1 enables TA#/READY# input. Writing a 0 disables TA#/READY# input.	Yes	Yes	1
7	**Memory Space 1 BTERM# Input Enable.** Writing a 1 enables BTERM# input. Writing a 0 disables BTERM# input. For more information, refer to Section 2.2.5 for M mode or Section 4.2.5 for C and J modes.	Yes	Yes	0
8	**Memory Space 1 Burst Enable.** Writing a 1 enables bursting. Writing a 0 disables bursting.	Yes	Yes	0
9	**Memory Space 1 Prefetch Disable.** When mapped into Memory space, writing a 0 enables Read prefetching. Writing a 1 disables prefetching. If prefetching is disabled, the PCI 9056 disconnects after each Memory read.	Yes	Yes	0
10	**Prefetch Count Enable.** When set to 1 and Memory prefetching is enabled, the PCI 9056 prefetches up to the number of Lwords specified in prefetch count. When set to 0, the PCI 9056 ignores the count and continues prefetching until it is terminated by the PCI Bus.	Yes	Yes	0
14:11	**Prefetch Counter.** Number of Lwords to prefetch during Memory Read cycles (0-15). A count of zero selects a prefetch of 16 Lwords.	Yes	Yes	0h
31:15	***Reserved.***	Yes	No	0h

Register 11-56. (DMDAC; PCI:FCh, LOC:17Ch) Direct Master PCI Dual Address Cycle Upper Address

Bit	Description	Read	Write	Value after Reset
31:0	**Upper 32 Bits of PCI Dual Address Cycle PCI Address during Direct Master Cycles.** If set to 0, the PCI 9056 performs 32-bit Direct Master Address access.	Yes	Yes	0h

Register 11-57. (PCIARB; PCI:100h, LOC:1A0h) PCI Arbiter Control

Bit	Description	Read	Write	Value after Reset
0	**Refer to the document, *PCI 9056 Blue Book Revision 0.91 Correction*, for the corrected version of this table entry.**			
1	**PCI 9056 High Priority.** Value of 0 indicates the PCI 9056 participates in round-robin arbitration with the other PCI Masters. Value of 1 indicates a two-level, round-robin arbitration scheme is enabled. The other PCI Bus Masters participate in their own round-robin arbitration. The winner of this arbitration then arbitrates for the PCI Bus with the PCI 9056 (when using the Internal PCI arbiter).	PCI/ Local	Yes	0
2	**Early Grant Release.** Value of 0 indicates the PCI 9056 keeps GNT# asserted until another Master requests use of the PCI Bus. Value of 1 indicates the PCI 9056 always de-asserts the current GNT# when FRAME# is asserted (when using the internal PCI arbiter).	PCI/ Local	Yes	0
3	**PCI Arbiter Parking on PCI 9056.** Value of 1 indicates the PCI arbiter parks the grant on the PCI 9056. Value of 0 indicates the PCI arbiter parks the grant on the current PCI Master (when using the internal PCI arbiter).	PCI/ Local	PCI/ Local	0
31:4	***Reserved.***	Y	N	0

Register 11-58. (PABTADR; PCI:104h, LOC:1A4h) PCI Abort Address

Bit	Description	Read	Write	Value after Reset
31:0	**PCI Abort Address.** When a PCI Master/Target Abort occurs, the starting address of the current access is returned to this register.	Yes	No	0000h

11.5 RUNTIME REGISTERS

Register 11-59. (MBOX0; PCI:40h or 78h, LOC:C0h) Mailbox Register 0

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.** ***Note:*** *Inbound Queue Port replaces Mailbox Register 0 when the I_2O function is enabled (QSR[0]=1). Mailbox Register 0 is always accessible at PCI address 78h and Local address C0h.*	Yes	Yes	0h

Register 11-60. (MBOX1; PCI:44h or 7Ch, LOC:C4h) Mailbox Register 1

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.** ***Note:*** *Mailbox Register 1 is replaced by Outbound Queue Port when the I_2O function is enabled (QSR[0]=1). Mailbox Register 1 is always accessible at PCI address 7Ch and Local address C4h.*	Yes	Yes	0h

Register 11-61. (MBOX2; PCI:48h, LOC:C8h) Mailbox Register 2

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.**	Yes	Yes	0h

Register 11-62. (MBOX3; PCI:4Ch, LOC:CCh) Mailbox Register 3

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.**	Yes	Yes	0h

Register 11-63. (MBOX4; PCI:50h, LOC:D0h) Mailbox Register 4

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.**	Yes	Yes	0h

Register 11-64. (MBOX5; PCI:54h, LOC:D4h) Mailbox Register 5

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.**	Yes	Yes	0h

Register 11-65. (MBOX6; PCI:58h, LOC:D8h) Mailbox Register 6

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.**	Yes	Yes	0h

Register 11-66. (MBOX7; PCI:5Ch, LOC:DCh) Mailbox Register 7

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.**	Yes	Yes	0h

Register 11-67. (P2LDBELL; PCI:60h, LOC:E0h) PCI-to-Local Doorbell

Bit	Description	Read	Write	Value after Reset
31:0	**Doorbell Register.** The PCI Bus Master can write to this register and assert a Local interrupt to the Local processor. The Local processor can then read this register to determine which doorbell bit was set. The PCI Bus Master sets the doorbell by writing a 1 to a particular bit. The Local processor can clear a doorbell bit by writing a 1 to that bit position.	Yes	Yes/Clr	0h

Register 11-68. (L2PDBELL; PCI:64h, LOC:E4h) Local-to-PCI Doorbell

Bit	Description	Read	Write	Value after Reset
31:0	**Doorbell Register.** The Local processor can write to this register and assert a PCI interrupt. The PCI Bus Master can then read this register to determine which doorbell bit was set. The Local processor sets the doorbell by writing a 1 to a particular bit. The PCI Bus Master can clear a doorbell bit by writing a 1 to that bit position.	Yes	Yes/Clr	0h

Register 11-69. (INTCSR; PCI:68h, LOC:E8h) Interrupt Control/Status

Bit	Description	Read	Write	Value after Reset
0	**Enable Local Bus TEA#/LSERR#.** Writing a 1 enables PCI 9056 to assert TEA#/LSERR# interrupt when the Received Master Abort bit is set (PCISR[13]=1 or INTCSR[6]=1).	Yes	Yes	0
1	Enable Local Bus TEA#/LSERR# when a PCI parity error occurs during a PCI 9056 Master Transfer or a PCI 9056 Slave access.	Yes	Yes	0
2	**Generate PCI Bus SERR# Interrupt.** When set to 0, writing 1 asserts the PCI Bus SERR# interrupt.	Yes	Yes	0
3	**Mailbox Interrupt Enable.** Writing a 1 enables a Local Interrupt to be asserted when the PCI Bus writes to MBOX0 through MBOX3. To clear a Local Interrupt, the Local Bus Master must read the Mailbox. Used in conjunction with the Local Interrupt Output Enable bit (INTCSR[16]).	Yes	Yes	0
4	**Power Management Interrupt Enable.** Writing a 1 enables a Local Interrupt to be asserted when the Power Management Power State changes.	Yes	Yes	0
5	**Power Management Interrupt.** When set to 1, indicates a Power Management interrupt is pending. A Power Management interrupt is caused by a change in the Power State register (PMCSR). Writing a 1 clears the interrupt.	Yes	Yes/Clr	0
6	**Direct Master Write/Direct Slave Read Local Data Parity Check Error Enable.** Writing a 1 enables a Local Data Parity error signal to be asserted through the LSERR#/TEA# pin. INTCSR[0] must be enabled for this to have an effect.	Yes	Yes	0
7	Direct Master **Write/Direct Slave Read Local Data Parity Check Error Status.** When set to 1, indicates the PCI 9056 has detected a Local Data Parity check error, even if the Check Parity Error bit is disabled. Writing 1 clears this bit to 0.	Yes	Yes/Clr	0
8	**PCI Interrupt Enable.** Writing a 1 enables PCI interrupts.	Yes	Yes	1
9	**PCI Doorbell Interrupt Enable.** Writing a 1 enables Doorbell interrupts. Used in conjunction with the PCI Interrupt Enable bit (INTCSR[8]). Clearing the doorbell interrupt bits that caused the interrupt also clears the interrupt.	Yes	Yes	0
10	**PCI Abort Interrupt Enable.** Values of 1 enables Master Abort or Master detect of a Target Abort to assert a PCI interrupt. Used in conjunction with the PCI Interrupt Enable bit (INTCSR[8]). Clearing the abort status bits also clears the PCI interrupt.	Yes	Yes	0
11	**Local Interrupt Input Enable.** Writing a 1 enables a Local interrupt input to assert a PCI interrupt. Used in conjunction with the PCI Interrupt Enable bit (INTCSR[8]). Clearing the Local Bus cause of the interrupt also clears the interrupt.	Yes	Yes	0
12	**Retry Abort Enable.** Writing a 1 enables the PCI 9056 to treat 256 Master consecutive Retries to a Target as a Target Abort. Writing a 0 enables the PCI 9056 to attempt Master Retries indefinitely.	Yes	Yes	0
13	**PCI Doorbell Interrupt Active.** When set to 1, indicates the PCI Doorbell interrupt is active.	Yes	No	0
14	**PCI Abort Interrupt Active.** When set to 1, indicates the PCI Abort interrupt is active.	Yes	No	0
15	**Local Input Interrupt Active.** When set to 1, indicates the Local Input interrupt is active.	Yes	No	0
16	**Local Interrupt Output Enable.** Writing a 1 enables Local interrupt output. Used in conjunction with the Mailbox Interrupt Enable bit (INTCSR[3]).	Yes	Yes	1
17	**Local Doorbell Interrupt Enable.** Writing a 1 enables Doorbell interrupts. Used in conjunction with the Local Interrupt Enable bit. Clearing the Local Doorbell Interrupt bits that caused the interrupt also clears the interrupt.	Yes	Yes	0

Register 11-69. (INTCSR; PCI:68h, LOC:E8h) Interrupt Control/Status (Continued)

Bit	Description	Read	Write	Value after Reset
18	**Local DMA Channel 0 Interrupt Enable.** Writing a 1 enables DMA Channel 0 interrupts. Used in conjunction with the Local Interrupt Enable bit. Clearing the DMA status bits also clears the interrupt.	Yes	Yes	0
19	**Local DMA Channel 1 Interrupt Enable.** Writing a 1 enables DMA Channel 1 interrupts. Used in conjunction with the Local Interrupt Enable bit. Clearing the DMA status bits also clears the interrupt.	Yes	Yes	0
20	**Local Doorbell Interrupt Active.** Reading a 1 indicates the Local Doorbell interrupt is active.	Yes	No	0
21	**DMA Channel 0 Interrupt Active.** Reading a 1 indicates the DMA Channel 0 interrupt is active.	Yes	No	0
22	**DMA Channel 1 Interrupt Active.** Reading a 1 indicates the DMA Channel 1 interrupt is active.	Yes	No	0
23	**BIST Interrupt Active.** Reading a 1 indicates the BIST interrupt is active. The BIST (built-in self test) interrupt is asserted by writing a 1 to bit 6 of the PCI Configuration BIST register. Clearing bit 6 clears the interrupt. Refer to the PCIBISTR register for a description of the self test.	Yes	No	0
24	Reading a 0 indicates the Direct Master was the Bus Master during a Master or Target Abort.	Yes	No	1
25	Reading a 0 indicates DMA Channel 0 was the Bus Master during a Master or Target Abort.	Yes	No	1
26	Reading a 0 indicates DMA Channel 1 was the Bus Master during a Master or Target Abort.	Yes	No	1
27	Reading a 0 indicates a Target Abort was asserted by the PCI 9056 after 256 consecutive Master retries to a Target.	Yes	No	1
28	Reading a 1 indicates the PCI Bus wrote data to MBOX0. Enabled only if the Mailbox Interrupt Enable bit is set (INTCSR[3]=1).	Yes	No	0
29	Reading a 1 indicates the PCI Bus wrote data to MBOX1. Enabled only if the Mailbox Interrupt Enable bit is set (INTCSR[3]=1).	Yes	No	0
30	Reading a 1 indicates the PCI Bus wrote data to MBOX2. Enabled only if the Mailbox Interrupt Enable bit is set (INTCSR[3]=1).	Yes	No	0
31	Reading a 1 indicates the PCI Bus wrote data to MBOX3. Enabled only if the Mailbox Interrupt Enable bit is set (INTCSR[3]=1).	Yes	No	0

Register 11-70. (CNTRL; PCI:6Ch, LOC:ECh) Serial EEPROM Control, PCI Command Codes, User I/O Control, and Init Control

