PLX TECHNOLOGY®
THE I/O INTERCONNECT SOLUTION

PCI 9656BA Data Book

Version	1.3
January	**2009**
Website	www.plxtech.com
Technical Support	www.plxtech.com/support
Phone	800 759-3735
	408 774-9060
FAX	408 774-2169

COPYRIGHT INFORMATION

PLX Technology provides this documentation without warranty, term or condition of any kind, either express or implied, including, but not limited to, express and implied warranties of merchantability, fitness for a particular purpose, and non-infringement. While the information contained herein is believed to be accurate, such information is preliminary, and no representations or warranties of accuracy or completeness are made. In no event will PLX Technology be liable for damages arising directly or indirectly from any use of or reliance upon the information contained in this document. PLX Technology may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

PLX Technology retains the right to make changes to this product at any time, without notice. Products may have minor variations to this publication, known as errata. PLX Technology assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of PLX Technology products.

Document Number: 9656BA-SIL-DB-P1-1.3

Contents

FIGURES

TABLES

THIS PAGE INTENTIONALLY LEFT BLANK.

REGISTERS

THIS PAGE INTENTIONALLY LEFT BLANK.

TIMING DIAGRAMS

THIS PAGE INTENTIONALLY LEFT BLANK.

PREFACE

The information provided 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 additional information:

- PLX Technology, Inc.
  870 W Maude Avenue, Sunnyvale, CA 94085 USA
  Tel: 800 759-3735 (domestic only) or 408 774-9060, Fax: 408 774-2169, www.plxtech.com

  The PLX PCI 9656 Toolbox includes this data book, as well as other PCI 9656 documentation, including the Errata.
- PCI Special Interest Group (PCI-SIG)
  3855 SW 153rd Drive, Beaverton, OR 97006 USA
  Tel: 503 619-0569, Fax: 503 644-6708, www.pcisig.com
  - *PCI Local Bus Specification, Revision 2.1*
  - *PCI Local Bus Specification, Revision 2.2*
  - *PCI Hot-Plug Specification, Revision 1.1*
  - *PCI Bus Power Management Interface Specification, Revision 1.1*
- PCI Industrial Computer Manufacturers Group (PICMG)
  c/o Virtual Inc., 401 Edgewater Place, Suite 600, Wakefield, MA 01880, USA
  Tel: 781 246-9318, Fax: 781 224-1239, www.picmg.org
  - *PICMG 2.0, R3.0, CompactPCI Specification*
  - *PICMG 2.1, R2.0, CompactPCI Hot Swap Specification*
- I_2O Special Interest Group (I_2O SIG®)
  www.developer.osdl.org/dev/opendoc/Online/Local/I20/index.html
  - *Intelligent I/O (I_2O) Architecture Specification, Revision 1.5*
- The Institute of Electrical and Electronics Engineers, Inc.
  445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331, USA
  Tel: 800 678-4333 (domestic only) or 732 981-0060, Fax: 732 981-1721, www.ieee.org
  - *IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture, 1990*

Note: *In this data book, shortened titles are provided to the previously listed documents. 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 r1.1	*PCI Hot-Plug Specification, Revision 1.1*
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
- **RoHS** – Restrictions on the use of certain Hazardous Substances (RoHS) Directive

Data Assignment Conventions

Data Width	PCI 9656 Convention
Half byte (4 bits)	Nybble
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
03/2003/	1.0	Production Release, Silicon Revision BA.
10/2003	1.1	Update to Version 1.1. Miscellaneous updates, including the following: **Miscellaneous** • Preface, added description for “Nybble.” • Section 7.1.10, added (PCIBAR0) minimum range. • Section 9.2.2.2, revised names of HS_CSR[7:6] and inserted new first sentence to third paragraph. • Section A.1, 2nd to last sentence, inserted the word “pitch” preceding “PBGA.” **Sections 2 and 4** • Table 2-13, added, “and BREQo Control” to Offsets 28h and 2Ah. • Tables 2-14 and 4-16, reversed the Description and Register Bits Affected listed for Offset 56h. • Section 2.2.5.2: 3rd paragraph, last sentence, changed “Lword boundary” to “4-Lword boundary.” • Section 2.4.4, paragraph preceding Figure 2-3, changed “TA# asserted” to “TA# de-asserted.” • Section 4.2.4, added “assertion” to 3rd paragraph, 2nd sentence, now reads, “BTERM# assertion….” • Section 4.2.4.1, 3rd paragraph, added C and J mode, as appropriate, regarding ADS# and ALE. • Section 4.2.5.2, end of 1st paragraph, added, “to a 32-, 16-, or 8-bit bus, respectively).” • Section 4.2.9, end of paragraph, added, “In J mode, LAD[1,0] also provide these address bits during the Address phase.” • Section 4.4.4, paragraph preceding Figure 4-3, changed “READY# asserted” to “READY# de-asserted.”

Date	Revision	Comments
10/2003	1.1	*Update to Version 1.1 (continued):* **Sections 3 and 5** • Tables 3-3 and 5-3, added Footnote 5. • Section 3.4, text following bullets, changed “four” to “three.” • Sections 3.4.1.2 and 5.4.1.2, added Qword equivalents for consistency to the FIFO Qword sizes. • Section 3.4.2.2, 6th paragraph, and Section 5.4.2.2, 5th paragraph, removed “single cycle” from text in parenthesis. • Figure 3-14 is now Figure 3-11 and Figure 5-14 is now Figure 5-11, subsequent figures renumbered. • Sections 3.4.4.2 and 5.4.4.2, added to second to last bullet, following “BIOS,” “according to the PCI Maximum Latency (PCIMLR) register.” • Section 3.4.4.2, removed bullet, “Special cycle BI# input is asserted.” • Sections 3.4.4.3 and 5.4.4.3, last paragraph before Notes, added “transfer” to read as “DMA transfer.” • Section 3.4.4.15, changed “Pause Timer” to “PCI 9656” in last sentence of 1st paragraph. Reworded last paragraph for clarity. • Section 3.5, added, “or is irrelevant” to end of second to last sentence. Added to end of paragraph, “When the PCI 9656 is not executing a transfer on that bus, the value is not shown because it is either irrelevant or unknown (any or all signals may be 1, 0, or Z).” • Timing Diagrams 3-24 and 3-25, reversed sequence. • Timing Diagrams 3-26 and 3-27, changed R/W# to RD/WR#. Added “Local Bus” to BB# text for TD 3-26. • Timing Diagram 3-34, Changed title from “DMA PCI-to-Local…” to “DMA Local-to-PCI….” • Figure 5-3, moved BLAST# into text block with other signals from Local Bus to PCI 9656 (top arrow), and removed BLAST# from lower arrow for consistency with other Figures of similar content. • Section 5.4.1.6, added to end of first paragraph, “Direct Master Read Ahead mode functions can be used with or without Direct Master Delayed Read mode.” • Section 5.4.1.10, 1st paragraph, 1st sentence, inserted “Direct Master” between “PCI Bus for” and “transfers.” • Section 5.4.4.14, 3rd paragraph, replaced with “The DMA Local Bus Latency Timer starts after the Local Bus is granted to the PCI 9656, and the Local Bus Pause Timer starts after the external Local Bus Arbiter de-asserts LHOLDA.” • Section 5.5, added, “or is irrelevant” to end of second to last sentence. Added C and J mode, as appropriate, regarding LD and LAD signals. **Section 10** • Section 10.2.2, revised VPD address/serial EEPROM word address information. • Section 10.5, corrected step 1 CNTRL[31] value (two places). **Section 11** • Register 11-33, revised names of HS_CSR[7:6]. • Registers 11-34 through 11-37, changed Read column from “PCI” to “Yes”. • Register 11-40, MARBR[26], replaced, “When PCI Compliance is enabled, value…” with “Value….” • Register 11-53, added PCIBAR0 minimum range. **Section 12** • Table 12-1, revised DTS pin type description. • Section 12.2, added “to V_{RING}” to first and second paragraphs, and revised Hot Swap pin/resistor information for Non-CompactPCI systems. • Table 12-6, revised CPCISW function description. **Section 13** • Section 13, removed Preliminary watermark. • Section 13.1, added the following sentence before second to last sentence, “It is recommended that V_{CORE}, 2.5V_{AUX}, and V_{IO} receive power first or as close to V_{RING} and Card_V_{AUX} as reasonable (up to the 10 ms maximum).” • Section 13.2, Table 13-1, removed Maximum Package Power Dissipation row from table.

Date	Revision	Comments
10/2008	1.2	Production update, Silicon Revision BA. Miscellaneous updates, including the following: • Applied miscellaneous corrections, changes, and enhancements throughout data book. • Incorporated all items from the *PCI 9656BA Data Book Addendum, Revision 1.0*. • Incorporated item 3 from the *PCI 9656BA Design Note, Revision 1.2*, which replaced the content in Sections 3.4.4.11 and 5.4.4.10. • Register 11-40, expanded descriptions of MARBR[15:8, 7:0]. • Table 12-12, corrected DT/R# description. • Replaced mechanical dimensions illustration in Figure 14-1. • Omitted Index (from end of document).
1/2009	1.3	Production update, Silicon Revision BA. Updated Note 2 of Table 13-5. Corrected LMISC1[3]-related text.

PCI 9656

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

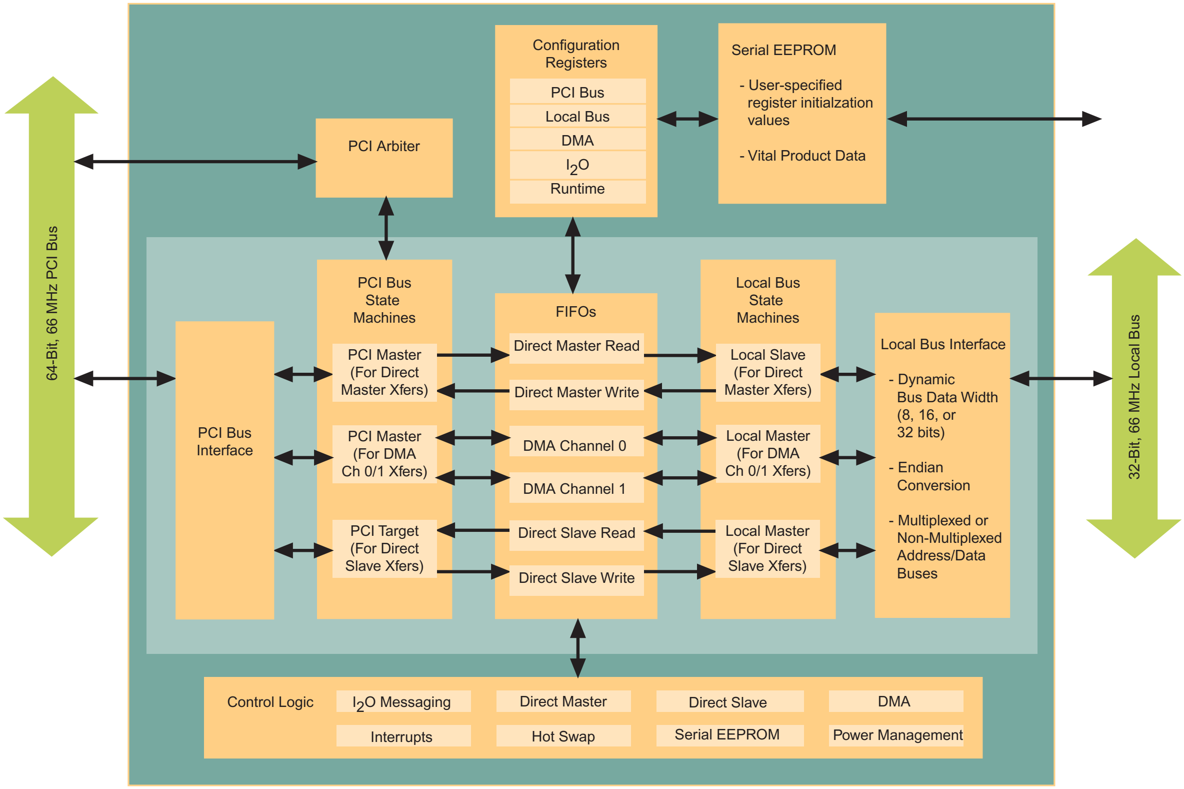
for PowerQUICC and Generic
64-Bit, 66 MHz Local Bus Designs

January, 2009
Version 1.3

FEATURE SUMMARY

- Bus Mastering interface between a 64-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 Mgmt. r1.1*
  - *PICMG 2.1 R2.0* Hot Swap Silicon
    - Programming Interface 0 (PI = 0)
    - Precharge Voltage support
    - Early Power support
    - Initially Not Respond support
    - 64-Bit Initialization support
  - *Hot-Plug r1.1*-compatible
  - Direct connection to three processor Local Bus types
    - **M Mode** – Motorola MPC850 and MPC860, IBM 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, and others
  - Asynchronous clock inputs for PCI and Local Buses
- 272-pin, 27 x 27 mm^2, 1.27 mm ball pitch PBGA
  - Low-power CMOS 2.5V core, 3.3V I/O
  - 3.3V, 5V 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 to the PCI Bus – one to PCI memory and one to PCI I/O
    - Generates all PCI Memory and I/O transaction types, including Memory Write and Invalidate (MWI) and Type 0 and Type 1 configuration
    - Read Ahead, Programmable Read Prefetch Counter(s) (all modes)
    - MPC850 and 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 Local Bus Ready timeout and recovery
  - **DMA** – The PCI 9656 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 descriptor DMA Ring Management mode 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 Channel 1
- Advanced features common to Direct Master, Direct Slave, and DMA
  - Zero wait state burst operation
  - 528 MB/s bursts on the PCI Bus
  - 264 MB/s bursts on the 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
- Internal PCI Arbiter supports seven external masters in addition to the PCI 9656
- 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 9054, PCI 9056, PCI 9060, and PCI 9080



PCI 9656 Block Diagram

1 INTRODUCTION

1.1 COMPANY AND PRODUCT BACKGROUND

PLX Technology, Inc. (www.plxtech.com), is a leading supplier of high-speed, I/O interconnect silicon for the server, storage, communications and industrial control industries.

PLX's expansive, I/O interconnect product offering ranges from I/O Accelerators, Switched-PCI controllers, and HyperTransport™ bridges to the PLX PCI Express-based family of switches and bridges, currently under development. These PCI Express devices are part of the first wave of PCI Express products. PCI Express-based systems are expected to appear in early 2004.

In addition to a broad product offering, PLX provides development tool support through Software Development Kits (SDKs), Rapid Development Kits (RDKs), and third-party tool support through the PLX Partner Program. The complete tool offering, combined with PLX silicon, enables hardware designers and software developers to maximize system input/output (I/O), lower development costs, minimize system design risk, and provide faster time to market. The PLX commitment to meeting customers' requirements extends beyond product solution, to participation in industry associations.

PLX participates in and contributes to several industry standard-setting bodies, including the Arapahoe Working Group (AWG), CompactPCI, HyperTransport Consortium, PCI-SIG, PICMG, and Server Blade (SBTA) trade associations. As an active member of the AWG, PLX is defining and contributing its extensive experience to the Advanced Switching and Bridging working groups. Additionally, PLX chairs subcommittees within PCI-SIG, the special interest group responsible for the release of all PCI Express specifications. PLX cochairs and is in the process of editing the AdvancedTCA (PICMG) 3.4 initiative, driving compliancy and standardization for chassis in next-generation systems. Furthermore, PLX is a key developer for the PCI Express technology and a member of the Intel Developers Network for PCI Express Technology.

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

PLX is publicly traded (NASDAQ:PLXT) and headquartered in Sunnyvale, CA, USA, with other domestic offices in Utah and Southern California. PLX European operations are based in the United Kingdom and Asian operations are based in China and Japan.

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 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 mapping of one or more independent Direct Master Local Bus address spaces to PCI addresses, as shown in Figure 1-1.

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

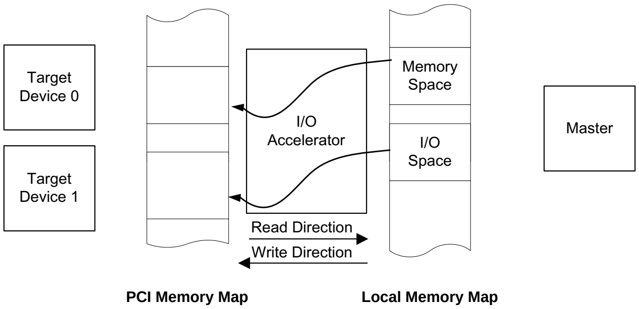


Figure 1-1. Direct Master Address Mapping

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 mapping of one or more independent Direct Slave PCI Address spaces to Local Bus addresses, as shown in Figure 1-2.

With software, Direct Slave transfers support Local Bus Data transfers of various widths (*for example*, on 32-bit Local Buses, data widths of 8, 16, and 32 bits are supported). Direct Slave Read transfers also support PCI Delayed Reads.

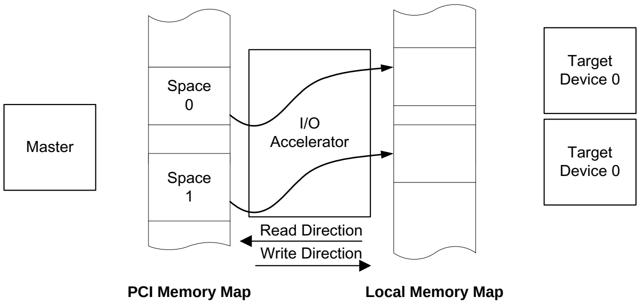


Figure 1-2. Direct Slave Address Mapping

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 offloaded 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 and Scatter/Gather.

1.2.1.2.1 DMA Block Mode

DMA Block mode is the simplest DMA mode. The Master simply programs the description of a single transfer in the I/O accelerator and sets the Start bit(s) (DMACSR*x*[1]=1). The I/O accelerator signals DMA completion to the Master, either by setting a Done bit in one of its registers that the Master polls (DMACSR*x*[4]) or by asserting an interrupt.

1.2.1.2.2 DMA 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, DMA Scatter/Gather mode is used to enable I/O accelerator list processing with minimal master intervention.

With DMA Scatter/Gather mode, the Master simply tells the I/O accelerator the location of the first descriptor in its list, sets the Start bit(s) (DMACSR*x*[1]=1), then waits for the I/O accelerator to service the entire list. This offloads both data and DMA descriptor transfer responsibilities from the Master.

Data Pipe Architecture technology supports Scatter/Gather mode descriptor lists in PCI or Local Bus memory. It also supports linear and circular descriptor lists, the latter is called *DMA Ring Management mode*.

DMA Ring Management 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 Ring Management Valid bit(s) when done (DMASIZ*x*[31]=1), and the I/O accelerator following behind servicing valid descriptors, 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(s).

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 of the Data transfer.

With End of Transfer (EOT), an EOT# signal is asserted to the I/O accelerator to end the transfer. When 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 *DMA Demand mode*, a hardware DREQ*x*#/DACK*x*# signal pair is used to pause and resume the DMA transfer. Data Pipe Architecture technology provides one DREQ*x*#/DACK*x*# signal pair for each DMA channel. Demand mode provides a means for a peripheral device with its own FIFO to control DMA transfers. The peripheral device uses Demand mode both to pause the transfer when the FIFO is full on a write or empty on a read and to resume the transfer when the FIFO condition changes to allow the Data transfer to continue. Demand mode can also be used for many non-FIFO-based applications.

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 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 the general-purpose Mailbox and Doorbell registers.

The second method is provided through Intelligent I/O (I_2O) support. 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 systems that include PCI or CompactPCI expansion slots.

1.3 PCI 9656 I/O ACCELERATOR

The PCI 9656, a 64-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 9656 with other PLX Bus Mastering I/O Accelerators.)

The PCI 9656 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 528 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 9656 continues the PLX tradition of extending its product capabilities to meet the leading edge requirements of I/O intensive embedded- processor applications. The PCI 9656 builds upon the industry-leading PLX PCI 9054 and PCI 9080 products, providing an easy upgrade path to 64-bit, 66 MHz PCI Bus operation, and 32-bit, 66 MHz Local Bus operation. The PCI 9656 supports all legacy processors and designs using the M, C, and J Local Bus interfaces. Additionally, the PCI 9656 adds several important new features that expand its applicability and performance.

1.3.1.1 High-Performance Motorola MPC850 and MPC860 PowerQUICC Designs

A key application for the PCI 9656 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 9656 simplifies these designs by providing an industry-leading enhanced direct-connect interface to the MPC850 or MPC860 processor. The flexible PCI 9656 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 9656 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 9656 makes use of the advanced Data Pipe Architecture technology, allowing unlimited burst capability, as shown in Figure 1-3.

Regarding items 1 and 2 in Figure 1-3:

1. For PowerQUICC IDMA operation, the PCI 9656 transfers data to the PCI Bus under the control of the IDMA handshake protocol using Direct Master transfers (1).

2. Simultaneously, the PCI 9656 DMA can be operated bi-directionally, with the PCI 9656 as the Master for both buses, to manage transfers of data between the PCI and Local Buses (2).

This is a prime example of how the PCI 9656 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 9656 has unlimited bursting capability, which enhances most MPC850 or MPC860 PowerQUICC designs.

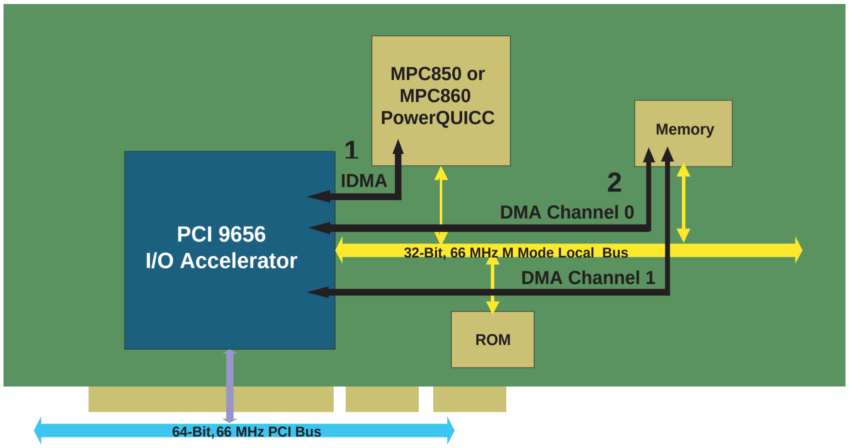


Figure 1-3. High-Performance MPC850 or MPC860 PowerQUICC Adapter Design

1.3.1.2 High-Performance CompactPCI Adapter Cards

Another key application for the PCI 9656 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 9656 is the perfect choice for adding CompactPCI connection capabilities to a variety of processor platforms.

The PCI 9656 has integrated key features to enable live insertion of CompactPCI Hot Swap adapters:

- *PCI r2.2*-compliant
- 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 precharge 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 status LED which indicates insertion and removal to the user
- Precharge Voltage support, with integrated 10KΩ 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
- 64-Bit Initialization support

Figure 1-4 illustrates a CompactPCI peripheral card that uses an MPC860 CPU for communication I/O and the PCI 9656 for PCI-based I/O.

The PCI 9656, with its internal PCI Arbiter, reset signal direction control, and Type 0 and Type 1 PCI configuration support, is also an ideal choice for CompactPCI system cards.

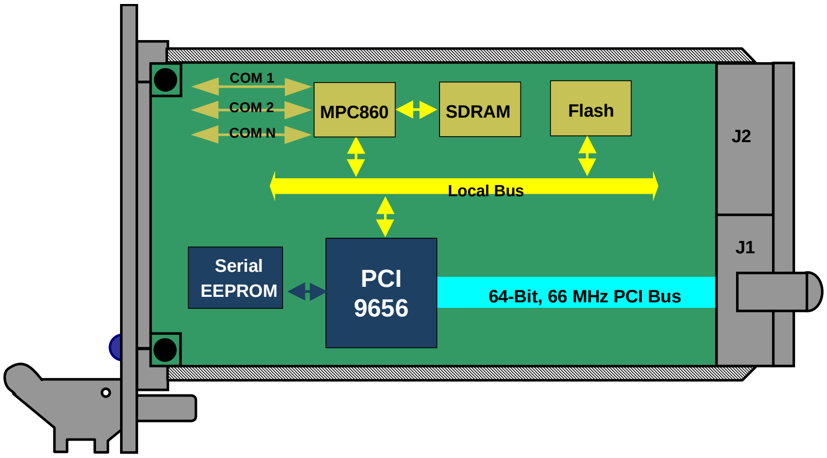


Figure 1-4. PCI 9656 CompactPCI Peripheral Card

1.3.1.3 High-Performance PCI Adapter Cards

The PCI 9656 is also designed for traditional PCI adapter card applications requiring 64-bit, 66 MHz PCI operation and bandwidth. Specific applications include high-performance communications, networking, disk control, and data encryption adapters. As such, the PCI 9656 enables easy migration of existing 32-bit, 33 MHz PCI I/O accelerator designs to 64-bit, 66 MHz capability.

Today, Power Management and Green PCs are major initiatives in traditional PCI applications. The PCI 9656 supports PCI Power Management.

Figure 1-5 illustrates the PCI 9656 in a PCI adapter card application with a CPU, using the C or J Local Bus mode.

The C and J Local Bus modes, in addition to supporting Intel 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 9656 provides 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 9656 and a PCI-to-PCI bridge interfacing to the system PCI Bus. The PCI 9656 internal PCI Arbiter provides arbitration services to the devices on the local PCI Bus.

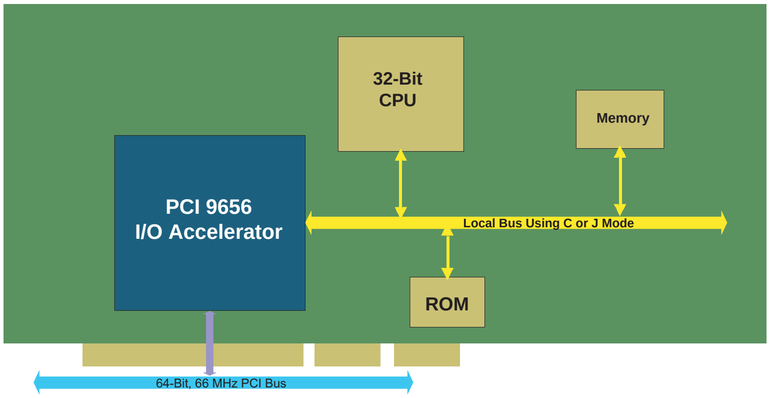


Figure 1-5. PCI 9656 PCI Adapter Card with C or J Mode Processor

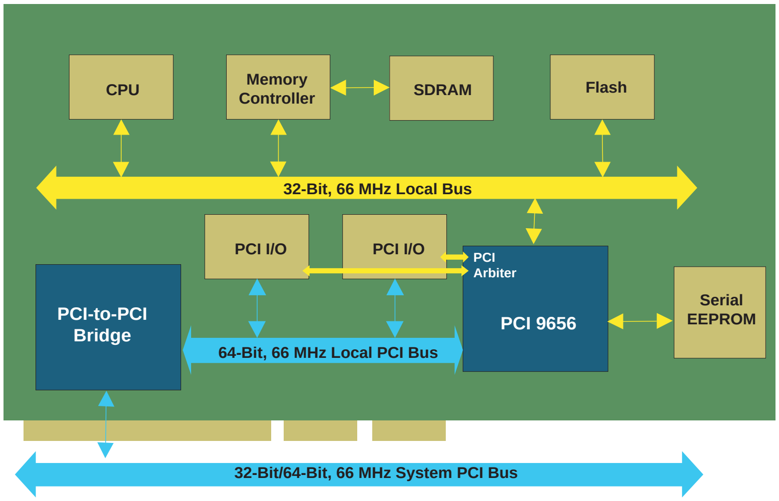


Figure 1-6. PCI 9656 PCI Adapter Card with Local PCI I/O and PCI-to-PCI Bridge

1.3.1.4 High-Performance Embedded Host Designs

I/O intensive embedded host designs are another major application of the PCI 9656. 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 9656 supports PCI Type 0 and Type 1 Configuration cycles to accomplish this.

Second, the host is responsible for providing PCI Bus arbitration services. The PCI 9656 includes an internal PCI Arbiter that supports seven external PCI masters in addition to the PCI 9656. 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 9656 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 1-7 illustrates the PCI 9656 in an embedded host system.

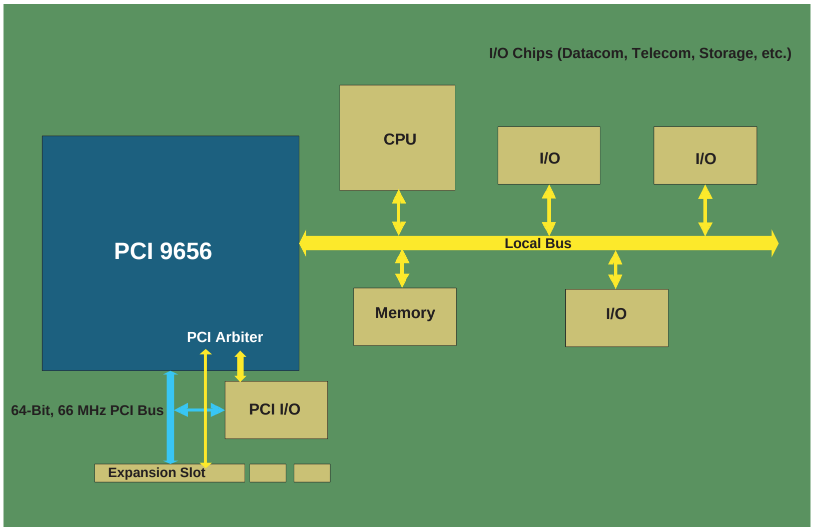


Figure 1-7. PCI 9656 Embedded Host System with Generic Host CPU

1.4 MAJOR FEATURES

1.4.1 Interfaces

The PCI 9656 is a PCI Bus Master interface chip that connects a 64-bit, 66 MHz PCI Bus to one of three 32-bit, 66 MHz Local Bus types.

***PCI r2.1-* and *PCI r2.2*-Compliant.** Compliant with *PCI r2.1 and PCI r2.2*, including 64-bit and 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}.

***PICMG 2.1 R2.0* Hot Swap Silicon.** Compliant with *PICMG 2.1 R2.0*, including support for Programming Interface 0 (PI = 0), Precharge Voltage, Early Power, and 64-Bit Initialization, and an option to Initially Not Respond during chip initialization.

PCI Hot Plug-Compliant. Compliant with *Hot-Plug r1.1*.

Subsystem and Subsystem Vendor IDs. Includes Subsystem and Subsystem Vendor IDs in the PCI Configuration register space, in addition to Device and Vendor IDs. The PCI 9656 also includes a permanent Device ID (9656h) and Vendor ID (10B5h).

RST# Timing. Supports response to first Configuration accesses after RST# de-assertion 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 IBM PowerPC 401 and 801 families, the Motorola MPC850 and MPC860 PowerQUICC families, the Intel i960 family, Texas Instruments DSPs, and other similar bus-protocol devices.

Local Bus Types. Local bus type selected through a pin strapping option, as listed in Table 1-1.

Table 1-1. Local Bus Types

Mode	Description
M	Motorola 32-bit address and data, non-multiplexed, direct connect interface to MPC850 or MPC860 PowerQUICC
C	Intel/Generic 32-bit address and data, non-multiplexed
J	Intel/Generic 32-bit address and 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. Table 1-2 enumerates the FIFO depth.

Table 1-2. FIFO Depth

FIFO	Depth
Direct Master Read	16 Qwords
Direct Master Write	32 Qwords
Direct Slave Read	16 Qwords
Direct Slave Write	32 Qwords
DMA Channel 0	32 Qwords
DMA Channel 1	32 Qwords

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

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.*

M Mode Data Transfers. Communicates with the MPC850 or MPC860, using three possible Data Transfer modes:

- Direct Master Operation, including support for MPC850 and MPC860 IDMA/SDMA Operation
- Direct Slave Operation
- DMA Operation

C and J Mode Data Transfers. Communicates with these processors, using three possible Data Transfer modes:

- 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-mapped and I/O devices.

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 data width, burst capabilities, and so forth.

Direct Master and Direct Slave 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. This feature allows for increased bandwidth utilization through reduced data latency.