Bit	Description	Read	Write	Value after Reset
3:0	**PCI Read Command Code for DMA.**	Yes	Yes	1110
7:4	**PCI Write Command Code for DMA.**	Yes	Yes	0111
11:8	**PCI Memory Read Command Code for Direct Master.**	Yes	Yes	0110
15:12	**PCI Memory Write Command Code for Direct Master.**	Yes	Yes	0111
16	**General Purpose Output.** Writing a 1 causes USERo output to go high. Writing a 0 causes USERo output to go low.	Yes	Yes	1
17	**General Purpose Input.** Reading a 1 indicates the USERi input pin is high. Reading a 0 indicates the USERi pin is low.	Yes	No	—
18	Writing a 1 selects USERi to be an input to the chip. Writing a 0 selects LLOCKi# as an input. Enables the user to select between the USERi and LLOCKi# functions when USERi is chosen to be an input. The select bit(s) for the pin is DMAMODE0[12] and/or DMAMODE1[12].	Yes	Yes	1
19	Writing a 1 selects USERo to be an output from the chip. Writing a 0 selects LLOCKo# as an output. Enables the user to select between the USERo and LLOCKo# functions when USERo is chosen to be an output. The select bit(s) for the pin is DMAMODE0[12] and/or DMAMODE1[12].	Yes	Yes	1
20	**LINTo# Interrupt Status.** When HOSTEN# is enabled, reading a 1 indicates the LINTo# interrupt is active by way of the INTA# PCI interrupt. Writing a 1 clears this bit.	Yes	Yes/Clr	0
21	**TEA#/LSERR# Interrupt Status.** When HOSTEN# is enabled, reading a 1 indicates the TEA#/LSERR# interrupt is active by way of the SERR# PCI System Error. Writing a 1 clears this bit.	Yes	Yes/Clr	0
23:22	***Reserved.***	Yes	No	00
24	**Serial EEPROM Clock for Local or PCI Bus Reads or Writes to Serial EEPROM.** Toggling this bit asserts the serial EEPROM clock. (Refer to manufacturer's data sheet for particular serial EEPROM being used.)	Yes	Yes	0
25	**Serial EEPROM Chip Select.** For Local or PCI Bus reads or writes to the serial EEPROM, setting this bit to 1 provides the serial EEPROM chip select.	Yes	Yes	0
26	**Write Bit to Serial EEPROM.** For writes, this output bit is input to the serial EEPROM. Clocked into the serial EEPROM by the serial EEPROM clock.	Yes	Yes	0
27	**Read Bit from Serial EEPROM.** (Refer to Sections 2.4.2 and 2.4.2.1 for M mode or Sections 4.4.2 and 4.4.2.1 for C and J modes.)	Yes	No	—
28	**Programmed Serial EEPROM Present.** When set to 1, indicates that a blank or programmed serial EEPROM is present.	Yes	No	0
29	**Reload Configuration Registers.** When set to 0, writing a 1 causes the PCI 9056 to reload the Local Configuration registers from the serial EEPROM.	Yes	Yes	0
30	**PCI Adapter Software Reset when HOSTEN#=1.** Writing a 1 holds the PCI 9056 Local Bus logic in a reset state, and asserts LRESET# output. Contents of the PCI Configuration registers and the shared Runtime registers are not reset. A software reset can only be cleared from the PCI Bus. **PCI Host Software Reset when HOSTEN#=0.** Writing a 1 holds the PCI 9056 PCI Bus logic in a reset state, and asserts RST# output. Contents of the Local Configuration, shared Runtime, DMA, and Messaging Queue registers are not reset. A software reset can only be cleared from the Local Bus.	Yes	Yes	0
31	**EEDO Input Enable.** When set to 1, the EEDI/EEDO I/O buffer is placed in Bus high-impedance state, enabling the serial EEPROM data to be read. The serial EEPROM data resides in CNTRL[27].	Yes	No	0

Register 11-71. (PCIHIDR; PCI:70h, LOC:F0h) PCI Hardwired Configuration ID

Bit	Description	Read	Write	Value after Reset
15:0	**Vendor ID.** Identifies manufacturer of device. Hardwired to the PCI SIG-issued Vendor ID of PLX (10B5h).	Yes	No	10B5h
31:16	**Device ID.** Identifies particular device. Hardwired to the PLX part number for PCI interface chip 9056h.	Yes	No	9056h

Register 11-72. (PCIHREV; PCI:74h, LOC:F4h) PCI Hardwired Revision ID

Bit	Description	Read	Write	Value after Reset
7:0	**Revision ID.** Hardwired silicon revision of the PCI 9056.	Yes	No	Current Rev # (AA)

11.6 DMA REGISTERS

Register 11-73. (DMAMODE0; PCI:80h, LOC:100h) DMA Channel 0 Mode

Bit	Description	Read	Write	Value after Reset
1:0	**Local Bus Width.** Writing a 00 indicates an 8-bit bus width. Writing a 01 indicates a 16-bit bus width. Writing a 10 or 11 indicates a 32-bit bus width.	Yes	Yes	M = 11 J = 11 C = 11
5:2	**Internal Wait States (data-to-data).**	Yes	Yes	0h
6	**TA#/READY# Input Enable.** Writing a 1 enables TA#/READY# input. Writing a 0 disables TA#/READY# input.	Yes	Yes	1
7	**BTERM# Input Enable.** Writing a 1 enables BTERM# input. Writing a 0 disables BTERM# input. For more information, refer to Section 2.2.5 for M mode or Section 4.2.5 for C and J modes.	Yes	Yes	0
8	**Local Burst Enable.** Writing a 1 enables Local bursting. Writing a 0 disables Local bursting.	Yes	Yes	0
9	**Scatter/Gather Mode.** Writing a 1 indicates Scatter/Gather mode is enabled. For Scatter/Gather mode, DMA source address, destination address, and byte count are loaded from memory in PCI or Local Address spaces. Writing a 0 indicates Block mode is enabled.	Yes	Yes	0
10	**Done Interrupt Enable.** Writing a 1 enables an interrupt when done. Writing a 0 disables an interrupt when done. If DMA Clear Count mode is enabled, the interrupt does not occur until the byte count is cleared.	Yes	Yes	0
11	**Local Addressing Mode.** Writing a 1 holds the Local Address bus constant. Writing a 0 indicates the Local Address is incremented.	Yes	Yes	0
12	**Demand Mode.** Writing a 1 causes the DMA controller to operate in Demand mode. In Demand mode, the DMA controller transfers data when its DREQ0# input is asserted. Asserts DACK0# to indicate the current Local Bus transfer is in response to DREQ0# input. DMA controller transfers Lwords (32 bits) of data. This may result in multiple transfers for an 8- or 16-bit bus.	Yes	Yes	0
13	**Memory Write and Invalidate Mode for DMA Transfers.** When set to 1, the PCI 9056 performs Memory Write and Invalidate cycles to the PCI Bus. The PCI 9056 supports Memory Write and Invalidate sizes of 8 or 16 Lwords. Size is specified in the System Cache Line Size bits (PCICLSR[7:0]). If a size other than 8 or 16 is specified, the PCI 9056 performs Write transfers rather than Memory Write and Invalidate transfers. Transfers must start and end at cache line boundaries.	Yes	Yes	0
14	**DMA EOT# Enable.** Writing a 1 enables the EOT# input pin. Writing a 0 disables the EOT# input pin.	Yes	Yes	0

Register 11-73. (DMAMODE0; PCI:80h, LOC:100h) DMA Channel 0 Mode (Continued)

Bit	Description	Read	Write	Value after Reset
15	**Fast/Slow Terminate Mode Select.** Writing a 0 sets PCI 9056 into the Slow Terminate mode. As a result in C or J modes, BLAST# is asserted on the last Data transfer to terminate DMA transfer. As a result in M mode, BDIP# is de-asserted at the nearest 16-byte boundary and stops the DMA transfer. Writing a 1 indicates that if EOT# is asserted or DREQ0# is de-asserted in Demand mode during DMA will immediately terminate the DMA transfer. In M mode, writing a 1 indicates BDIP# output is disabled. As a result, the PCI 9056 DMA transfer terminates immediately when EOT# is asserted or when DREQ0# is de-asserted in Demand mode.	Yes	Yes	0
16	**DMA Clear Count Mode.** Writing a 1 clears the byte count in each Scatter/Gather descriptor when the corresponding DMA transfer is complete.	Yes	Yes	0
17	**DMA Channel 0 Interrupt Select.** Writing a 1 routes the DMA Channel 0 interrupt to the PCI Bus interrupt. Writing a 0 routes the DMA Channel 0 interrupt to the Local Bus interrupt.	Yes	Yes	0
18	**DAC Chain Load.** When set to 1, enables the descriptor to load the PCI Dual Address Cycle value. Otherwise, it uses the register contents.	Yes	Yes	0
19	**EOT# End Link.** Used only for Scatter/Gather DMA transfers. When EOT# is asserted, value of 1 indicates the DMA transfer ends the current Scatter/Gather link and continues with the remaining Scatter/Gather transfers. When EOT# is asserted, value of 0 indicates the DMA transfer ends the current Scatter/Gather transfer and does not continue with the remaining Scatter/Gather transfers.	Yes	Yes	0
20	**Valid Mode Enable.** Value of 0 indicates the Valid bit (DMASIZ0[31]) is ignored. Value of 1 indicates the DMA descriptors are processed only when the Valid bit is set (DMASIZ0[31]). If the Valid bit is set, the transfer count is 0, and the descriptor is not the last descriptor in the chain. The DMA controller then moves to the next descriptor in the chain.	Yes	Yes	0
21	**Valid Stop Control.** Value of 0 indicates the DMA Chaining controller continuously polls a descriptor with the Valid bit set to 0 (invalid descriptor) if the Valid Mode Enable bit is set (bit [20]=1). Value of 1 indicates the Chaining controller stops polling when the Valid bit with a value of 0 is detected (DMASIZ0[31]=0). In this case, the CPU must restart the DMA controller by setting the Start bit (DMACSR0[1]=1). A pause sets the DMA Done bit (DMACSR0[4]).	Yes	Yes	0
31:22	***Reserved.***	Yes	No	0h

Register 11-74. (DMAPADR0; (PCI:84h, LOC:104h when DMAMODE0[20]=0 or PCI:88h, LOC:108h when DMAMODE0[20]=1) DMA Channel 0 PCI Address

Bit	Description	Read	Write	Value after Reset
31:0	**PCI Address Register.** Indicates from where in PCI Memory space DMA transfers (reads or writes) start.	Yes	Yes	0h

Register 11-75. (DMALADR0; PCI:88h, LOC:108h when DMAMODE0[20]=0 or PCI:8Ch, LOC:10Ch when DMAMODE0[20]=1) DMA Channel 0 Local Address

Bit	Description	Read	Write	Value after Reset
31:0	**Local Address Register.** Indicates from where in Local Memory space DMA transfers (reads or writes) start.	Yes	Yes	0h

Register 11-76. (DMASIZ0; PCI:8Ch, LOC:10Ch when DMAMODE0[20]=0 or PCI:84h, LOC:104h when DMAMODE0[20]=1) DMA Channel 0 Transfer Size (Bytes)

Bit	Description	Read	Write	Value after Reset
22:0	**DMA Transfer Size (Bytes).** Indicates the number of bytes to transfer during a DMA operation.	Yes	Yes	0h
30:23	***Reserved.***	Yes	No	0h
31	**Valid.** When the Valid Mode Enable bit is set (DMAMODE0[20]=1), indicates the validity of this DMA descriptor.	Yes	Yes	0h

Register 11-77. (DMADPR0; PCI:90h, LOC:110h) DMA Channel 0 Descriptor Pointer

Bit	Description	Read	Write	Value after Reset
0	**Descriptor Location.** Writing a 1 indicates PCI Address space. Writing a 0 indicates Local Address space.	Yes	Yes	0
1	**End of Chain.** Writing a 1 indicates end of chain. Writing a 0 indicates not end of chain descriptor. (Same as Block mode.)	Yes	Yes	0
2	**Interrupt after Terminal Count.** Writing a 1 causes an interrupt to be asserted after the terminal count for this descriptor is reached. Writing a 0 disables interrupts from being asserted.	Yes	Yes	0
3	**Direction of Transfer.** Writing a 1 indicates transfers from the Local Bus to the PCI Bus. Writing a 0 indicates transfers from the PCI Bus to the Local Bus.	Yes	Yes	0
31:4	**Next Descriptor Address.** Qword-aligned (bits [3:0]=0000).	Yes	Yes	0h

Register 11-78. (DMAMODE1; PCI:94h, LOC:114h) DMA Channel 1 Mode

Bit	Description	Read	Write	Value after Reset
1:0	**Local Bus Width.** Writing a 00 indicates an 8-bit bus width. Writing a 01 indicates a 16-bit bus width. Writing a 10 or 11 indicates a 32-bit bus width.	Yes	Yes	M = 11 J = 11 C = 11
5:2	**Internal Wait States (data-to-data).**	Yes	Yes	0h
6	**TA#/READY# Input Enable.** Writing a 1 enables TA#/READY# input. Writing a 0 disables TA#/READY# input.	Yes	Yes	1
7	**BTERM# Input Enable.** Writing a 1 enables BTERM# input. Writing a 0 disables BTERM# input. For more information, refer to Section 2.2.5 for M mode or Section 4.2.5 for C and J modes.	Yes	Yes	0
8	**Local Burst Enable.** Writing a 1 enables Local bursting. Writing a 0 disables Local bursting.	Yes	Yes	0
9	**Scatter/Gather Mode.** Writing a 1 indicates Scatter/Gather mode is enabled. For Scatter/Gather mode, the DMA source address, destination address, and byte count are loaded from memory in PCI or Local Address spaces. Writing a 0 indicates Block mode is enabled.	Yes	Yes	0
10	**Done Interrupt Enable.** Writing a 1 enables interrupt when done. Writing a 0 disables the interrupt when done. If DMA Clear Count mode is enabled, the interrupt does not occur until the byte count is cleared.	Yes	Yes	0
11	**Local Addressing Mode.** Writing a 1 holds the Local address bus constant. Writing a 0 indicates the Local address is incremented.	Yes	Yes	0

Register 11-78. (DMAMODE1; PCI:94h, LOC:114h) DMA Channel 1 Mode (Continued)

Bit	Description	Read	Write	Value after Reset
12	**Demand Mode.** Writing a 1 causes the DMA controller to operate in Demand mode. In Demand mode, the DMA controller transfers data when its DREQ1# input is asserted. Asserts DACK1# to indicate the current Local Bus transfer is in response to DREQ1# input. DMA controller transfers Lwords (32 bits) of data. This may result in multiple transfers for an 8- or 16-bit bus.	Yes	Yes	0
13	**Memory Write and Invalidate Mode for DMA Transfers.** When set to 1, the PCI 9056 performs Memory Write and Invalidate cycles to the PCI Bus. The PCI 9056 supports Memory Write and Invalidate sizes of 8 or 16 Lwords. Size is specified in the System Cache Line Size bits (PCICLSR[7:0]). If a size other than 8 or 16 is specified, the PCI 9056 performs Write transfers rather than Memory Write and Invalidate transfers. Transfers must start and end at cache line boundaries.	Yes	Yes	0
14	**DMA EOT# Enable.** Writing a 1 enables the EOT# input pin. Writing a 0 disables the EOT# output pin.	Yes	Yes	0
15	**Fast/Slow Terminate Mode Select.** Writing a 0 sets the PCI 9056 into Slow Terminate mode. As a result in C or J modes, BLAST# is asserted to terminate the DMA transfer. As a result in M mode, BDIP# is de-asserted at the nearest 16-byte boundary and stops the DMA transfer. Writing a 1 indicates that asserting EOT# during DMA will terminate the DMA transfer. In M mode, writing a 1 indicates BDIP# output is disabled. As a result, the PCI 9056 DMA transfer terminates immediately when EOT# is asserted.	Yes	Yes	0
16	**DMA Clear Count Mode.** When set to 1, the byte count in each Scatter/Gather descriptor is cleared when the corresponding DMA transfer is complete.	Yes	Yes	0
17	**DMA Channel 1 Interrupt Select.** Writing a 1 routes the DMA Channel 1 interrupt to the PCI Bus interrupt. Writing a 0 routes the DMA Channel 1 interrupt to the Local Bus interrupt.	Yes	Yes	0
18	**DAC Chain Load.** When set to 1, enables the descriptor to load the PCI Dual Address Cycle value. Otherwise, it uses the register contents.	Yes	Yes	0
19	**EOT# End Link.** Used only for DMA Scatter/Gather transfers. When EOT# is asserted, value of 1 indicates the DMA transfer ends the current Scatter/Gather link and continues with the remaining Scatter/Gather transfers. When EOT# is asserted, value of 0 indicates the DMA transfer completes the current Scatter/Gather transfer, but does not continue with the remaining Scatter/Gather transfers.	Yes	Yes	0
20	**Valid Mode Enable.** Value of 0 indicates the Valid bit (DMASIZ1[31]) is ignored. Value of 1 indicates the DMA descriptors are processed only when the Valid bit is set (DMASIZ1[31]). If the Valid bit is set, the transfer count is 0, and the descriptor is not the last descriptor in the chain. The DMA controller then moves to the next descriptor in the chain.	Yes	Yes	0
21	**Valid Stop Control.** Value of 0 indicates the DMA Scatter/Gather controller continuously polls a descriptor with the Valid bit set to 0 (invalid descriptor) if the Valid Mode Enable bit is set (bit [20]=1). Value of 1 indicates the Scatter/Gather controller stops polling when the Valid bit with a value of 0 is detected (DMASIZ1[31]=0). In this case, the CPU must restart the DMA controller by setting the Start bit (DMACSR1[1]=1). A pause sets the DMA Done bit (DMASCR1[4]).	Yes	Yes	0
31:22	***Reserved.***	Yes	No	0h

Register 11-79. (DMAPADR1;PCI:98h, LOC:118h when DMAMODE1[20]=0 or PCI:9Ch, LOC:11Ch when DMAMODE1[20]=1) DMA Channel 1 PCI Address

Bit	Description	Read	Write	Value after Reset
31:0	**PCI Address Register.** Indicates from where in PCI Memory space DMA transfers (reads or writes) start.	Yes	Yes	0h

Register 11-80. (DMALADR1;PCI:9Ch, LOC:11Ch when DMAMODE1[20]=0 or PCI:A0h, LOC:120h when DMAMODE1[20]=1) DMA Channel 1 Local Address

Bit	Description	Read	Write	Value after Reset
31:0	**Local Address Register.** Indicates from where in Local Memory space DMA transfers (reads or writes) start.	Yes	Yes	0h

Register 11-81. (DMASIZ1; PCI:A0h, LOC:120h when DMAMODE1[20]=0 or PCI:98h, LOC:118h when DMAMODE1[20]=1) DMA Channel 1 Transfer Size (Bytes)

Bit	Description	Read	Write	Value after Reset
22:0	**DMA Transfer Size (Bytes).** Indicates the number of bytes to transfer during a DMA operation.	Yes	Yes	0h
30:23	***Reserved.***	Yes	No	0h
31	**Valid.** When the Valid Mode Enable bit is set (DMAMODE1[20]=1), indicates the validity of this DMA descriptor.	Yes	Yes	0h

Register 11-82. (DMADPR1; PCI:A4h, LOC:124h) DMA Channel 1 Descriptor Pointer

Bit	Description	Read	Write	Value after Reset
0	**Descriptor Location.** Writing a 1 indicates PCI Address space. Writing a 0 indicates Local Address space.	Yes	Yes	0
1	**End of Chain.** Writing a 1 indicates end of chain. Writing a 0 indicates not end of chain descriptor. (Same as Block mode.)	Yes	Yes	0
2	**Interrupt after Terminal Count.** Writing a 1 causes an interrupt to be asserted after the terminal count for this descriptor is reached. Writing a 0 disables interrupts from being asserted.	Yes	Yes	0
3	**Direction of Transfer.** Writing a 1 indicates transfers from the Local Bus to the PCI Bus. Writing a 0 indicates transfers from the PCI Bus to the Local Bus.	Yes	Yes	0
31:4	**Next Descriptor Address.** Qword-aligned (bits [3:0]=0000).	Yes	Yes	0h

Register 11-83. (DMACSR0; PCI:A8h, LOC:128h) DMA Channel 0 Command/Status

Bit	Description	Read	Write	Value after Reset
0	**Channel 0 Enable.** Writing a 1 enables channel to transfer data. Writing a 0 disables the channel from starting a DMA transfer, and if in the process of transferring data, suspends the transfer (pause).	Yes	Yes	0
1	**Channel 0 Start.** Writing a 1 causes the channel to start transferring data if the channel is enabled.	No	Yes/Set	0
2	**Channel 0 Abort.** Writing a 1 causes the channel to abort current transfer. Channel 0 Enable bit must be cleared (bit [0]=0). Sets Channel 0 Done (bit [4]=1) when abort is complete.	No	Yes/Set	0
3	**Channel 0 Clear Interrupt.** Writing a 1 clears Channel 0 interrupts.	No	Yes/Clr	0
4	**Channel 0 Done.** Reading a 1 indicates a channel transfer is complete. Reading a 0 indicates a channel transfer is not complete.	Yes	No	1
7:5	***Reserved.***	Yes	No	000

Register 11-84. (DMACSR1; PCI:A9h, LOC:129h) DMA Channel 1 Command/Status

Bit	Description	Read	Write	Value after Reset
0	**Channel 1 Enable.** Writing a 1 enables channel to transfer data. Writing a 0 disables the channel from starting a DMA transfer, and if in the process of transferring data, suspends the transfer (pause).	Yes	Yes	0
1	**Channel 1 Start.** Writing a 1 causes channel to start transferring data if the channel is enabled.	No	Yes/Set	0
2	**Channel 1 Abort.** Writing a 1 causes channel to abort current transfer. Channel 1 Enable bit must be cleared (bit [0]=0). Sets Channel 1 Done (bit [4]=1) when abort is complete.	No	Yes/Set	0
3	**Channel 1 Clear Interrupt.** Writing a 1 clears Channel 1 interrupts.	No	Yes/Clr	0
4	**Channel 1 Done.** Reading a 1 indicates a channel transfer is complete. Reading a 0 indicates a channel transfer is not complete.	Yes	No	1
7:5	***Reserved.***	Yes	No	000

Register 11-85. (DMAARB; PCI:ACh, LOC:12Ch) DMA Arbitration

Same as Register 11-40 “(MARBR; PCI:08h or ACh, LOC:88h or 12Ch) Mode/DMA Arbitration,” on page 11-21.

Register 11-86. (DMATHR; PCI:B0h, LOC:130h) DMA Threshold

Bit	Description	Read	Write	Value after Reset
3:0	**DMA Channel 0 PCI-to-Local Almost Full (C0PLAF).** Number of full entries (divided by two, minus one) in the FIFO before requesting the Local Bus for writes. (C0PLAF+1) + (C0PLAE+1) should be ≤ a FIFO Depth of 32.	Yes	Yes	0h
7:4	**DMA Channel 0 Local-to-PCI Almost Empty (C0LPAE).** Number of empty entries (divided by two, minus one) in the FIFO before requesting the Local Bus for reads. (C0LPAF+1) + (C0LPAE+1) should be ≤ a FIFO depth of 32.	Yes	Yes	0h
11:8	**DMA Channel 0 Local-to-PCI Almost Full (C0LPAF).** Number of full entries (divided by two, minus one) in the FIFO before requesting the PCI Bus for writes.	Yes	Yes	0h
15:12	**DMA Channel 0 PCI-to-Local Almost Empty (C0PLAE).** Number of empty entries (divided by two, minus one) in the FIFO before requesting the PCI Bus for reads.	Yes	Yes	0h
19:16	**DMA Channel 1 PCI-to-Local Almost Full (C1PLAF).** Number of full entries (divided by two, minus one) in the FIFO before requesting the Local Bus for writes. (C1PLAF+1) + (C1PLAE+1) should be ≤ a FIFO Depth of 32.	Yes	Yes	0h
23:20	**DMA Channel 1 Local-to-PCI Almost Empty (C1LPAE).** Number of empty entries (divided by two, minus one) in the FIFO before requesting the Local Bus for reads. (C1LPAF+1) + (C1LPAE+1) should be ≤ a FIFO depth of 32.	Yes	Yes	0h
27:24	**DMA Channel 1 Local-to-PCI Almost Full (C1LPAF).** Number of full entries (divided by two, minus one) in the FIFO before requesting the PCI Bus for writes.	Yes	Yes	0h
31:28	**DMA Channel 1 PCI-to-Local Almost Empty (C1PLAE).** Number of empty entries (divided by two, minus one) in the FIFO before requesting the PCI Bus for reads.	Yes	Yes	0h

Note: For DMA Channel 0 only, *if number of entries needed is x, then the value is one less than half the number of entries (that is, x/2 - 1).*

Register 11-87. (DMADAC0; PCI:B4h, LOC:134h) DMA Channel 0 PCI Dual Address Cycle Upper Address

Bit	Description	Read	Write	Value after Reset
31:0	**Upper 32 Bits of the PCI Dual Address Cycle PCI Address during DMA Channel 0 Cycles.** If set to 0h, the PCI 9056 performs a 32-bit DMA Channel 0 Address access.	Yes	Yes	0h

Register 11-88. (DMADAC1; PCI:B8h, LOC:138h) DMA Channel 1 PCI Dual Address Cycle Upper Address

Bit	Description	Read	Write	Value after Reset
31:0	**Upper 32 Bits of the PCI Dual Address Cycle PCI Address during DMA Channel 1 Cycles.** If set to 0h, the PCI 9056 performs a 32-bit DMA Channel 1 Address access.	Yes	Yes	0h

11.7 MESSAGING QUEUE REGISTERS

Register 11-89. (OPQIS; PCI:30h, LOC:B0h) Outbound Post Queue Interrupt Status

Bit	Description	Read	Write	Value after Reset
2:0	***Reserved.***	Yes	No	000
3	**Outbound Post Queue Interrupt.** Set when the Outbound Post Queue is not empty. Not affected by the Interrupt Mask bit.	Yes	No	0
31:4	***Reserved.***	Yes	No	0h

Register 11-90. (OPQIM; PCI:34h, LOC:B4h) Outbound Post Queue Interrupt Mask

Bit	Description	Read	Write	Value after Reset
2:0	***Reserved.***	Yes	No	000
3	**Outbound Post Queue Interrupt Mask.** Writing a 1 masks the interrupt.	Yes	Yes	1
31:4	***Reserved.***	Yes	No	0h

Register 11-91. (IQP; PCI:40h) Inbound Queue Port

Bit	Description	Read	Write	Value after Reset
31:0	Value written by the PCI Master is stored into the Inbound Post Queue, which is located in Local memory at the address pointed to by the Queue Base Address + Queue Size + Inbound Post Head Pointer. From the time of the PCI write until the Local Memory write and update of the Inbound Post Queue Head Pointer, further accesses to this register result in a Retry. A Local interrupt is asserted when the Inbound Post Queue is not empty. When the port is read by the PCI Master, the value is read from the Inbound Free Queue, which is located in Local memory at the address pointed to by the Queue Base Address + Inbound Free Tail Pointer. If the queue is empty, FFFFFFFFh is returned.	PCI	PCI	0h

Register 11-92. (OQP; PCI:44h) Outbound Queue Port

Bit	Description	Read	Write	Value after Reset
31:0	Value written by the PCI Master is stored into the Outbound Free Queue, which is located in Local memory at the address pointed to by the Queue Base Address + 3*Queue Size + Outbound Free Head Pointer. From the time of the PCI write until the Local Memory write and update of the Outbound Free Queue Head Pointer, further accesses to this register result in a Retry. If the queue fills up, a Local NMI interrupt is asserted. When the port is read by the PCI Master, the value is read from the Outbound Post Queue, which is located in Local memory at the address pointed to by the Queue Base Address + 2*Queue Size + Outbound Post Tail Pointer. If the queue is empty, FFFFFFFFh is returned. A PCI interrupt is asserted if the Outbound Post Queue is not empty.	PCI	PCI	0h

Register 11-93. (MQCR; PCI:C0h, LOC:140h) Messaging Queue Configuration

Bit	Description	Read	Write	Value after Reset
0	**Queue Enable.** Writing a 1 allows accesses to the Inbound and Outbound Queue ports. If cleared to 0, writes are accepted but ignored and reads return FFFFFFFFh.	Yes	Yes	0
5:1	**Circular Queue Size.** Contains the size of one of the circular FIFO queues. Each of the four queues are the same size. Queue Size Encoding values: Bits [5:1] Number of entries Total size 00001 4-KB entries 64 KB 00010 8-KB entries 128 KB 00100 16-KB entries 256 KB 01000 32-KB entries 512 KB 10000 64-KB entries 1 MB	Yes	Yes	00001
31:6	***Reserved.***	Yes	No	0h