Programmable Prefetch Counters. Includes programmable control to prefetch data during Direct Slave and Direct Master prefetches (known or unknown size). 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 Direct Master or Direct Slave 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 Posted Memory Writes 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 DMA Block and Scatter/Gather modes, including DMA Ring Management (Valid mode), as well as EOT and DMA Demand modes.

PCI Dual-Address Cycles Support (64-bit Address Space). Supports PCI Dual Address Cycles (DAC) beyond the low 4-GB Address space. PCI DAC can be used during PCI 9656 PCI Bus Master operation (Direct Master and DMA).

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 *I_2O r1.5* PCI Extension.

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 9656.

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 272-pin, 27 x 27 mm^2 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 9656 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 an optional serial EEPROM interface that can be used to load configuration information. This is useful for loading information unique to a particular adapter, such as the Device or Vendor ID, especially in designs that do not include a Local processor.

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 9656 is ***not*** pin compatible with other PLX Bus Master Accelerator or Target I/O Accelerators.

1.5.2 Register Compatibility

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

The PCI 9656 is ***not*** register-compatible with the following PLX Target I/O Accelerators – PCI 9030, PCI 9050, nor PCI 9052.

1.5.3 PCI 9656 Comparison with Other PLX Chips

Table 1-3 is a comparison of the PCI 9656 with the PCI 9054 and PCI 9056 Bus Master I/O Accelerator chips.

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

Features	PCI 9054	PCI 9056	PCI 9656
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
	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
			C: Generic, 32-Bit A, 32-Bit D, non-mux
			J: Generic, 32-Bit A, 32-Bit D, 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 Signaling	✔	✔	✔
5V Tolerant PCI Bus	✔	✔	✔
3.3V Local Bus Signaling	✔	✔	✔
5V Tolerant Local Bus	✔	✔	✔
PICMG 2.1 R2.0	Programming Interface (PI = 0)	Programming Interface 0 (PI = 0) Precharge Voltage Support Early Power Support Initially Not Respond Support	Programming Interface 0 (PI = 0) Precharge Voltage Support Early Power Support Initially Not Respond Support 64-Bit Initialization Support
Package Size/Type(s): Pin/Ball Count External Dimensions (mm^2) Pin/Ball Pitch (mm) Package Type	176-Pin, 26 x 26, 0.5 PQFP 225-Pin, 27 x 27, 1.5 PBGA	256-Ball, 17 x 17, 1.00 Fine Pitch PBGA (FPBGA)	272-Ball, 27 x 27, 1.27 PBGA
Industrial Temperature Range Operation	✔	✔	✔

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

Features	PCI 9054	PCI 9056	PCI 9656
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 (Direct Master, Direct Slave)	✔	✔	✔
Programmable Processor Bus 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)
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 (Valid Mode)	–	✔	✔
Programmable Prefetch Counter	✔	✔	✔
Dual Address Cycles 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*
Internal PCI Arbiter	–	✔ Seven external masters	✔ Seven external masters
PCI Type 0 and Type 1 Configuration Cycles	✔	✔	✔
PCI Power Management	✔ *r1.1*	✔ *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 9656 is *PCI r2.2*-compliant. Refer to *PCI r2.2* for specific PCI Bus functions.

2.1.1 Direct Slave Command Codes

As a target, the PCI 9656 allows access to the PCI 9656 internal registers and Local Bus, using the commands listed in Table 2-1.

All Direct Slave Read or Write accesses to the PCI 9656 can be Byte (8-bit), Word (16-bit), Lword (32-bit), or Qword (64-bit) accesses. All memory commands are aliased to basic memory commands. All I/O accesses to the PCI 9656 are decoded to an Lword boundary. The PCI Byte Enables (C/BE[7:0]#) 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. All Configuration register Read or Write accesses to the PCI 9656 can be Byte, Word, or Lword accesses, with Byte Enables used to determine which bytes are read or written.

Table 2-1. Direct Slave Command Codes

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

2.1.2 PCI Master Command Codes

The PCI 9656 becomes the PCI Bus Master to perform DMA or Direct Master transfers. The PCI command code used by the PCI 9656 during a Direct Master or DMA transfer is specified by the value(s) contained in the CNTRL[15:0] register bits. Except when in Memory Write and Invalidate (MWI) mode (PCICR[4]=1; DMPBAM[9]=1 or DMAMODE*x*[13]=1; PCICLSR[7:0] = cache line size of 8 or 16 Lwords). In MWI mode for a Direct Master or DMA transfer, if the starting address alignment is aligned with the cache line size and upon determining it can transfer at least one cache line of data, the PCI 9656 uses a PCI command code of Fh regardless of the value(s) contained in the CNTRL[15:0] register bits.

Note: *DMA can only perform Memory accesses. DMA* ***cannot*** *perform I/O or Configuration accesses.*

2.1.2.1 DMA Master Command Codes

The PCI 9656 DMA Controllers can assert the Memory cycles listed in Table 2-2.

Table 2-2. DMA Master Command Codes

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

Note: *During each command phase, the PCI 9656 drives C/BE[7:4]# to unspecified values.*

2.1.2.2 Direct Master Local-to-PCI Command Codes

For Direct Master Local-to-PCI Bus accesses, the PCI 9656 asserts the cycles listed in Tables 2-3 through 2-5.

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

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

Note: *During each command phase, the PCI 9656 drives C/BE[7:4]# to unspecified values.*

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

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

Note: *During each command phase, the PCI 9656 drives C/BE[7:4]# to unspecified values.*

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

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

Note: *During each command phase, the PCI 9656 drives C/BE[7:4]# to unspecified values.*

2.1.3 PCI Arbitration

The PCI 9656 asserts REQ# to request the PCI Bus. The PCI 9656 can be programmed using the PCI Request Mode bit (MARBR[23]) to de-assert REQ# when the PCI 9656 asserts FRAME# during a Bus Master cycle, or to hold REQ# asserted for the entire Bus Master cycle. The PCI 9656 always de-asserts REQ# for a minimum of two PCI clocks between Bus Master ownership that includes a Target disconnect.

During a Direct Master Write cycle, the PCI 9656 PCI REQ# assertion can be delayed by programming the Direct Master Delayed Write Mode bits (DMPBAM [15:14]). DMPBAM can be programmed to wait 0, 4, 8, or 16 PCI Bus clocks after the PCI 9656 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 9656 Direct Master Write FIFO, which provides for better use of the PCI Bus.

2.1.4 PCI Bus Wait States

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

2.2 LOCAL BUS CYCLES

The PCI 9656 interfaces a PCI Host Bus to several Local Bus types, as listed in Table 2-6. It operates in one of three modes – M, C, and J, selected through the MODE[1:0] pins – corresponding to the three bus types.

In M mode, the PCI 9656 provides a direct connection to the MPC850 or MPC860 address and data lines, regardless of the PCI 9656 Big or Little Endian modes.

Table 2-6. MODE Pin-to-Bus Mode Cross-Reference

MODE1	MODE0	Bus Mode	Bus Type
1	1	M	Motorola, 32-bit non-multiplexed
1	0	***Reserved***	–
0	0	C	Intel/Generic, 32-bit non-multiplexed
0	1	J	Intel/Generic, 32-bit multiplexed

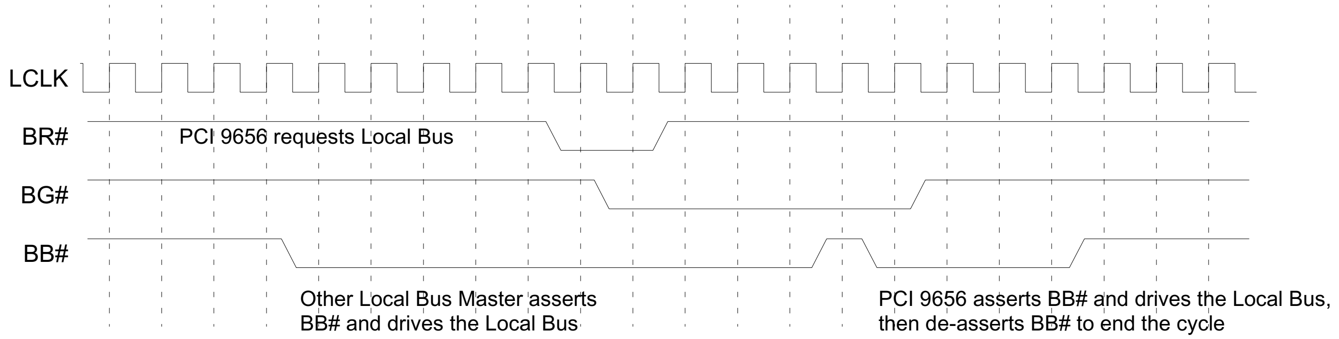
2.2.1 Local Bus Arbitration

The PCI 9656 asserts BR# to request the Local Bus for a Direct Slave or DMA transfer, then waits for the external Local Bus Arbiter to assert BG#. Upon receiving BG#, the PCI 9656 waits for BB# to de-assert (to ensure that no other master is driving the bus) before it asserts BB# (on the next rising edge of the Local clock) to become the Local Bus Master and drive the bus. As the Local Bus Master, the PCI 9656 continues to assert BB# until either the transaction is complete or the Local Bus Latency Timer (MARBR[7:0]), if enabled (MARBR[16]=1), expires (whichever occurs first).

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

2.2.1.1 Local Bus Arbitration Timing Diagram

Timing Diagram 2-1. Local Bus Arbitration (BR#, BG#, and BB#)


LCLK
BR#
PCI 9656 requests Local Bus
BG#
BB#
Other Local Bus Master asserts BB# and drives the Local Bus
PCI 9656 asserts BB# and drives the Local Bus, then de-asserts BB# to end the cycle

Note: *Timing Diagram 2-1 was created using the Timing Designer tool. It is accurate and adheres to its specified protocol(s). This diagram shows transfers and signal transitions, but should not be relied upon to show exactly, on a clock-for-clock basis, where PCI 9656-driven signal transitions will occur.*

2.2.2 Direct Master

Local Bus cycles can be single or Burst cycles. As a Local Bus target, the PCI 9656 allows access to the PCI 9656 internal registers and the PCI Bus.

Local Bus Direct Master accesses to the PCI 9656 must be for a 32-bit non-pipelined bus. Non-32-bit Direct Master accesses to the PCI 9656 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 9656 is a PCI Bus target and a Local Bus master).

2.2.4 Wait State Control

Figure 2-1 illustrates the PCI 9656 wait states for M mode.

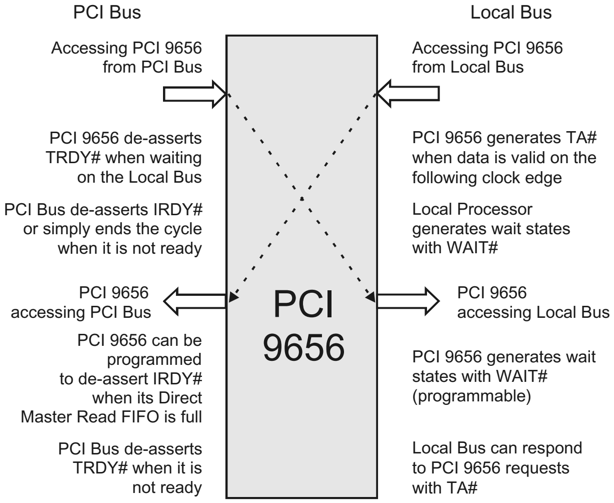

PCI Bus
Local Bus
Accessing PCI 9656 from PCI Bus
Accessing PCI 9656 from Local Bus
PCI 9656 de-asserts TRDY# when waiting on the Local Bus
PCI 9656 generates TA# when data is valid on the following clock edge
PCI Bus de-asserts IRDY# or simply ends the cycle when it is not ready
Local Processor generates wait states with WAIT#
PCI 9656 accessing PCI Bus
PCI 9656 accessing Local Bus
PCI 9656
PCI 9656 can be programmed to de-assert IRDY# when its Direct Master Read FIFO is full
PCI 9656 generates wait states with WAIT# (programmable)
PCI Bus de-asserts TRDY# when it is not ready
Local Bus can respond to PCI 9656 requests with TA#

Figure 2-1. Wait States

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

2.2.4.1 Local Bus Wait States

In Direct Master mode, when accessing the PCI 9656 registers or transferring data, the PCI 9656 is a Local Bus slave. As a Local Bus slave, the PCI 9656 inserts external wait states with the TA# signal.

In Direct Slave and DMA modes, the PCI 9656 acts as a Local Bus master. When TA# input is disabled, the Internal Wait State Counter(s) can be used to program the number of internal wait states between the first address-to-data state, and subsequent data-to-data in Burst mode. (Refer to Table 2-7.)

In Direct Slave and DMA modes, if TA# is enabled (LBRD0[6]=1 for Space 0, LBRD1[6]=1 for Space 1, LBRD0[22]=1 for Expansion ROM, and/or DMAMODE*x*[6]=1 for Channel *x*), the Wait State Counter(s) and the WAIT# pin should ***not*** be used. The external memory control can use the TA# input to insert wait states.

Table 2-7. Internal Wait State Counters

Bits	Description
LBRD0[5:2]	Local Address Space 0 Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)
LBRD1[5:2]	Local Address Space 1 Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)
LBRD0[21:18]	Expansion ROM Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)
DMAMODE0[5:2]	DMA Channel 0 Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)
DMAMODE1[5:2]	DMA Channel 1 Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)

2.2.5 Data Transfer Modes

The PCI 9656 supports the MPC850 and MPC860 with three Data Transfer modes:

- Single Cycle
- Burst-4
- Continuous Burst

Single Cycle mode is the default Data Transfer mode, Burst-4 mode is MPC850- and MPC860-compatible, and Continuous Burst mode provides the highest throughput.

Table 2-8 summarizes the register settings used to select Local Bus Data Transfer modes. It also indicates the data quantity transferred per Address cycle (TS#).

Notes: *The BI#/BTERM# pin is referred to as the BI# pin in M mode, and as the BTERM# pin in C and J modes.*

The BTERM# Input Enable bits (LBRD0[7] for Space 0, LBRD1[7] for Space 1, LBRD0[23] for Expansion ROM, and/or DMAMODEx[7] for Channel x) function as BI mode bits in M mode.

The term "Burst Forever" was formerly used to describe "Continuous Burst."

Table 2-8. Local Bus Data Transfer Modes

Mode	Burst Enable Bit(s)	BI Mode Bit(s)	Result
Single Cycle	0	X	One TS# per data.
Burst-4	1	0	One TS# per 16 bytes (recommended for MPC850 or MPC860).
Continuous Burst	1	1	One TS# per data burst or until BI# is asserted.

Notes: *Single cycle is the default Data Transfer mode.*

"X" is "Don't Care."

2.2.5.1 Single Cycle Mode

Single Cycle mode is the default Data Transfer mode. In Single Cycle mode, the PCI 9656 issues one TS# per data cycle. The starting address for a single cycle Data transfer can be on any address. If a starting address in a Direct Slave or DMA PCI-to-Local transfer is ***not*** aligned to an Lword boundary, the PCI 9656 performs a single cycle until the next 4-Lword boundary.

For single cycle Data transfers, Burst mode is disabled (LBRD0[24]=0 for Space 0, LBRD1[8]=0 for Space 1, LBRD0[26]=0 for Expansion ROM, and/or DMAMODE*x*[8]=0 for Channel *x*).

2.2.5.1.1 Partial Data Accesses

Partial Data accesses (not all Byte Enables are asserted) are broken into single cycles. If there is remaining data that is ***not*** Lword-aligned during DMA PCI-to-Local transfers, it results in a single cycle Data transfer.

2.2.5.2 Burst-4 Mode

Burst-4 mode forces the PCI 9656 to perform Data transfers as bursts of 16 bytes, regardless of the Local Bus data width.

Burst-4 mode Data transfers are set up by enabling bursting and clearing the BI mode bit(s) (LBRD0[24, 7]=10b for Space 0, LBRD1[8:7]=10b for Space 1, LBRD0[26, 23]=10b for Expansion ROM, and/or DMAMODE*x*[8:7]=10b for Channel *x*, respectively).

Burst-4 mode bursting starts on any 4-Lword boundary and bursts to the next 4-Lword boundary. The PCI 9656 can continue to burst by asserting TS# and performing another burst. (Refer to Table 2-9.) If a starting address in a Direct Slave or DMA PCI-to-Local transfer is ***not*** aligned to a 4-Lword boundary, the PCI 9656 performs a single cycle until the next 4-Lword boundary.

For DMA, if Burst-4 mode is implemented with Address Increment disabled (DMAMODE*x*[11]=1), the PCI 9656 defaults to Continuous Burst mode.

Table 2-9. Burst-4 Mode

Local Bus Data Width	Burst-4
32 bit	4 Lwords start/stop at a 4-Lword boundary
16 bit	8 words start/stop at a 4-Lword boundary
8 bit	16 bytes start/stop at a 4-Lword boundary

2.2.5.2.1 Partial Data (<4 Bytes) Accesses

Partial Data accesses occur when one or more PCI Byte Enables (C/BE[7:0]#) are ***not*** asserted. The accesses are broken in single cycles until the next 4-Lword boundary.

2.2.5.3 Continuous Burst Mode

Continuous Burst mode enables the PCI 9656 to perform data bursts of longer than 4 Lwords. However, special external M mode interface devices are required that can accept bursts longer than 4 Lwords.

Continuous Burst mode Data transfers are set up by enabling bursting and setting the BI mode bit(s) (LBRD0[24, 7]=11b for Space 0, LBRD1[8:7]=11b for Space 1, LBRD0[26, 23]=11b for Expansion ROM, and/or DMAMODE*x*[8:7]=11b for Channel *x*, respectively).

The PCI 9656 asserts one TS# cycle and continues to burst data. If a slave device requires a new Address cycle (TS#), it can assert the BI# input. The PCI 9656 completes the current Data transfer and stops the burst. The PCI 9656 continues the transfer by asserting TS# and beginning a new burst at the next address.

For DMA, if Burst-4 mode is implemented with Address Increment disabled (DMAMODE*x*[11]=1), the PCI 9656 defaults to Continuous Burst mode.

2.2.6 Local Bus Read Accesses

For all single cycle Local Bus Read accesses, when the PCI 9656 is the Local Bus Master, the PCI 9656 reads only bytes corresponding to Byte Enables requested by the PCI Master applicable to 32 bits. For a 64-bit PCI Bus single cycle read, the Local Bus performs multiple Read cycles (with all Byte Enables on). For all Burst Read cycles, the PCI 9656 passes all the bytes and can be programmed to:

- Prefetch
- Perform Direct Slave 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 PCI 9656 DMA Controller(s) are written.

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

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

2.2.9 Local Bus Data Parity

Generation or use of Local Bus data parity is optional. The Local Bus Parity Check is passive and provides only 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 9656 data bus (DP[0:3]). “Even data parity” is asserted for each lane during Local Bus reads from the PCI 9656 and during PCI 9656 Master writes to the Local Bus.

Even data parity is checked for Direct Slave reads, Direct Master writes, and DMA Local Bus reads. If an error is detected, the PCI 9656 sets the Direct Master Write/Direct Slave Read Local Data Parity Check Error Status bit (INTCSR[7]=1) and asserts an interrupt (LINTo#), if enabled (INTCSR[16, 6]=11b, respectively). This occurs in the Clock cycle following the data being checked.

For applications in which TA# is disabled in the PCI 9656 registers, an external pull-down resistor is required for TA# to allow INTCSR[7] and the LINTo# interrupt to be set and asserted, respectively.

2.3 BIG ENDIAN/LITTLE ENDIAN

2.3.1 PCI Bus Data Bits Mapping onto Local Bus

Tables 2-10 and 2-11 clarify PCI 9656 signal mapping between the PCI and Local Buses during Big/Little Endian conversion.

Notes:

1. *During 64-bit PCI transfers, the lower 32 bits of the PCI Bus (AD[31:0]) are always mapped first.*

2. *In each Local Bus pin entry, **n-m** denotes that that row's PCI pin maps to Local Bus pin **m** during Local Bus cycle **n** that results from the PCI cycle (PCI-to-Local Bus transfers) or in the PCI cycle (Local-to-PCI Bus transfers). Examples follow.*

   In Table 2-10 (refer to shaded entry), a Local Bus pin of "2-LD26" for PCI pin AD21 during 16-bit Little Endian Local Bus transfers, denotes that during a PCI-to-Local Bus transfer, the value of PCI pin AD21 during each 32-bit PCI transfer occurs on Local Bus pin LD26 of the second resulting 16-bit Local Bus transfer. During a Local-to-PCI Bus transfer, it denotes that the value of PCI pin AD21 results from the value of Local Bus pin LD26 during the second 16-bit Local Bus transfer.

   In Table 2-11 (refer to shaded entry), a Local Bus pin of "2-LD10" for PCI pin AD21 during 16-bit Little Endian Local Bus transfers denotes that during a PCI-to-Local Bus transfer, the value of PCI pin AD21 during each 32-bit PCI transfer occurs on Local Bus pin LD10 of the second resulting 16-bit Local Bus transfer. During a Local-to-PCI Bus transfer, it denotes that the value of PCI pin AD21 results from the value of Local Bus pin LD10 during the second 16-bit Local Bus transfer.

3. *Mappings occur only during Data phases. Addresses always map to/from PCI AD[31:0], as indicated in the 32-bit Little Endian column after the address translation specified in the Configuration registers is performed.*

4. *Big/Little Endian modes are selected by register bits and pin signals, depending on the Data phase type (Direct Master read/write, Direct Slave read/write, DMA PCI-to-Local Bus/ Local-to-PCI Bus, and Configuration register read/write). [For further details, refer to the BIGEND register, and to the BIGEND# pin description in Table 12-10, "M Mode Local Bus Pins."]*

Table 2-10. PCI Bus Data Bits Mapping onto Local Bus M Mode Low Byte Lane

PCI Pins Mapped 2nd (64-Bit Transfers Only)	PCI Pins Mapped 1st	M Mode Local Bus Pin Byte Lane Mode = 0 (BIGEND[4]=0)					
		Little Endian			Big Endian		
		32-Bit	16-Bit	8-Bit	32-Bit	16-Bit	8-Bit
AD32	AD0	1-LD31	1-LD31	1-LD31	1-LD7	1-LD23	1-LD31
AD33	AD1	1-LD30	1-LD30	1-LD30	1-LD6	1-LD22	1-LD30
AD34	AD2	1-LD29	1-LD29	1-LD29	1-LD5	1-LD21	1-LD29
AD35	AD3	1-LD28	1-LD28	1-LD28	1-LD4	1-LD20	1-LD28
AD36	AD4	1-LD27	1-LD27	1-LD27	1-LD3	1-LD19	1-LD27
AD37	AD5	1-LD26	1-LD26	1-LD26	1-LD2	1-LD18	1-LD26
AD38	AD6	1-LD25	1-LD25	1-LD25	1-LD1	1-LD17	1-LD25
AD39	AD7	1-LD24	1-LD24	1-LD24	1-LD0	1-LD16	1-LD24
AD40	AD8	1-LD23	1-LD23	2-LD31	1-LD15	1-LD31	2-LD31
AD41	AD9	1-LD22	1-LD22	2-LD30	1-LD14	1-LD30	2-LD30
AD42	AD10	1-LD21	1-LD21	2-LD29	1-LD13	1-LD29	2-LD29
AD43	AD11	1-LD20	1-LD20	2-LD28	1-LD12	1-LD28	2-LD28
AD44	AD12	1-LD19	1-LD19	2-LD27	1-LD11	1-LD27	2-LD27
AD45	AD13	1-LD18	1-LD18	2-LD26	1-LD10	1-LD26	2-LD26
AD46	AD14	1-LD17	1-LD17	2-LD25	1-LD9	1-LD25	2-LD25
AD47	AD15	1-LD16	1-LD16	2-LD24	1-LD8	1-LD24	2-LD24
AD48	AD16	1-LD15	2-LD31	3-LD31	1-LD23	2-LD23	3-LD31
AD48	AD17	1-LD14	2-LD30	3-LD30	1-LD22	2-LD22	3-LD30
AD50	AD18	1-LD13	2-LD29	3-LD29	1-LD21	2-LD21	3-LD29
AD51	AD19	1-LD12	2-LD28	3-LD28	1-LD20	2-LD20	3-LD28
AD52	AD20	1-LD11	2-LD27	3-LD27	1-LD19	2-LD19	3-LD27
AD53	AD21	1-LD10	2-LD26	3-LD26	1-LD18	2-LD18	3-LD26
AD54	AD22	1-LD9	2-LD25	3-LD25	1-LD17	2-LD17	3-LD25
AD55	AD23	1-LD8	2-LD24	3-LD24	1-LD16	2-LD16	3-LD24
AD56	AD24	1-LD7	2-LD23	4-LD31	1-LD31	2-LD31	4-LD31
AD57	AD25	1-LD6	2-LD22	4-LD30	1-LD30	2-LD30	4-LD30
AD58	AD26	1-LD5	2-LD21	4-LD29	1-LD29	2-LD29	4-LD29
AD59	AD27	1-LD4	2-LD20	4-LD28	1-LD28	2-LD28	4-LD28
AD60	AD28	1-LD3	2-LD19	4-LD27	1-LD27	2-LD27	4-LD27
AD61	AD29	1-LD2	2-LD18	4-LD26	1-LD26	2-LD26	4-LD26
AD62	AD30	1-LD1	2-LD17	4-LD25	1-LD25	2-LD25	4-LD25
AD63	AD31	1-LD0	2-LD16	4-LD24	1-LD24	2-LD24	4-LD24

Table 2-11. PCI Bus Data Bits Mapping onto Local Bus M Mode High Byte Lane

PCI Pins Mapped 2nd (64-Bit Transfers Only)	PCI Pins Mapped 1st	M Mode Local Bus Pin Byte Lane Mode = 1 (BIGEND[4]=1)					
		Little Endian			Big Endian		
		32-Bit	16-Bit	8-Bit	32-Bit	16-Bit	8-Bit
AD32	AD0	1-LD31	1-LD15	1-LD7	1-LD7	1-LD7	1-LD7
AD33	AD1	1-LD30	1-LD14	1-LD6	1-LD6	1-LD6	1-LD6
AD34	AD2	1-LD29	1-LD13	1-LD5	1-LD5	1-LD5	1-LD5
AD35	AD3	1-LD28	1-LD12	1-LD4	1-LD4	1-LD4	1-LD4
AD36	AD4	1-LD27	1-LD11	1-LD3	1-LD3	1-LD3	1-LD3
AD37	AD5	1-LD26	1-LD10	1-LD2	1-LD2	1-LD2	1-LD2
AD38	AD6	1-LD25	1-LD9	1-LD1	1-LD1	1-LD1	1-LD1
AD39	AD7	1-LD24	1-LD8	1-LD0	1-LD0	1-LD0	1-LD0
AD40	AD8	1-LD23	1-LD7	2-LD7	1-LD15	1-LD15	2-LD7
AD41	AD9	1-LD22	1-LD6	2-LD6	1-LD14	1-LD14	2-LD6
AD42	AD10	1-LD21	1-LD5	2-LD5	1-LD13	1-LD13	2-LD5
AD43	AD11	1-LD20	1-LD4	2-LD4	1-LD12	1-LD12	2-LD4
AD44	AD12	1-LD19	1-LD3	2-LD3	1-LD11	1-LD11	2-LD3
AD45	AD13	1-LD18	1-LD2	2-LD2	1-LD10	1-LD10	2-LD2
AD46	AD14	1-LD17	1-LD1	2-LD1	1-LD9	1-LD9	2-LD1
AD47	AD15	1-LD16	1-LD0	2-LD0	1-LD8	1-LD8	2-LD0
AD48	AD16	1-LD15	2-LD15	3-LD7	1-LD23	2-LD7	3-LD7
AD48	AD17	1-LD14	2-LD14	3-LD6	1-LD22	2-LD6	3-LD6
AD50	AD18	1-LD13	2-LD13	3-LD5	1-LD21	2-LD5	3-LD5
AD51	AD19	1-LD12	2-LD12	3-LD4	1-LD20	2-LD4	3-LD4
AD52	AD20	1-LD11	2-LD11	3-LD3	1-LD19	2-LD3	3-LD3
AD53	AD21	1-LD10	2-LD10	3-LD2	1-LD18	2-LD2	3-LD2
AD54	AD22	1-LD9	2-LD9	3-LD1	1-LD17	2-LD1	3-LD1
AD55	AD23	1-LD8	2-LD8	3-LD0	1-LD16	2-LD0	3-LD0
AD56	AD24	1-LD7	2-LD7	4-LD7	1-LD31	2-LD15	4-LD7
AD57	AD25	1-LD6	2-LD6	4-LD6	1-LD30	2-LD14	4-LD6
AD58	AD26	1-LD5	2-LD5	4-LD5	1-LD29	2-LD13	4-LD5
AD59	AD27	1-LD4	2-LD4	4-LD4	1-LD28	2-LD12	4-LD4
AD60	AD28	1-LD3	2-LD3	4-LD3	1-LD27	2-LD11	4-LD3
AD61	AD29	1-LD2	2-LD2	4-LD2	1-LD26	2-LD10	4-LD2
AD62	AD30	1-LD1	2-LD1	4-LD1	1-LD25	2-LD9	4-LD1
AD63	AD31	1-LD0	2-LD0	4-LD0	1-LD24	2-LD8	4-LD0

2.4 SERIAL EEPROM

This section describes the use of a serial EEPROM with the PCI 9656.

The PCI 9656 supports 2K bit or 4K bit serial EEPROMs. The PCI 9656 requires serial EEPROMs that support sequential reads. (The PLX PCI 9656 Toolbox includes the latest information on supported serial EEPROMs.)

The PCI 9656 supports two serial EEPROM load lengths – long serial EEPROM and extra long serial EEPROM (refer to Figure 2-2):

- **Long Load Mode** – Default. The PCI 9656 loads 34, 16-bit words from the serial EEPROM if the Extra Long Load from Serial EEPROM bit is cleared (LBRD0[25]=0)
- **Extra Long Load Mode** – The PCI 9656 loads 50, 16-bit words from the serial EEPROM if the Extra Long Load from Serial EEPROM bit is set (LBRD0[25]=1)

The serial EEPROM can be read or written from the PCI or Local Buses, through either the Serial EEPROM Control register bits (CNTRL[31, 27:24]) or VPD capability.

The 3.3V serial EEPROM clock (EESK) is derived from the PCI clock. The PCI 9656 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.

During serial EEPROM loading, the PCI 9656 responds to Direct Slave accesses with a Retry or allows a Master Abort (refer to Section 2.4.2); Local Processor accesses are delayed by holding TA# de-asserted.

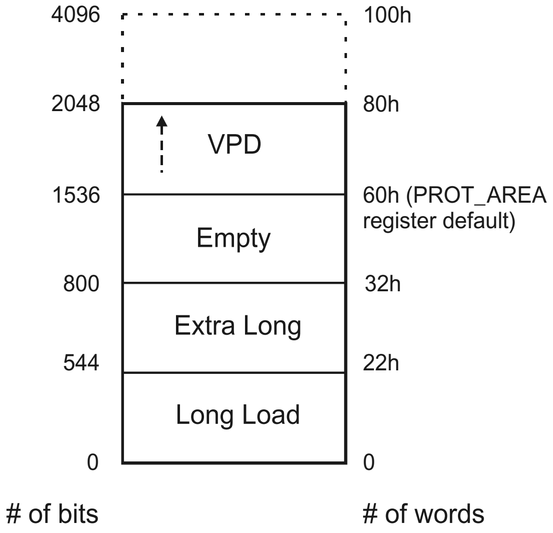


Figure 2-2. Serial EEPROM Memory Map

2.4.1 PCI 9656 Initialization from Serial EEPROM

After reset, the PCI 9656 attempts to read the serial EEPROM to determine its presence. A first-returned bit cleared to 0 indicates a serial EEPROM is present. The first word is then checked to verify that the serial EEPROM is programmed. If the first word (16 bits) is all ones (1), a blank serial EEPROM is present. If the first word (16 bits) is all zeros (0), no serial EEPROM is present. For both conditions, the PCI 9656 reverts to the default values. (Refer to Table 2-12.) The Serial EEPROM Present bit is set (CNTRL[28]=1) if the serial EEPROM is detected as present and is programmed or blank.