Register 11-94. (QBAR; PCI:C4h, LOC:144h) Queue Base Address

Bit	Description	Read	Write	Value after Reset
19:0	***Reserved.***	Yes	No	0h
31:20	**Queue Base Address.** Local Memory base address of circular queues. Queues must be aligned on a 1-MB boundary.	Yes	Yes	0h

Register 11-95. (IFHPR; PCI:C8h, LOC:148h) Inbound Free Head Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00
19:2	**Inbound Free Head Pointer.** Local Memory Offset for the Inbound Free Queue. Maintained by the Local CPU software.	Yes	Yes	0h
31:20	**Queue Base Address.**	Yes	No	0h

Register 11-96. (IFTPR; PCI:CCh, LOC:14Ch) Inbound Free Tail Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00
19:2	**Inbound Free Tail Pointer.** Local Memory offset for the Inbound Free Queue. Maintained by the hardware and incremented modulo the queue size.	Yes	Yes	0h
31:20	**Queue Base Address.**	Yes	No	0h

Register 11-97. (IPHPR; PCI:D0h, LOC:150h) Inbound Post Head Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00
19:2	**Inbound Post Head Pointer.** Local Memory offset for the Inbound Post Queue. Maintained by the hardware and incremented modulo the queue size.	Yes	Yes	0h
31:20	**Queue Base Address.**	Yes	No	0h

Register 11-98. (IPTPR; PCI:D4h, LOC:154h) Inbound Post Tail Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00
19:2	**Inbound Post Tail Pointer.** Local Memory offset for the Inbound Post Queue. Maintained by the Local CPU software.	Yes	Yes	0h
31:20	**Queue Base Address.**	Yes	No	0h

Register 11-99. (OFHPR; PCI:D8h, LOC:158h) Outbound Free Head Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00
19:2	**Outbound Free Head Pointer.** Local Memory offset for the Outbound Free Queue. Maintained by the hardware and incremented modulo the queue size.	Yes	Yes	0h
31:20	**Queue Base Address.**	Yes	No	0h

Register 11-100. (OFTPR; PCI:DCh, LOC:15Ch) Outbound Free Tail Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00
19:2	**Outbound Free Tail Pointer.** Local Memory offset for the Outbound Free Queue. Maintained by the Local CPU software.	Yes	Yes	0h
31:20	**Queue Base Address.**	Yes	No	0h

Register 11-101. (OPHPR; PCI:E0h, LOC:160h) Outbound Post Head Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00
19:2	**Outbound Post Head Pointer.** Local Memory offset for the Outbound Post Queue. Maintained by the Local CPU software.	Yes	Yes	0h
31:20	**Queue Base Address.**	Yes	No	0h

Register 11-102. (OPTPR; PCI:E4h, LOC:164h) Outbound Post Tail Pointer

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00
19:2	**Outbound Post Tail Pointer.** Local Memory offset for the Outbound Post Queue. Maintained by the hardware and incremented modulo the queue size.	Yes	Yes	0h
31:20	**Queue Base Address.**	Yes	No	0h

Register 11-103. (QSR; PCI:E8h, LOC:168h) Queue Status/Control

Bit	Description	Read	Write	Value after Reset
0	**I_2O Decode Enable.** When set, replaces the MBOX0 and MBOX1 registers with the Inbound and Outbound Queue Port registers and redefines Space 1 as PCI Base Address 0 to be accessed by PCIBAR0. Former Space 1 registers LAS1RR, LAS1BA, and LBRD1 should be programmed to configure their shared I_2O Memory space, defined as PCI Base Address 0.	Yes	Yes	0
1	**Queue Local Space Select.** When set to 0, use the Local Address Space 0 Bus Region descriptor for Queue accesses. When set to 1, use the Local Address Space 1 Bus Region descriptor for Queue accesses.	Yes	Yes	0
2	**Outbound Post Queue Prefetch Enable.** Writing a 1 causes prefetching to occur from the Outbound Post Queue if it is not empty.	Yes	Yes	0
3	**Inbound Free Queue Prefetch Enable.** Writing a 1 causes prefetching to occur from the Inbound Free Queue if it is not empty.	Yes	Yes	0
4	**Inbound Post Queue Interrupt Mask.** Writing a 1 masks the interrupt.	Yes	Yes	1
5	**Inbound Post Queue Interrupt Not Empty.** Set when the Inbound Post Queue is not empty. Not affected by the Interrupt Mask bit.	Yes	No	0
6	**Outbound Free Queue Overflow Interrupt Mask.** When set to 1, masks the interrupt. Value of 0 clears the mask.	Yes	Yes	1
7	**Outbound Free Queue Overflow Interrupt Full.** Set when the Outbound Free Queue becomes full. A Local TEA#/LSERR# (NMI) interrupt is asserted. Writing a 1 clears the interrupt.	Yes	Yes/Clr	0
31:8	***Unused.***	Yes	No	0h

12 PIN DESCRIPTION

12.1 PIN SUMMARY

Tables in this section describe each PCI 9056 pin. Table 12-4 through Table 12-9 provide pin information common to all Local Bus modes of operation:

- PCI System Bus Interface
- JTAG
- CompactPCI Hot Swap
- System
- Serial EEPROM Interface
- Power and Ground

Pins in Table 12-10 through Table 12-12 correspond to the PCI 9056 Local Bus modes—M, C, and J:

- M Bus Mode Interface Pin Description (32-bit address/32-bit data, non-multiplexed)
- C Bus Mode Interface Pin Description (32-bit address/32-bit data, non-multiplexed)
- J Bus Mode Interface Pin Description (32-bit address/32-bit data, multiplexed)

For a visual view of the chip pinout, refer to Section 14, “Physical Specs.”

All Local Bus internal pull-up resistors go through a 50K-ohm resistor. All Local Bus internal pull-down resistors go through a 50K-ohm resistor.

All Local I/O pins should have external pull-up or pull-down resistors, which depend upon the application and pin polarity. (Use approximately 3K to 10K ohms.) This is recommended due to the weak value of the internal pull-up and pull-down resistors.

Unspecified pins are not connected (NC).

Note for PCI Pins: ***DO NOT*** *pull any pins up or down unless the PCI 9056 is being used in an embedded design. Refer to PCI r2.2, page 138.*

The IDDQEN# pin has an internal pull-down resistor.

The pins in the following table have internal pull-up resistors.

Table 12-1. Pins with Internal Pull-Up Resistors

ADS#	BDIP#	BI#	BIGEND#/ WAIT#
BLAST#	BTERM#	BURST#	CCS#
DACK0#	DACK1#	DP[3:0]	DREQ0#
DREQ1#	EEDI/EEDO	HOSTEN#	LA[28:2] C, J Modes
LA[31:30] C, J Modes	LA[0:1] M Mode	LA[3:31] M Mode	LAD[31:0] J Mode
LBE[3:0]#	LD[31:0] C Mode	LD[0:31] M Mode	LINTi#
LINTo#	LRESET#	LSERR#	LW/R#
MDREQ#/ DMPAF/EOT#	PMEREQ#	RD/WR#	READY#
TA#	TEA#	TS#	TSIZ[0:1]
WAIT#			

The pins in the following table have no internal resistors. A pull-up or pull-down resistor is recommended, based upon the pin functionality.

Table 12-2. Pins with No Internal Resistors

BB#	BG#	BR#	BREQi
BREQo	EECS	EESK	LA29 C, J Modes
LA2 M Mode	LHOLD	LHOLDA	MODE0
MODE1	RETRY#	USERo/ LLOCKo#	USERi/ LLOCKi#

Notes: *Due to the complexity of pin multiplexing, LA[29] (C mode) or LA[2] (M mode), requires an external pull-up resistor. In J mode, ALE requires a pull-down resistor.*

Refer to Table 2-18 or Table 4-18 for pull-up and pull-down resistor requirements for the EEDI/EEDO pin.

The following table lists abbreviations used in this section to represent various pin types.

Table 12-3. Pin Type Abbreviations

Abbreviation	Pin Type
I/O	Input and output
I	Input only
O	Output only
TS	Three-state
OC or OD	Open collector or open drain
TP	Totem pole
STS	Sustained three-state, driven high for one CLK before float
DTS	Driven three-state, driven high for one-half CLK before float

12.2 PINOUT COMMON TO ALL BUS MODES

Table 12-4. PCI System Bus Interface Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
AD[31:0]	Address and Data	32	I/O TS PCI	C4, A2, D4, C3, B2, B1, A1, C2, C1, D2, E3, D1, F3, E1, F2, F1, K3, L1, L2, L3, M1, N1, L4, M3, P1, N3, M4, P2, T1, R1, R2, P3	PCI multiplexed address/data bus.
C/BE[3:0]#	Bus Command and Byte Enables	4	I/O TS PCI	E4, G3, K2, N2	All multiplexed on the same PCI pins. During the Address phase of a transaction, defines the bus command. During the Data phase, used as byte enables. Refer to the *PCI r2.2* for further detail if needed.
DEVSEL#	Device Select	1	I/O STS PCI	H3	When actively driven, indicates the driving device has decoded its address as the Target of the current access. As input, indicates whether any device on the bus is selected.
FRAME#	Cycle Frame	1	I/O STS PCI	G2	Driven by the current Master to indicate beginning and duration of an access. FRAME# is asserted to indicate a bus transaction is beginning. While FRAME# is asserted, Data transfers continue. When FRAME# is de-asserted, the transaction is in the final Data phase.
GNT0# REQ#	Internal Arbiter Grant 0 External Arbiter Request	1	O O STS PCI	A3	GNT0#: When the internal PCI arbiter is enabled, the PCI GNT0# signal is an output to an arbitrating master. The PCI 9056 arbiter asserts GNT0# to grant the PCI Bus to the master. REQ#: When the internal PCI arbiter is disabled, GNT0# becomes the REQ# output from the PCI 9056 to an external arbiter. The PCI 9056 asserts REQ# to request the PCI Bus.
GNT[6:1]#	Internal Arbiter Grant 6-1	6	O TP	T5, R5, R4, N5, T3, T2	When the internal PCI arbiter is enabled, the PCI GNT[6:1]# signals are outputs, one each to an arbitrating master. The PCI 9056 arbiter asserts one of the GNT# signals to grant the PCI Bus to the corresponding master. ***Note:*** *PCI Arbiter pins are type "TP" when the PCI arbiter is enabled. Otherwise, they are left floating.*
IDSEL	Initialization Device Select	1	I	D3	Used as a chip select during Configuration Read and Write transactions.
INTA#	Interrupt A	1	I/O OC PCI	B4	As an input, it is available only if HOSTEN# is asserted (drives LINTo# onto a Local Bus to a Local processor). The PCI 9056 is a PCI Host. As an output, the PCI 9056 drives INTA# to perform a PCI Interrupt request.
IRDY#	Initiator Ready	1	I/O STS PCI	G1	Indicates ability of the initiating agent (Bus Master) to complete the current Data phase of the transaction.

Table 12-4. PCI System Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LOCK#	Lock	1	I/O STS PCI	H1	Indicates an atomic operation that may require multiple transactions to complete.
PAR	Parity	1	I/O TS PCI	K1	Even parity across AD[31:0] and C/BE[3:0]#. All PCI agents require parity generation. PAR is stable and valid one clock after the Address phase. For Data phases, PAR is stable and valid one clock after either IRDY# is asserted on a Write transaction or TRDY# is asserted on a Read transaction. Once PAR is valid, it remains valid until one clock after the current Data phase completes.
PCLK	Clock	1	I	J1	Provides timing for all transactions on PCI and is an input to every PCI device. The PCI 9056 PCI Bus operates up to 66 MHz.
PERR#	Parity Error	1	I/O STS PCI	J2	Reports data parity errors during all PCI transactions, except during a special cycle.
PME#	Power Management Event	1	O OC PCI	A6	Asserted to alert system to a power management event.
REQ0# GNT#	Internal Arbiter Request 0 External Arbiter Grant	1	I	D5	Multiplexed input pin. REQ0#: When the internal PCI arbiter is enabled, the PCI REQ0# signal is an input from an arbitrating master. REQ0# is asserted to the PCI 9056 arbiter by the master to request the PCI Bus. GNT#: When the internal PCI arbiter is disabled, REQ0# becomes the GNT# input to the PCI 9056 from an external arbiter. The arbiter asserts GNT# to grant the PCI Bus to the PCI 9056. REQ[6:1]# are not used.
REQ[6:1]#	Internal Arbiter Request 6-1	6	I	R6, N6, T4, P5, R3, P4	When the internal PCI arbiter is enabled, the PCI REQ[6:1]# signals are inputs, one each from an arbitrating master. REQ[6:1]# is asserted to the PCI 9056 arbiter by the corresponding master to request the PCI Bus.
RST#	Reset	1	I/O	C5	As an input, used to bring PCI-specific registers, sequencers, and signals to a consistent state. As an output, available only if HOSTEN# is asserted, causing the entire PCI Bus to reset by way of LRESET# assertion. The PCI 9056 is a PCI Host.
SERR#	Systems Error	1	I/O OC PCI	J3	As an input, available only if HOSTEN# is asserted, causing TEA#/LSERR# to be asserted any time the PCI error occurs. The PCI 9056 is a PCI Host. As an output, reports address parity errors, data parity errors on the Special Cycle command, or any other system error where the result is catastrophic.