Table 2-12. Serial EEPROM Guidelines

Local Processor	Serial EEPROM	System Boot Condition	Device which Sets LMISC1[2]
None	None	The PCI 9656 uses default values. The EEDI/EEDO pin must be pulled low – a 1KΩ resistor is required (rather than pulled high, which is typically done for this pin). If the PCI 9656 detects all zeros (0), it reverts to default values and sets the Local Init Status bit (LMISC1[2]=1) by itself.	PCI 9656
None	Programmed	Boot with serial EEPROM values. The Local Init Status bit must be set (LMISC1[2]=1) by the serial EEPROM.	Serial EEPROM
None	Blank	The PCI 9656 detects a blank device, reverts to default values, and sets the Local Init Status bit (LMISC1[2]=1) by itself.	PCI 9656
Present	None	The Local processor programs the PCI 9656 registers, then sets the Local Init Status bit (LMISC1[2]=1). A 1KΩ or greater value pull-up resistor or EEDI/EEDO is recommended, but not required. The EEDI/EEDO pin has an internal pull-up resistor. (Refer to Table 12-2, “Pins with Internal Pull-Up Resistors.”) ***Note:*** *Some systems may avoid configuring devices that do not complete PCI Configuration accesses in less than 2^{15} PCI clocks after RST# is de-asserted. In addition, some systems may hang if Direct Slave reads and writes take too long (during initialization, the PCI Host also performs Direct Slave accesses). The value of the Direct Slave Retry Delay Clocks (LBRD0[31:28]) may resolve this.*	Local CPU
Present	Programmed	Load serial EEPROM, but the Local processor can reprogram the PCI 9656. Either the Local processor or the serial EEPROM must set the Local Init Status bit (LMISC1[2]=1).	Serial EEPROM or Local CPU
Present	Blank	The PCI 9656 detects a blank serial EEPROM and reverts to default values. ***Notes:*** *In some systems, the Local processor may be overly late to reconfigure the PCI 9656 registers before the BIOS configures them.* *The serial EEPROM can be programmed through the PCI 9656 after the system boots in this condition.*	PCI 9656

Note: *If the serial EEPROM is missing and a Local processor is present with blank Flash, the condition None/None (as listed in Table 2-12) applies, until the processor’s Flash is programmed.*

2.4.2 Local Initialization and PCI Bus Behavior

The PCI 9656 issues a Retry or allows a Master Abort to all PCI accesses until the Local Init Status bit is set (LMISC1[2]=1). This bit can be programmed three ways:

1. By the Local processor, through the Local Configuration register.
2. By the serial EEPROM, during a serial EEPROM load, if the Local processor does not set this bit.
3. If the Local processor and the serial EEPROM are missing, or the serial EEPROM is blank (with or without a Local processor), the PCI 9656 uses defaults and sets this bit. (Refer to Table 2-12.)

PCI r2.2, Section 3.5.1.1, states:

"If the target is accessed during initialization-time, it is allowed to do any of the following:

1. Ignore the request (except if it is a boot device). This results in a Master Abort.
2. Claim the access and hold in wait sates until it can complete the request, not to exceed the end of initialization-time.
3. Claim the access and terminate with [PCI] Retry."

The PCI 9656 supports Option 1 *(Initially Not Respond)*, and Option 3 *(Initially Retry)*, above. For CompactPCI Hot Swap live insertion systems, the preferred method for the silicon is not to respond to PCI Configuration accesses during initialization. For legacy systems, Retries are usually preferred for compatibility. However, it is ultimately the designer's choice which option to use.

The PCI 9656 determines the option, as follows. The USERi pin is sampled at the rising edge of RST# to determine the selected PCI Bus response mode during Local Bus initialization. If USERi is low (through an external 1KΩ pull-down resistor), the PCI 9656 does not respond to PCI activity until the device's Local Bus initialization is complete. This results in a Master Abort (the preferred method for CompactPCI Hot Swap systems). If USERi is high (through an external 1KΩ to 4.7KΩ pull-up resistor), the PCI 9656 responds to PCI accesses with PCI Retry cycles until the device's Local Bus initialization is complete. Local Bus initialization is complete when the Local Init Status bit is set (LMISC1[2]=1).

During run time, USERi can be used as a general-purpose input. (Refer to Table 12-10, "M Mode Local Bus Pins.")

Refer to Section 9, "CompactPCI Hot Swap," for specifics on using this feature in *PICMG 2.1 R2.0*-compliant systems.

2.4.2.1 Long Serial EEPROM Load

If the Extra Long Load from Serial EEPROM bit is ***not*** set (LBRD0[25]=0), the registers listed in Table 2-13 are loaded from the serial EEPROM after a reset is de-asserted. The serial EEPROM is organized in words (16 bits). The PCI 9656 first loads the Most Significant Word bits (MSW[31:16]), starting from the Most Significant Bit (MSB[31]). The PCI 9656 then loads the Least Significant Word bits (LSW[15:0]), starting again from the Most Significant Bit (MSB[15]). Therefore, the PCI 9656 loads the Device ID, Vendor ID, Class Code, and so forth.

The serial EEPROM values can be programmed using a serial EEPROM programmer. The values can also be programmed using the PCI 9656 VPD function [refer to Section 10 "PCI Vital Product Data (VPD)"] or through the Serial EEPROM Control register (CNTRL).

Using the CNTRL register or VPD, the designer can perform reads and writes from/to the serial EEPROM, 32 bits at a time. Program serial EEPROM values in the order listed in Table 2-13. The 34, 16-bit words listed in the table are stored sequentially in the serial EEPROM.

2.4.2.2 Extra Long Serial EEPROM Load

If the Extra Long Load from Serial EEPROM bit is set (LBRD0[25]=1), the registers listed in Tables 2-13 and 2-14 are loaded from the serial EEPROM, with the Table 2-13 values loaded first.

Program serial EEPROM values in the order listed in Table 2-14. The 50 16-bit words listed in Tables 2-13 and 2-14 should be stored sequentially in the serial EEPROM, with the Table 2-13 values loaded first.

Table 2-13. Long Serial EEPROM Load Registers

Serial EEPROM Byte Offset	Description	Register Bits Affected
0h	PCI Device ID	PCIIDR[31:16]
2h	PCI Vendor ID	PCIIDR[15:0]
4h	PCI Class Code	PCICCR[23:8]
6h	PCI Class Code / PCI Revision ID	PCICCR[7:0] / PCIREV[7:0]
8h	PCI Maximum Latency / PCI Minimum Grant	PCIMLR[7:0] / PCIMGR[7:0]
Ah	PCI Interrupt Pin / PCI Interrupt Line	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 Direct Slave Local Address Space 0 Range	LAS0RR[31:16]
16h	LSW of Direct Slave Local Address Space 0 Range	LAS0RR[15:0]
18h	MSW of Direct Slave Local Address Space 0 Local Base Address (Remap)	LAS0BA[31:16]
1Ah	LSW of Direct Slave Local Address Space 0 Local Base Address (Remap)	LAS0BA[15:2, 0], ***Reserved*** [1]
1Ch	MSW of Mode/DMA Arbitration	MARBR[31, 29:16] or DMAARB[31, 29:16],***Reserved*** [30]
1Eh	LSW of Mode/DMA Arbitration	MARBR[15:0] or DMAARB[15:0]
20h	Local Miscellaneous Control 2 / Serial EEPROM Write-Protected Address Boundary	LMISC2[5:0], ***Reserved*** [7:6] / PROT_AREA[6:0], ***Reserved*** [7]
22h	Local Miscellaneous Control 1 / Local Bus Big/Little Endian Descriptor	LMISC1[7:0] / BIGEND[7:0]
24h	MSW of Direct Slave Expansion ROM Range	EROMRR[31:16]
26h	LSW of Direct Slave Expansion ROM Range	EROMRR[15:11, 0], ***Reserved*** [10:1]
28h	MSW of Direct Slave Expansion ROM Local Base Address (Remap) and BREQo Control	EROMBA[31:16]
2Ah	LSW of Direct Slave Expansion ROM Local Base Address (Remap) and BREQo Control	EROMBA[15:11, 5:0], ***Reserved*** [10:6]
2Ch	MSW of Local Address Space 0/Expansion ROM Bus Region Descriptor	LBRD0[31:16]
2Eh	LSW of Local Address Space 0/Expansion ROM Bus Region Descriptor	LBRD0[15:0]
30h	MSW of Local Range for Direct Master-to-PCI	DMRR[31:16]
32h	LSW of Local Range for Direct Master-to-PCI ***(Reserved)***	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 ***(Reserved)***	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 ***(Reserved)***	DMLBAI[15:0]
3Ch	MSW of PCI Base Address (Remap) for Direct Master-to-PCI Memory	DMPBAM[31:16]
3Eh	LSW of PCI Base Address (Remap) for Direct Master-to-PCI Memory	DMPBAM[15:0]
40h	MSW of PCI Configuration Address for Direct Master-to-PCI I/O Configuration	DMCFGA[31, 23:16], ***Reserved*** [30:24]
42h	LSW of PCI Configuration Address for Direct Master-to-PCI I/O Configuration	DMCFGA[15:0]

Table 2-14. Extra Long Serial EEPROM Load Registers

Serial EEPROM Byte Offset	Description	Register Bits Affected
44h	PCI Subsystem ID	PCISID[15:0]
46h	PCI Subsystem Vendor ID	PCISVID[15:0]
48h	MSW of Direct Slave Local Address Space 1 Range (1 MB)	LAS1RR[31:16]
4Ah	LSW of Direct Slave Local Address Space 1 Range (1 MB)	LAS1RR[15:0]
4Ch	MSW of Direct Slave Local Address Space 1 Local Base Address (Remap)	LAS1BA[31:16]
4Eh	LSW of Direct Slave Local Address Space 1 Local Base Address (Remap)	LAS1BA[15:2, 0], ***Reserved*** [1]
50h	MSW of Local Address Space 1 Bus Region Descriptor	LBRD1[31:16]
52h	LSW of Local Address Space 1 Bus Region Descriptor ***(Reserved)***	LBRD1[15:0]
54h	Hot Swap Control/Status ***(Reserved)***	***Reserved***
56h	Hot Swap Next Capability Pointer / Hot Swap Control	HS_NEXT[7:0] / HS_CNTL[7:0]
58h	***Reserved***	***Reserved***
5Ah	PCI Arbiter Control	PCIARB[3:0], ***Reserved*** [15:4]
5Ch	Power Management Capabilities	PMC[15:9, 2:0]
5Eh	Power Management Next Capability Pointer ***(Reserved)*** / Power Management Capability ID ***(Reserved)***	***Reserved***
60h	Power Management Data / PMCSR Bridge Support Extension ***(Reserved)***	PMDATA[7:0] / ***Reserved***
62h	Power Management Control/Status	PMCSR[14:8]

2.4.3 Serial EEPROM Access

The CNTRL[31, 27:24] register bits are programmed to control the EESK, EECS, and EEDI/EEDO settings. Bit 24 is used to generate EESK (clock), bit 25 controls the chip select, bit 26 controls the EEDI output logic level, 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 output to go high. A pull-up resistor is required on the EEDI/EEDO pin for the pin to go high when bit 31 is set. When reading the serial EEPROM, bit 31 must be set (CNTRL[31]=1).

To perform the read, the basic approach is to set the EECS and EEDO bits (bits [25, 31]=11b, 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 the EECS, EEDI, EEDO, and EESK Input Enable bits.
2. Set EECS high.
3. Toggle EESK high, then low.
4. Set the EEDI bit high (Start bit).
5. Toggle EESK high, then low.
6. Repeat step 5.
7. Clear EEDI.
8. Toggle EESK high, then low.
9. Toggle EESK high, then low 8 times (clock in Serial EEPROM Address 0).
10. Set EEDO Input Enable to 1 (CNTRL[31]=1) to float the EEDO input for reading.
11. Toggle EESK high, then low 16 times (clock in 1 word from serial EEPROM).
12. After each clock pulse, read bit 27 and save.
13. Clear the EECS bit.
14. Toggle EESK high, then low. Read is now complete.

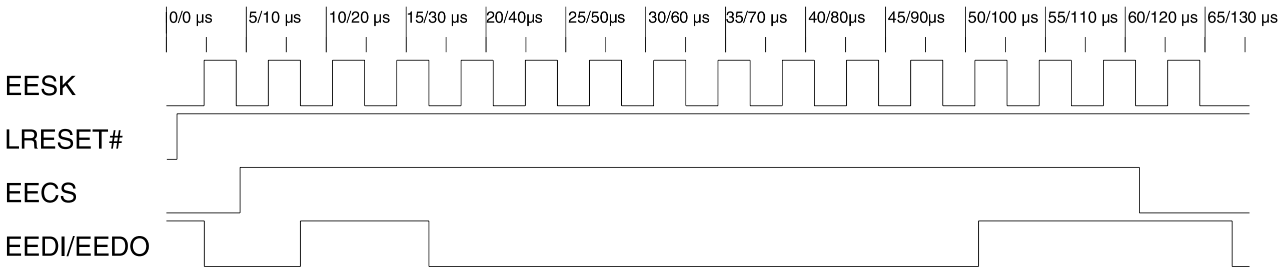
The serial EEPROM uses word addressing with 8-bit sequential address values. Due to PCI 9656 EEDI/EEDO signal multiplexing and to avoid contention between the least-significant address bit and the serial EEPROM's Start bit, Read access to an odd-numbered word address (bit 0 is set to 1) must be performed by using the even-numbered word address (bit 0 is cleared to 0), along with the serial EEPROM's sequential read functionality, clocking in 32 (or more) data bits while keeping EECS continually asserted during the Read access.

During a serial EEPROM Write operation, the serial EEPROM's programming cycle is triggered by EECS de-assertion. When EECS is re-asserted, the serial EEPROM drives its Ready status on its DO pin connected to the PCI 9656 EEDI/EEDO pin. (Refer to manufacturer's data sheet for Ready functionality.) Ready status (DO=1) indicates internal write completion. If the PCI 9656 is driving the EEDI (EEDI/EEDO pin) signal to a logic low when EECS is re-asserted after a Write operation, contention exists on the multiplexed EEDI/EEDO bus. The contention is released when Ready functionality is cleared, by clocking in a Start bit (causing the serial EEPROM to float its DO output). To remove contention following a Write cycle, re-assert EECS after waiting the Minimum CS Low Time, and, after Ready status goes high or the Write Cycle Time expires (typically 10 to 15 µs), set the EEDI output to logic high, then clock the Start bit into the serial EEPROM by toggling EESK high, then low. This Start bit can be used as part of the next serial EEPROM instruction, or, the cycle can be terminated by de-asserting EECS. Contention can also be avoided by setting the EEDI output high after clocking out the LSB of the data after each Write sequence.

The serial EEPROM can also be read or written, using the VPD function. [Refer to Section 10, "PCI Vital Product Data (VPD)."]

2.4.4 Serial EEPROM Initialization Timing Diagram

Timing Diagram 2-2. Serial EEPROM Start-Up (Serial EEPROM Absence Detected)



Notes: *The time scales provided are based on a 66 MHz/33 MHz PCI clock.*
The EEDI/EEDO signal is not sampled by the serial EEPROM until after EECS is asserted.
Upon LRESET# de-assertion, the PCI 9656 issues a Read command of Serial EEPROM Address 0.

For this timing diagram, no serial EEPROM is present; however, the Local processor is present to perform initialization. Therefore, the PCI 9656 does not detect a serial EEPROM Start bit because the EEDI/EEDO signal is sampled high (due to the internal pull-up resistor). Thereafter, the EESK, EECS and EEDI/EEDO signal pins are driven to 0.

If the PCI 9656 detects a serial EEPROM Start bit and the first 16 bits (which correspond to PCIIDR[31:16]) are not all zeros (0) or all ones (1), the PCI 9656 continues to assert EECS and generate the serial EEPROM clock (EESK) to load its registers with the programmed serial EEPROM values.
The first bit downloaded after the Start bit is PCIIDR[31].

Figure 2-3 illustrates the approximate amount of time required to detect the Start Bit and how much time is required for either a Long or Extra Long Serial EEPROM Load. While the PCI 9656 is downloading serial EEPROM data, PCI accesses to the PCI 9656 are Retried and Local Master accesses are accepted, but do not complete (TA# de-asserted) until the serial EEPROM download completes.

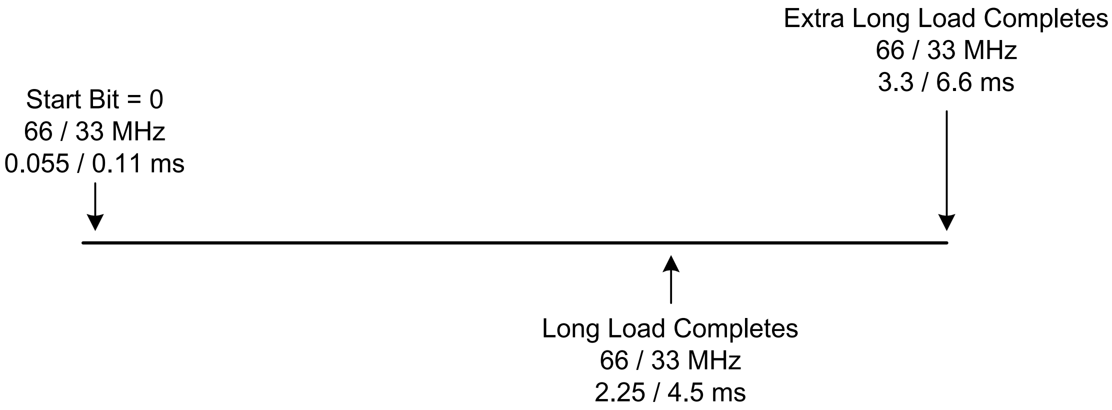


Figure 2-3. Serial EEPROM Timeline

2.5 INTERNAL REGISTER ACCESS

The PCI 9656 internal registers provide maximum flexibility in bus-interface control and performance. These registers are accessible from the PCI and Local Buses (refer to Figure 2-4) and include the following:

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

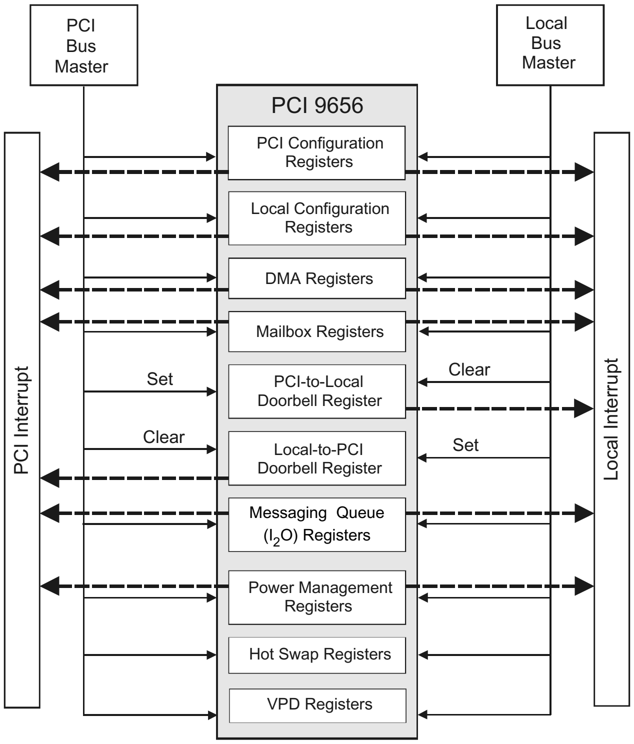

PCI
Bus
Master
Local
Bus
Master
PCI 9656
PCI Configuration Registers
Local Configuration Registers
DMA Registers
Mailbox Registers
PCI-to-Local Doorbell Register
Local-to-PCI Doorbell Register
Messaging Queue (I_2O) Registers
Power Management Registers
Hot Swap Registers
VPD Registers
Set
Clear
Clear
Set
PCI Interrupt
Local Interrupt

Figure 2-4. PCI 9656 Internal Register Access

2.5.1 PCI Bus Access to Internal Registers

The PCI 9656 PCI Configuration registers can be accessed from the PCI Bus with a Type 0 Configuration cycle.

All other PCI 9656 internal registers can be accessed by a Memory cycle, with the PCI Bus address that matches the Memory Base Address specified in the PCI Base Address for Memory Accesses to Local, Runtime, DMA, and Messaging Queue Registers register (PCIBAR0[31:9]). The PCI 9656 decodes 512 bytes for Memory-Mapped Configuration register accesses. These registers can also be accessed by an I/O cycle, with the PCI Bus address matching the I/O Base Address specified in the PCI Base Address for I/O Accesses to Local, Runtime, DMA, and Messaging Queue Registers register (PCIBAR1 [31:8]). The PCI 9656 decodes the first 256 bytes for I/O-Mapped Configuration register accesses.

All PCI Read or Write accesses to the PCI 9656 registers can be Byte, Word, or Lword accesses. All PCI Memory accesses to the PCI 9656 registers can be Burst or Non-Burst accesses. The PCI 9656 responds with a PCI disconnect for all Burst I/O accesses (PCIBAR1[31:8]) to the PCI 9656 internal registers.

2.5.1.1 New Capabilities Function Support

The New Capabilities Function Support includes PCI Power Management, Hot Swap, and VPD features, as listed in Table 2-15.

If a New Capabilities Function is not used and passed over in the Capability Pointer's linked list, the unused New Capabilities Function registers are set to default values.

Table 2-15. New Capabilities Function Support Features

New Capability Function	PCI Register Offset Location
First (Power Management)	40h, if the New Capabilities Function Support bit is enabled (PCISR[4]=1; default). (Refer to Section 8, "PCI Power Management," for details about this feature.)
Second (Hot Swap)	48h, which is pointed to from PMNEXT[7:0]. (Refer to Section 9, "CompactPCI Hot Swap," for details about this feature.)
Third (VPD)	4Ch, which is pointed to from HS_NEXT[7:0]. Because PVPD_NEXT[7:0] defaults to 0h, this indicates that VPD is the last New Capability Function Support feature of the PCI 9656. [Refer to Section 10, "PCI Vital Product Data (VPD)," for details about this feature.]

2.5.2 Local Bus Access to Internal Registers

The Local processor can access all PCI 9656 internal registers through an external chip select. The PCI 9656 responds to a Local Bus access when the PCI 9656 Configuration Chip Select input (CCS#) is asserted low during the Address phase.

Local Read or Write accesses to the PCI 9656 internal registers can be Byte, Word, or Lword accesses. Local accesses to the PCI 9656 internal registers can be Burst or Non-Burst accesses.

The PCI 9656 TA# signal indicates that the Data transfer is complete.

THIS PAGE INTENTIONALLY LEFT BLANK.

3 M MODE FUNCTIONAL DESCRIPTION

The functional operation described in this section can be modified through the PCI 9656 programmable internal registers.

3.1 RESET OPERATION

3.1.1 Adapter Mode

The PCI 9656 operates in Adapter mode when the HOSTEN# pin is strapped high.

3.1.1.1 PCI Bus RST# Input

The PCI Bus RST# input pin is a PCI Host reset. It causes all PCI Bus outputs to float, resets the entire PCI 9656 and causes Local LRESET# to assert. LEDon# is asserted during PCI Bus RST# assertion for CompactPCI Hot Swap systems. Most Local Bus output signals are set to float while LRESET# remains asserted, with the exceptions listed in Table 3-1.

Table 3-1. Local Bus Output Signals that Do Not Float when RST# Is Asserted

Signal	Value when RST# Is Asserted
EECS	0
EESK	0
LEDon#	0
LRESET#	0
PME#	X (undefined)
USERo	X (tri-stated)

Note: *When HOSTEN#=1 and during a hard reset (RST# = 0) USERo is tri-stated. On the second rising edge of LCLK after RST# is de-asserted, LRESET# de-asserts. On the next falling edge of LCLK, the USERo pin is driven to 0 (from tri-state). On the next rising edge of LCLK, USERo is driven to the value specified in the General-Purpose Output bit (CNTRL[16]=1, default).*

3.1.1.2 JTAG Reset TRST# Input

The TRST# input pin is the asynchronous JTAG logic reset. TRST# assertion causes the PCI 9656 Test Access Port (TAP) controller to initialize. In addition, when the TAP controller is initialized, it selects the PCI 9656 normal logic path (core-to-I/O).

It is recommended that the designer take the following into consideration when implementing the asynchronous JTAG logic reset on a board:

- If JTAG functionality is required, the TRST# input signal should use a low-to-high transition once during the PCI 9656 boot-up, along with the RST# signal.
- If JTAG functionality is not required, the TRST# signal should always be directly connected to ground.

3.1.1.3 Software Reset

When the Software Reset bit is set (CNTRL[30]=1), the following functional blocks are reset:

- All Local Bus logic
- Local Configuration and Messaging Queue registers
- PCI and Local Bus DMA logic
- FIFOs

The PCI Bus logic, PCI Configuration DMA and Runtime registers, and the Local Init Status bit (LMISC1[2]) are ***not*** reset.

When the Software Reset bit is set (CNTRL[30]=1), the PCI 9656 responds to PCI Configuration, Runtime, and DMA registers' accesses, initiated from the PCI Bus. Because the Local Bus is in reset, accesses by the Local Bus are ***not*** allowed. The PCI 9656 remains in this reset condition until the PCI Host clears the bit (CNTRL[30]=0). 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, although the Local processor does not use LRESET# to reset.*

3.1.1.4 Power Management Reset

When the Power Management reset is asserted (transition from D_3 to any other power state), the PCI 9656 resets as if a PCI reset was asserted. (Refer to Section 8, “PCI Power Management.”)

3.1.2 Host Mode

The PCI 9656 operates in Host mode when the HOSTEN# pin is strapped low.

3.1.2.1 Local Reset

The PCI Bus RST# output is driven when Local LRESET# is asserted or the Software Reset bit is set (CNTRL[30]=1). Most PCI and Local Bus output signals are set to float, with the exceptions listed in Table 3-2.

Table 3-2. PCI and Local Bus Output Signals that Do Not Float when LRESET# Is Asserted

Signal	Value when LRESET# Is Asserted
AD[63:0]	0
C/BE[7:0]#	0
EECS	0
EESK	0
LEDon#	0
PAR	0
PAR64	0
PME#	X (undefined)
REQ64#	0
RST#	0

3.1.2.2 Software Reset

When the Software Reset bit is set (CNTRL[30]=1), the following functional blocks are reset:

- PCI Master logic
- PCI Slave logic
- PCI and Local Bus DMA logic
- PCI 9656 PCI Configuration, Mailbox, and DMA registers
- FIFOs
- PCI interrupts/lock
- Internal PCI Arbiter
- Power Management interrupts

The Local Bus logic, Local Configuration, Runtime, and Messaging Queue registers are ***not*** reset. A software reset can be cleared only from a host on the Local Bus, and the PCI 9656 remains in this reset condition until a Local host clears the bit (CNTRL[30]=0). When the Software Reset bit is set (CNTRL[30]=1), the PCI 9656 responds to the Local Configuration, Runtime, and DMA registers’ accesses initiated from the Local Bus.

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

When HOSTEN#=1 and the Soft Reset bit is set (CNTRL[30]=1), the LRESET# and USERo signals are asserted low (0). When the Soft Reset bit is cleared (CNTRL[30]=0), USERo reverts to the value specified in the General-Purpose Output bit (CNTRL[16]), one LCLK after the LRST# signal is de-asserted (driven to 1).

3.1.2.3 Power Management Reset

Power Management reset is ***not*** applicable for Host mode.

3.2 PCI 9656 INITIALIZATION

The PCI 9656 Configuration registers can be programmed by an optional serial EEPROM and/or by a Local processor, as listed in Table 2-12, “Serial EEPROM Guidelines.” The serial EEPROM can be reloaded by setting the Reload Configuration Registers bit (CNTRL[29]=1).

The PCI 9656 Retries all PCI cycles or allows Master Aborts 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 3-3 lists the PCI 9656 response to full and empty FIFOs.

Table 3-3. Response to FIFO Full or Empty

Mode	Direction	FIFO	PCI Bus	Local Bus
Direct Master Write	Local-to-PCI	Full	Normal	De-asserts TA#, RETRY#[1]
		Empty	De-asserts REQ# (releases the PCI Bus)	Normal[5]
Direct Master Read	PCI-to-Local	Full	De-asserts REQ# or throttles IRDY#[2]	Normal[5]
		Empty	Normal	De-asserts TA#
Direct Slave Write	PCI-to-Local	Full	Disconnects or throttles TRDY#[3]	Normal[5]
		Empty	Normal	Retains the Local Bus
Direct Slave Read	Local-to-PCI	Full	Normal	Retains the Local Bus
		Empty	Throttles TRDY#[4]	Normal[5]
DMA	Local-to-PCI	Full	Normal	Normal[5]
		Empty	De-asserts REQ#	Normal[5]
	PCI-to-Local	Full	De-asserts REQ#	Normal[5]
		Empty	Normal	Normal[5]

[1] *RETRY# assertion depends upon the Direct Master Write FIFO Almost Full RETRY# Output Enable bit being set (LMISC1[6]=1).*

[2] *Throttle IRDY# depends upon the Direct Master PCI Read Mode bit being set (DMPBAM[4]=1).*

[3] *Throttle TRDY# depends upon the Direct Slave PCI Write Mode bit being set (LBRD0[27]=1).*

[4] *If PCI Compliance is enabled (MARBR[24]=1), PCI Retry is issued instead of throttling TRDY#.*

[5] *Release Local Bus ownership per MPC850/MPC860 protocol.*

3.4 DIRECT DATA TRANSFER MODES

In M mode, the PCI 9656 supports three Direct Data Transfer modes:

- **Direct Master** – Local CPU accesses PCI memory or I/O, including support for MPC850 and MPC860 Independent (IDMA) and Serial (SDMA) Operation
- **Direct Slave** – PCI Master accesses Local memory or I/O
- **DMA** – PCI 9656 DMA Controller(s) reads/writes PCI memory to/from Local memory

All three Direct Data Transfer modes generate dummy cycles. The PCI 9656 generates dummy cycles on the PCI and Local Buses. The PCI Bus C/BE[7:0]# pins and the Local Bus TSIZ[0:1] pins must be monitored so that dummy cycles can be recognized on either bus.

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

The PCI 9656 supports direct access of the PCI Bus by the Local processor or an intelligent controller. Master mode must be enabled in the PCI Command register (PCICR[2]=1). The following registers define Local-to-PCI accesses (refer to Figure 3-1):

- 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)

Important Note: The PCI 9656 generates dummy cycles on the PCI Bus. (Refer to Sections 3.4 and 3.4.1.3.1.)

3.4.1.1 Direct Master Memory and I/O Decode

The Range register and the Local Base Address specify the Local Address bits to use for decoding a Local-to-PCI access (Direct Master). The Memory or I/O space range must be a power of 2 and the Range register value must be two’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 9656. All Direct Master cycles are then decoded as PCI Memory, I/O, or Type 0 or Type 1 Configuration. Moreover, a Direct Master Memory or I/O cycle is remapped according to the Remap register value, which must be a multiple of the Direct Master range value (***not*** the Range register value).

The PCI 9656 can accept Memory cycles only from a Local processor. The Local Base Address and range determine whether PCI Memory or PCI I/O transactions occur.

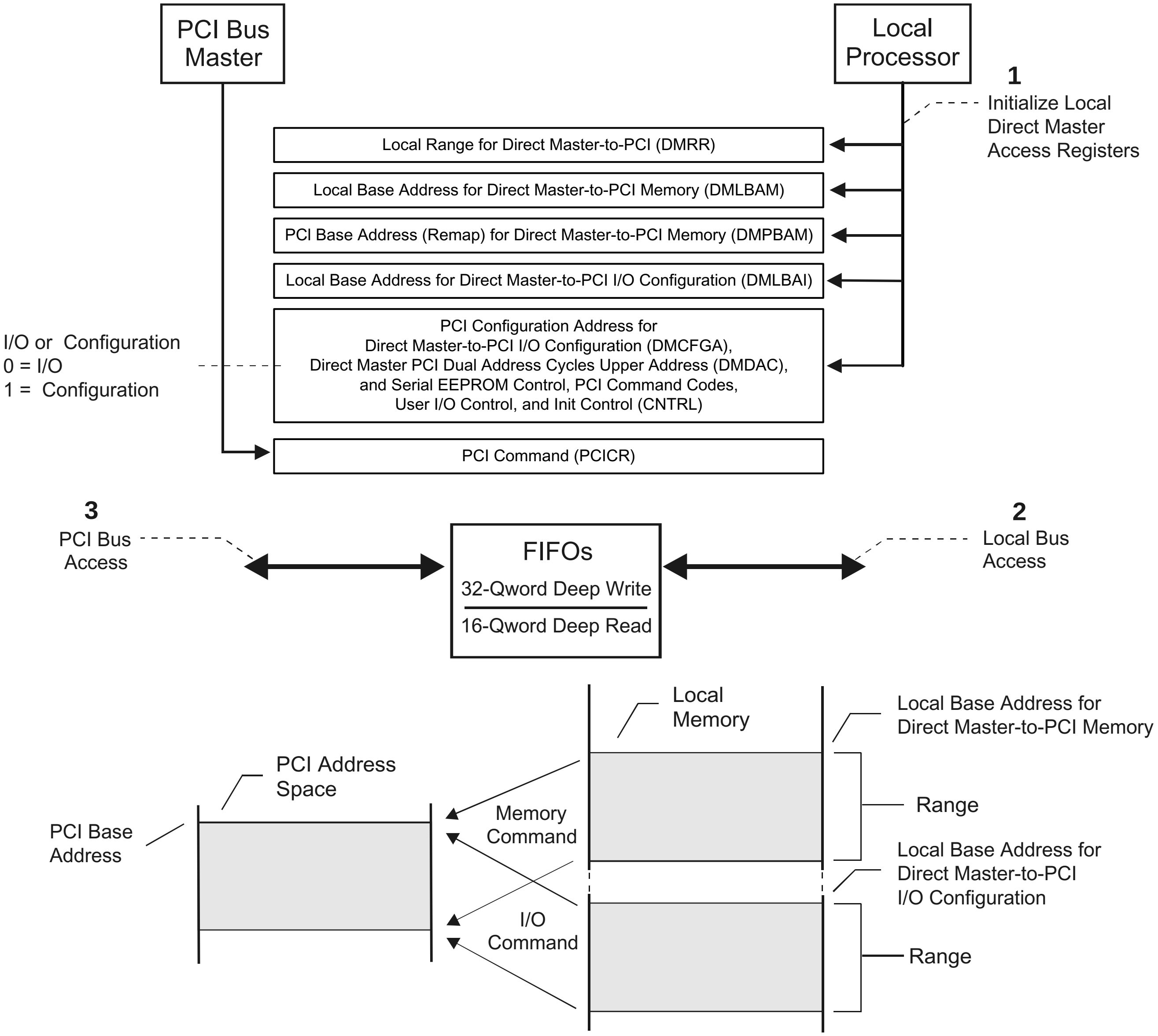

PCI Bus Master
Local Processor
1
Initialize Local Direct Master Access Registers
Local Range for Direct Master-to-PCI (DMRR)
Local Base Address for Direct Master-to-PCI Memory (DMLBAM)
PCI Base Address (Remap) for Direct Master-to-PCI Memory (DMPBAM)
Local Base Address for Direct Master-to-PCI I/O Configuration (DMLBAI)
PCI Configuration Address for Direct Master-to-PCI I/O Configuration (DMCFGA), Direct Master PCI Dual Address Cycles Upper Address (DMDAC), and Serial EEPROM Control, PCI Command Codes, User I/O Control, and Init Control (CNTRL)
I/O or Configuration
0 = I/O
1 = Configuration
PCI Command (PCICR)
3
PCI Bus Access
2
Local Bus Access
FIFOs
32-Qword Deep Write
16-Qword Deep Read
Local Memory
Local Base Address for Direct Master-to-PCI Memory
PCI Address Space
PCI Base Address
Memory Command
Range
Local Base Address for Direct Master-to-PCI I/O Configuration
I/O Command
Range

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

3.4.1.2 Direct Master FIFOs

For Direct Master Memory access to the PCI Bus, the PCI 9656 has a 64-Lword/32-Qword (256-byte) Write FIFO and a 32-Lword/16-Qword (128-byte) Read FIFO. The FIFOs enable the Local Bus to operate independently 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 Figures 3-2 and 3-3.

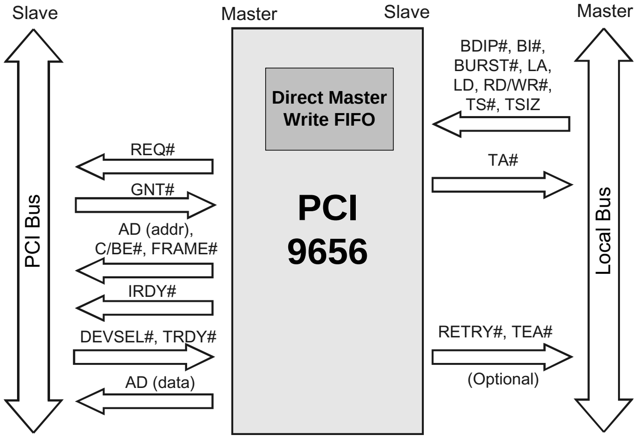


Figure 3-2. Direct Master Write

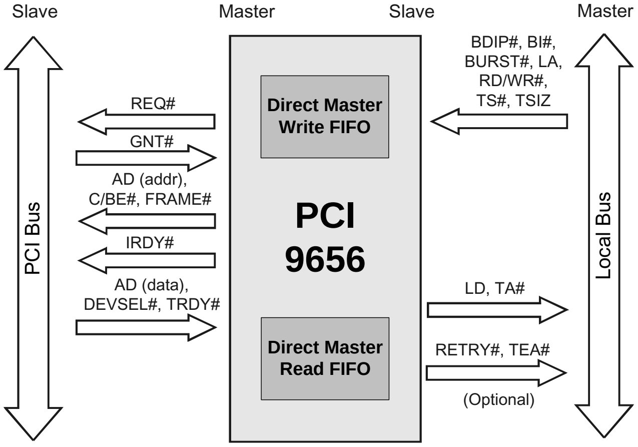


Figure 3-3. Direct Master Read

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 9656 and PCI Bus. The MPC850 or MPC860 IDMA/SDMA accesses to the PCI 9656 appear as Direct Master operations. (Refer to Section 3.4.1.13 for further details about IDMA/SDMA accesses.)

The PCI 9656 converts the Local Read/Write access to PCI Bus accesses. The Local Address space starts from the Direct Master Local Base Address up to the range. Remap (PCI Base Address) defines the PCI starting address.

An MPC850 or MPC860 single cycle causes a single cycle PCI transaction. An MPC850 or MPC860 Burst cycle causes a PCI burst transaction. Local bursts are limited to 16 bytes (4 Lwords) in the MPC850 or MPC860 bus protocol.

The PCI 9656 supports bursts beyond the 16-byte boundary (PLX Continuous Burst mode) when BDIP# input remains asserted beyond a 16-byte boundary by an external Local Bus Master. To finish the continuous burst, the external Master should de-assert BDIP# in the last Data phase.

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.

3.4.1.3.1 Direct Master Writes

For 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 9656 becomes the PCI Bus Master, arbitrates for the PCI Bus, and writes data to the PCI Slave device. The PCI 9656 continues to accept writes and returns TA# until the Direct Master Write FIFO is full. It then holds TA# de-asserted until space becomes available in the Direct Master Write FIFO. A programmable Direct Master “almost full” status output is provided (DMPAF signal). (Refer to DMPBAM[10, 8:5].) The PCI 9656 asserts RETRY# when the Direct Master Write FIFO is full, implying that the Local Master can relinquish Local Bus ownership and finish the Write operation at a later time (LMISC1[6]).

MPC850 or MPC860 single cycle Write transactions result in PCI 9656 transfers of 1 Lword of data onto a 32-bit PCI Bus. A Qword-aligned single cycle write to a 64-bit PCI Bus results in a 64-bit PCI Write cycle, with PCI Byte Enables (C/BE[7:0]#) asserted as F0h. If the single cycle write is ***not*** Qword-aligned (*for example*, X4h or XCh), the PCI Bus Write cycle is 32 bits (REQ64# de-asserted).

MPC850 or MPC860 Burst cycle Write transactions of 4 Lwords result in PCI 9656 Burst transfers of 4 Lwords to a 32-bit PCI Bus. The same type of transfer results in PCI 9656 Burst transfers of 2 Qwords with all PCI Byte Enables (C/BE[7:0]#=00h) asserted onto the 64-bit PCI Bus.

A Local processor (MPC850 or MPC860), with no burst limitations and a Burst cycle Write transaction of 2 Lwords, causes the PCI 9656 to perform two single writes to a 32-bit PCI Bus. The same Local transfer to a 64-bit PCI Bus results in the PCI 9656 transferring 1 Qword, with all PCI Byte Enables (C/BE[7:0]#=00h) asserted. Three (or more) Lword Burst cycles of any type result in the PCI 9656 bursting data onto the PCI Bus.

Direct Master writes on the Local Bus may generate Dummy Cycle writes on the PCI Bus (dummy cycles are C/BE[7:0]#=FFh for the 64-bit PCI Bus, and C/BE[7:0]#=XFh for the 32-bit PCI Bus).

- **64-bit PCI Bus** – No dummy cycles, although one-half of the Qword on the PCI Bus may have its C/BE[3:0]#=Fh or C/BE[7:4]#=Fh.
- **32-bit PCI Bus, with a Local Bus burst greater than two** – If the Local starting address is Qword-aligned, and the burst count is odd, the last PCI Write cycle is a dummy cycle, and the PCI starting address is Qword-aligned.

  If the Local starting address is Lword-aligned, but not Qword-aligned, and the burst count is even, the first and last PCI Write cycles are dummy cycles and the PCI starting address is Qword-aligned.

  If the Local starting address is Lword-aligned, but not Qword-aligned, and the burst count is odd, the first PCI Write cycle is a dummy cycle and the PCI starting address is Qword-aligned.

Other 32-bit PCI Bus cases do ***not*** generate dummy cycles.

3.4.1.3.2 Direct Master Reads

For a Local Bus read, the PCI 9656 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. The PCI 9656 asserts Local Bus TA# to indicate that the requested data is on the Local Bus.

The PCI 9656 holds TA# de-asserted while receiving data from the PCI Bus. Programmable Prefetch modes are available if prefetch is enabled – 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 9656 does not prefetch PCI data for single cycle (Local BURST# input is not asserted during the first Data phase) Direct Master reads unless Read Ahead mode is enabled and prefetch length is set to continuous (DMPBAM[2, 11]=10b, respectively). (Refer to Section 3.4.1.7 for details on Read Ahead mode.) For single cycle Direct Master reads of a 32-bit PCI Bus (Read Ahead mode disabled, DMPBAM[2]=0), the PCI 9656 reads 1 Lword from the PCI Bus. For single cycle Direct Master reads of a 64-bit PCI Bus (Read Ahead mode disabled) starting at a Qword boundary, the PCI 9656 reads 1 Qword from the PCI Bus, with C/BE[7:0]#=F0h. If the starting address is not a Qword boundary (*for example*, X4h or XCh), the PCI 9656 performs a 32-bit PCI read (REQ64# de-asserted).

For single cycle Direct Master reads, the PCI 9656 derives the PCI Byte Enables from the Local Bus address and TSIZ[0:1] signal.

For Burst Cycle reads, the PCI 9656 reads entire Lwords or Qwords (all PCI Byte Enables are asserted), independent of the PCI Bus data width.

If the Direct Master Prefetch Limit bit is enabled (DMPBAM[11]=1), the PCI 9656 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 (DMPBAM[11]=0), the prefetch crosses the 4-KB boundaries.

If the 4-KB Prefetch Limit bit is enabled and the PCI 9656 has started a Direct Master read to a 64-bit PCI Bus, PCI Address FF8h (Qword-aligned, 1 Qword before the 4-KB boundary), without ACK64# acknowledgment from the PCI Slave, the PCI 9656 does ***not*** perform a Burst prefetch of 2 Lwords. The PCI 9656 instead performs a prefetch of two single cycle Lwords to prevent crossing the PCI 4-KB boundary limit. If the PCI Slave responds with ACK64#, the PCI 9656 performs a single cycle read of one Qword and terminates to prevent crossing a 4-KB limit boundary. The cycle then restarts at the new boundary.

3.4.1.4 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 Configuration Enable bit is set (DMCFGA[31]=1), the PCI cycle becomes a PCI Configuration cycle. (Refer to Section 3.4.1.9.1 and the DMCFGA register for details.)

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. When the I/O Remap Select is cleared (DMPBAM[13]=0), DMPBAM[31:16] are used as the remap addresses for the PCI address.

For writes, data is loaded into the Write FIFO and TA# is returned to the Local Bus. For reads, the PCI 9656 holds TA# de-asserted while receiving an Lword from the PCI Bus.

Local Burst accesses are broken into single PCI I/O (Address/Data) cycles. The PCI 9656 does ***not*** prefetch Read data for I/O and Configuration reads.

For Direct Master I/O and Configuration cycles, the PCI 9656 derives the PCI Byte Enables from the Local Bus address and TSIZ[0:1] signal.

3.4.1.5 Direct Master Delayed Read Mode

The PCI 9656 supports Direct Master Delayed Read transactions. When the Direct Master Delayed Read Enable bit is set (LMISC1[4]=1), the PCI 9656 asserts RETRY# and prefetches Read data each time the Local Master requests a read. During a PCI data prefetch, the Local Master is capable of performing other transactions and free to return for requested data at a later time. In this mode, it is required that a Local processor repeat the Retried transaction exactly as the original request, including TSIZ[0:1] assertion, and read at least one data. Otherwise, the PCI 9656 indefinitely Retries the Local Bus Master. When Direct Master Delayed Read mode is disabled (LMISC1[4]=0), the Local Master must retain the Local Bus and wait for the requested data (TA# remains de-asserted until data is available to the Local Bus).

3.4.1.6 Direct Master Delayed Write Mode

The PCI 9656 supports Direct Master Delayed Write mode transactions, in which posted Write data accumulates in the Direct Master Write FIFO before the PCI 9656 requests the PCI Bus. Direct Master Delayed Write mode is programmable to delay REQ# assertion for the amount of PCI clocks selected 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 9656 only uses the Delay Counter and accumulates data in the Direct Master Write FIFO for Burst transactions on the Local Bus. A single cycle write on the Local Bus immediately starts a PCI cycle (ignores delay setting).

3.4.1.7 Direct Master Read Ahead Mode

The PCI 9656 only supports Direct Master Read Ahead mode (DMPBAM[2]=1), when the Direct Master Read Prefetch Size Control bits are set to continuous prefetch (DMPBAM[12, 3]=00b). Other Direct Master Read Prefetch Size Control bit settings force Direct Master Read Ahead mode off. Direct Master Read Ahead mode allows prefetched data to be read from the PCI 9656 internal FIFO instead of the PCI Bus. The address must be subsequent to the previous address and 32-bit-aligned (next address = current address + 4). Direct Master Read Ahead mode functions can be used with or without Direct Master Delayed Read mode.

A Local Bus single cycle Direct Master transaction, with Direct Master Read Ahead mode enabled results in the PCI 9656 processing continuous PCI Bus Read Burst data with all bytes enabled (C/BE[7:0]#=00h), if the Direct Master Read Prefetch Size Control bits are set to continuous prefetch (DMPBAM[12, 3]=00b). (Refer to Table 3-4.)

The Direct Master Read Ahead stops on the PCI Bus after one of the following occurs:

- Local Bus completes reading data
- Prefetch count has been reached

Caution: *To prevent constant locking of the PCI Bus, do **not** simultaneously enable Direct Master Read Ahead and Direct Master PCI Read modes (DMPBAM[2, 4]≠11b, respectively).*

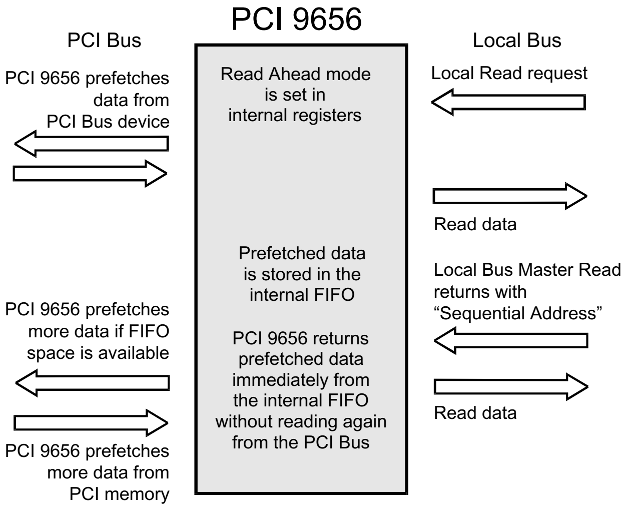

PCI 9656
PCI Bus
Local Bus
PCI 9656 prefetches data from PCI Bus device
Read Ahead mode is set in internal registers
Local Read request
Read data
Prefetched data is stored in the internal FIFO
Local Bus Master Read returns with "Sequential Address"
PCI 9656 prefetches more data if FIFO space is available
PCI 9656 returns prefetched data immediately from the internal FIFO without reading again from the PCI Bus
Read data
PCI 9656 prefetches more data from PCI memory

Figure 3-4. Direct Master Read Ahead Mode

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

Table 3-4. Example of PCI Bus Behavior with Direct Master Read Ahead Mode Enabled

Direct Master Local Bus Cycle to the PCI 9656	Direct Master PCI Bus Behavior with Direct Master Read Prefetch Size Control Bits Not Set to Continuous Prefetch (DMPBAM[12, 3]≠00b)	Direct Master PCI Bus Behavior with Direct Master Read Prefetch Size Control Bits Set to Continuous Prefetch (DMPBAM[12, 3]=00b)
Single cycle read followed by single cycle read	Two single cycle reads	Burst cycle with Read Ahead
Single cycle read followed by consecutive address Burst Read cycle	One single cycle followed by Burst cycle	Burst cycle followed by restart of Burst cycle with Read Ahead
Burst cycle read of four data followed by consecutive address Burst cycle read of four data	Burst cycle read followed by restart of Burst cycle read	Burst read followed by a second burst read with a contiguous address
Burst cycle read of four data followed by consecutive address single cycle read	Burst cycle read followed by restart of a single cycle read	Burst read followed by a second burst read with a contiguous address

3.4.1.8 RETRY# Capability

The PCI 9656 supports a Direct Master Write FIFO full condition in one of two ways:

- When FIFO Almost Full Retry is enabled (LMISC1[6]=1), the PCI 9656 asserts Local RETRY# to force the Local Master to stop its Write cycle. The Local Master can later return and complete the write.
- If FIFO Almost Full Retry is disabled (LMISC1[6]=0), the PCI 9656 de-asserts TA# (and does not assert RETRY#) until there is space in the FIFO for additional Write data.

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

If the Configuration Enable bit is set (DMCFGA[31]=1), a Configuration access is made to the PCI Bus. In addition to enabling this bit, the user must provide all register information. (Refer to the DMCFGA register.) The Register Number and Device Number bits (DMCFGA[7:2, 15:11], respectively) must be modified and a new Direct Master Read/Write cycle must be performed at the Direct Master I/O base address for each Configuration register. Before performing Non-Configuration cycles, the Configuration Enable bit must be cleared (DMCFGA[31]=0).

If the PCI Configuration Address register selects a Type 0 command, DMCFGA[10:0] are copied to address bits [10:0]. DMCFGA[15:11] (Device Number) are translated into a single bit being set in PCI Address bits [31:11]. The PCI Address bits [31:11] can be used as a device select. For a Type 1 command, DMCFGA[23:0] are copied to PCI address bits [23:0]. The PCI Address bits [31:24] are cleared 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 9656 holds TA# de-asserted while receiving an Lword from the PCI Bus.

3.4.1.9.1 Direct Master Configuration Cycle Example

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

1. The PCI 9656 must be configured to allow Direct Master access to the PCI Bus. The PCI 9656 must also be enabled to master the PCI Bus (PCICR[2]=1).

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

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 two's complement of 00100000h:

   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 user must know which PCI device and PCI Configuration register the PCI Configuration cycle is accessing. This example assumes the Target PCI device IDSEL signal is connected to AD[21] (logical device #10=0Ah). Also, access PCIBAR0 (the fourth register, counting from 0; use Table 11-2, "PCI Configuration Register Address Mapping," for reference). Set DMCFGA[31, 23:0], as listed in Table 3-5.

After these registers are configured, a single Local Master Memory cycle to the I/O base address is necessary to generate a PCI Configuration Read or Write cycle. The PCI 9656 takes the Local Bus Master Memory cycle and checks for the Configuration Enable bit (DMCFGA[31]). If set to 1, the PCI 9656 converts the current cycle to a PCI Configuration cycle, using the DMCFGA register and the Read/Write signal (RD/WR#).

Table 3-5. Direct Master Configuration Cycle Example – DMCFGA[31, 23:0] Settings

Bits	Description	Value
1:0	Type 0 Configuration.	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.	0h
31	Configuration Enable.	1

3.4.1.10 Direct Master PCI Dual Address Cycles

The PCI 9656 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 64-bit address to devices that support 64-bit addressing when the address is not in the lower 4-GB address space. The PCI 9656 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]#) “Dh” and the second PCI address will be a Hi-Addr with the command (C/BE[3:0]#) “6h” or “7h”, depending upon it being a PCI Read or a PCI Write cycle. (Refer to Figure 3-5.) When the DMDAC register contains a value of 0h (feature is enabled when DMDAC register value is ***not*** 0h), the PCI 9656 performs a Single Address Cycle (SAC) on the PCI Bus.

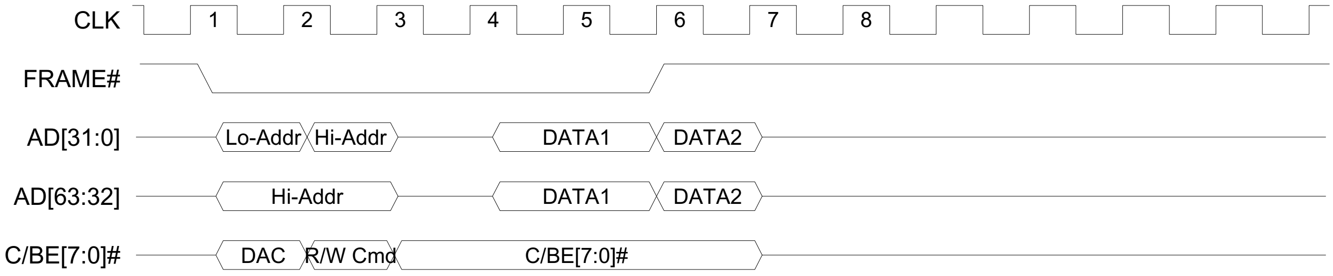


Figure 3-5. PCI Dual Address Cycle Timing

3.4.1.11 PCI Master/Target Abort

The following describes how the PCI 9656 logic handles a PCI Master/Target Abort.

As a PCI master, if the PCI 9656 attempts an 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 (PCISR[13]=0) and continue by processing the next task.

If a PCI Master/Target Abort or 256 consecutive Master Retry timeout is encountered during a transfer, the PCI 9656 asserts TEA# [which can be used as a Non-Maskable Interrupt (NMI)]. (Refer to Section 6.1.12 for specific TEA# behavior.) The PCI 9656 also asserts LINTo#, if enabled (INTCSR [16, 12]=11b), which can be used as an interrupt.

If a Direct Master write is posted when a PCI Master/ Target Abort occurs, the PCI 9656 asserts TEA# on the following Direct Master cycle. If a PCI Master/ Target Abort occurs during a Direct Master read, the PCI 9656 asserts TEA# during the Direct Master read or upon the following Direct Master access. For Direct Master reads, TA# is asserted during PCI Master/Target Aborts if TEA# assertion is disabled (INTCSR[0]=0). The Local Bus Master's interrupt handler can take the appropriate application-specific action. It can then clear the Received Master or Target Abort bit (PCISR[13 or 12]=1, respectively; writing 1 clears the bit to 0) to clear the Master or Target Abort and re-enable Direct Master transfers.

If a PCI Master/Target Abort or Retry Timeout is encountered during a Direct Master transfer with multiple Direct Master operations posted in the Direct Master Write FIFO (*for example*, an address, followed by data, followed by another address, followed by data), the PCI 9656 flushes the entire Direct Master Write FIFO if Direct Master Write FIFO Flush during PCI Master Abort is disabled (LMISC1[3]=0). If the bit is enabled (LMISC1[3]=1), the PCI 9656 flushes only the aborted Direct Master operation in the Direct Master Write FIFO and skips to the next available address in the FIFO entry. The PCI 9656 does not arbitrate on the PCI Bus until the Received Master/Target Abort bits are cleared (PCISR[13:12]=00b, respectively).

If a Local Bus master is attempting a Burst read from a non-responding PCI device (PCI Master/Target Abort), the PCI 9656 asserts TEA# (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 9656 stores the PCI Abort Address into PABTADR[31:0] and sets the Received Master/Target Abort bits (PCISR [13:12]=11b, respectively). The PCI 9656 disables all PCI Master capabilities when PCISR[13:12]=11b. Thus, in this condition, the PCI 9656 does not arbitrate for the PCI Bus to execute new or pending Direct Master or DMA transfers. The Received Master/ Target Abort bits should be read and then cleared before starting new Direct Master or DMA transfers. The PCI Abort Address register bits (PABTADR[31:0]) contents are updated for each PCI Master/Target Abort. (For details about DMA PCI Master/Target Abort, refer to Section 3.4.4.17.)

The PCI 9656 PCI Master/Target Abort logic can be used by 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).

3.4.1.12 Direct Master Memory Write and Invalidate

The PCI 9656 can be programmed to perform Memory Write and Invalidate (MWI) cycles to the PCI Bus for Direct Master transfers, as well as DMA transfers. (Refer to Section 3.4.4.4.) The PCI 9656 supports MWI 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 9656 performs Write transfers rather than MWI transfers.

Direct Master MWI transfers are enabled when the MWI Mode and the MWI Enable bits are set (DMPBAM[9]=1 and PCICR[4]=1, respectively).

In MWI mode, if the start address of the Direct Master transfer is on a cache line boundary, the PCI 9656 waits until the number of Lwords required for the specified cache line size are written from the Local Bus before starting a PCI MWI 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 9656 starts a PCI Write access using the PCI command code from the CNTRL[15:12] register bits and the PCI 9656 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 9656 completes the remainder of that cache line, using normal writes.

If the Direct Master write is less than the cache line size, the PCI 9656 waits until the next Direct Master write begins before starting the PCI write. This PCI write retains the MWI command, regardless of the number of Lwords in the Direct Master Write FIFO, which may result in the PCI Bus completing an MWI cycle with less than a cache line worth of data. For this reason, Burst writes should be equal to, or multiples of, the cache line size. Because the MPC850 or MPC860 is limited to bursts of 4 Lwords, MWI should be used only with DMA (DMA does not have this issue) or with Direct Master cycles in Continuous Burst mode.

3.4.1.13 IDMA/SDMA Operation

3.4.1.13.1 IDMA Operation

The PCI 9656 supports the MPC850 or MPC860 Independent DMA (IDMA) mode, using the MDREQ# signal and operating in Direct Master mode. MDREQ# is connected to the MPC850 or MPC860 DREQ*x*# input pins. After programming the MPC850 or MPC860 IDMA channel, the PCI 9656 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 9656 MDREQ# signal. When the PCI 9656 Direct Master Write 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 9656 must be set to a value of at least 5 Qwords below the full capacity of the FIFO – 27 Qwords to prevent FIFO overflow (DMPBAM[10, 8:5]). The Retry function can be used to communicate to the Local Bus Master that it should relinquish Local Bus ownership.

Note: *The Direct Master Write FIFO Almost Full RETRY# Output Enable bit can be disabled (LMISC1[6]=0) to prevent RETRY# assertion.*

In IDMA reads (PCI 9656 to the Local Bus), MDREQ# is asserted (indicating data is available), although the Read FIFO is empty. Any Local Bus read of the PCI Bus causes the PCI 9656 to become a PCI Bus Master and fills the Direct Master Read FIFO buffer. When sufficient data is in the FIFO, the PCI 9656 completes the Local Bus cycle by asserting Transfer Acknowledge (TA#).

In IDMA writes (Local-to-PCI Bus), MDREQ# is asserted (indicating that the Direct Master FIFO is capable of accepting additional data). After the IDMA has transferred all required bytes [MPC850 or MPC860 Byte counter decrements to zero (0)], the MPC850 or MPC860 generates an internal interrupt, which in turn should execute the code to disable the IDMA channel (the PCI 9656 may still assert MDREQ# output). The PCI 9656 does ***not*** use the MPC850 or MPC860 SDACK[1:0]# signals (no connection).

Refer to Section 3.4.1 for further details about Direct Master Data transfers.

3.4.1.13.2 SDMA Operation

The PCI 9656 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 Local Bus ownership. The Direct Master Write FIFO Almost Full RETRY# Output can be disabled (LMISC1[6]=0) to prevent RETRY# assertion.

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

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

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

For Direct Slave reads, the PCI Bus Master reads data from the Local Bus Slave. Direct Slave or Direct Master preempts DMA; however, Direct Slave does ***not*** preempt Direct Master. (Refer to Section 3.4.2.10.)

The PCI 9656 supports Burst Memory-Mapped Transfer accesses and I/O-Mapped, Single-Transfer PCI-to-Local Bus accesses through a 32-Lword/16-Qword (128-byte) Direct Slave Read FIFO and a 64-Lword/32-Qword (256-byte) Direct Slave Write FIFO. The PCI Base Address registers are provided to set the adapter location in PCI Memory and I/O space. In addition, Local mapping registers allow address translation from the PCI Address Space to the Local Address Space. Three Local Address Spaces are available:

- Space 0
- Space 1
- Expansion ROM (Expansion ROM is intended to support a bootable Host ROM device)

Direct Slave supports on-the-fly Endian conversion for Space 0, Space 1, and Expansion ROM space.

Each Local space can be programmed to operate with an 8-, 16-, or 32-bit Local Bus data width. The PCI 9656 includes an internal wait state generator and TA# Input signal that can be used to externally assert wait states. TA# can be selectively enabled or disabled for each Local Address Space (LBRD0[6] for Space 0, LBRD1[6] for Space 1, and/or LBRD0[22] for Expansion ROM).

Independent of the PCI Bus, the Local Bus can perform:

- Bursts as long as data is available (Continuous Burst mode)
- Bursts of 4 Lwords at a time (Burst-4 mode, recommended)
- Continuous single cycles (default)
- Dummy cycles, which may be generated on the Local Bus (refer to Sections 3.4 and 3.4.2.1)

The Direct Slave Retry Delay Clock bits (LBRD0 [31:28] can be used to program the period of time in which the PCI 9656 holds TRDY# de-asserted. The PCI 9656 issues a Retry to the PCI Bus Transaction Master when the programmed time period expires. This occurs when the PCI 9656 cannot gain control of the Local Bus and return TRDY# within the programmed time period.

3.4.2.1 Direct Slave Writes

For 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 9656 arbitrates for the Local Bus, becomes the Local Bus Master, and writes data to a Local slave device. The PCI 9656 continues to accept writes and asserts TRDY# until the Direct Slave Write FIFO is full. It then holds TRDY# de-asserted until space becomes available in the Write FIFO, or asserts STOP# and Retries the PCI Bus Master, dependent upon the LBRD0[27] register bit setting.

The PCI 9656 transfers dummy data by way of a dummy cycle, with TSIZ[0:1]=11b at the end of each transfer when the following conditions are true:

- **64-bit PCI Bus with BI mode enabled (Continuous Burst mode)** – If the last Qword on the PCI Bus has Byte Enables of F0h, the last Lword on the Local Bus has a TSIZ of 3h (dummy cycle).
- **32-bit PCI Bus with BI mode enabled (Continuous Burst mode)** – If the number of Lwords written on the PCI Bus is odd and adequately large for the Local Bus to burst data, the last Lword on the Local Bus has a TSIZ of 3h (dummy cycle).

  If any Lword on the PCI Bus has Byte Enables of Fh that are not for the first or last Lword, the Local burst is broken and the burst restarts with a new TS#. If the Lword on the PCI Bus is an odd count (third Lword, fifth Lword, and so forth), the last Lword before the burst stops is a dummy cycle. If the Lword on the PCI Bus is an even count (second Lword, fourth Lword, and so forth), the last Lword before the burst stops is ***not*** a dummy cycle.
- If the PCI data is of insufficient length to cause the Local Bus to burst data, there are no dummy cycles.