Table 12-4. PCI System Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
STOP#	Stop	1	I/O STS PCI	H2	Indicates the current Target is requesting that the Master stop the current transaction.
TRDY#	Target Ready	1	I/O STS PCI	H4	Indicates ability of the Target agent (selected device) to complete the current Data phase of the transaction.
Total		64			

Table 12-5. JTAG Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
TCK	Test Clock Input	1	I	B5	Clock source for the PCI 9056 test access port (TAP). Legal rates for TCK are either equal to the LCLK rate or less than one-half the LCLK rate.
TDI	Test Data Input	1	I	D6	Used to input data into the TAP. When the TAP enables this pin, it is sampled on the rising edge of TCK and the sampled value is input to the selected TAP shift register.
TDO	Test Data Output	1	O TS PCI	A4	Used to transmit serial data from the PCI 9056 TAP. Data from the selected shift register is shifted out of TDO.
TMS	Test Mode Select	1	I	A5	Sampled by the TAP on the rising edge of TCK. The TAP state machine uses the TMS pin to determine the TAP mode.
TRST#	Test Reset	1	I	C6	Resets JTAG.
Total		5			

Table 12-6. CompactPCI Hot Swap Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
BD_SEL#	CompactPCI Board Select	1	I	B6	CompactPCI board select for Hot Swap system. For non-CompactPCI systems, this pin should be grounded.
CPCISW	CompactPCI Switch Sense	1	I	T6	Input that monitors CompactPCI board latch status. For non-CompactPCI systems, this pin should be pulled high.
ENUM#	Enumeration	1	O OC PCI	P6	Interrupt output asserted when an adapter using PCI 9056 has been inserted or is ready to be removed from a PCI slot.
LEDon#	CompactPCI LED On	1	O TP 24 mA	N7	Activates the CompactPCI Hot Swap board indicator LED.
Total		4			

Table 12-7. System Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
IDDQEN#	Buffered PCI Frame	1	I	B7	Provides main power status to the PCI 9056 D_{3cold} Power Management logic. For all normal operations, this pin should be connected directly to the 3.3V power line. For IDDQ' tests. the pin should be grounded.
MODE[1:0]	Bus Mode	2	I	A15, B15	Selects the PCI 9056 bus operation mode: **Mode 0** **Mode 1** **Bus Mode** 1 1 M 0 0 C 1 0 J 0 1 ***Reserved***
HOSTEN#	Host Enable	1	I	B12	When asserted, configures the PCI 9056 as a host bridge, setting reset and interrupt signal directions for system board applications. When de-asserted, configures the PCI 9056 as a peripheral bridge, setting reset and interrupt signal directions for peripheral board applications.
Total		4			

Table 12-8. Serial EEPROM Interface Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
EECS	Serial EEPROM Chip Select	1	O TP 12 mA	A8	Serial EEPROM chip select.
EEDI/EEDO	Serial EEPROM Data IN/ Serial EEPROM Data OUT	1	I/O TP (TS if CNTRL[31]=1) 12 mA	C8	Multiplexed Write and Read data to the serial EEPROM pin.
EESK	Serial Data Clock	1	O TP 12 mA	B8	Serial EEPROM clock pin.
Total		3			

Table 12-9. Power and Ground Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
$2.5V_{AUX}$	PME 2.5V D_{3cold} Power	1	I	D7	2.5V to PME logic during D_{3cold} state. For Power Management systems, connect directly to 2.5V regulated power line from Card_V_{AUX}. Otherwise, connect directly to 2.5V power line.
Card_V_{AUX}	PME 3.3V D_{3cold} Power	1	I	A7	3.3V to PME logic during D_{3cold} state. Refer to the *PCI Power Mgmt. r1.1*, Figure 12. For non-Power Management systems, connect directly to 3.3V.
PRESENT_DET	3.3V V_{AUX} Present Detect Power	1	I	D8	When sampled as 1, $3.3V_{AUX}$ power is present and PME# assertion in D_{3cold} is supported. When sampled as 0, $3.3V_{AUX}$ power is not present and PME# assertion in D_{3cold} is not supported by the PME_Support D_{3cold} bit (PMC[15]). Refer to the *PCI Power Mgmt. r1.1* Figure 12. For non-Power Management systems, connect directly to ground.
V_{CORE}	Core Power	6	I	D9, F4, F13, K4, K13, N9	2.5V to core.
V_{IO}	PCI System Voltage	4	I	B3, E2, M2, N4	System voltage select, 3.3 or 5V, from PCI Bus.
V_{RING}	I/O Ring Power	13	I	E7, E8, E10, G5, G12, H5, H12, J12, K5, K12, M7, M8, M10	3.3V to I/O ring.
V_{SS}	Ground	55	I	E5, E6, E9, E11, E12, F5-F12, G4, G6-G11, G13, H6-H11, J4-J11, J13, K6-K11, L5-L12, M5, M6, M9, M11, M12	Ground.
Total		81			

12.3 M BUS MODE PINOUT

Table 12-10. M Mode Local Bus Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
BB#	Bus Busy	1	I/O OC 24 mA	A13	As an input, the PCI 9056 monitors this signal to determine whether an external Master has ended a Bus cycle. As an output, the PCI 9056 asserts this signal after an external arbiter has granted ownership of the Local Bus and BB# is inactive from another Master. It is recommended to use an external pull-up resistor value of 4.7K ohms be applied to guarantee a fast transition to the inactive state when the PCI 9056 relinquishes ownership of the Local Bus.
BDIP#	Burst Data in Progress	1	I/O TS 24 mA	C12	As an input, driven by the Bus Master during a Burst transaction. The Master de-asserts before the last Data phase on the bus. As an output, driven by the PCI 9056 during the Data phase of a Burst transaction. The PCI 9056 de-asserts before the last Burst Data phase on the bus.
BG#	Bus Grant	1	I	B14	Asserted by the Local Bus arbiter in response to BR#. Indicates the requesting Master is next.
BI#	Burst Inhibit	1	I	E13	When asserted, indicates that the Target device does not support Burst transactions.
BIGEND# WAIT#	Big Endian Select WAIT Input/Output Select	1	I I/O TS 24 mA	A9	Multiplexed input/output pin. BIGEND#: Can be asserted during the Local Bus Address phase of a Direct Master transfer or Configuration register access to specify use of Big Endian Byte ordering. Big Endian Byte order for Direct Master transfers or Configuration register accesses is also programmable through the Configuration registers. WAIT#: If wait is selected, then the PCI 9056 issues WAIT# when it is a Master on the Local Bus and has internal wait states setup. As a Slave, the PCI 9056 accepts WAIT# as an input from the Bus Master.
BR#	Bus Request	1	O TP 24 mA	A16	Asserted by the Master to request use of the Local Bus. The Local Bus arbiter asserts BG# when the Master is next in line for bus ownership.
BURST#	Burst	1	I/O TS 24 mA	A14	As an input, driven by the Master along with address and data indicating a Burst transfer is in progress. As an output, driven by the PCI 9056 along with address and data indicating a Burst transfer is in progress.
CCS#	Configuration Register Select	1	I	B9	Internal PCI 9056 registers are selected when CCS# is asserted low.

Table 12-10. M Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
DACK0#	DMA Channel 0 Demand Mode Acknowledge	1	O TP 24 mA	C9	When a channel is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is being executed. Corresponds to PCI 9056 DMA Channel 0.
DACK1#	DMA Channel 1 Demand Mode Acknowledge	1	O TP 24 mA	B10	When a channel is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is being executed. Corresponds to PCI 9056 DMA Channel 1.
DP[0:3]	Data Parity	4	I/O TS 24 mA	D13, C14, B16, D14	Parity is even for each of up to four byte lanes on the Local Bus. Parity is checked for writes or reads to the PCI 9056. Parity is asserted for reads from or writes by the PCI 9056.
DREQ0#	DMA Channel 0 Demand Mode Request	1	I	A10	When a channel is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request. Corresponds to PCI 9056 DMA Channel 0.
DREQ1#	DMA Channel 1 Demand Mode Request	1	I	C10	When a channel is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request. Corresponds to PCI 9056 DMA Channel 1.
LA[0:31]	Address Bus	32	I/O TS 24 mA	P7, R7, T7, N8, P8, R8, T8, T9, R9, P9, T10, R10, P10, T11, N10, P11, R11, T12, R12, T13, N11, P12, T14, R13, N12, P13, T15, R14, R15, N13, R16, N14	Carries the 32 bits of the physical Address Bus.
LCLK	Local Processor Clock	1	I	D16	Local clock input.
LD[0:31]	Data Bus	32	I/O TS 24 mA	P16, M13, N15, M14, L13, N16, M15, L14, L15, M16, L16, K14, K15, K16, J14, J15, J16, H16, H15, H14, H13, G16, G15, G14, F16, E16, F15, F14, E15, D15, E14, C16	Carries 8-, 16-, or 32-bit data quantities, depending upon bus-width configuration. All Master accesses to the PCI 9056 are 32 bits only.
LINTi#	Local Interrupt Input	1	I	C11	When asserted, causes a PCI interrupt.
LINTo#	Local Interrupt Output	1	O OC 24 mA	B11	Synchronous level output that remains asserted as long as an interrupt condition exists. If an edge-level interrupt is required, disabling and then enabling Local interrupts through INTCSR creates an edge if an interrupt condition continues or a new interrupt condition occurs.

Table 12-10. M Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LRESET#	Local Bus Reset	1	I/O TP 24 mA	D11	As an input, available only if HOSTEN# is asserted, causing RST# to be asserted on the PCI Bus. The PCI 9056 is a PCI Host. As an output, asserted when the PCI 9056 chip is in reset. Can be used to reset the back-end logic on the board.
MDREQ# DMPAF EOT#	IDMA Data Transfer Request *(Available at this location in M mode only)* Direct Master Programmable Almost Full End of Transfer for Current DMA Channel	1	O TS 24 mA O TS 24 mA I	A12	Multiplexed input/output pin. MDREQ#: IDMA M mode Data transfer request start. Always asserted, indicating Data transfer should start. De-asserted only when the Direct Master FIFO becomes full. Programmable through a Configuration register. DMPAF: Direct Master Write FIFO Almost Full status output. Programmable through a Configuration register. EOT#: Terminates the current DMA transfer. ***Note:*** *EOT# serves as a general purpose EOT. Before asserting EOT#, user should be aware of DMA channel activity.*
PMEREQ#	PME Request	1	I	C7	Request a Power Management Event during a D_{3cold} power state. Other Power Management Events should be done through the PCI 9056 Power Management registers.
RD/WR#	Read/Write	1	I/O TS 24 mA	P14	Asserted high for reads and low for writes.
RETRY#	Retry	1	O OC 24 mA	B13	Driven by the PCI 9056 when it is a Slave to indicate a Local Master must back off and restart the cycle. In Delayed Read mode, indicates the Master should return for requested data.
TA#	Transfer Acknowledge	1	I/O DTS 24 mA	C15	As an input, when the PCI 9056 is a Bus Master, indicates a Write Data transfer is complete or that Read data on the bus is valid. As an output, when a Local Bus access is made to the PCI 9056, indicates a Write Data transfer is complete or that Read data on the bus is valid.
TEA#	Transfer Error Acknowledge	1	I/O OC 24 mA	C13	Driven by the Target device, indicating an error condition occurred during a Bus cycle.
TS#	Address Strobe	1	I/O TS 24 mA	D12	Indicates the valid address and start of new Bus access. Asserted for the first clock of a Bus access.
TSIZ[0:1]	Transfer Size	2	I/O TS 24 mA	T16, P15	Driven by current Master along with the address, indicating the data-transfer size. Refer to Section 3.4.3.7.3 for more information.

Table 12-10. M Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
USERi LLOCKi#	User Input Local Lock Input	1	I I	A11	Multiplexed input pin. USERi: General-purpose input that can be read by way of the PCI 9056 Configuration registers. LLOCKi#: Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9056 for direct Local access to the PCI Bus.
USERo LLOCKo#	User Output Local Lock Output	1	O TS 24 mA O	D10	Multiplexed output pin. USERo: General-purpose output controlled from the PCI 9056 Configuration registers. LLOCKo#: Indicates an atomic operation for a Direct Slave PCI-to-Local Bus access may require multiple transactions to complete.
Total		95			

12.4 C BUS MODE PINOUT

Table 12-11. C Mode Local Bus Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
ADS#	Address Strobe	1	I/O TS 24 mA	D12	Indicates valid address and start of new Bus access. Asserted for first clock of Bus access.
BIGEND#	Big Endian Select	1	I	A9	Can be asserted during the Local Bus Address phase of a Direct Master transfer or Configuration register access to specify use of Big Endian Byte ordering. Big Endian Byte order for Direct Master transfers or Configuration register accesses is also programmable through the Configuration registers.
BLAST#	Burst Last	1	I/O TS 24 mA	A14	Signal driven by the current Local Bus Master to indicate the last transfer in a Bus access. As an output, asserted by the PCI 9056 after internal wait states have expired (WAIT# de-asserted). Internal wait states are programmed in LAS0BRD[5:2] and/or LAS1BRD[5:2].
BREQi	Bus Request	1	I	A13	Asserted to indicate a Local Bus Master requires the bus. If enabled through the PCI 9056 Configuration registers, the PCI 9056 releases the bus during a DMA transfer if this signal is asserted.
BREQo	Bus Request Out	1	O OC 24 mA	B13	Asserted to indicate the PCI 9056 requires the bus to perform a Direct Slave PCI-to-Local Bus access while a Direct Master access is pending on the Local Bus. Can be used with external logic to assert back off to a Local Bus Master. Operational parameters are set up through the PCI 9056 Configuration registers.
BTERM#	Burst Terminate	1	I/O DTS 24 mA	E13	As input to the PCI 9056: For processors that burst up to four Lwords. If the BTERM# Mode bit is disabled through the PCI 9056 Configuration registers, the PCI 9056 also bursts up to four Lwords. If enabled, the PCI 9056 continues to burst until a BTERM# input is asserted. BTERM# is a Ready input that breaks up a Burst cycle and causes another Address cycle to occur. Used in conjunction with the PCI 9056 programmable wait state generator. As output from the PCI 9056: Asserted, along with READY#, to request break up of a burst and start of a new Address cycle (PCI aborts only).
CCS#	Configuration Register Select	1	I	B9	Internal PCI 9056 registers are selected when CCS# is asserted low.
DACK0#	DMA Channel 0 Demand Mode Acknowledge	1	O TP 24 mA	C9	When a channel is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is being executed. Corresponds to PCI 9056 DMA Channel 0.
DACK1#	DMA Channel 1 Demand Mode Acknowledge	1	O TP 24 mA	B10	When a channel is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is being executed.

Table 12-11. C Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
DMPAF EOT#	Direct Master Programmable Almost Full End of Transfer for Current DMA Channel	1	O TS 24 mA I	A12	Multiplexed input/output pin. DMPAF: Direct Master Write FIFO Almost Full status output. Programmable through a Configuration register. EOT#: Terminates the current DMA transfer. ***Note:*** *EOT# serves as a general purpose EOT. Before asserting EOT#, user should be aware of DMA channel activity.*
DP[3:0]	Data Parity	4	I/O TS 24 mA	D13, C14, B16, D14	Parity is even for each of up to four byte lanes on the Local Bus. Parity is checked for writes or reads to the PCI 9056. Parity is asserted for reads from or writes by the PCI 9056.
DREQ0#	DMA Channel 0 Demand Mode Request	1	I	A10	When a channel is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request. Corresponds to PCI 9056 DMA Channel 0.
DREQ1#	DMA Channel 1 Demand Mode Request	1	I	C10	When a channel is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request. Corresponds to PCI 9056 DMA Channel 1.
LA[31:2]	Address Bus	30	I/O TS 24 mA	P7, R7, T7, N8, P8, R8, T8, T9, R9, P9, T10, R10, P10, T11, N10, P11, R11, T12, R12, T13, N11, P12, T14, R13, N12, P13, T15, R14, R15, N13	Carries the upper 30 bits of physical Address Bus. During bursts, LA[31:2] increment to indicate successive Data cycles.
LBE[3:0]#	Byte Enables	4	I/O TS 24 mA	T16, P15, R16, N14	Encoded, based on the bus-width configuration, as follows: **32-Bit Bus** The four byte enables indicate which of the four bytes are active during a Data cycle: LBE3# Byte Enable 3—LD[31:24] LBE2# Byte Enable 2—LD[23:16] LBE1# Byte Enable 1—LD[15:8] LBE0# Byte Enable 0—LD[7:0] **16-Bit Bus** LBE[3, 1:0]# are encoded to provide BHE#, LA1, and BLE#, respectively: LBE3# Byte High Enable (BHE#)—LD[15:8] LBE2# not used LBE1# Address bit 1 (LA1) LBE0# Byte Low Enable (BLE#)—LD[7:0] **8-Bit Bus** LBE[1:0]# are encoded to provide LA[1:0], respectively: LBE3# not used LBE2# not used LBE1# Address bit 1 (LA1) LBE0# Address bit 0 (LA0)

Table 12-11. C Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LCLK	Local Processor Clock	1	I	D16	Local clock input.
LD[31:0]	Data Bus	32	I/O TS 24 mA	P16, M13, N15, M14, L13, N16, M15, L14, L15, M16, L16, K14, K15, K16, J14, J15, J16, H16, H15, H14, H13, G16, G15, G14, F16, E16, F15, F14, E15, D15, E14, C16	Carries 8-, 16-, or 32-bit data quantities, depending upon bus-width configuration. All Master accesses to the PCI 9056 are 32 bits only.
LHOLD	Hold Request	1	O TP 24 mA	A16	Asserted to request use of the Local Bus. The Local Bus arbiter asserts LHOLDA when control is granted.
LHOLDA	Hold Acknowledge	1	I	B14	Asserted by the Local Bus arbiter when control is granted in response to LHOLD. Bus should not be granted to the PCI 9056 unless requested by LHOLD.
LINTi#	Local Interrupt Input	1	I	C11	When asserted, causes a PCI interrupt.
LINTo#	Local Interrupt Output	1	O OC 24 mA	B11	Synchronous level output that remains asserted as long as an interrupt condition exists. If an edge-level interrupt is required, disabling and then enabling Local interrupts through INTCSR creates an edge if an interrupt condition continues or a new interrupt condition occurs.
LRESET#	Local Bus Reset	1	I/O TP 24 mA	D11	As an input, available only if HOSTEN# is asserted, causing RST# to be asserted on the PCI Bus. The PCI 9056 is a PCI Host. As an output, asserted when the PCI 9056 chip is in reset. Can be used to reset the back-end logic on the board.
LSERR#	System Error Interrupt Output	1	O OC 24 mA	C13	Synchronous level output asserted when the PCI Bus Target Abort bit is set (PCISR[11]=1) or Received Master Abort bit is set (PCISR[13]=1). If an edge level interrupt is required, disabling and then enabling LSERR# interrupts through the interrupt/control status creates an edge if an interrupt condition still exists or a new interrupt condition occurs.
LW/R#	Write/Read	1	I/O TS 24 mA	P14	Asserted low for reads and high for writes.
PMEREQ#	PME Request	1	I	C7	Request a Power Management Event during a D_{3cold} power state. Other Power Management Events should be done through the PCI 9056 Power Management registers.

Table 12-11. C Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
READY#	Ready Input/Output	1	I/O DTS 24 mA	C15	When the PCI 9056 is a Bus Master, indicates that Read data on the bus is valid or that a Write Data transfer is complete. READY# input is not sampled until the internal wait state counter expires (WAIT# output de-asserted). When a Local Bus access is made to the PCI 9056, indicates that Read data on the bus is valid or that a Write Data transfer is complete. READY# output is not asserted until the Local Master de-asserts the WAIT# input (requesting wait states).
USERi LLOCKi#	User Input Local Lock Input	1	I I	A11	Multiplexed input pin. USERi: General-purpose input that can be read by way of the PCI 9056 Configuration registers. LLOCKi#: Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9056 for direct Local access to the PCI Bus.
USERo LLOCKo#	User Output Local Lock Output	1	O TS 24 mA O	D10	Multiplexed output pin. USERo: General-purpose output controlled from the PCI 9056 Configuration registers. LLOCKo#: Indicates an atomic operation for a Direct Slave PCI-to-Local Bus access may require multiple transactions to complete.
WAIT#	Wait Input/Output Select	1	I/O TS 24 mA	C12	As an input, can be asserted to cause the PCI 9056 to insert wait states for Local Direct Master accesses to the PCI Bus. Can be thought of as a Ready input from an external Master for Direct Master accesses. As an output, asserted by the PCI 9056 when internal wait state generator causes wait states. Can be thought of as an output providing PCI 9056 Ready status.
Total		95			

12.5 J BUS MODE PINOUT

Table 12-12. J Mode Local Bus Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
ADS#	Address Strobe	1	I/O TS 24 mA	D12	Indicates valid address and start of new Bus access. Asserted for first clock of Bus access.
ALE	Address Latch Enable	1	I/O TS 24 mA	T7	Asserted during Address phase and de-asserted before Data phase and before next LCLK rising edge.
BIGEND#	Big Endian Select	1	I	A9	Can be asserted during the Local Bus Address phase of a Direct Master transfer or Configuration register access to specify use of Big Endian Byte ordering. Big Endian Byte order for Direct Master transfers or Configuration register accesses is also programmable through the Configuration registers.
BLAST#	Burst Last	1	I/O TS 24 mA	A14	Signal driven by the current Local Bus Master to indicate the last transfer in a Bus access. As an output, asserted by the PCI 9056 after internal wait states have expired (WAIT# de-asserted). Internal wait states are programmed in LAS0BRD[5:2] and/or LAS1BRD[5:2].
BREQi	Bus Request	1	I	A13	Asserted to indicate a Local Bus Master requires the bus. If enabled through the PCI 9056 Configuration registers, the PCI 9056 releases the bus during a DMA transfer if this signal is asserted.
BREQo	Bus Request Out	1	O OC 24 mA	B13	Asserted to indicate the PCI 9056 requires the bus to perform a Direct Slave PCI-to-Local Bus access while a Direct Master access is pending on the Local Bus. Can be used with external logic to assert back off to a Local Bus Master. Operational parameters are set up through the PCI 9056 Configuration registers.
BTERM#	Burst Terminate	1	I/O DTS 24 mA	E13	As input to the PCI 9056: For processors that burst up to four Lwords. If the BTERM# Mode bit is disabled through the PCI 9056 Configuration registers, the PCI 9056 also bursts up to four Lwords. If enabled, the PCI 9056 continues to burst until a BTERM# input is asserted. BTERM# is a Ready input that breaks up a Burst cycle and causes another Address cycle to occur. Used in conjunction with the PCI 9056 programmable wait state generator. As output from the PCI 9056: Asserted, along with READY#, to request break up of a burst and start of a new Address cycle (PCI aborts only).
CCS#	Configuration Register Select	1	I	B9	Internal PCI 9056 registers are selected when CCS# is asserted low.
DACK0#	DMA Channel 0 Demand Mode Acknowledge	1	O TP 24 mA	C9	When a channel is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is being executed. Corresponds to PCI 9056 DMA Channel 0.

Table 12-12. J Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
DACK1#	DMA Channel 1 Demand Mode Acknowledge	1	O TP 24 mA	B10	When a channel is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is being executed.
DEN#	Data Enable	1	O TS 24 mA	R7	Used in conjunction with DT/R# to provide control for data transceivers attached to the Local Bus.
DMPAF EOT#	Direct Master Programmable Almost Full End of Transfer for Current DMA Channel	1	O TS 24 mA I	A12	Multiplexed input/output pin. DMPAF: Direct Master Write FIFO Almost Full status output. Programmable through a Configuration register. EOT#: Terminates the current DMA transfer. ***Note:*** *EOT# serves as a general purpose EOT. Before asserting EOT#, user should be aware of DMA channel activity.*
DP[3:0]	Data Parity	4	I/O TS 24 mA	D13, C14, B16, D14	Parity is even for each of up to four byte lanes on the Local Bus. Parity is checked for writes or reads to the PCI 9056. Parity is asserted for reads from or writes by the PCI 9056.
DREQ0#	DMA Channel 0 Demand Mode Request	1	I	A10	When a channel is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request. Corresponds to PCI 9056 DMA Channel 0.
DREQ1#	DMA Channel 1 Demand Mode Request	1	I	C10	When a channel is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request. Corresponds to PCI 9056 DMA Channel 1.
DT/R#	Data Transmit/Receive	1	O TS 24 mA	P7	Used in conjunction with DEN# to provide control for data transceivers attached to the Local Bus. When asserted, indicates the PCI 9056 receives data.
LA[28:2]	Local Address Bus	27	I/O TS 24 mA	N8, P8, R8, T8, T9, R9, P9, T10, R10, P10, T11, N10, P11, R11, T12, R12, T13, N11, P12, T14, R13, N12, P13, T15, R14, R15, N13	Carries the middle 27 bits of the physical address bus. During bursts, it is incremented to indicate successive Data cycles. The lowest two bits, LA[3:2], carry the Word address of the 32-bit Memory address. All bits are incremented during a Burst access.
LAD[31:0]	Address/Data Bus	32	I/O TS 24 mA	P16, M13, N15, M14, L13, N16, M15, L14, L15, M16, L16, K14, K15, K16, J14, J15, J16, H16, H15, H14, H13, G16, G15, G14, F16, E16, F15, F14, E15, D15, E14, C16	During an Address phase, the bus carries the upper 30 bits of the physical Address Bus. During a Data phase, the bus carries 32 bits of data.

Table 12-12. J Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LBE[3:0]#	Byte Enables	4	I/O TS 24 mA	T16, P15, R16, N14	Encoded, based on the bus-width configuration, as follows: **32-Bit Bus** The four byte enables indicate which of the four bytes are active during a Data cycle: LBE3# Byte Enable 3—LAD[31:24] LBE2# Byte Enable 2—LAD[23:16] LBE1# Byte Enable 1—LAD[15:8] LBE0# Byte Enable 0—LAD[7:0] **16-Bit Bus** LBE[3, 1:0]# are encoded to provide BHE#, LAD1, and BLE#, respectively: LBE3# Byte High Enable (BHE#)—LAD[15:8] LBE2# not used LBE1# Address bit 1 (LAD1) LBE0# Byte Low Enable (BLE#)—LAD[7:0] **8-Bit Bus** LBE[1:0]# are encoded to provide LAD[1:0], respectively: LBE3# not used LBE2# not used LBE1# Address bit 1 (LAD1) LBE0# Address bit 0 (LAD0)
LCLK	Local Processor Clock	1	I	D16	Local clock input.
LHOLD	Hold Request	1	O TP 24 mA	A16	Asserted to request use of the Local Bus. The Local Bus arbiter asserts LHOLDA when control is granted.
LHOLDA	Hold Acknowledge	1	I	B14	Asserted by the Local Bus arbiter when control is granted in response to LHOLD. Bus should not be granted to the PCI 9056 unless requested by LHOLD.
LINTi#	Local Interrupt Input	1	I	C11	When asserted, causes a PCI interrupt.
LINTo#	Local Interrupt Output	1	O OC 24 mA	B11	Synchronous level output that remains asserted as long as an interrupt condition exists. If an edge-level interrupt is required, disabling and then enabling Local interrupts through INTCSR creates an edge if an interrupt condition still exists or a new interrupt condition occurs.
LRESET#	Local Bus Reset	1	I/O TP 24 mA	D11	As an input, available only if HOSTEN# is asserted, causing RST# to be asserted on the PCI Bus. The PCI 9056 is a PCI Host. As an output, asserted when the PCI 9056 chip is in reset. Can be used to reset the back-end logic on the board.
LSERR#	System Error Interrupt Output	1	O OC 24 mA	C13	Synchronous level output asserted when the PCI Bus Target Abort bit is set (PCISR[11]=1) or Received Master Abort bit is set (PCISR[13]=1). If an edge level interrupt is required, disabling and then enabling LSERR# interrupts through the interrupt/control status creates an edge if an interrupt condition still exists or a new interrupt condition occurs.