A 32-bit PCI Bus Master single cycle Write transaction results in a PCI 9656 transfer of 1 Lword of data onto the Local Bus. A 64-bit PCI Bus Master Qword data single cycle write results in PCI 9656 Burst transfers of 2 Lwords [when the BI Mode bit(s) is enabled] onto the Local Bus, if the Burst bit(s) is enabled (LBRD0[24]=1 for Space 0, LBRD1[8]=1 for Space 1, and/or LBRD0[26]=1 for Expansion ROM).

The PCI 9656 can be programmed to retain the PCI Bus by generating a wait state(s) and de-asserting TRDY#, if the Direct Slave Write FIFO becomes full. If the Local Bus Latency Timer is enabled (MARBR [16]=1) and expires (MARBR[7:0]), the Local Bus is dropped.

A Burst Write cycle from the PCI Bus through the PCI 9656 performs an MPC850 or MPC860 Burst transaction, if the following conditions are true:

- The address is Qword-aligned (64-bit PCI Bus) or Lword-aligned (32-bit PCI Bus),
- The FIFO contains at least 4 Lwords, and
- All PCI Byte Enables are set.

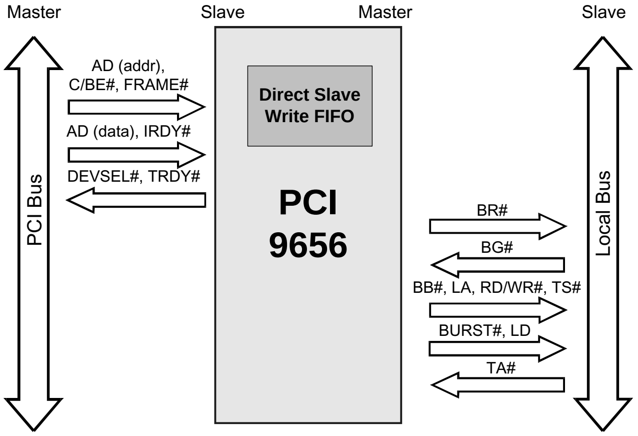


Figure 3-6. Direct Slave Write

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

3.4.2.2 Direct Slave Reads

For a PCI Bus read, the PCI Bus Master reads data from the Direct Slave Read FIFO.

The PCI 9656 holds TRDY# de-asserted while receiving an Lword from the Local Bus, unless the PCI Compliance Enable bit is set (MARBR[24]=1). (Refer to Section 3.4.2.4.) Programmable Prefetch modes are available, if prefetch is enabled – prefetch, 1 to 16, or continuous data – until the Direct Slave read ends. The Local prefetch continues until several clocks after FRAME# is de-asserted or the PCI 9656 issues a Retry or disconnect. Unused data is flushed from the FIFO unless Direct Slave Read Ahead mode is enabled (MARBR[28]=1).

For the highest Data transfer rate, the PCI 9656 can be programmed to prefetch data during a PCI Burst read on the Local Bus. When Prefetch is enabled (LBRD0[8]=0 for Space 0, LBRD1[9]=0 for Space 1, and/or LBRD0[9]=0 for Expansion ROM), prefetch can be from 1 to 16 Lwords or until the PCI Bus stops requesting. When the PCI 9656 prefetches, it drops the Local Bus after reaching the Prefetch Counter limit. In Continuous Prefetch mode, the PCI 9656 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 9656 disconnects after each Read transfer.

The PCI 9656 64-bit PCI Bus Direct Slave unaligned Qword Data Prefetch Read transfers are special cases that result in prefetching one additional Lword (32 bits) of Local Bus data than specified in the Prefetch Counter(s) (LBRD0[14:11] and/or LBRD1[14:11]), to sustain zero wait state 64-bit PCI Data transfers. For 64-bit Qword-aligned and all 32-bit PCI Bus Direct Slave Prefetch Read transfers, the PCI 9656 prefetches the amount specified in the Prefetch Counter(s). If a Direct Slave Prefetch Burst Read transfer is performed to a Local Bus device that cannot burst, and TA# is asserted for only one clock period, the PCI 9656 retains the Read data in the Direct Slave Read FIFO and does not complete the Read transfer to the PCI Bus. This occurs because the PCI 9656 is built with 64-bit FIFO architecture, and a prefetch mechanism is required to prefetch further data for transferring to occur. Under such conditions, it is necessary to disable a Prefetch or Burst feature.

In addition to Prefetch mode, the PCI 9656 supports Direct Slave Read Ahead mode (MARBR[28]=1). (Refer to Section 3.4.2.5.)

Only 32-bit PCI Bus single cycle Direct Slave Read transactions result in the PCI 9656 passing requested PCI Byte Enables (C/BE[3:0]#) to a Local Bus target device by way of TSIZ[0:1] assertion. This transaction results in the PCI 9656 reading 1 Lword or partial Lword data. For all other types of Read transactions (64-bit PCI Bus Burst transfers or Unaligned), the PCI 9656 Local Bus reads one or more complete Lwords (4 bytes).

For Read accesses mapped to PCI I/O space, the PCI 9656 does not prefetch Read data. Rather, it breaks each read of a Burst cycle into a single Address/Data cycle on the Local Bus.

If the Local Bus Latency Timer is enabled (MARBR[16]=1) and expires (MARBR[7:0]), and M mode protocol requirements are met, the Local Bus is dropped.

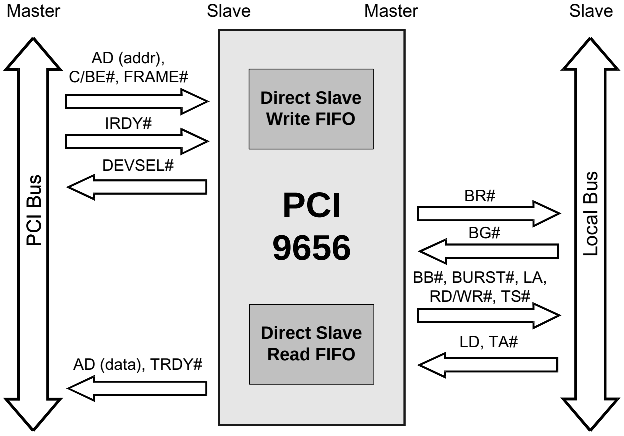


Figure 3-7. Direct Slave Read

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

3.4.2.3 Direct Slave Lock

The PCI 9656 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 PCI LOCK# Input Enable bit (MARBR[22]) for PCI-to-Local accesses.

3.4.2.4 *PCI r2.2*-Compliance Enable

The PCI 9656 can be programmed through the PCI Compliance Enable bit to perform all PCI Read/Write transactions in compliance with *PCI r2.2* (MARBR [24]=1). The following sections describe the behavior of the PCI 9656 when MARBR[24]=1.

3.4.2.4.1 Direct Slave Delayed Read Mode

PCI Bus single cycle aligned or unaligned 32-bit Direct Slave Read transactions always result in a 1-Lword single cycle transfer on the Local Bus, with corresponding Local Address and TSIZ[0:1] to reflect the PCI Byte Enables (C/BE[3:0]#) unless the PCI Read No Flush Mode bit is enabled (MARBR[28]=1). (Refer to Section 3.4.2.5.) This causes the PCI 9656 to Retry all PCI Bus Read requests that follow, until the original PCI Byte Enables are matched.

PCI Bus single cycle aligned or unaligned 64-bit Direct Slave Delayed Read transactions always result in 2-Lword Prefetch transfers on the Local Bus, with the aligned Local Address and TSIZ[0:1]=00b asserted to successfully complete Read Data transfers to a 64-bit PCI Bus. This causes the PCI 9656 to always return requested data to a 64-bit PCI master, although the PCI Byte Enables do not match the original request.

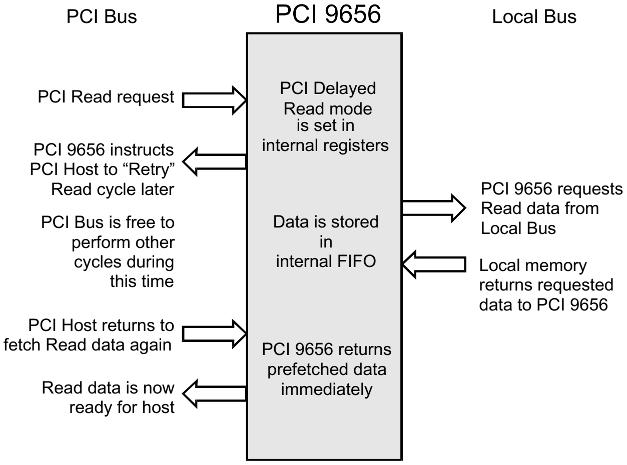

PCI Bus
PCI 9656
Local Bus
PCI Read request
PCI Delayed Read mode is set in internal registers
PCI 9656 instructs PCI Host to "Retry" Read cycle later
PCI Bus is free to perform other cycles during this time
Data is stored in internal FIFO
PCI 9656 requests Read data from Local Bus
Local memory returns requested data to PCI 9656
PCI Host returns to fetch Read data again
PCI 9656 returns prefetched data immediately
Read data is now ready for host

Figure 3-8. Direct Slave Delayed Read Mode

***Note:** Figure 3-8 represents a sequence of Bus cycles.*

3.4.2.4.2 2^{15} PCI Clock Timeout

If the PCI Master does not complete the originally requested Direct Slave Delayed Read transfer, the PCI 9656 flushes the Direct Slave Read FIFO after 2^{15} PCI clocks and grants an access to a new Direct Slave Read access. All other Direct Slave Read accesses before the 2^{15} PCI clock timeout are Retried by the PCI 9656.

3.4.2.4.3 *PCI r2.2* 16- and 8-Clock Rule

The PCI 9656 guarantees that if the first Direct Slave Write data cannot be accepted by the PCI 9656 and/or the first Direct Slave Read data cannot be returned by the PCI 9656 within 16 PCI clocks from the beginning of the Direct Slave cycle (FRAME# asserted), the PCI 9656 issues a Retry (STOP# asserted) to the PCI Bus.

During successful Direct Slave Read and/or Write accesses, the subsequent data after the first access must be accepted for writes or returned for reads within 8 PCI clocks (TRDY# asserted). Otherwise, the PCI 9656 issues a PCI disconnect (STOP# asserted) to the PCI Master.

In addition, the PCI 9656 supports the following *PCI r2.2* functions:

- No write while a Delayed Read is pending (PCI Retries for writes) (MARBR[25])
- Write and flush pending Delayed Read (MARBR[26])

3.4.2.5 Direct Slave Read Ahead Mode

The PCI 9656 also supports Direct Slave Read Ahead mode (MARBR[28]=1), where prefetched data can be read from the internal FIFO of the PCI 9656 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 and 64-bit-aligned (next address = current address + 8) for 64-bit Direct Slave transfers. Direct Slave Read Ahead mode functions with or without Direct Slave Delayed Read mode.

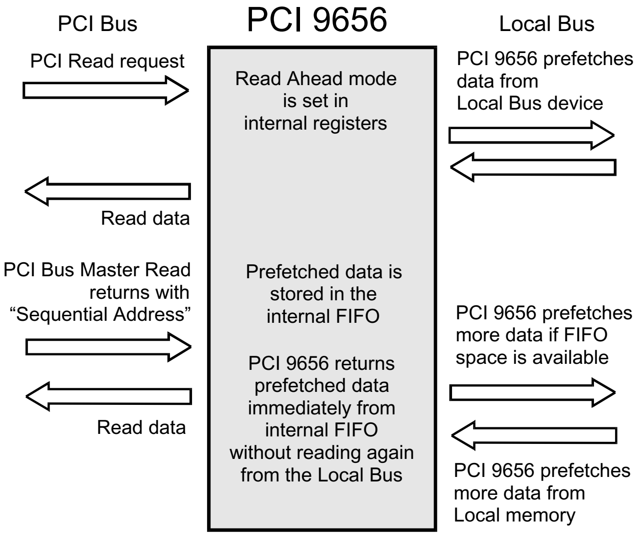

PCI Bus
PCI 9656
Local Bus
PCI Read request
Read Ahead mode is set in internal registers
PCI 9656 prefetches data from Local Bus device
Read data
PCI Bus Master Read returns with "Sequential Address"
Prefetched data is stored in the internal FIFO
PCI 9656 prefetches more data if FIFO space is available
PCI 9656 returns prefetched data immediately from internal FIFO without reading again from the Local Bus
Read data
PCI 9656 prefetches more data from Local memory

Figure 3-9. Direct Slave Read Ahead Mode

Note: *Figure 3-9 represents a sequence of Bus cycles.*

3.4.2.6 Direct Slave Delayed Write Mode

The PCI 9656 supports Direct Slave Delayed Write mode transactions, in which posted Write data accumulates in the Direct Slave Write FIFO before the PCI 9656 requests ownership of the Local Bus (BR# assertion). The Direct Slave Delayed Write mode is programmable to delay the BR# assertion for the amount of Local clocks specified in 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.

3.4.2.7 Direct Slave Local Bus TA# Timeout Mode

Direct Slave Local Bus TA# Timeout mode is enabled/disabled by LMISC2[0].

If Direct Slave Local Bus TA# Timeout mode is disabled (LMISC2[0]=0), *and* the PCI 9656 is mastering the Local Bus during a Direct Slave Read or Write Data transfer, *and* no Local Bus device asserts TA# in response to the Local Bus address generated by the PCI 9656, the PCI 9656 hangs on the Local Bus waiting for TA# assertion. To recover, reset the PCI 9656.

If Direct Slave Local Bus TA# Timeout mode is enabled (LMISC2[0]=1), *and* the PCI 9656 is mastering the Local Bus during a Direct Slave Read or Write Data transfer, the PCI 9656 uses the TA# Timeout Select bit (LMISC2[1]) to determine when to timeout while waiting for a Local Bus device to assert TA# in response to the Local Bus address generated by the PCI 9656. When LMISC2[1]=0, the PCI 9656 waits 32 Local Bus clocks for TA# assertion before timing out. When LMISC2[1]=1, the PCI 9656 waits 1,024 Local Bus clocks for TA# assertion before timing out.

If a Direct Slave Local Bus TA# Timeout occurs during a Direct Slave read, the PCI 9656 issues a Target Abort to the PCI Bus Master. If the PCI Bus Master is currently mastering the PCI 9656 (*that is*, the PCI 9656 is not awaiting a PCI Bus Master Retry of the Direct Slave read), the Target Abort is immediately issued. If the PCI Bus Master is not currently mastering the PCI 9656 (*that is*, the PCI 9656 is awaiting a PCI Bus Master Retry of the Direct Slave read), the PCI 9656 issues a Target Abort the next time the PCI Bus Master repeats the Direct Slave read.

If a Direct Slave Local Bus TA# Timeout occurs during a Direct Slave write *and* the PCI Bus Master is currently mastering the PCI 9656 (*that is*, the PCI Bus Master has not posted the Direct Slave write to the PCI 9656), the PCI 9656 immediately issues a Target Abort to the PCI Bus Master. If the PCI Bus Master is not currently mastering the PCI 9656 (*that is*, the PCI Bus Master has posted the Direct Slave write to the PCI 9656), the PCI 9656 does ***not*** issue to the PCI Bus Master any indication that the timeout occurred.

Caution: *Only use the PCI 9656 Direct Slave Local Bus TA# Timeout feature during design debug. This feature allows illegal Local Bus addresses to be easily detected and debugged. Once the design is debugged and legal addresses are guaranteed, disable Direct Slave Local Bus TA# Timeout mode to avoid unreported timeouts during Direct Slave writes.*

3.4.2.8 Direct Slave Transfer Error

The PCI 9656 supports Local Bus error conditions that use TEA# as an input. A Local Bus device may assert TEA#, either before or simultaneously with TA#. In either case, the PCI 9656 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 the Direct Slave FIFOs). To prevent bursting, hold TEA# asserted until the Direct Slave transfer completes. After sensing TEA# is asserted, the PCI 9656 asserts PCI SERR# and sets an error flag, using the Signaled System Error (SERR# Status) 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).

3.4.2.9 Direct Slave PCI-to-Local Address Mapping

Note: *Not applicable in I_2O mode.*

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:

- Direct Slave Local Address Space Range (LAS0RR, LAS1RR, and/or EROMRR)
- Direct Slave Local Address Space Local Base Address (Remap) (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. (Refer to Section 3.4.2.9.1.) These Local Bus characteristics can be modified at any time before actual data transactions.

3.4.2.9.1 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.

3.4.2.9.2 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 9656 returns zeros (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.)

3.4.2.9.3 Direct Slave PCI Initialization Example

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

1. Local initialization software or serial EEPROM contents set 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 (LAS*x*BA[0]=1, where *x* is the Local Address Space number)]
2. PCI Initialization software writes all ones (1) to the PCI Base Address, then reads it back again.
   - The PCI 9656 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).

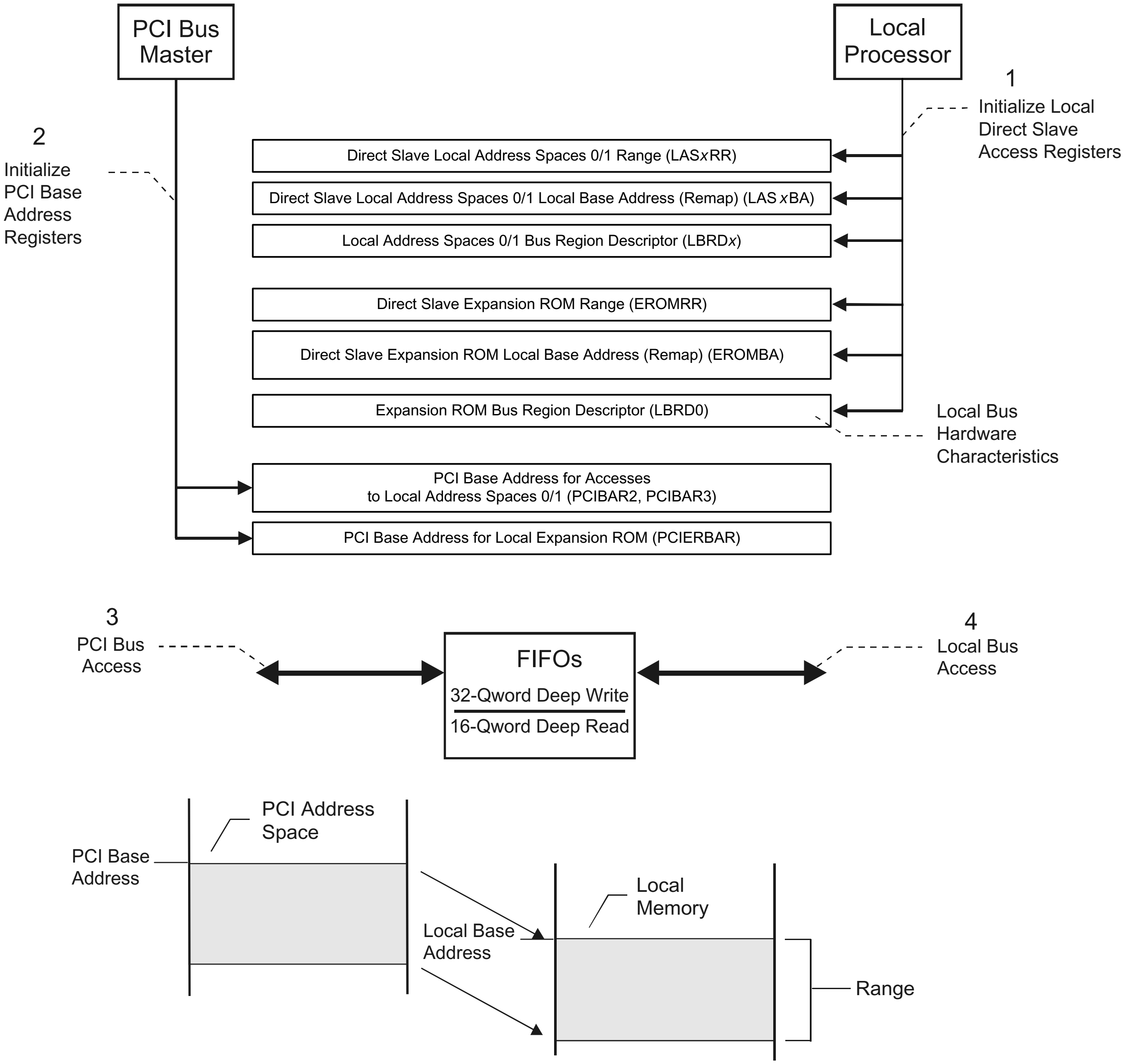

PCI Bus Master
Local Processor
1
Initialize Local Direct Slave Access Registers
2
Initialize PCI Base Address Registers
Direct Slave Local Address Spaces 0/1 Range (LASxRR)
Direct Slave Local Address Spaces 0/1 Local Base Address (Remap) (LASxBA)
Local Address Spaces 0/1 Bus Region Descriptor (LBRDx)
Direct Slave Expansion ROM Range (EROMRR)
Direct Slave Expansion ROM Local Base Address (Remap) (EROMBA)
Expansion ROM Bus Region Descriptor (LBRD0)
Local Bus Hardware Characteristics
PCI Base Address for Accesses to Local Address Spaces 0/1 (PCIBAR2, PCIBAR3)
PCI Base Address for Local Expansion ROM (PCIERBAR)
3
PCI Bus Access
FIFOs
32-Qword Deep Write
16-Qword Deep Read
4
Local Bus Access
PCI Address Space
PCI Base Address
Local Memory
Local Base Address
Range

Figure 3-10. Local Bus Direct Slave Access

3.4.2.9.4 Direct Slave Transfer Size

The TSIZ[0:1] pins correspond to the data-transfer size on the Local Bus, as listed in Tables 3-6 and 3-7.

Table 3-6. Data Bus TSIZ[0:1] Contents for Single Write 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	–
	1	0	1	0	–	–	OP2	OP3	OP2	OP3	–
Lword	0	0	0	0	OP0	OP1	OP2	OP3	–	–	–

Note: *The "–" symbol indicates that a byte is* ***not*** *required to be delivered during that Read cycle. However, the PCI 9656 drives these byte lanes, although the data is not used.*

Table 3-7. 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	–
	1	0	1	0	–	–	OP2	OP3	OP2	OP3	–
Lword	0	0	0	0	OP0	OP1	OP2	OP3	–	–	–

Note: *The "–" symbol indicates that a valid byte is* ***not*** *required during that Write cycle. However, the PCI 9656 drives these byte lanes, although the data is not used.*

3.4.2.10 Direct Slave Priority

Direct Slave accesses have a higher priority than DMA accesses, thereby preempting DMA transfers. During a DMA transfer, if the PCI 9656 detects a pending Direct Slave access, it releases the Local Bus. The PCI 9656 resumes DMA operation after the Direct Slave access completes.

When the PCI 9656 DMA Controller(s) owns the Local Bus, its BR# output and BG# input are asserted. When a Direct Slave access occurs, the PCI 9656 releases the Local Bus by de-asserting BB# and floating the Local Bus outputs. After the PCI 9656 senses that BG# is de-asserted, it requests the Local Bus for a Direct Slave transfer by asserting BR#. When the PCI 9656 receives BG#, and BB# is de-asserted, it drives the bus and performs the Direct Slave transfer. To complete the Direct Slave transfer, the PCI 9656 releases the Local Bus by de-asserting BB# and floating the Local Bus outputs. After the PCI 9656 senses that BG# is de-asserted and the Local Bus Pause Timer Counter is zero (MARBR[15:8]=0h) or disabled (MARBR[17]=0), it requests a DMA transfer from the Local Bus by re-asserting BR#. When the PCI 9656 senses that BG# is asserted and BB# de-asserted, it drives the bus and continues the DMA transfer.

3.4.3 Deadlock Conditions

A deadlock can occur when a PCI Bus Master must access the PCI 9656 Local Bus at the same time a Master on the PCI 9656 Local Bus must access the PCI Bus.

There are two types of deadlock:

- **Partial Deadlock** – Occurs when the PCI device “A” Local Master tries to access the PCI 9656 Local Bus concurrently with the PCI 9656 Local Master trying to access the PCI device “B” Local Bus.
- **Full Deadlock** – Occurs when the PCI device “A” Local Master tries to access the PCI 9656 Local Bus concurrently with the PCI 9656 Local Master trying to access the PCI device “A” Local Bus and neither Local Master relinquishes Local Bus ownership.

This applies only to Direct Master and Direct Slave accesses through the PCI 9656. Deadlock does ***not*** occur in transfers through the PCI 9656 DMA channels or internal registers (*such as*, Mailboxes).

For partial deadlock, the PCI access to the Local Bus times out (Direct Slave Retry Delay Clocks, LBRD0 [31:28]) and the PCI 9656 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 external PCI Bus Arbiter to grant the PCI Bus to the PCI 9656 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 timeout is undesirable, or a full deadlock condition exists.

When a full deadlock occurs, it is necessary to back off the Local Bus Master to prevent a system hang.

3.4.3.1 Backoff

The backoff sequence is used to break a deadlocked condition. The PCI 9656 Local RETRY# signal indicates that a deadlock condition has occurred.

The PCI 9656 starts the Backoff Timer (refer to the EROMBA register for further details) 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*, 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 9656 Direct Master Write FIFO cannot accept another Write cycle.

If Local Bus Backoff is enabled (EROMBA[4]=1), the Backoff Timer expires, and the PCI 9656 has not received BG#, the PCI 9656 asserts RETRY#. External bus logic can use RETRY# 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 9656 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 9656 detects BG#, it proceeds with the PCI Master-to-Local Bus access. When this access completes and the PCI 9656 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 9656 Direct Master Write FIFO retains all accepted data (*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 data the PCI 9656 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). The PCI 9656 is ***not*** allowed to perform a new read.

3.4.3.1.1 Software/Hardware Solution for Systems without Backoff Capability

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

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 Local Bus Arbiter not grant the bus to any Local Bus Master except the PCI 9656. Status output from the Local Bus Arbiter can be connected to 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.

3.4.3.1.2 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#.

3.4.3.2 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.

3.4.4 DMA Operation

The PCI 9656 supports two independent DMA channels capable of transferring data from the PCI-to-Local Bus and Local-to-PCI Bus.

Each channel consists of a DMA Controller and a dedicated, bi-directional FIFO. Both channels support DMA Block, Scatter/Gather, and Demand Mode transfers, with or without End of Transfer (EOT#). Master mode must be enabled (PCICR[2]=1) before the PCI 9656 can become a PCI Bus master. In addition, both DMA channels can be programmed to:

- Operate with 8-, 16-, or 32-bit Local Bus data widths
- Use zero to 15 internal wait states (Local Bus)
- Enable/disable external wait states (Local Bus)
- Set single cycle/Burst mode (refer to Table 3-8)
- Limit Local Bus bursts to four (Burst-4 mode)
- Hold Local address constant (Local Slave is FIFO) or incremented
- Perform PCI Memory Write and Invalidate (command code = Fh) or normal PCI Memory Write (command code = 7h)
- Stop/pause Local transfer with or without BDIP# (DMA Fast/Slow Terminate mode)
- Operate in DMA Clear Count mode

The PCI 9656 supports PCI Dual Address Cycles (DAC) with the upper 32-bit register(s) (DMADAC*x*). The PCI 9656 also generates dummy cycles on the PCI and Local Buses. (Refer to Section 3.4.)

The Local Bus Latency Timer (MARBR[7:0]) determines the number of Local clocks the PCI 9656 can burst data before relinquishing Local Bus ownership. The Local Pause Timer (MARBR[15:8]) sets how soon (after the Latency Timer times out) the DMA channel can again request the Local Bus.

Table 3-8. Local Bus Data Transfer Modes

Mode	Burst Enable Bit(s)	BI Mode Bit(s)	Result
Single Cycle	0	X	One TS# per data.
Burst-4	1	0	One TS# per 16 bytes (recommended for MPC850 or MPC860).
Continuous Burst	1	1	One TS# per data burst or until BI# is asserted.

Notes: *Single cycle is the default Data transfer mode.*
"X" is "Don't Care."

3.4.4.1 DMA PCI Dual Address Cycles

The PCI 9656 supports PCI Dual Address Cycles (DAC) when it is a PCI Bus Master using the DMADAC*x* register(s) for DMA Block transactions. DMA Scatter/Gather can use the DAC function by way of the DMADAC*x* register(s) or DMAMODE*x*[18]. The DAC command is used to transfer a 64-bit address to devices that support 64-bit addressing when the address is above the 4-GB address space. The PCI 9656 performs a DAC within two PCI clock periods, when the first PCI address is a Lo-Addr, with the command (C/BE[3:0]#) "Dh", and the second PCI address is a Hi-Addr, with the command (C/BE[3:0]#) "6h" or "7h", depending upon whether it is a PCI Read or PCI Write cycle. (Refer to Figure 3-11.)

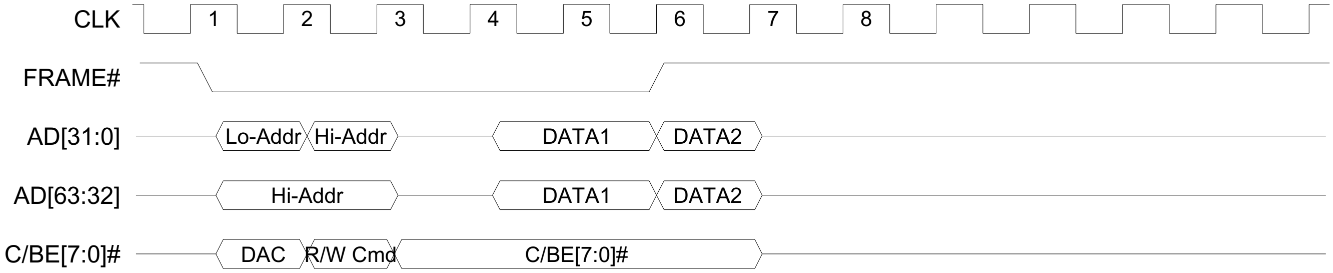


Figure 3-11. PCI Dual Address Cycle Timing

3.4.4.2 DMA Block Mode

The Host processor or the Local processor sets the PCI and Local starting addresses, transfer byte count, and transfer direction. The Host or Local processor then sets the DMA Channel Start and Enable bits (DMACSR*x*[1:0]=11b) to initiate a transfer. The PCI 9656 requests the PCI and Local Buses and transfers data. Once the transfer completes, the PCI 9656 sets the Channel Done bit(s) (DMACSR*x*[4]=1) and asserts an interrupt(s) (INTCSR[16], INTCSR[8], DMAMODE*x*[17], and/or DMAMODE*x*[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 3-12.)