Table 12-12. J Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LW/R#	Write/Read	1	I/O TS 24 mA	P14	Asserted low for reads and high for writes.
PMEREQ#	PME Request	1	I	C7	Request a Power Management Event during a D_{3cold} power state. Other Power Management Events should be done through the PCI 9056 Power Management registers.
READY#	Ready Input/Output	1	I/O DTS 24 mA	C15	When the PCI 9056 is a Bus Master, indicates that Read data on the bus is valid or that a Write Data transfer is complete. READY# input is not sampled until the internal wait state counter expires (WAIT# output de-asserted). When a Local Bus access is made to the PCI 9056, indicates that Read data on the bus is valid or that a Write Data transfer is complete. READY# output is not asserted until the Local Master de-asserts the WAIT# input (requesting wait states).
USERi LLOCKi#	User Input Local Lock Input	1	I I	A11	Multiplexed input pin. USERi: General-purpose input that can be read by way of the PCI 9056 Configuration registers. LLOCKi#: Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9056 for direct Local access to the PCI Bus.
USERo LLOCKo#	User Output Local Lock Output	1	O TS 24 mA O	D10	Multiplexed output pin. USERo: General-purpose output controlled from the PCI 9056 Configuration registers. LLOCKo#: Indicates an atomic operation for a Direct Slave PCI-to-Local Bus access may require multiple transactions to complete.
WAIT#	Wait Input/Output Select	1	I/O TS 24 mA	C12	As an input, can be asserted to cause the PCI 9056 to insert wait states for Local Direct Master accesses to the PCI Bus. Can be thought of as a Ready input from an external Master for Direct Master accesses. As an output, asserted by the PCI 9056 when internal wait state generator causes wait states. Can be thought of as an output providing PCI 9056 Ready status.
Total		95			

12.6 DEBUG INTERFACE

The PCI 9056 provides a JTAG Boundary Scan interface which can be utilized to debug a pin's connectivity to the board.

12.6.1 IEEE 1149.1 Test Access Port (JTAG Debug Port)

The IEEE 1149.1 Test Access Port (TAP), commonly called the JTAG (Joint Test Action Group) debug port, is an architectural standard described in IEEE Standard 1149.1–1990, *IEEE Standard Test Access Port and Boundary-Scan Architecture.* The standard describes a method for accessing internal chip facilities using a four- or five-signal interface.

The JTAG debug port, originally designed to support scan-based board testing, is enhanced to support the attachment of debug tools. The enhancements, which comply with IEEE Standard 1149.1-1990 for vendor-specific extensions, are compatible with standard JTAG hardware for boundary-scan system testing.

- **JTAG Signals**—JTAG debug port implements the four required JTAG signals, TCLK, TMS, TDI, TDO, and the optional TRST# signal.
- **JTAG Clock Requirements**—The TCLK signal frequency can range from DC to one-half of the internal chip clock frequency.
- **JTAG Reset Requirements**—JTAG debug port logic is reset at the same time as a system reset. Upon receiving TRST#, the JTAG TAP controller returns to the Test-Logic Reset state.

12.6.2 JTAG Instructions

The JTAG debug port provides the standard **extest, sample/preload,** and **bypass** instructions. Invalid instructions behave as the **bypass** instruction. There are three private instructions.

The following tables list the JTAG instructions and infrared (IR) outputs.

Table 12-13. JTAG Instructions

Instruction	Input Code	Comments
Extest	0000	IEEE 1149.1 standard
Sample/Preload	0100	IEEE 1149.1 standard
Bypass	1111	IEEE 1149.1 standard

Table 12-14. JTAG Infrared Outputs

Instruction	IR Output	Comments
Extest	0001	IEEE 1149.1 standard
Sample/Preload	0101	IEEE 1149.1 standard
Bypass	1101	IEEE 1149.1 standard

12.6.3 JTAG Boundary Scan

Boundary Scan Description Language (BSDL), IEEE Standard 1149.1b-1994, is a supplement to IEEE Standard 1149.1-1990 and IEEE Standard 1149.1a-1993, *IEEE Standard Test Access Port and Boundary-Scan Architecture*. BSDL, a subset of the IEEE Standard 1076-1993 VHSIC Hardware Description Language (VHDL), allows a rigorous description of testability features in components which comply with the standard. It is used by automated test pattern generation tools for package interconnect tests and electronic design automation (EDA) tools for synthesized test logic and verification. BSDL supports robust extensions that can be used for internal test generation and to write software for hardware debug and diagnostics.

The primary components of BSDL include the logical port description, physical pin map, instruction set, and boundary register description.

The logical port description assigns symbolic names to the pins of a chip. Each pin has a logical type of in, out, inout, buffer, or linkage that defines the logical direction of signal flow.

The physical pin map correlates the logical ports of the chip to the physical pins of a specific package. A BSDL description can have several physical pin maps; each map is given a unique name.

Instruction set statements describe the bit patterns that must be shifted into the Instruction Register to place the chip in the various test modes defined by the standard. Instruction set statements also support descriptions of instructions that are unique to the chip.

The boundary register description lists each cell or shift stage of the Boundary Register. Each cell has a unique number; the cell numbered 0 is the closest to the Test Data Out (TDO) pin and the cell with the highest number is closest to the Test Data In (TDI) pin. Each cell contains additional information, including:

- Cell type
- Logical port associated with the cell
- Logical function of the cell
- Safe value
- Control cell number
- Disable value
- Result value

13 ELECTRICAL SPECIFICATIONS

13.1 GENERAL ELECTRICAL SPECIFICATIONS

Table 13-1. Absolute Maximum Ratings

Specification	Maximum Rating
Storage Temperature	-55 to +125 °C
Ambient Temperature with Power Applied	-40 to +85 °C
Supply Voltage to Ground (I/O V_{DD})	-0.5 to +4.6V
Supply Voltage to Ground (Core V_{DD})	-0.5 to +3.6V
Supply Voltage to Ground (V_{IO})	-0.5 to +6.5V
Input Voltage (V_{IN})	V_{SS} -0.5 to 6.5V
Output Voltage (V_{OUT})	V_{SS} -0.5V to V_{DD} +0.5
Maximum Package Power Dissipation	1.5W

Table 13-2. Operating Ranges

Ambient Temperature	Supply Voltage (I/O V_{DD})	Supply Voltage (Core V_{DD})	Input Voltage (V_{IN})	
			Min	Max
-40 to +85 °C	3.0 to 3.6V	2.3 to 2.7V	V_{SS}	V_{IO}

Table 13-3. Capacitance (Sample Tested Only)

Parameter	Test Conditions	Pin Type	Value		Units
			Typical	Maximum	
C_{IN}	V_{IN} = 0V	Input	4	6	pF
C_{OUT}	V_{OUT} = 0V	Output	6	10	pF

The following table lists the package thermal resistance in °**C/W** (Θ_{j-a}).

Table 13-4. Package Thermal Resistance

Linear Air Flow			
0m/s	1m/s	2m/s	3m/s
30	22	19	17

Table 13-5. Electrical Characteristics over Operating Range

Parameter	Description	Test Conditions	Min	Max	Units	Buffer Type
Refer to the document, *PCI 9056 Blue Book Revision 0.91 Correction*, for the corrected version of this table.						

Table 13-5. Electrical Characteristics over Operating Range (Continued)

Parameter	Description	Test Conditions	Min	Max	Units	Buffer Type
Refer to the document, *PCI 9056 Blue Book Revision 0.91 Correction*, for the corrected version of this table.						

13.2 LOCAL INPUTS

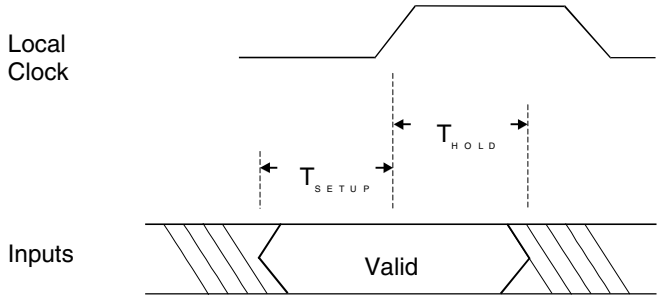


Figure 13-1. PCI 9056 Local Input Setup and Hold Waveform

Table 13-6. AC Electrical Characteristics (Local Inputs) over Operating Range (M Mode)

Signals (Synchronous Inputs) C_L = 50 pF, V_{CC} = 3.0V, Ta = 85 °C	T_{SETUP} (ns) (WORST CASE)	T_{HOLD} (ns) (WORST CASE)
BB#	4.5	1
BDIP#/WAIT#	4.5	1
BG#	4.5	1
BI#	4.5	1
BIGEND#/WAIT#	4.5	1
BURST#	4.5	1
CCS#	4.5	1
DP[0:3]	4.5	1
LA[0:31]	4.5	1
LD[0:31]	4.5	1
MDREQ#/DMPAF/EOT#	4.5	1
RD/WR#	4.5	1
TA#	4.5	1
TS#	4.5	1
TSIZ[0:1]	4.5	1
USERi/LLOCKi#	4.5	1
DREQ0#	4.5	1
Input Clocks	**Min**	**Max**
Local Clock Input Frequency	0	66 MHz
PCI Clock Input Frequency	0	66 MHz

Table 13-7. AC Electrical Characteristics (Local Inputs) over Operating Range (C and J Modes)

Signals (Synchronous Inputs) C_L = 50 pF, V_{CC} = 3.0V, Ta = 85 °C	Bus Mode	T_{SETUP} (ns) (WORST CASE)	T_{HOLD} (ns) (WORST CASE)
ADS#	C, J	4.5	1
ALE	J	4.5	1
BIGEND#	C, J	4.5	1
BLAST#	C, J	4.5	1
BREQi	C, J	4.5	1
BTERM#	C, J	4.5	1
CCS#	C, J	4.5	1
DMPAF/EOT#	C, J	4.5	1
DP[3:0]	C, J	4.5	1
LAD[31:0]	J	4.5	1
LBE[3:0]#	C, J	4.5	1
LD[31:0]	C	4.5	1
LHOLDA	C, J	4.5	1
LW/R#	C, J	4.5	1
READY#	C, J	4.5	1
USERi/LLOCKi#	C, J	4.5	1
DREQ0#	C, J	4.5	1
WAIT#	C, J	4.5	1
Input Clocks	**Bus Mode**	**Min**	**Max**
Local Clock Input Frequency	C, J	0	66 MHz
PCI Clock Input Frequency	C, J	0	66 MHz

13.3 LOCAL OUTPUTS

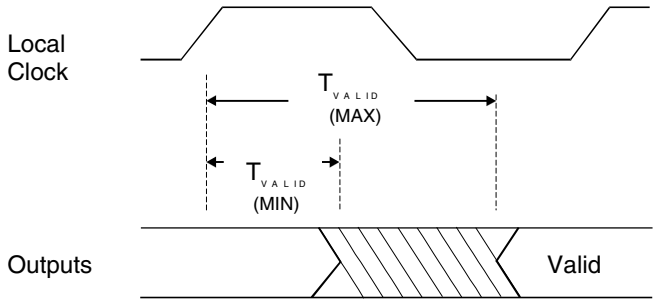


Figure 13-2. PCI 9056 Local Output Delay

Table 13-8. AC Electrical Characteristics (Local Outputs) over Operating Range (M Mode)

Signals (Synchronous Outputs) C_L = 50 pF, V_{CC} = 3.0V, Ta = 85 °C	Clock to Out Worst Case (ns) T_{VALID} (Max)
BB#	9.0
BDIP#	9.0
BI#	9.0
BIGEND#/WAIT#	9.0
BR#	9.0
BURST#	9.0
DP[0:3]	9.0
LA[0:31]	9.0
LD[0:31]	9.0
MDREQ#/DMPAF/EOT#	9.0
RD/WR#	9.0
RETRY#	9.0
TA#	9.0
TEA#	9.0
TS#	9.0
TSIZ[0:1]	9.0
DACK0#	9.0
USERo/LLOCKo#	9.0

Notes: *All T_{VALID} (Min) values are greater than 5 ns.*

Timing derating for loading is ±35 PS/PF.

Table 13-9. AC Electrical Characteristics (Local Outputs) over Operating Range (C and J Modes)

Signals (Synchronous Outputs) C_L = 50 pF, V_{CC} = 3.0V, Ta = 85 °C	Bus Mode	Output T_{VALID} (Max)
ADS#	C, J	9.0
BLAST#	C, J	9.0
BREQo	C, J	9.0
BTERM#	C, J	9.0
DEN#	J	9.0
DMPAF/EOT#	C, J	9.0
DP[3:0]	C, J	9.0
DT/R#	J	9.0
LA[31:2]	C	9.0
LA[28:2]	J	9.0
LAD[31:0]	J	9.0
LBE[3:0]#	C, J	9.0
LD[31:0]	C	9.0
LHOLD	C, J	9.0
LSERR#	C, J	9.0
LW/R#	C, J	9.0
READY#	C, J	9.0
DACK0#	C, J	9.0
USERo/LLOCKo#	C, J	9.0
WAIT#	C, J	9.0

Notes: *All T_{VALID} (Min) values are greater than 5 ns.*

Timing derating for loading is ±35 PS/PF.

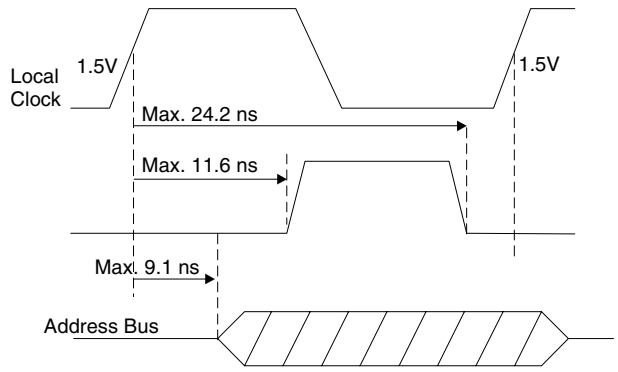
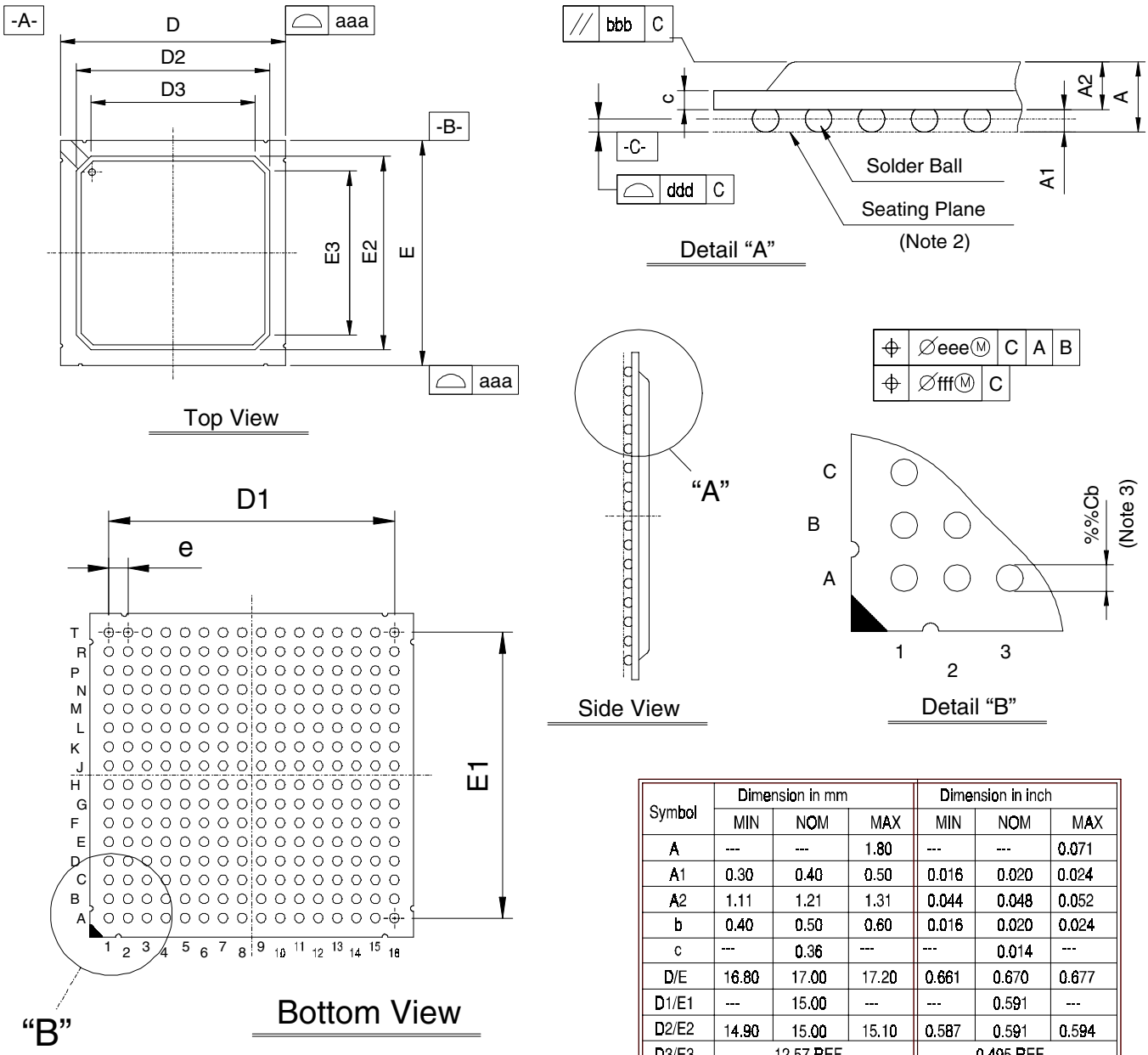


Figure 13-3. PCI 9056 ALE Output Delay to the Local Clock at 33 MHz Clock

14 PHYSICAL SPECS

14.1 MECHANICAL DIMENSIONS



Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	---	---	1.80	---	---	0.071
A1	0.30	0.40	0.50	0.016	0.020	0.024
A2	1.11	1.21	1.31	0.044	0.048	0.052
b	0.40	0.50	0.60	0.016	0.020	0.024
c	---	0.36	---	---	0.014	---
D/E	16.80	17.00	17.20	0.661	0.670	0.677
D1/E1	---	15.00	---	---	0.591	---
D2/E2	14.90	15.00	15.10	0.587	0.591	0.594
D3/E3	12.57 REF			0.495 REF		
e	---	1.00	---	---	0.039	---
aaa	0.20			0.008		
bbb	0.25			0.010		
ddd	0.20			0.008		
eee	0.25			0.010		
fff	0.10			0.039		
MD/ME	16/16			16/16		

Notes:
1. Controlling Dimension: Millimeter
2. Primary Datum C and Seating Plane are defined by the spherical crowns of the solder balls.
3. Dimension b is measured at the maximum solder ball diameter, parallel to Primary Datum C.
4. There shall be a minimum clearance of 0.25 mm between the edge of the solder ball and the body edge.
5. Reference document: JEDEC MO-192.

Figure 14-1. Mechanical Dimensions—Top, Side, and Bottom Views

14.2 BALL GRID ASSIGNMENTS

	1	2	3	4	5	6	7	8
A	AD25	AD30	GNT0# or REQ#	TDO	TMS	PME#	CARD_V_{AUX}	EECS
B	AD26	AD27	V_{IO}	INTA#	TCK	BD_SEL#	IDDQEN#	EESK
C	AD23	AD24	AD28	AD31	RST#	TRST#	PMEREQ#	EEDI/ EEDO
D	AD20	AD22	IDSEL	AD29	REQ0# or GNT#	TDI	2.5V_{AUX}	PRESENT_DET
E	AD18	V_{IO}	AD21	C/BE3#	V_{SS}	V_{SS}	V_{RING}	V_{RING}
F	AD16	AD17	AD19	V_{CORE}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
G	IRDY#	FRAME#	C/BE2#	V_{SS}	V_{RING}	V_{SS}	V_{SS}	V_{SS}
H	LOCK#	STOP#	DEVSEL#	TRDY#	V_{RING}	V_{SS}	V_{SS}	V_{SS}
J	PCLK	PERR#	SERR#	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
K	PAR	C/BE1#	AD15	V_{CORE}	V_{RING}	V_{SS}	V_{SS}	V_{SS}
L	AD14	AD13	AD12	AD9	V_{SS}	V_{SS}	V_{SS}	V_{SS}
M	AD11	V_{IO}	AD8	AD5	V_{SS}	V_{SS}	V_{RING}	V_{RING}
N	AD10	C/BE0#	AD6	V_{IO}	GNT3#	REQ5#	LEDon#	LA3 (M) LA28 (C,J)
P	AD7	AD4	AD0	REQ1#	REQ3#	ENUM#	LA0 (M) LA31 (C) DT/R# (J)	LA4 (M) LA27 (C,J)
R	AD2	AD1	REQ2#	GNT4#	GNT5#	REQ6#	LA1 (M) LA30 (C) DEN# (J)	LA5 (M) LA26 (C,J)
T	AD3	GNT1#	GNT2#	REQ4#	GNT6#	CPCISW	LA2 (M) LA29 (C) ALE (J)	LA6 (M) LA25 (C,J)

Figure 14-2. Ball Grid Assignments (A1-A8 through T1-T8)

9	10	11	12	13	14	15	16	
BIGEND# (all) WAIT# (M)	DREQ0#	USERi/ LLOCKi#	MDREQ# (M) DMPAF (all) EOT# (all)	BB# (M) BREQi (C,J)	BURST# (M) BLAST# (C,J)	MODE1	BR# (M) LHOLD (C,J)	A
CCS#	DACK1#	LINTo#	HOSTEN#	RETRY# (M) BREQo (C,J)	BG# (M) LHOLDA (C,J)	MODE0	DP2 (M) DP1 (C,J)	B
DACK0#	DREQ1#	LINTi#	BDIP# (M) WAIT# (C,J)	TEA# (M) LSERR# (C,J)	DP1 (M) DP2 (C,J)	TA# (M) READY# (C,J)	LD31 (M) LD0 (C) LAD0 (J)	C
V_{CORE}	USERo/ LLOCKo#	LRESET#	TS# (M) ADS# (C,J)	DP0 (M) DP3 (C,J)	DP3 (M) DP0 (C,J)	LD29 (M) LD2 (C) LAD2 (J)	LCLK	D
V_{SS}	V_{RING}	V_{SS}	V_{SS}	BI# (M) BTERM# (C,J)	LD30 (M) LD1 (C) LAD1 (J)	LD28 (M) LD3 (C) LAD3 (J)	LD25 (M) LD6 (C) LAD6 (J)	E
V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{CORE}	LD27 (M) LD4 (C) LAD4 (J)	LD26 (M) LD5 (C) LAD5 (J)	LD24 (M) LD7 (C) LAD7 (J)	F
V_{SS}	V_{SS}	V_{SS}	V_{RING}	V_{SS}	LD23 (M) LD8 (C) LAD8 (J)	LD22 (M) LD9 (C) LAD9 (J)	LD21 (M) LD10 (C) LAD10 (J)	G
V_{SS}	V_{SS}	V_{SS}	V_{RING}	LD20 (M) LD11 (C) LAD11 (J)	LD19 (M) LD12 (C) LAD12 (J)	LD18 (M) LD13 (C) LAD13 (J)	LD17 (M) LD14 (C) LAD14 (J)	H
V_{SS}	V_{SS}	V_{SS}	V_{RING}	V_{SS}	LD14 (M) LD17 (C) LAD17 (J)	LD15 (M) LD16 (C) LAD16 (J)	LD16 (M) LD15 (C) LAD15 (J)	J
V_{SS}	V_{SS}	V_{SS}	V_{RING}	V_{CORE}	LD11 (M) LD20 (C) LAD20 (J)	LD12 (M) LD19 (C) LAD19 (J)	LD13 (M) LD18 (C) LAD18 (J)	K
V_{SS}	V_{SS}	V_{SS}	V_{SS}	LD4 (M) LD27 (C) LAD27 (J)	LD7 (M) LD24 (C) LAD24 (J)	LD8 (M) LD23 (C) LAD23 (J)	LD10 (M) LD21 (C) LAD21 (J)	L
V_{SS}	V_{RING}	V_{SS}	V_{SS}	LD1 (M) LD30 (C) LAD30 (J)	LD3 (M) LD28 (C) LAD28 (J)	LD6 (M) LD25 (C) LAD25 (J)	LD9 (M) LD22 (C) LAD22 (J)	M
V_{CORE}	LA14 (M) LA17 (C,J)	LA20 (M) LA11 (C,J)	LA24 (M) LA7 (C,J)	LA29 (M) LA2 (C,J)	LA31 (M) LBE0# (C,J)	LD2 (M) LD29 (C) LAD29 (J)	LD5 (M) LD26 (C) LAD26 (J)	N
LA9 (M) LA22 (M)	LA12 (M) LA19 (C,J)	LA15 (M) LA16 (C,J)	LA21 (M) LA10 (C,J)	LA25 (M) LA6 (C,J)	RD/WR#(M) LW/R# (C,J)	TSIZ1 (M) LBE2# (C,J)	LD0 (M) LD31 (C) LAD31 (J)	P
LA8 (M) LA23 (C,J)	LA11 (M) LA20 (C,J)	LA16 (M) LA15 (C,J)	LA18 (M) LA13 (C,J)	LA23 (M) LA8 (C,J)	LA27 (M) LA4 (C,J)	LA28 (M) LA3 (C,J)	LA30 (M) LBE1# (C,J)	R
LA7 (M) LA24 (C,J)	LA10 (M) LA21 (C,J)	LA13 (M) LA18 (M)	LA17 (M) LA14 (C,J)	LA19 (M) LA12 (C,J)	LA22 (M) LA9 (C,J)	LA26 (M) LA5 (C,J)	TSIZ0 (M) LBE3# (C,J)	T

Figure 14-3. Ball Grid Assignments (A9-A16 through T9-T16)

A GENERAL INFORMATION

A.1 ORDERING INSTRUCTIONS

The PCI 9056 is a 32-bit, 66 MHz PCI I/O Accelerator featuring advanced PLX proprietary Data Pipe Architecture technology, which includes two DMA engines, programmable Direct Slave and Direct Master Data Transfer modes, and PCI messaging functions. The PCI 9056 offers 3.3V, 5V tolerant PCI and Local signaling, and supports Universal PCI Adapter designs, 3.3V core, low-power CMOS offered in a 256-pin (ball) PBGA. The device is designed to operate at Industrial Temperature range.

Table A-1. Available Package

Package	Ordering Part Number
256-pin PBGA	PCI 9056-AA66BI

A.2 UNITED STATES AND INTERNATIONAL REPRESENTATIVES, AND DISTRIBUTORS

A list of PLX Technology, Inc., representatives and distributors can be found at http://www.plxtech.com.

A.3 TECHNICAL SUPPORT

PLX Technology, Inc., technical support information is listed at http://www.plxtech.com; or call 408 774-9060 or 800 759-3735.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Z