During DMA transfers, the PCI 9656 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 9656 releases the PCI Bus, if one of the following conditions occur (refer to Figures 3-13 and 3-14):

- 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 according to the PCI Maximum Latency (PCIMLR) register value – and PCI GNT# is de-asserted
- PCI Target asserts STOP#

The PCI 9656 releases the Local Bus, if one of the following conditions occur:

- FIFO is empty (PCI-to-Local Bus)
- FIFO is full (Local-to-PCI Bus)
- Terminal count is reached
- Local Bus Latency Timer is enabled (MARBR[16]=1) and expires (MARBR[7:0])
- Direct Slave request is pending

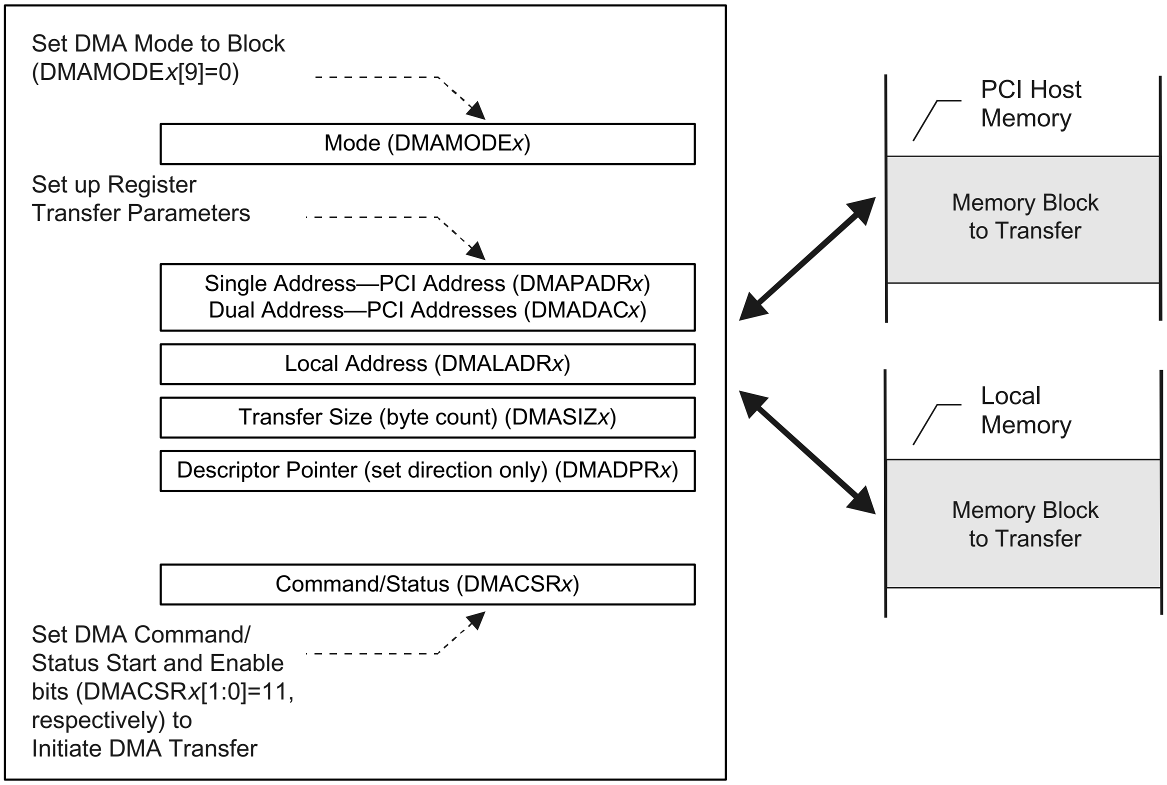

Set DMA Mode to Block (DMAMODE*x*[9]=0)
Mode (DMAMODE*x*)
Set up Register Transfer Parameters
Single Address—PCI Address (DMAPADR*x*)
Dual Address—PCI Addresses (DMADAC*x*)
Local Address (DMALADR*x*)
Transfer Size (byte count) (DMASIZ*x*)
Descriptor Pointer (set direction only) (DMADPR*x*)
Command/Status (DMACSR*x*)
Set DMA Command/ Status Start and Enable bits (DMACSR*x*[1:0]=11, respectively) to Initiate DMA Transfer
PCI Host Memory
Memory Block to Transfer
Local Memory
Memory Block to Transfer

Figure 3-12. DMA Block Mode Initialization (Single Address or Dual Address PCI)

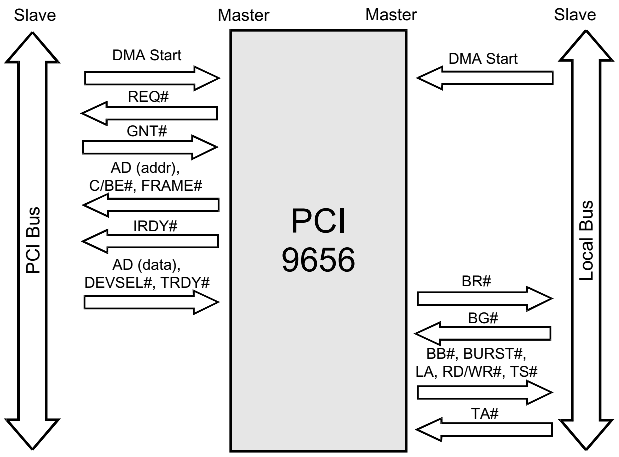


Figure 3-13. DMA, PCI-to-Local Bus

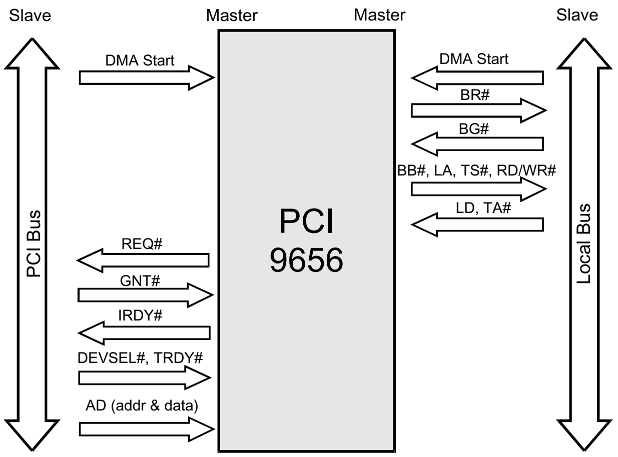


Figure 3-14. DMA, Local-to-PCI Bus

Note: *Figures 3-13 and 3-14 represent a sequence of Bus cycles.*

Table 3-9. DMA Function

BI Mode Bit(s)	Fast/Slow Terminate Mode Select Bit(s)	PCI 9656 BDIP# Output	EOT# Enabled
Continuous Burst DMAMODE*x*[7]=1	Fast DMAMODE*x*[15]=1	BDIP# is asserted on the last Data transfer, or when BI# is asserted for one CLK cycle. (Refer to Section 2.2.5.2.)	Transfer immediately terminated by EOT#.
Continuous Burst DMAMODE*x*[7]=1	Slow DMAMODE*x*[15]=0	BDIP# is asserted until the last Data transfer, or when BI# is asserted for one CLK cycle. (Refer to Section 2.2.5.2.) ***Note:*** *BI# input assertion is **not** supported during DMA Scatter/Gather Descriptor Load phase.*	Transfers up to two additional Lwords after EOT# is asserted.
Burst-4 DMAMODE*x*[7]=0	Fast DMAMODE*x*[15]=1	BDIP# is ***not*** asserted.	Transfer immediately terminated by EOT#.
Burst-4 DMAMODE*x*[7]=0	Slow DMAMODE*x*[15]=0	BDIP# is asserted by the PCI 9656. Transfers up to the nearest 16-byte boundary, then terminates (MPC850 or MPC860 compatible).	Transfers up to two additional Lwords after EOT# is asserted.

Notes: *If bursting is disabled (DMAMODEx[8]=0), the PCI 9656 performs single cycle transfers on the Local Bus.*

3.4.4.2.1 DMA Block Mode PCI Dual Address Cycles

The PCI 9656 supports the DAC feature in DMA Block mode. When the DMADAC*x* register(s) contains a value of 0h, the PCI 9656 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 3-11.)

3.4.4.3 DMA Scatter/Gather Mode

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

- Enables the Scatter/Gather mode bit(s) (DMAMODE*x*[9]=1)
- Sets up the address of initial descriptor block in the PCI 9656 Descriptor Pointer register(s) (DMADPR*x*)
- Initiates the transfer by setting a control bit(s) (DMACSR*x*[1:0]=11b)

The PCI 9656 supports zero wait state Descriptor Block bursts from the Local Bus when Local Burst is enabled (DMAMODE*x*[8]=1).

The PCI 9656 loads the first descriptor block and initiates the Data transfer. The PCI 9656 continues to load descriptor blocks and transfer data until it detects the End of Chain bit(s) is set (DMADPR*x*[1]=1). When the End of Chain bit(s) is detected, the PCI 9656 completes the current descriptor block, sets the DMA Done bit(s) (DMACSR*x*[4]=1), and asserts a PCI or Local interrupt (INTA# or LINTo#, respectively) if the interrupt(s) is enabled (DMAMODE*x*[10]=1).

The PCI 9656 can also be programmed to assert PCI or Local interrupts after each descriptor is loaded, then the corresponding Data transfer is finished.

The DMA controller(s) can be programmed to clear the transfer size at completion of each DMA transfer, using the DMA Clear Count Mode bit(s) (DMAMODE*x* [16]=1).

Notes: *In DMA Scatter/Gather 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 (DMAMODEx[18]=1). Otherwise, the DMADACx register values are used.*

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

DMA descriptors stored on the Local Bus are accessed using the same bus data width as programmed for the Data transfer.

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

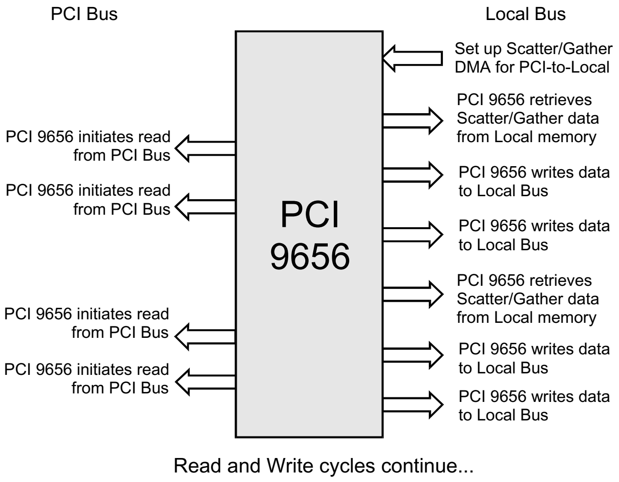

PCI Bus
Local Bus
Set up Scatter/Gather DMA for PCI-to-Local
PCI 9656 retrieves Scatter/Gather data from Local memory
PCI 9656 initiates read from PCI Bus
PCI 9656 writes data to Local Bus
PCI 9656 initiates read from PCI Bus
PCI 9656 writes data to Local Bus
PCI 9656
PCI 9656 retrieves Scatter/Gather data from Local memory
PCI 9656 initiates read from PCI Bus
PCI 9656 writes data to Local Bus
PCI 9656 initiates read from PCI Bus
PCI 9656 writes data to Local Bus
Read and Write cycles continue...

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

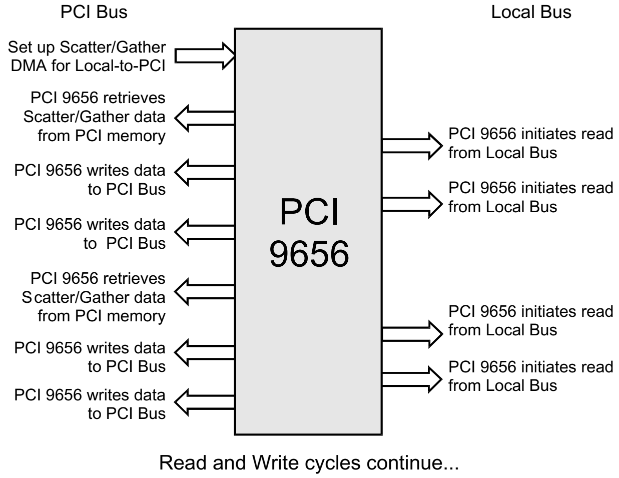

PCI Bus
Local Bus
Set up Scatter/Gather DMA for Local-to-PCI
PCI 9656 retrieves Scatter/Gather data from PCI memory
PCI 9656 initiates read from Local Bus
PCI 9656 writes data to PCI Bus
PCI 9656 initiates read from Local Bus
PCI 9656 writes data to PCI Bus
PCI 9656
PCI 9656 retrieves Scatter/Gather data from PCI memory
PCI 9656 initiates read from Local Bus
PCI 9656 writes data to PCI Bus
PCI 9656 initiates read from Local Bus
PCI 9656 writes data to PCI Bus
Read and Write cycles continue...

Figure 3-16. DMA Scatter/Gather 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.*

3.4.4.3.1 DMA Scatter/Gather PCI Dual Address Cycles

The PCI 9656 supports the DAC feature in DMA Scatter/Gather mode for Data transfers only. The descriptor blocks should reside below the 4-GB Address space.

The PCI 9656 offers three different options of how PCI DAC DMA Scatter/Gather is used. Assuming the descriptor blocks are located on the PCI Bus:

- DMADAC*x* contain(s) a non-zero value. DMAMODE*x*[18] is cleared to 0. The PCI 9656 performs a Single Address Cycle (SAC) 4-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. (Refer to Figure 3-17.) The DMA descriptor block starting addresses should be 16-byte-aligned.
- DMADAC*x* contain(s) a value of 0h. DMAMODE*x*[18] is set to 1. The PCI 9656 performs a SAC 5-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. (Refer to Figure 3-18.) The DMA descriptor block starting addresses should be 32-byte-aligned.
- DMADAC*x* contain(s) a non-zero value. DMAMODE*x*[18] is set to 1. The PCI 9656 performs a SAC 5-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. The fifth descriptor overwrites the value of the DMADAC*x* register(s). (Refer to Figure 3-18.) The DMA descriptor block starting addresses should be 32-byte-aligned.

3.4.4.3.2 DMA Clear Count Mode

The PCI 9656 supports DMA Clear Count mode (Write-Back feature, DMAMODE*x*[16]). The PCI 9656 clears each transfer size descriptor to zero (0) by writing to its location on the PCI and/or Local Bus memory at the end of each transfer chain. This feature is available for DMA descriptors located on the PCI and Local Buses.

DMA Clear Count mode can also be used in conjunction with the EOT feature (DMAMODE*x*[14]). EOT# assertion during DMA Data transfers causes the PCI 9656 to write back the amount of data bytes not transferred to the destination bus.

When encountering a PCI Master/Target Abort on a DMA Data transfer, the PCI 9656 writes random values when the descriptor is on the Local Bus. No write occurs if the descriptor is on the PCI Bus.

Note: *DMA Clear Count mode works only if there is more than one descriptor in the descriptor chain, because the first descriptor is written directly into the PCI 9656 DMA Configuration registers and the remainder are loaded from PCI or Local memory.*

3.4.4.3.3 DMA Ring Management (Valid Mode)

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

When the Ring Management Valid Stop Control bit(s) is cleared to 0 (DMAMODE*x*[21]=0), the DMA Scatter/Gather controller continuously polls the descriptor with the Valid bit(s) cleared to 0 (invalid descriptor) until the Valid bit(s) is read as 1, which initiates the DMA transfer. When the Ring Management Valid Stop Control bit(s) is set to 1 (DMAMODE*x*[21]=1), the DMA Scatter/Gather controller pauses if a Valid bit(s) with a value of 0 is detected. In this case, the processor must restart the DMA controller(s) by setting the DMA Channel Start bit(s) (DMACSR*x*[1]=1). The DMA Clear Count mode bit(s) must be enabled (DMAMODE*x*[16]=1) for the Ring Management Valid bit(s) (DMASIZ*x*[31]) to be cleared at the completion of each descriptor. (Refer to Figure 3-19 and/or Figure 3-20 for the DMA Ring Management descriptor load sequence for SAC and DAC PCI addresses.)

Notes: *In DMA Ring Management Scatter/Gather mode, the descriptor includes the transfer size with a valid descriptor bit, PCI and Local Address Space, and next descriptor pointer. It also includes a DAC value if the DAC Chain Load bit(s) is enabled (DMAMODEx[18]=1). Otherwise, the DMADACx register value is used.*

In Ring Management mode, the transfer size is loaded before the PCI address.

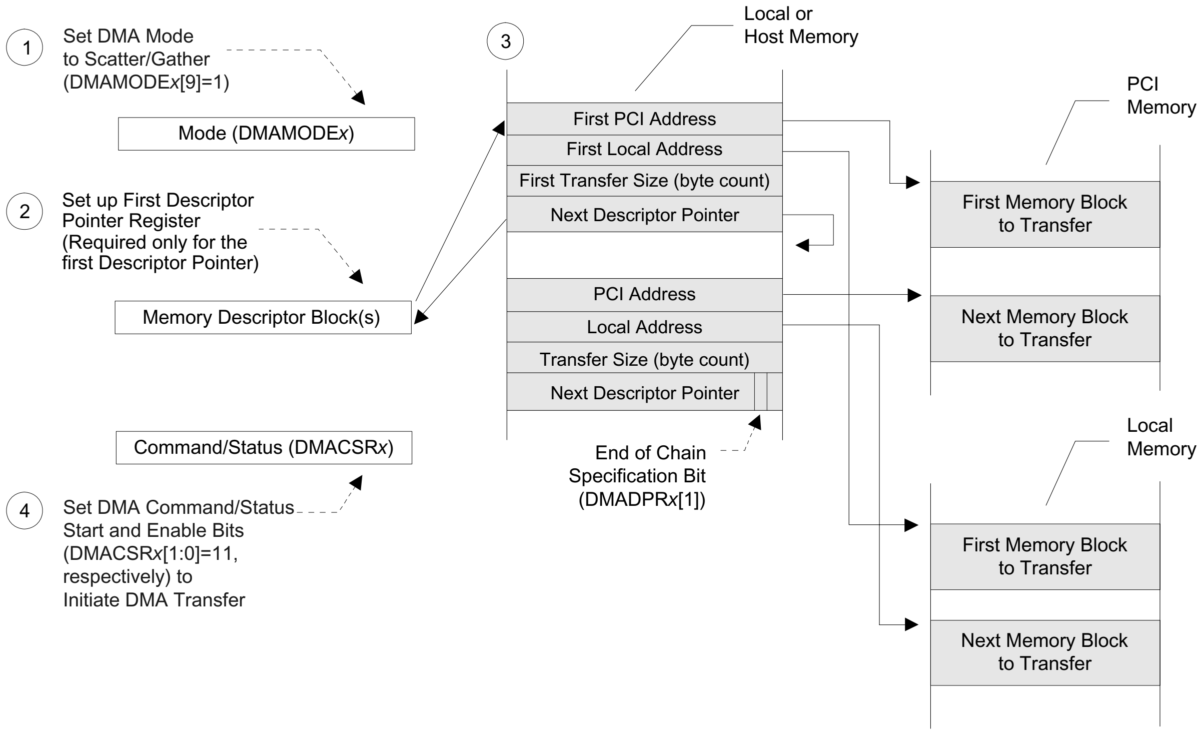


Figure 3-17. DMA Scatter/Gather Mode Descriptor Initialization
[PCI SAC/DAC PCI Address (DMADACx) Register Dependent]

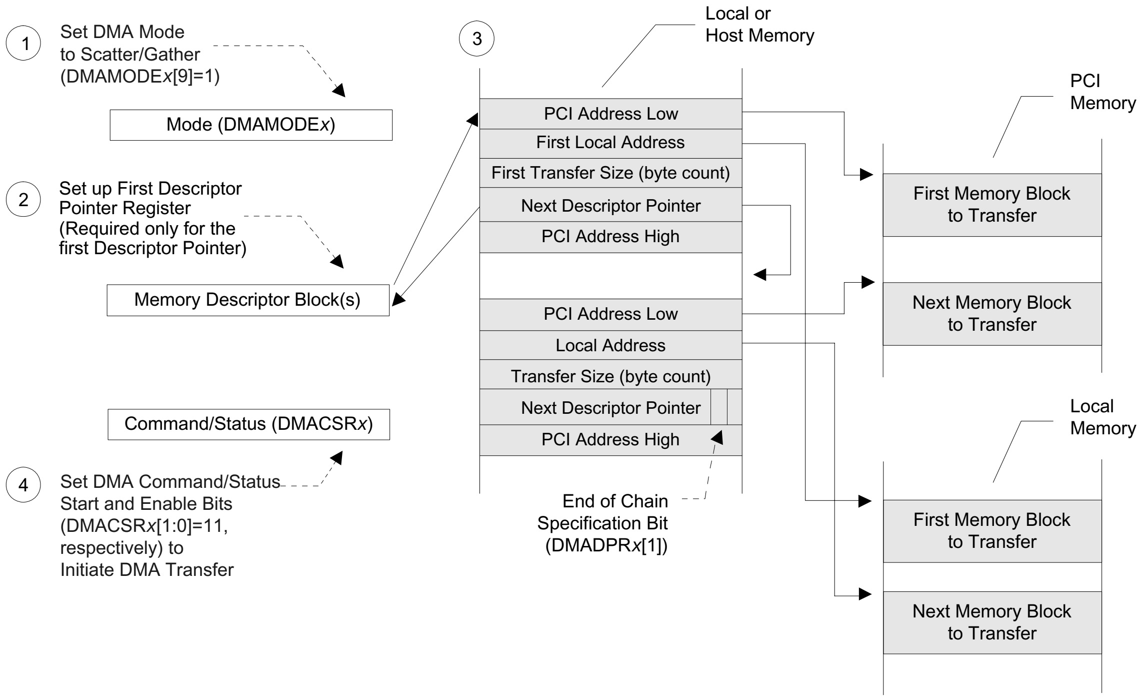


Figure 3-18. DMA Scatter/Gather Mode Descriptor Initialization
[DAC PCI Address (DMAMODEx[18]) Descriptor Dependent (PCI Address High Added)]

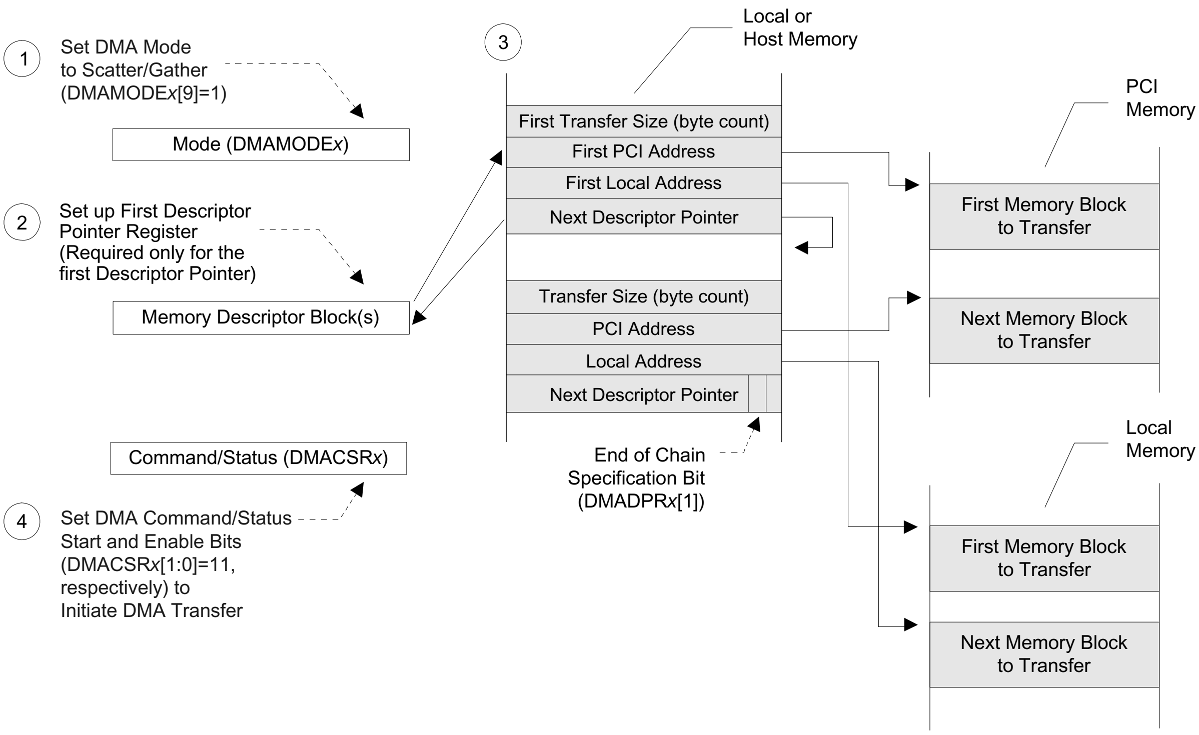


Figure 3-19. DMA Ring Management Scatter/Gather Mode Descriptor Initialization [PCI SAC/DAC PCI Address (DMADACx) Register Dependent]

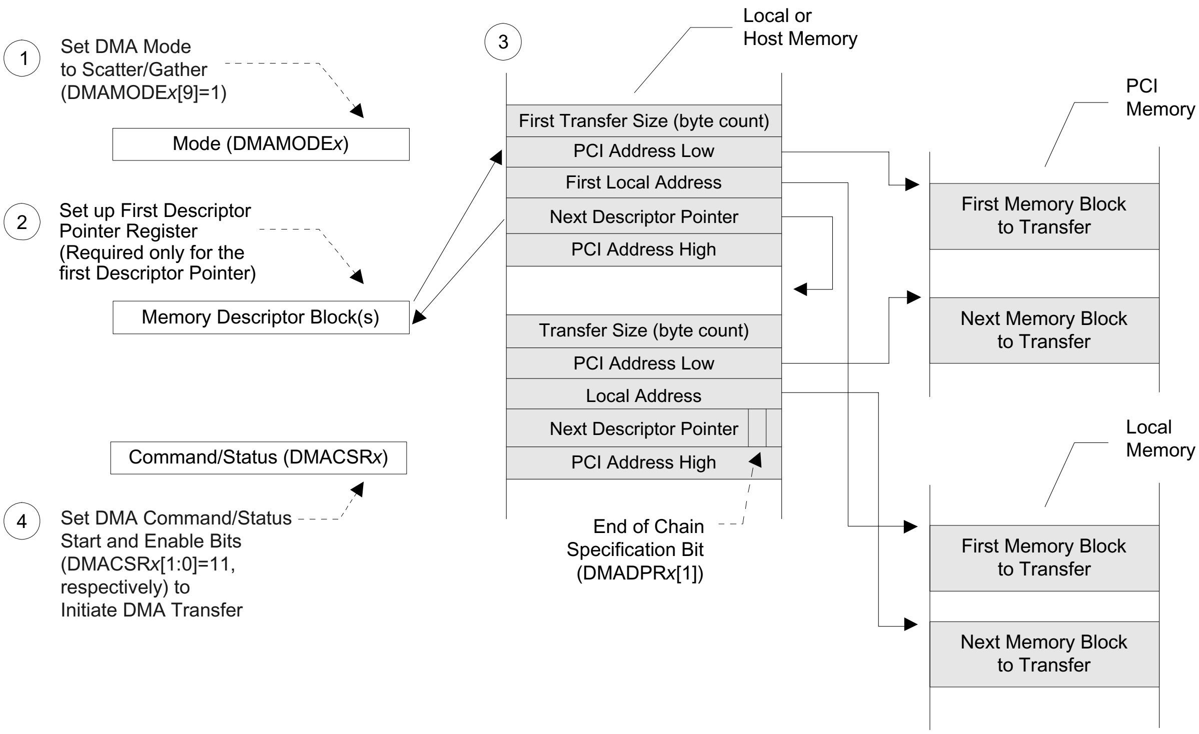


Figure 3-20. DMA Ring Management Scatter/Gather Mode Descriptor Initialization [DAC PCI Address (DMAMODEx[18]) Descriptor Dependent (PCI Address High Added)]

3.4.4.4 DMA Memory Write and Invalidate

The PCI 9656 can be programmed to perform Memory Write and Invalidate (MWI) cycles to the PCI Bus for DMA transfers, as well as Direct Master transfers. (Refer to Section 3.4.1.12.) The PCI 9656 supports MWI 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 9656 performs Write transfers (using the command code programmed in CNTRL[7:4]) rather than MWI transfers.

DMA MWI transfers are enabled when the Memory Write and Invalidate Mode for DMA Transfers and the Memory Write and Invalidate Enable bits are set (DMAMODE*x*[13]=1 and PCICR[4]=1, respectively).

In MWI mode, the PCI 9656 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 9656 completes the remainder of that cache line, using normal writes before resuming MWI transfers. If an MWI cycle is in progress, the PCI 9656 continues to burst if another cache line is read from the Local Bus before the cycle completes. Otherwise, the PCI 9656 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 9656 completes the DMA transfer, using normal writes.

An EOT# assertion, or DREQ*x*# de-assertion in Demand Mode occurring before the cache line is read from the Local Bus, results in a normal PCI Memory write of the data read into a DMA FIFO. If the DMA Data transfer starts from a non-cache line boundary, it first performs normal writes until it reaches the cache line boundary. If the DMA Data transfer possesses more than one cache line of data in the DMA FIFO, it starts the MWI transfer.

3.4.4.5 DMA Abort

DMA transfers may be aborted by software commands, or by issuing the EOT# signal. (Refer to Section 3.4.4.12 for further details about EOT#.) To abort a DMA transfer, follow these steps:

1. Clear the DMA Channel Enable bit(s) (DMACSR*x*[0]=0).
2. Abort the DMA transfer by setting the DMA Channel Abort bit(s) along with the corresponding DMA Channel Start bit(s) (DMACSR*x*[2:1]=11b, respectively).
3. Wait until the DMA Channel Done bit(s) is set (DMACSR*x*[4]=1).

Note: *One to two Data transfers occur after the Abort bit(s) is set. Software may simultaneously set DMACSRx[2:0]. Setting software commands to abort a DMA transfer when no DMA cycles are in progress causes the next DMA transfer to abort.*

3.4.4.6 DMA Channel Priority

The DMA Channel Priority bits (MARBR[20:19]) can be used to specify the priorities listed in Table 3-10.

Table 3-10. DMA Channel Priority Bit Specifications

MARBR[20:19]	Channel Priority
00b	Rotating
01b	DMA Channel 0
10b	DMA Channel 1
11b	***Reserved***

3.4.4.7 DMA Channel *x* Interrupts

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

The DMA Channel Done bit(s) (DMACSR*x*[4]) can be used to determine whether an interrupt is one of the following:

- DMA Done interrupt
- Transfer complete for current descriptor interrupt

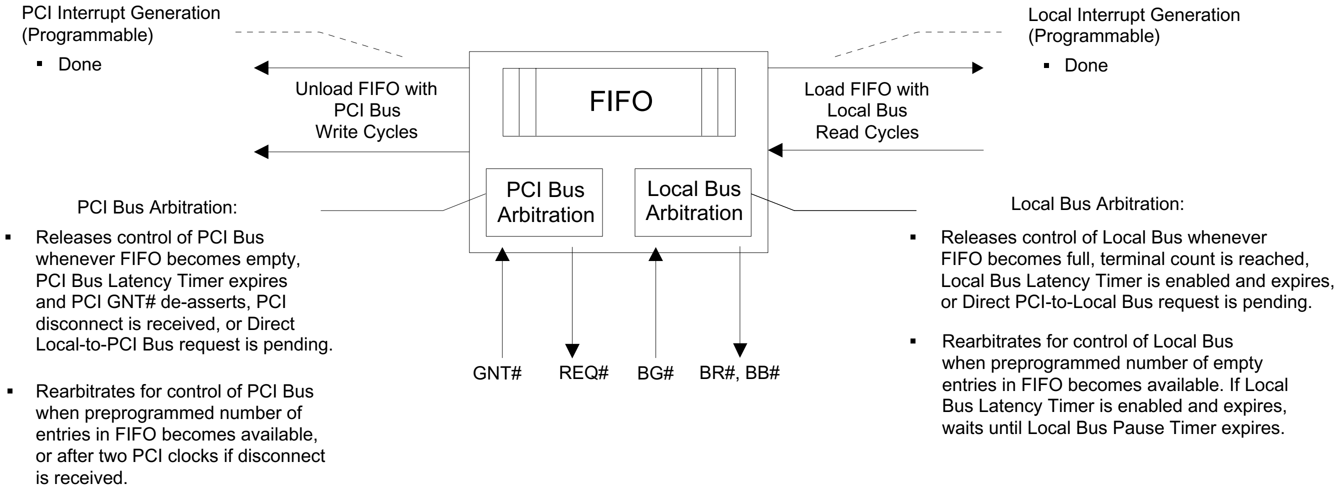
Setting DMAMODE*x*[10]=1 enables a DMA Channel Done interrupt. In DMA Scatter/Gather mode, the Descriptor Pointer register Interrupt after Terminal Count bit(s) (DMADPR*x*[2], loaded from Local memory) specifies whether to assert an interrupt at the end of the transfer for the current descriptor.

Setting DMACSR*x*[3]=1 clears a DMA Channel interrupt.

3.4.4.8 DMA Data Transfers

The PCI 9656 DMA controllers can be programmed to transfer data from the Local-to-PCI Bus or from the PCI-to-Local Bus, as illustrated in Figures 3-21 and 3-22, respectively.

3.4.4.8.1 Local-to-PCI Bus DMA Transfer



PCI Bus Arbitration:

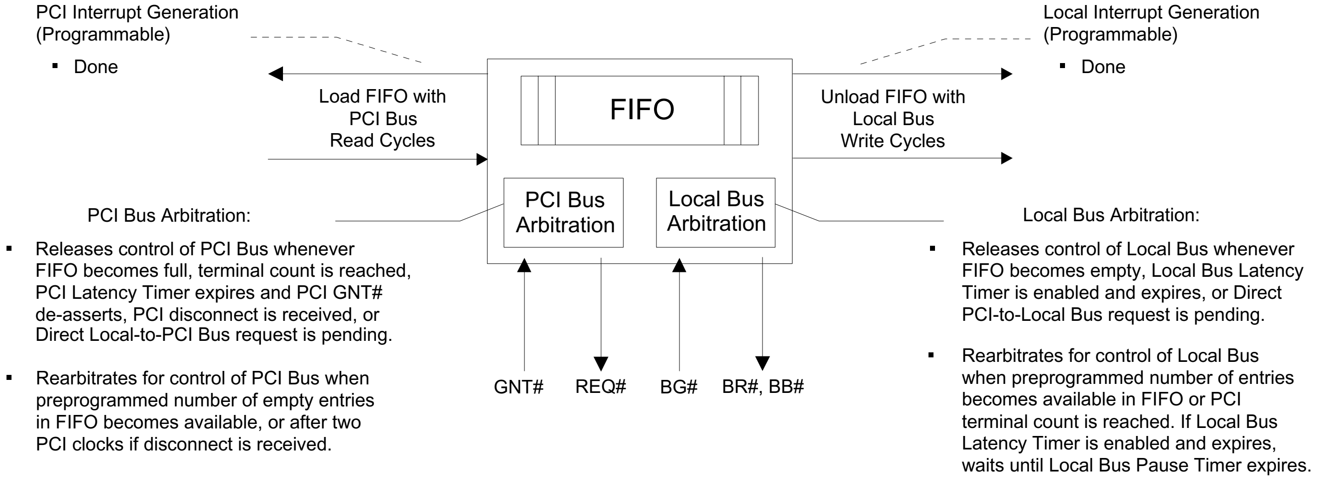
- Releases control of PCI Bus whenever FIFO becomes empty, PCI Bus Latency Timer expires and PCI GNT# de-asserts, PCI disconnect is received, or Direct Local-to-PCI Bus request is pending.
- Rearbitrates for control of PCI Bus when preprogrammed number of entries in FIFO becomes available, or after two PCI clocks if disconnect is received.

Local Bus Arbitration:

- Releases control of Local Bus whenever FIFO becomes full, terminal count is reached, Local Bus Latency Timer is enabled and expires, or Direct PCI-to-Local Bus request is pending.
- Rearbitrates for control of Local Bus when preprogrammed number of empty entries in FIFO becomes available. If Local Bus Latency Timer is enabled and expires, waits until Local Bus Pause Timer expires.

Figure 3-21. Local-to-PCI Bus DMA Data Transfer Operation

3.4.4.8.2 PCI-to-Local Bus DMA Transfer



PCI Bus Arbitration:

- Releases control of PCI Bus whenever FIFO becomes full, terminal count is reached, PCI Latency Timer expires and PCI GNT# de-asserts, PCI disconnect is received, or Direct Local-to-PCI Bus request is pending.
- Rearbitrates for control of PCI Bus when preprogrammed number of empty entries in FIFO becomes available, or after two PCI clocks if disconnect is received.

Local Bus Arbitration:

- Releases control of Local Bus whenever FIFO becomes empty, Local Bus Latency Timer is enabled and expires, or Direct PCI-to-Local Bus request is pending.
- Rearbitrates for control of Local Bus when preprogrammed number of entries becomes available in FIFO or PCI terminal count is reached. If Local Bus Latency Timer is enabled and expires, waits until Local Bus Pause Timer expires.

Figure 3-22. PCI-to-Local Bus DMA Data Transfer Operation

3.4.4.9 DMA Local Bus Error Condition

The PCI 9656 supports Local Bus error conditions with the TEA# signal. A device on the Local Bus may assert TEA#, either before or simultaneously with TA#. In either case, the PCI 9656 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 9656 asserts PCI SERR# and sets the Signaled System Error (SERR# Status) bit (PCISR[14]=1), if enabled (PCICR[8]=1), 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 9656 Local Bus Pause and Latency Timers (MARBR[15:0]) can be used to better use the Local Bus.

3.4.4.10 DMA Unaligned Transfers

For unaligned Local-to-PCI transfers, the PCI 9656 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 Local Bus bursting is enabled (DMAMODE*x*[8]=1), the PCI 9656 bursts thereafter. Lwords are assembled, aligned to the PCI Bus address, and loaded into the FIFO until they reach the nearest 16-byte boundary.

For PCI-to-Local transfers, Lwords/Qwords 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 9656 bursts thereafter. All PCI Bus unaligned reads start on either an Lword or Qword PCI Address boundary. Therefore, if DMA PCI-to-Local Bus transfers are programmed to start on an unaligned Lword PCI Address boundary (PCI Bus is 64 and/or 32 bits wide), the PCI 9656 starts a DMA read on the aligned Lword/Qword PCI Address boundary with C/BE[7:0]#=F0h, causing extra bytes of data to be read. The PCI 9656 internally masks extra bytes of data and does not drive them onto the Local Bus. The same applies to the end of the DMA PCI-to-Local Bus unaligned transfer, if the PCI 9656 completes a PCI Bus read on an unaligned Lword PCI Address boundary.

3.4.4.11 DMA Demand Mode, Channel *x*

DMA Demand mode allows the transfer of DMA data to be controlled by the DREQ*x*# input pin(s). When DREQ*x*# is asserted, the PCI 9656 starts a DMA transfer, based upon the values programmed into the DMA registers or by using internally stored values when resuming the transfer after it was paused in response to a DREQ*x*# de-assertion. The PCI 9656 asserts DACK*x*#, to indicate that the DMA transfer is in-progress on the Local Bus. DACK*x*# asserts with TS#, and remains asserted until one clock before the PCI 9656 de-asserts BB# to release the Local Bus. Data is only written to, or read from, the Local Bus while DACK*x*# is asserted. DACK*x*# de-assertion indicates that the DMA transfer has completed or is being paused in response to a DREQ*x*# de-assertion, or because the PCI 9656 needs to release the Local Bus as described in Section 3.4.4.2.

While processing in DMA Demand mode, the amount of data transferred during a Local Bus access depends upon the length and timing of the DREQ*x*# assertion. If DREQ*x*# is not asserted a sufficient length of time that allows all data to be transferred to complete the DMA transfer, the PCI 9656 transfers data in Lword-sized chunks until it releases the Local Bus. For an 8-bit Local Bus device, the PCI 9656 releases the Local Bus after transferring the last byte of an Lword. For a 16-bit Local Bus device, the PCI 9656 releases the bus after transferring the last word of an Lword, unless it is the last byte or word of the transfer.

Each DMA channel is capable of two types of transfers, PCI-to-Local or Local-to-PCI, as selected by the DMA Channel *x* Direction of Transfer bit (DMADPR*x*[3]). In either case, the PCI 9656 becomes a Master on the PCI and Local Buses, to execute the DMA transfer. If the DMA Local-to-PCI FIFO becomes full, DACK*x*# is de-asserted and the DREQ*x*# assertion is ignored. Subsequent transfers do not occur until DACK*x*# is re-asserted, indicating that additional room is now available in the FIFO.

The following can be used to modify the functionality of the DMA Demand mode transfer:

- DMA Channel *x* Continuous Burst Enable bit (DMAMODE*x*[7]) setting
- DMA Channel *x* Fast/Slow Terminate Mode Select bit (DMAMODE*x*[15]) setting
- PCI Bus starting or resuming address

- Local Bus starting or resuming address
- Local Bus widths

The functional changes that these items cause are explained, in detail, in the sections that follow.

DREQ*x*# can be asserted from one clock to more clocks than is necessary, to transfer all the DMA data. This, combined with the other factors that modify the PCI 9656 functionality, create a wide variety of responses to the assertion of DREQ*x*#. The simplest way to use DMA Demand mode is to use Slow Terminate Mode (DMAMODE*x*[15]=0) to assert the DREQ*x*# pin, wait for DACK*x*# to assert, allow some data to transfer before de-asserting DREQ*x*#, then continue to accept or provide data until the PCI 9656 releases the Local Bus. If the DREQ*x*# assertion for a DMA transfer will never resume for ongoing transfers, a DMA Abort procedure or the EOT# Input signal/pin assertion can be applied, to terminate/complete the DMA transfer and/or flush the DMA FIFO.

Note: *If bursting is enabled (DMAMODEx[8]=1) when DMA Constant Address mode is also enabled (DMAMODEx [11]=1), the PCI 9656 enables Continuous Burst mode, regardless of the Continuous Burst bit setting (DMADMODEx[7]=x).*

3.4.4.11.1 Fast/Slow Terminate Mode

In Slow Terminate mode (DMAMODEx[15]=0), the PCI 9656 adheres to the M mode protocol and de-asserts the BDIP# output, to indicate that the current Data transfer is the last Lword to be transferred of a burst before releasing the Local Bus while processing in DMA Demand mode.

In Fast Terminate Mode (DMAMODE*x*[15]=1) the PCI 9656 is ***not*** required to adhere to the M mode protocol regarding BDIP# output de-assertion (to indicate that the current data transfer is the last Lword to be transferred of a burst before releasing the Local Bus). Additionally, it forces the PCI 9656 into Continuous Burst mode, regardless of the DMAMODE*x*[7] register bit setting.

Slow Terminate mode is the default setting. Running in Slow Terminate mode is recommended, unless the user needs the PCI 9656 to immediately release the Local Bus, and is willing to account for a disconnection without BDIP# assertion on the last Data transfer. Running in Fast Terminate mode is usually combined with EOT# assertion, to terminate/complete the DMA transfer and/or flush the DMA FIFO.

3.4.4.11.2 PCI-to-Local DMA Demand Mode

For a PCI-to-Local Bus DMA Demand Mode transfer, independent of the DREQ*x*# assertion the PCI 9656 starts reading data from the PCI Bus into the DMA FIFO, upon the setting of the DMA Channel *x* Start bit (DMACSR*x*[1]=1). The PCI 9656 continued to read data from the PCI Bus, and attempts to keep the FIFO full, until the entire DMA transfer is complete. The PCI 9656 waits for at least 2 Lwords of data to be read into the DMA FIFO and DREQ*x*# to be asserted before attempting to arbitrate for, and write data to, the Local Bus. The PCI 9656's behavior in response to DREQ*x*# assertion depends upon the following:

- When, and for how long, DREQ*x*# is asserted
- Where DREQ*x*# is de-asserted with relation to DACK*x*#
- Data availability
- PCI and Local Bus starting or resumption addresses
- DMA Channel *x* Fast/Slow Terminate Mode Select bit (DMAMODE*x*[15]) setting
- DMA Channel *x* Continuous Burst Enable bit (DMAMODE*x*[7]) setting
- Transfer/protocol dependencies

At least 2 Lwords of data must be available in the FIFO, for the PCI 9656 to respond to DREQ*x*# assertion for one or more Local clock periods. The detailed response descriptions that follow assume that 2 Lwords have already been read into the FIFO, or that DREQ*x*# is being asserted for a sufficient length of time to read 2 Lwords into the FIFO.

EOT# assertion causes the PCI 9656 to terminate the ongoing DMA transfer and flush the FIFO. However, before the PCI 9656 releases the Local Bus, it may or may not need to write additional data. Details of the terminating Local Bus behavior, when EOT# is asserted, are also provided below.

Slow Terminate mode (DMAMODE*x*[15]=0) in Burst-4 mode (DMAMODE*x*[8, 7]=10b)

- If the Local Bus starting or resuming address is double Qword-aligned (LA[28:31]=0h), and DREQ*x*# is asserted for one Local Bus clock cycle, or is asserted then de-asserted before the fourth Data transfer of a 4-Lword burst, the PCI 9656 executes/completes a 4-Lword Burst write and releases the Local Bus.

  If DREQ*x*# remains asserted, the PCI 9656 completes the current 4-Lword burst and continues to execute and complete additional 4-Lword Burst writes.

  If DREQ*x*# is de-asserted on the fourth Data transfer of any 4-Lword burst, the PCI 9656 completes the current burst and must complete one additional 4-Lword burst.
- If the Local Bus starting or resuming address is not double Qword-aligned (LA[28:31]≠0h), the PCI 9656 single cycles up to next double Qword-aligned boundary, then starts bursting 4 Lwords at a time.

  If DREQ*x*# is de-asserted during a burst, the PCI 9656 completes the current burst and must complete one additional 4-Lword burst if DREQ*x*# is de-asserted on the fourth Data transfer.

  If DREQ*x*# is de-asserted during or before the start of a single cycle write, the PCI 9656 completes that write, then releases the Local Bus.

  If DREQ*x*# is de-asserted during TS# assertion, the PCI 9656 must complete the current single cycle write and one additional single cycle write or 4-Lword Burst write.
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is writing the fourth Data transfer of the 4-Lword burst, the PCI 9656 must complete the current 4-Lword burst and one additional 4-Lword burst before releasing the Local Bus. Otherwise, the PCI 9656 will complete the current 4-Lword burst, and then release the Local Bus.

  If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is executing a single cycle write, the PCI 9656 completes that write and must complete another single cycle write or a 4-Lword Burst write before terminating the ongoing DMA Data transfer, flushing the DMA FIFO, and releasing the Local Bus.

Slow Terminate mode (DMAMODE*x*[15]=0) in Continuous Burst mode (DMAMODE*x*[8, 7]=11b)

- Regardless of the Local Bus starting or resuming address, if DREQ*x*# is asserted for one Local Bus Clock cycle, or asserted then de-asserted before or with TS# assertion, the PCI 9656 executes a 1-Lword single cycle write before it releases the Local Bus.
- If the Local Bus starting or resuming address is double Qword-aligned (LA[28:31]=0h), and DREQ*x*# remains asserted past the assertion of TS#, the PCI 9656 starts a Burst write. The PCI 9656 continues bursting until DREQ*x*# is de-asserted. Upon DREQ*x*# de-assertion, the PCI 9656 completes the current Data cycle and one additional Data cycle, with BDIP# de-asserted, before releasing the Local Bus.
- If the Local Bus starting or resuming address is not double Qword-aligned (LA[28:31]≠0h), and DREQ*x*# remains asserted past TS# assertion, the PCI 9656 single cycles up to the next double Qword-aligned boundary, then starts a burst. After the PCI 9656 starts a burst, it continues to burst Write data until DREQ*x*# is de-asserted. Upon DREQ*x*# de-assertion, the PCI 9656 complete the current Data cycle and one additional Data cycle with BDIP# de-asserted, before releasing the Local Bus.
- If DREQ*x*# is de-asserted while the PCI 9656 is executing a single cycle transfer, the PCI 9656 must complete the current single cycle, and may or may not need to complete one additional single cycle or 1-Lword burst.
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is bursting (writing) data, the PCI 9656 must complete the current Data cycle and one additional Data cycle with BDIP# de-asserted, before releasing the Local Bus.

  If the PCI 9656 is executing a single cycle transfer when EOT# is asserted along with DREQ*x*# de-assertion, it must complete the current single cycle, and may or may not need to complete one additional single cycle or 1-Lword burst.

  If the PCI 9656 is asserting TS# for a Burst transfer when EOT# is asserted along with DREQ*x*# de-assertion, the PCI 9656 must complete a 2-Lword burst before terminating the ongoing DMA Data transfer, flushing the DMA FIFO, and releasing the Local Bus.

Fast Terminate mode (DMAMODE*x*[15]=1), Continuous Burst mode (DMAMODE*x*[8, 7]=1xb)

In Fast Terminate mode (DMAMODE*x*[15]=1), the PCI 9656 defaults into Continuous Burst mode, regardless of the DMA Channel *x* Continuous Burst Mode bit (DMAMODE*x*[7]) setting. With these settings, the PCI 9656 operates the same as Slow Terminate/ Continuous Burst mode, with the following differences:

- BDIP# is never asserted.
- If DREQ*x*# is de-asserted during a single cycle transfer, the PCI 9656 releases the Local Bus after the current write transfer completes. For a burst, the PCI 9656 writes one additional Lword with DREQ*x*# de-asserted and TA# asserted [or the internal Wait State Counter(s) has decremented to 0] or wait for TA# assertion to complete the Write cycle before releasing the Local Bus.
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is starting a transfer to write data or is waiting (TA# de-asserted) to write data during a burst, the PCI 9656 must write one additional Lword of data before it releases the Local Bus. The PCI 9656 completes the final data transfer if TA# is asserted with EOT#, or when TA# becomes asserted after EOT# assertion.

  If EOT# is asserted while waiting to complete a single cycle transfer, the PCI 9656 completes that transfer before releasing the Local Bus.

3.4.4.11.3 Local-to-PCI DMA Demand Mode

For a Local-to-PCI Bus DMA Demand Mode transfer, the PCI 9656 does not arbitrate on the Local Bus to read data into the DMA FIFO until the DMA Channel *x* Start bit (DMACSR*x*[1]=1) and the DREQ*x*# signal/pin is asserted. As long as DREQ*x*# is asserted, the PCI 9656 continues to read data from the Local Bus and attempts to keep the FIFO full, until the entire DMA transfer is complete. The PCI 9656 waits for at least 1 Lword of data to be read into the DMA FIFO before attempting to arbitrate for, and write data to, the PCI Bus. The PCI 9656 behavior in response to DREQ*x*# assertion depends upon the following:

- When, and for how long, DREQ*x*# is asserted
- Where DREQ*x*# is de-asserted with relation to DACK*x*#
- Data availability
- PCI and Local Bus starting or resumption addresses
- DMA Channel *x* Fast/Slow Terminate Mode Select bit (DMAMODE*x*[15]) setting
- DMA Channel *x* Continuous Burst Enable bit (DMAMODE*x*[7]) setting
- Transfer/protocol dependencies

For transfers that are reading data from a 32-bit Local Bus device, the PCI 9656 require 1 or 2 Lwords to be read (explanation follows) and put into the DMA FIFO before it arbitrates for, and writes data to, the PCI Bus. As a result, if 2 Lwords must be read, one Lword of data will remain in the DMA FIFO until the second Lword is read. Upon reading the required amount of data, the PCI 9656 will then arbitrate for, and write the data to, the PCI Bus.

For non-Lword aligned transfers that read data from 8- or 16-bit Local Bus devices, the PCI 9656 requires a minimum amount of bytes or words to be read from the Local Bus device before it arbitrates for, and writes data to, the PCI Bus. The number of bytes or words that it must read before transferring the data to the PCI Bus is determined by the PCI starting address. As a result, there could be seven or fewer bytes remaining in the DMA FIFO at any given time.

Therefore, once a DMA Demand Mode transfer is started (DREQ*x*# asserted), it may be necessary for the PCI 9656 to read more bytes, words or Lwords from the Local Bus than are required to complete the DMA transfer. If bytes, words, or an Lword remain in the DMA FIFO due to a DREQ*x*# de-assertion, when DREQ*x*# is re-asserted (and additional data is read), that data is transferred to the PCI Bus. If the DREQ*x*# assertion never resumes for ongoing transfers, a DMA Abort procedure or EOT# assertion can be applied to terminate/complete the ongoing DMA transfer.

Because there are 2 Lwords per DMA FIFO location, upon a DREQ*x*# de-assertion, the PCI 9656 may or may not need to continue to read or write additional data. It is recommended that DREQ*x*# be asserted until DACK*x*# is asserted and at least 4 Lwords of data are read into the DMA FIFO.

The PCI 9656's responses for allowing smaller transfers inside an ongoing DMA transfer are provided below.

EOT# assertion causes the PCI 9656 to terminate the ongoing DMA transfer. However, before the PCI 9656 releases the Local Bus, it may or may not need to read additional data. Details of the terminating Local Bus behavior when EOT# is asserted are also provided below. All data that is read into the DMA FIFO is written to the PCI Bus.

Slow Terminate mode (DMAMODE*x*[15]=0) in Burst-4 mode (DMAMODE*x*[8, 7]=10b)

- If the Local Bus starting or resuming address is double Qword-aligned (LA[28:31]=0h), and DREQ*x*# is asserted for one Local Bus clock cycle, or asserted then de-asserted before the fourth Data transfer of a 4-Lword burst, the PCI 9656 executes/completes a 4-Lword Burst read, and then releases the Local Bus.

  If DREQ*x*# remains asserted, the PCI 9656 completes the current 4-Lword burst and continues to execute and complete additional 4-Lword Burst reads.

  If DREQ*x*# is de-asserted on or after the third Data transfer of any 4-Lword burst, the PCI 9656 completes the current burst and must complete one additional 4-Lword burst.
- If the Local Bus starting address is not double Qword-aligned (LA[28:31]≠0h), the PCI 9656 single cycles up to next double Qword-aligned boundary, then starts bursting 4 Lwords at a time.

  If DREQ*x*# is de-asserted during a burst, the PCI 9656 completes the current burst and may need to complete an additional 4-Lword burst.

  If DREQ*x*# is de-asserted during or before the start of a single cycle read, the PCI 9656 completes that read and any additional single cycle reads, then completes a 4-Lword burst.
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is reading the fourth Data transfer of the 4-Lword burst, the PCI 9656 must complete the current 4-Lword burst and one additional 4-Lword burst, before releasing the Local Bus. Otherwise, it will complete the current 4-Lword burst, and then release the Local Bus.

Slow Terminate mode (DMAMODE*x*[15]=0) in Continuous Burst mode (DMAMODE*x*[8, 7]=11b)

- Depending upon the PCI and Local Bus starting or resuming addresses, where and how long DREQ*x*# is asserted and where it is de-asserted, the PCI 9656 must execute zero to three single cycle transfers, followed by bursts of any size. The size of the burst is determined by the length of time DREQ*x*# remains asserted. Any DREQ*x*# assertion causes the PCI 9656 to read at least one Lword of data. The only time the PCI 9656 starts or resumes the transfer with a burst is if the address is double Qword-aligned (LA[28:31]=0h).
- After DREQ*x*# is de-asserted, the DMA controller(s) release the Local Bus after the completion of one to two additional Lword Data transfers. The transfer are paused on an Lword boundary (LA[30:31]=00b). For a burst, the PCI 9656 reads one additional Lword with BDIP# asserted, then releases the Local Bus after another Lword is read with BDIP# de-asserted and TA# asserted [or the internal Wait State Counter(s) has decremented to 0].

  If DREQ*x*# is de-asserted while the PCI 9656 is single cycling, the PCI 9656 must complete the current single cycle read and one additional Lword read, before releasing the Local Bus.
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is reading data or waiting (TA# de-asserted) to read data during a burst, the PCI 9656 must read two more Lwords of data before it releases the Local Bus. One during the cycle that EOT# is asserted or when TA# becomes asserted, and one additional Lword with BDIP# de-asserted.

  If EOT# is asserted while waiting to complete a single cycle transfer, the PCI 9656 will complete that transfer and one additional Lword Data transfer before releasing the Local Bus.

Fast Terminate mode (DMAMODE*x*[15]=1), Continuous Burst mode (DMAMODE*x*[8, 7]=1xb)

In Fast Terminate mode (DMAMODE*x*[15]=1), the PCI 9656 defaults into Continuous Burst mode, regardless of the DMA Channel x Continuous Burst Mode bit (DMAMODE*x*[7]) setting. With these settings, the PCI 9656 operates the same as Slow Terminate/ Continuous Burst mode, with the following differences:

- BDIP# is never asserted.
- If DREQ*x*# is de-asserted during a single cycle, the PCI 9656 releases the Local Bus after the current read completes, and may or may not need to execute one additional single cycle read. For a burst, the PCI 9656 will read one additional Lword with DREQ*x*# de-asserted and TA# asserted [or the internal Wait State Counter(s) has decremented to 0] or wait for TA# assertion and read the last data before releasing the Local Bus.
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is starting a transfer to read data or is waiting (TA# de-asserted) to read data during a burst, the PCI 9656 must read one additional Lword of data before it releases the Local Bus. The PCI 9656 will complete the final data transfer if TA# is asserted with EOT# or when TA# becomes asserted after EOT# assertion.

  If EOT# is asserted while waiting to complete a single cycle transfer, the PCI 9656 completes that transfer before releasing the Local Bus.

3.4.4.12 End of Transfer (EOT#) Input

EOT# is a one-clock wide pulse that ends a DMA transfer. It should be asserted only when the PCI 9656 owns the Local Bus. Depending on whether Fast or Slow Terminate mode is selected, the DMA transfer ends without a BDIP# assertion (Fast Terminate mode) or with BDIP# asserted (Slow Terminate mode). The DMAMODE*x*[15:14] bits enable and control how the PCI 9656 responds to an EOT# input assertion.

In Fast Terminate mode, if EOT# is asserted while the PCI 9656 receives an external TA# signal assertion, the DMA Controller(s) ends the DMA transfer and releases the Local Bus. In Slow Terminate mode, if EOT# is asserted while the PCI 9656 receives an external TA# signal assertion, the DMA Controller(s) continues to transfer data to the nearest 16-byte boundary before terminating the DMA transfer. (Refer to Table 3-9.)

If Slow Terminate and Burst-4 modes are enabled (DMAMODE*x*[15, 7]=00b, respectively), and EOT# is asserted during the Data-Transfer phase of the last four bytes of a 16-byte transfer, the PCI 9656 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. (Refer to Table 3-9.)

Regardless of the BI Mode bit(s) setting with the Fast/ Slow Terminate Mode Select disabled (DMAMODE*x* [15, 7]=1xb), the DMA Controller(s) terminates a transfer on an Lword boundary after EOT# is asserted. For an 8-bit bus, the PCI 9656 terminates after transferring the last byte for the Lword. For a 16-bit bus, the PCI 9656 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. EOT# has no effect when loading the descriptor from the PCI Bus.

3.4.4.13 DMA Arbitration

The PCI 9656 asserts BR# when it is necessary for the device to be the Local Bus Master. Upon receiving BG#, the PCI 9656 waits for BB# to be de-asserted. The PCI 9656 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 9656 continues to assert BB# while acting as the Local Bus Master until one of the following conditions occur:

- Local Bus Latency Timer is enabled (MARBR[16]=1) and expires (MARBR[7:0])
- Direct Slave access is pending
- EOT# input is received (if enabled)

The DMA Controller(s) releases control of the PCI Bus when one of the following 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(s) de-asserts PCI REQ# for a minimum of two PCI clocks.

3.4.4.14 Local Bus DMA Priority

The PCI 9656 supports programmable Local Bus arbitration priority for DMA Channel 0 and Channel 1, when both channels are active (priority set with MARBR[20:19]). DMA Block and Scatter/Gather modes have priority over DMA Demand mode. DMA transfer direction does not influence DMA channel priority.

There are three types of priorities:

- **Channel 0 Priority** – DMA Channel 0 completes the transfer on the Local Bus before Channel 1. If Channel 1 is performing a Data transfer, with Channel 0 set as highest priority and started, Channel 1 continues its transfer until the Local Bus Latency Timer (MARBR[7:0]) expires, preempted by a Direct Slave Data transfer, or another termination occurs (EOT# assertion or DREQ*x*# de-assertion). Channel 0 then owns the Local Bus until transfer completion, before Channel 1 can continue the interrupted transfer, unless Channel 1 previously completed its transfer.
- **Channel 1 Priority** – DMA Channel 1 completes its transfer on the Local Bus before Channel 0. If Channel 0 is performing a Data transfer, with Channel 1 set as highest priority and started, Channel 0 continues its transfer until the Local Bus Latency Timer expires, preempted by a Direct Slave Data transfer, or another termination occurs (EOT# assertion or DREQ*x*# de-assertion). Channel 1 then owns the Local Bus until transfer completion, before Channel 0 can continue the interrupted transfer, unless Channel 0 previously completed its transfer.
- **Rotational Priority** – Depends on the transfer direction, however, if the starting bus is the same for both DMA channels, in the freshly started DMA, Channel 0 always starts first. Rotational priority does not start unless the ongoing DMA channel Data transfer is interrupted by the Local Bus Latency Timer expiration, preempted by a Direct Slave Data transfer, or another termination occurs (EOT# assertion or DREQ*x*# de-assertion). The other DMA channel then owns the Local Bus until the previously described interrupts or terminations occur. Rotational priority occurs each time a DMA channel loses Local Bus ownership, unless one of the DMA channels previously completed its transfer.

3.4.4.15 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 (MARBR[16]=1) and expires (MARBR[7:0]), the PCI 9656 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 the PCI 9656 receives BG#, it asserts BB# and continues to transfer until the FIFO is empty for a PCI-to-Local transfer or full for a Local-to-PCI transfer.

The DMA transfer can be paused by writing 0 to the Channel Enable bit(s) (DMACSR*x*[0]=0). To acknowledge the disable, the PCI 9656 receives at least one data from the bus before it stops. However, this is ***not*** recommended during a burst.

The DMA Local Bus Latency Timer starts after the Local Bus is granted to the PCI 9656, and the Local Bus Pause Timer starts after the PCI 9656 de-asserts BB#.

3.4.4.16 DMA FIFO Programmable Threshold

The PCI 9656 supports programmable DMA FIFO threshold (DMATHR). The DMATHR register can be programmed with any of four different FIFO Full/Empty conditions, for DMA Channel 0 and/or Channel 1, as listed in Table 3-11. (Refer to the DMATHR register and Table 11-7, “DMA Threshold Nybble Values,” for further details.)

Table 3-11. DMATHR FIFO Threshold

DMA Transfer	Condition	Description
PCI-to-Local	DMA Channel *x* FIFO Almost Full	Number of full Qword entries (plus 1, times 2) in the FIFO before requesting the Local Bus for writes.
	DMA Channel *x* FIFO Almost Empty	Number of empty Qword entries (plus 1, times 2) in the FIFO before requesting the PCI Bus for reads.
Local-to-PCI	DMA Channel *x* FIFO Almost Full	Number of full Qword entries (plus 1, times 2) in the FIFO before requesting the PCI Bus for writes.
	DMA Channel *x* FIFO Almost Empty	Number of empty Qword entries (plus 1, times 2) in the FIFO before requesting the Local Bus for reads.

3.4.4.17 DMA PCI Master/Target Abort

During PCI Bus mastering and upon encountering non-existing or “bad” devices, the PCI 9656 PCI Master/Target Abort logic enables the PCI 9656 to successfully recover during transfers to and from other PCI devices. As a PCI master, if the PCI 9656 attempts an 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 or Target Abort bit (PCISR[13 or 12]=0, respectively) and continue by processing the next task. Not clearing the bit prevents the PCI 9656 from ever becoming a PCI Bus master again.

If a PCI Master/Target Abort or 256 consecutive Master Retry timeout is encountered during a DMA transfer, the PCI 9656 asserts LINTo#, if enabled (INTCSR[16, 12]=11b, respectively), which can be used as an interrupt. The PCI 9656 also flushes all PCI Bus Master FIFOs (DMA and Direct Master).

The PCI 9656 DMA channels function independently when a PCI Master/Target Abort occurs. If one DMA channel encounters a PCI Master/Target Abort, the FIFO on the aborted channel is flushed. PCI Master capabilities for both DMA channels are disabled until the Received Master/Target Abort bits are cleared (PCISR[13:12]=00b, respectively). However, if the second DMA channel begins a new operation while the first DMA channel is receiving a PCI Master/Target Abort, the FIFO contents of the second DMA channel are not affected, and its new or pending Direct Master operations successfully complete. (For details about PCI Master/Target Abort, refer to Section 3.4.1.11.)

Note: *In applications where both DMA channels are used and one of the DMA channels encounters a PCI Master/Target Abort, the Received Master/Target Abort bits should be cleared (PCISR[13:12]=00b, respectively) and the non-aborted DMA channel should be allowed to complete its Data transfer before the aborted DMA channel’s registers are re-initialized and the transfer re-started.*

If a PCI Master/Target Abort is encountered during a DMA transfer, the PCI 9656 stores the PCI Abort Address into PABTADR[31:0]. PABTADR contents are updated for each PCI Master/Target Abort. The Received Master/Target Abort bits should be read and cleared before starting a new DMA or Direct Master, Type 0 or Type 1 Configuration transfer.

Table 3-12 outlines special PCI Master/Target Abort conditions for DMA mode Data transfers.

Table 3-12. DMA Mode Data Transfer Special PCI/Master/Target Abort Conditions

DMA Mode Data Transfer Type	PCI Master/Target Abort Condition
DMA Local-to-PCI in Slow Terminate Mode (DMAMODE*x*[15]=0	After encountering a PCI Master/Target Abort, the PCI 9656 immediately terminates data transfer on the PCI Bus and continues reading additional data from the Local Bus into the DMA Local-to-PCI FIFO until its full or transfer count is reached. (Refer to the DMATHR register.) The PCI 9656 then sets the DMA Channel Done bit (DMACSR*x*[4]=1) and the Received Master/Target Abort bits can be cleared (PCISR[13:12]=00b, respectively).
DMA Local-to-PCI in Fast Terminate Mode (DMAMODE*x*[15]=1)	After encountering a PCI Master/Target Abort, the PCI 9656 immediately terminates data transfer on the PCI and Local Buses and sets the DMA Channel Done bit (DMACSR*x*[4]=1).
DMA PCI-to-Local Independent of Fast/Slow Terminate Mode (DMAMODE*x*[15]=0 or 1)	After encountering a PCI Master/Target Abort on the PCI Bus, the PCI 9656 immediately terminates a Burst Data transfer on the Local Bus and single cycles data from the DMA PCI-to-Local FIFO until it is empty.
DMA Local-to-PCI, with EOT# assertion on the Local Bus	Upon encountering a PCI Master/Target Abort on the PCI Bus with EOT# assertion on the Local Bus, the PCI 9656 terminates data transfer on the PCI and Local Buses and sets the DMA Channel Done bit (DMACSR*x*[4]=1).
DMA PCI-to-Local, with EOT# assertion on the Local Bus	Upon encountering a PCI Master/Target Abort on the PCI Bus with EOT# assertion on the Local Bus, the PCI 9656 flushes the DMA PCI-to-Local FIFO before encountering the PCI Master/Target Abort.
DMA PCI-to-Local Scatter/Gather, Ring Management with EOT# assertion on the Local Bus	With the EOT# function enabled (DMAMODE*x*[14]=1), EOT# End Link enabled (DMAMODE*x*[19]=1), and the DMA Descriptor Links located on the Local Bus, the PCI 9656 starts the DMA Descriptor load and then transfers data after the DMA Channel Enable and Start bits are set (DMACSR*x*[0:1]=11b, respectively). Upon receiving the PCI Target Abort on the PCI Bus and EOT# assertion on the Local Bus during a Scatter/Gather DMA PCI-to-Local data transfer, the PCI 9656 pauses the transfer in response to the PCI Target Abort. The PCI 9656 then flushes the DMA FIFO and sets the DMA Channel Done bit (DMACSR*x*[4]=1) after sampling EOT# asserted. No further DMA Descriptor load is performed after the Received Target Abort bit is cleared (PCISR[12]=0), and the Scatter/Gather DMA transfer is terminated.

THIS PAGE INTENTIONALLY LEFT BLANK.

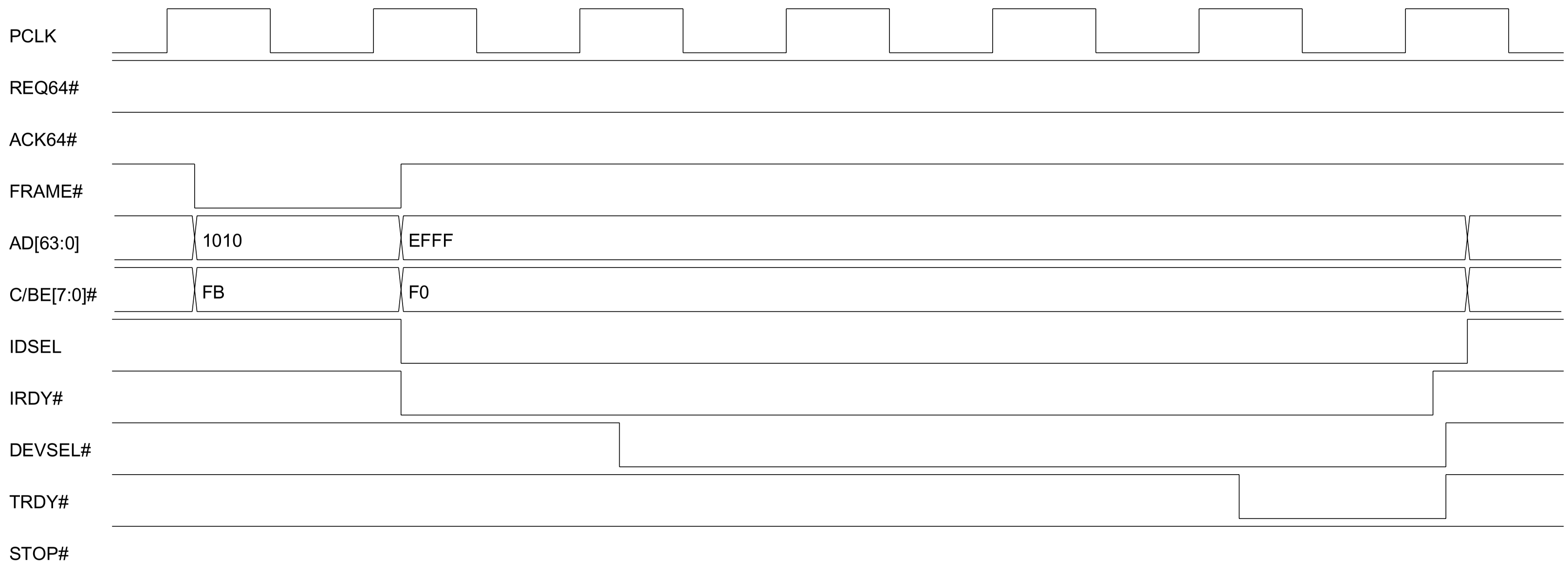
3.5 M MODE FUNCTIONAL TIMING DIAGRAMS

General notes about timing designer (graphical) waveforms:
The graphical Timing Diagrams were created using the Timing Designer tool. They are accurate and adhere to their specified protocol(s). These diagrams show transfers and signal transitions, but should not be relied upon to show exactly, on a clock-for-clock basis, where PCI 9656-driven signal transitions will occur.

General notes about captured waveforms:
The captured Timing Diagrams were captured from a simulation signal display tool that displays the results of stimulus run on the netlist. Using the netlist illustrates realistic delay on signals driven by the PCI 9656. Signals driven by the test environment to the PCI 9656 are quite fast. Leading zeros for buses such as AD[63:0] or LD[0:31], may or may not be shown. If no value for a bus is shown while the PCI 9656 is executing a transfer, it is because the entire value cannot be displayed in the available space or is irrelevant. When the PCI 9656 is not executing a transfer on that bus, the value is not shown because it is either irrelevant or unknown (any or all signals may be 1, 0, or Z). For these waveforms, the external Local Bus Arbiter "parks" the PCI 9656 on the Local Bus (BG# asserted throughout) or grants and "parks" the Local Bus to the PCI 9656 upon detecting that the PCI 9656 has requested the Local Bus by asserting BR#.

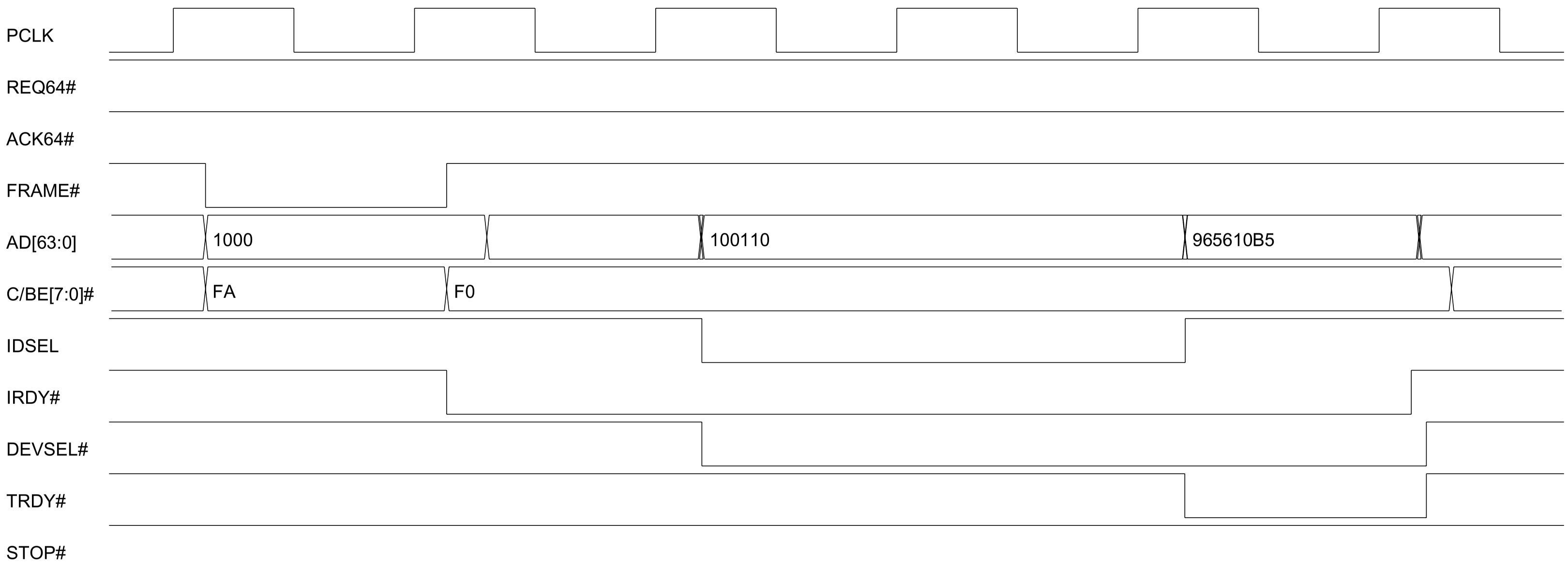
3.5.1 Configuration Timing Diagrams

Timing Diagram 3-1. PCI Configuration Write to PCI Configuration Register



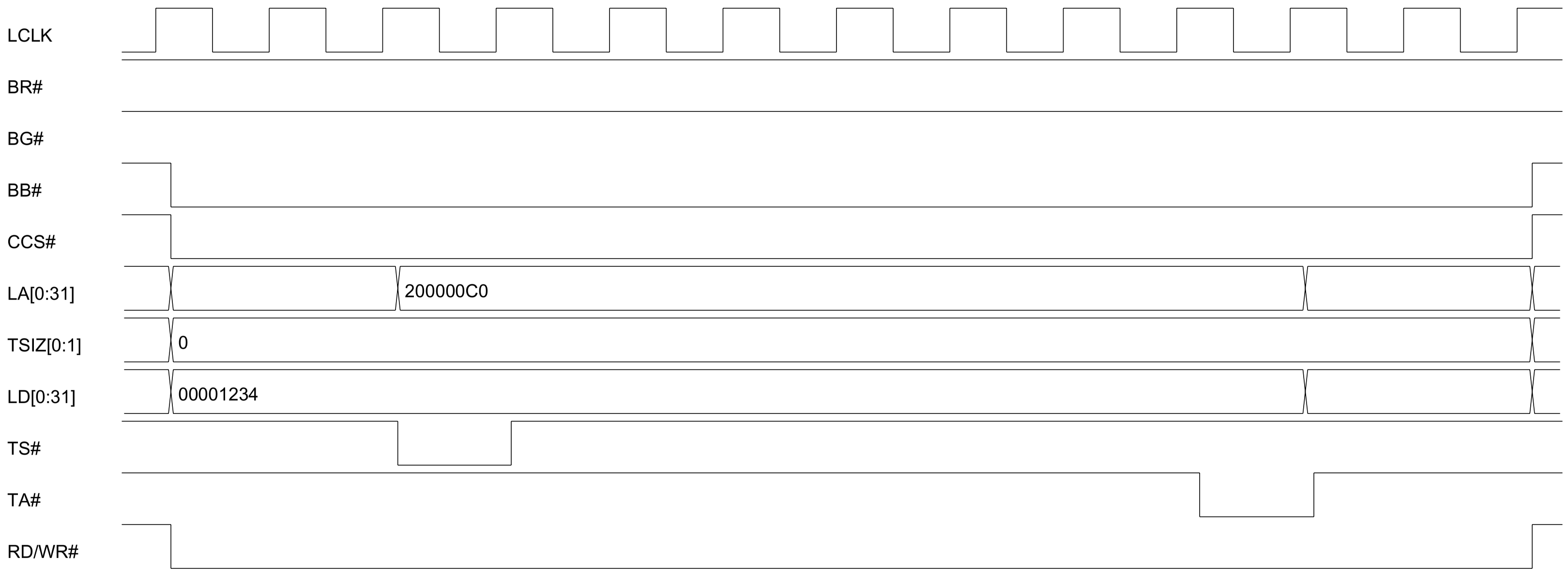
Notes: *PCI Configuration write to PCIBAR0 (PCI:10h) register.*
IDSEL = AD[12].
Leading zeros for AD[63:0] bus are not shown.

Timing Diagram 3-2. PCI Configuration Read of PCI Configuration Register



Notes: *PCI Configuration read of PCIIDR (PCI:00h) register.*
IDSEL = AD[12].
Leading zeros for AD[63:0] bus are not shown.

Timing Diagram 3-3. Local Write to Configuration Register

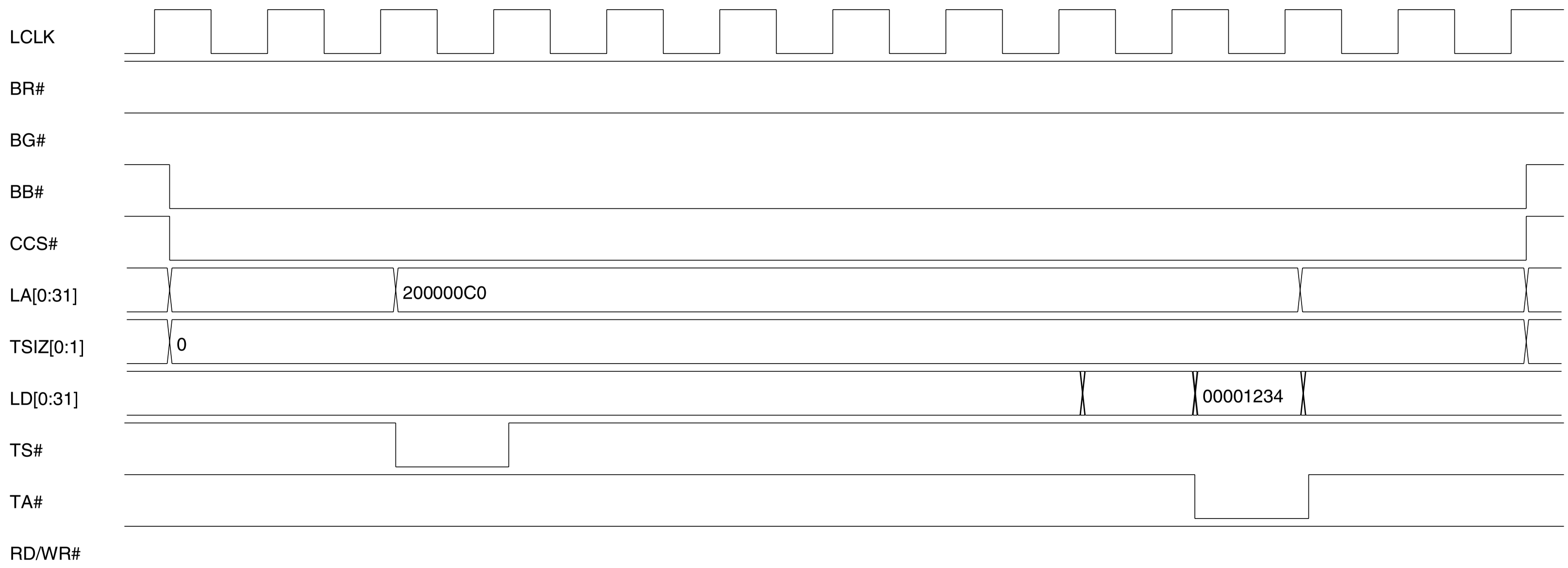


Notes: *Local Configuration write to MBOX0 (LOC:C0h) register.*

CCS# is asserted for multiple clock cycles, but it is required to be asserted only during the TS# clock cycle.

Only the LA[23:29] signals are used to decode the MBOX0 register.

Timing Diagram 3-4. Local Read of Configuration Register



Notes: *Local Configuration read of MBOX0 (LOC:C0h) register.*

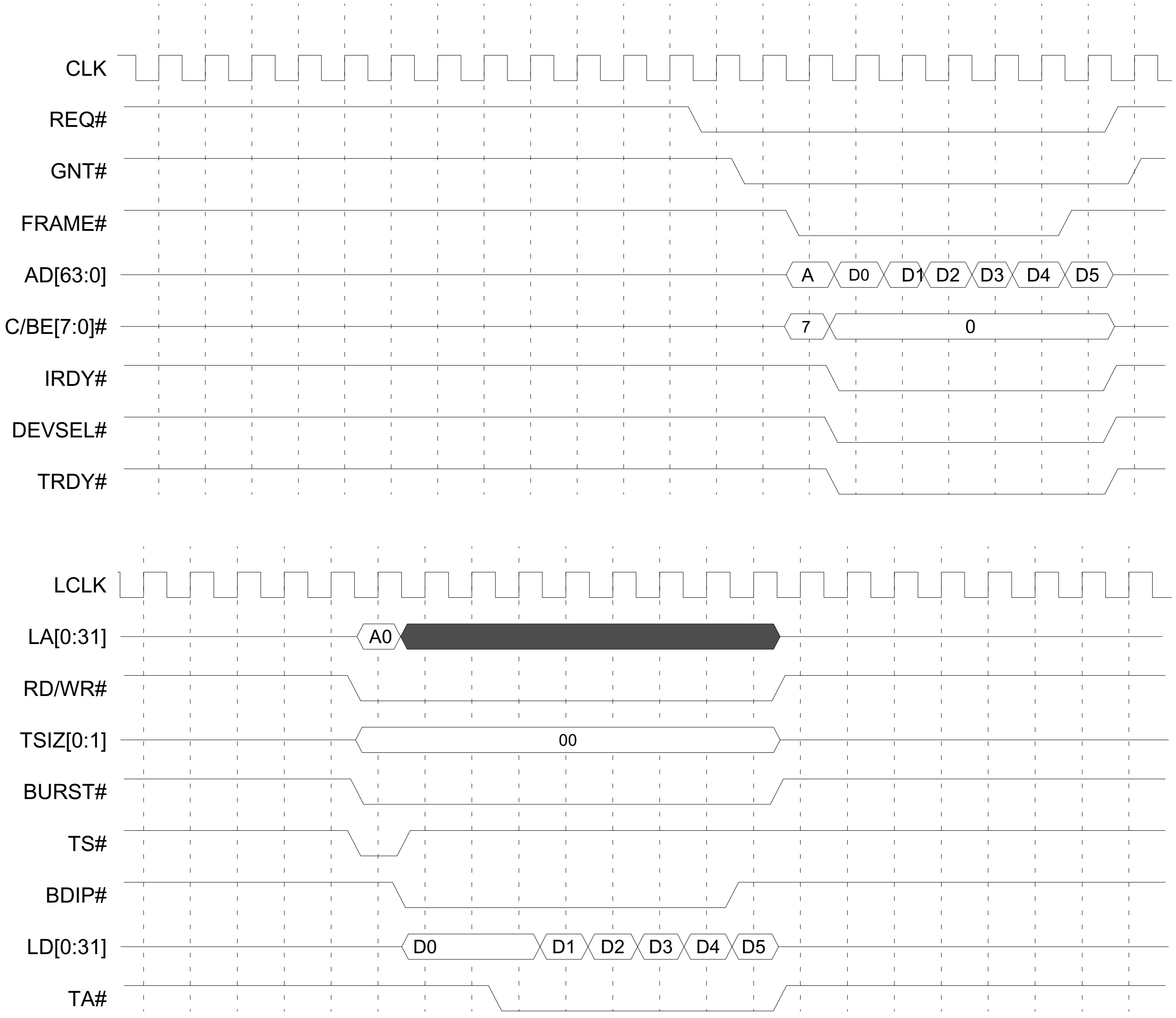
CCS# is asserted for multiple clock cycles, but it is required to be asserted only during the TS# clock cycle.

Only the LA[23:29] signals are used to decode the MBOX0 register.

The PCI 9656 starts driving the LD[0:31] bus with meaningless data one cycle before it asserts the TA# signal with the data read from the MBOX0 register.

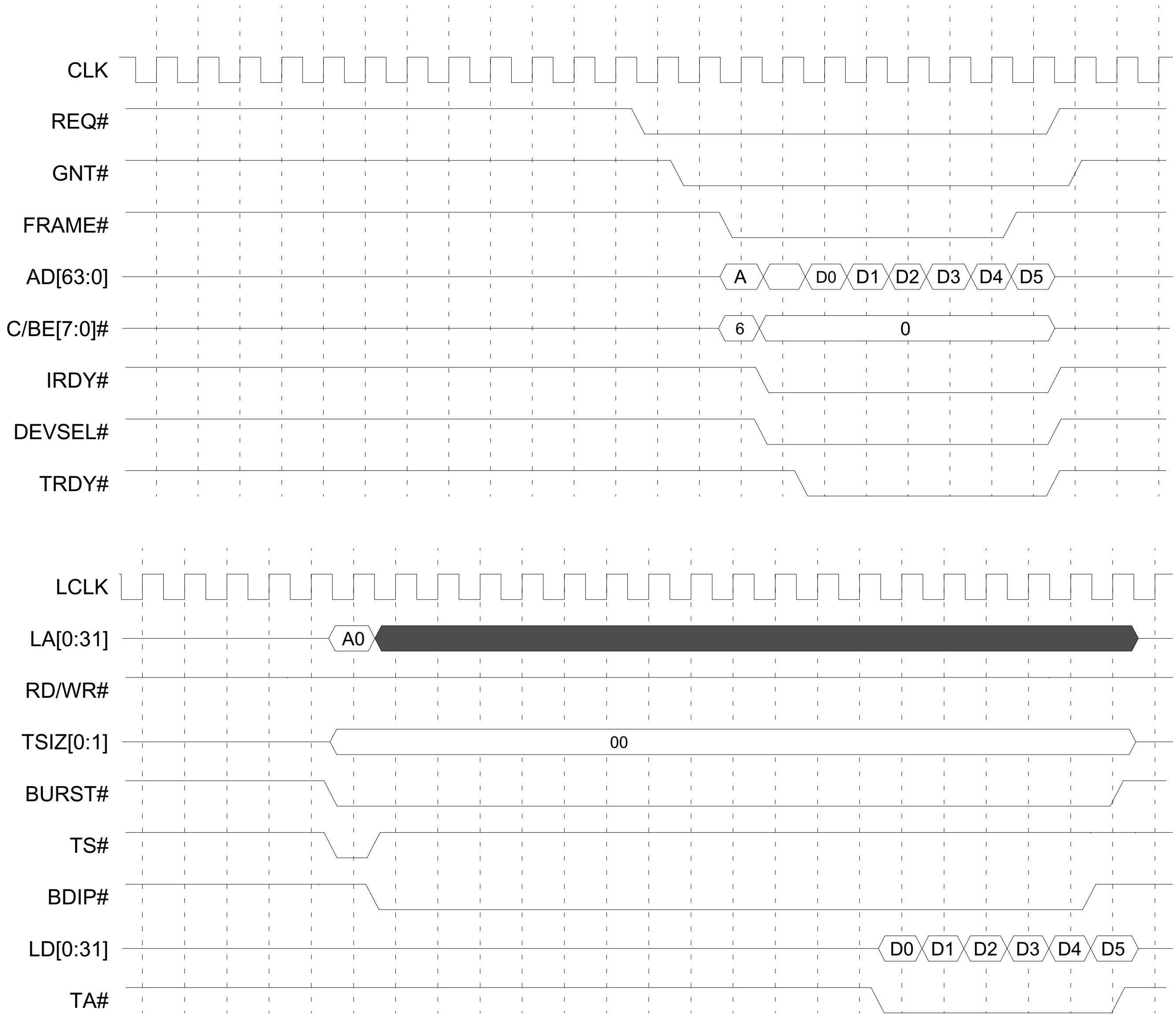
3.5.2 M Mode Direct Master Timing Diagrams

Timing Diagram 3-5. Direct Master Burst Write of 6 Lwords, Continuous Burst Mode



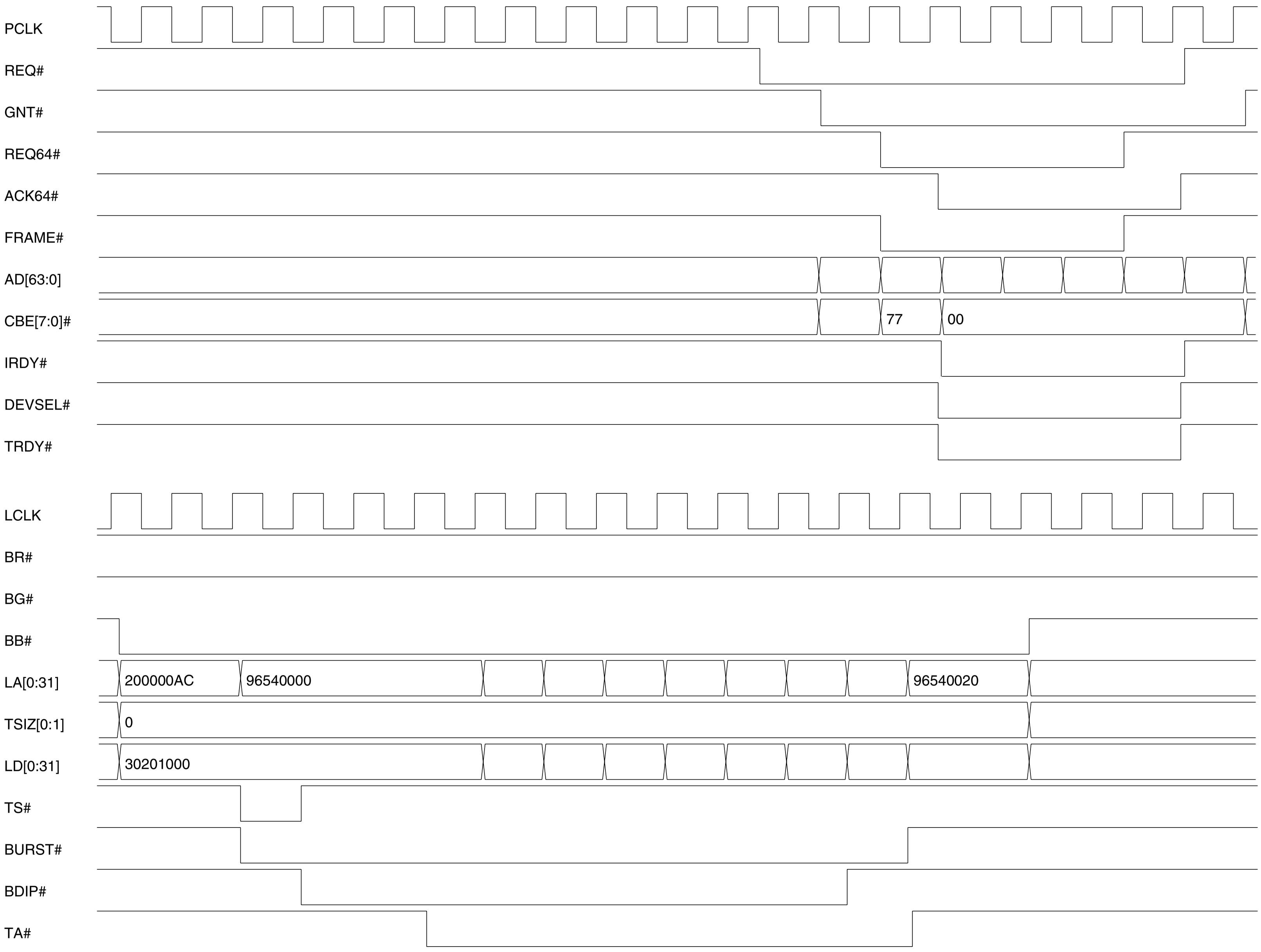
***Note:** In Timing Diagram 3-5, the PCI 9656 is configured to be a 32-bit PCI device [REQ64# de-asserted when RST# transitions high (not shown)].*

Timing Diagram 3-6. Direct Master Burst Read of 6 Lwords, Continuous Burst Mode



Note: *In Timing Diagram 3-5, the PCI 9656 is configured to be a 32-bit PCI device [REQ64# de-asserted when RST# transitions high (not shown)].*

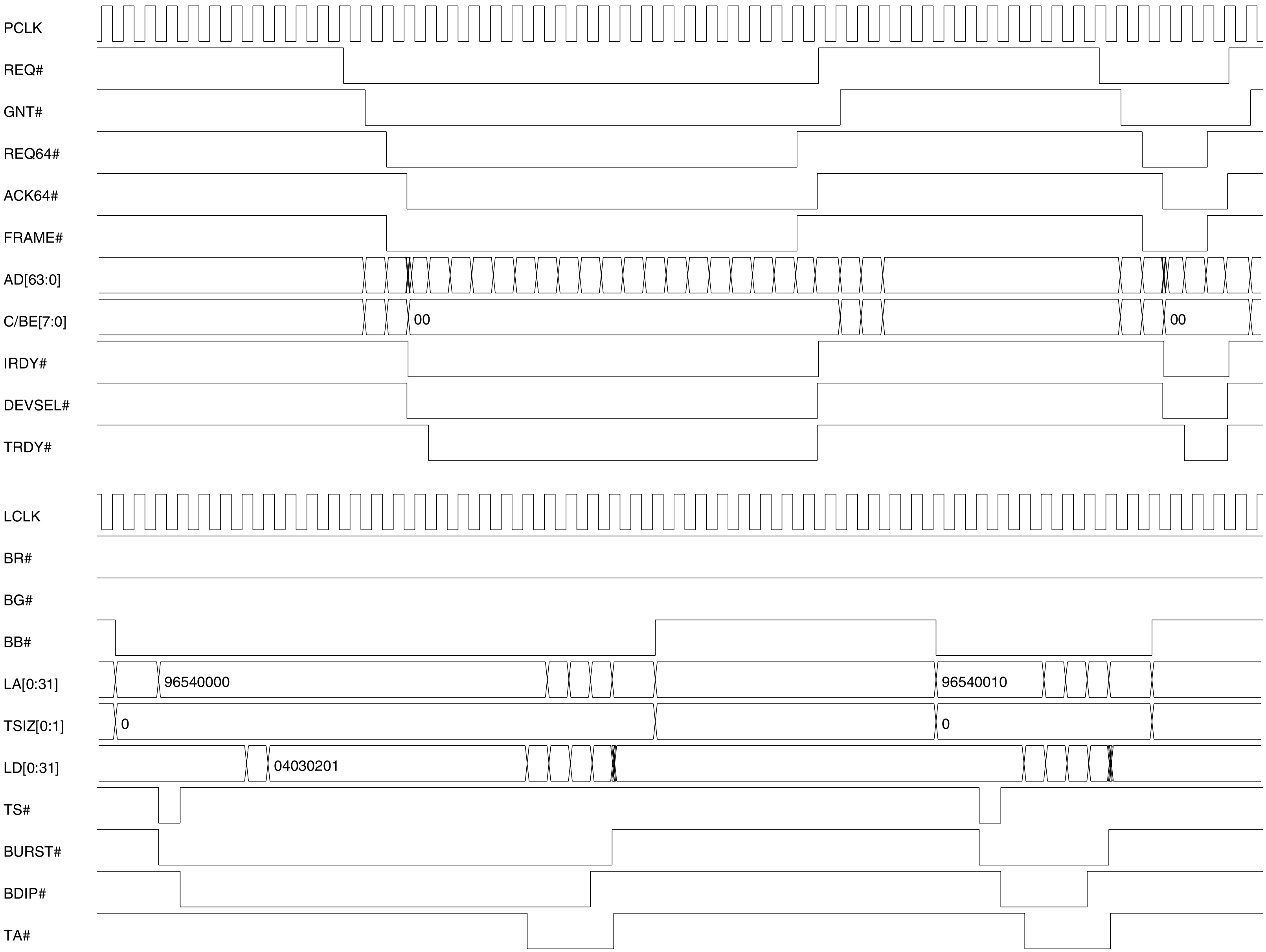
Timing Diagram 3-7. Direct Master Burst Write of 8 Lwords (64-Bit PCI Bus)



Notes: *Key register value is DMPBAM[15:0]=0007h.*

The PCI STOP# signal and M mode RD/WR# signal are equal to 1 and 0, respectively, for the duration of this simulation, but are not shown.

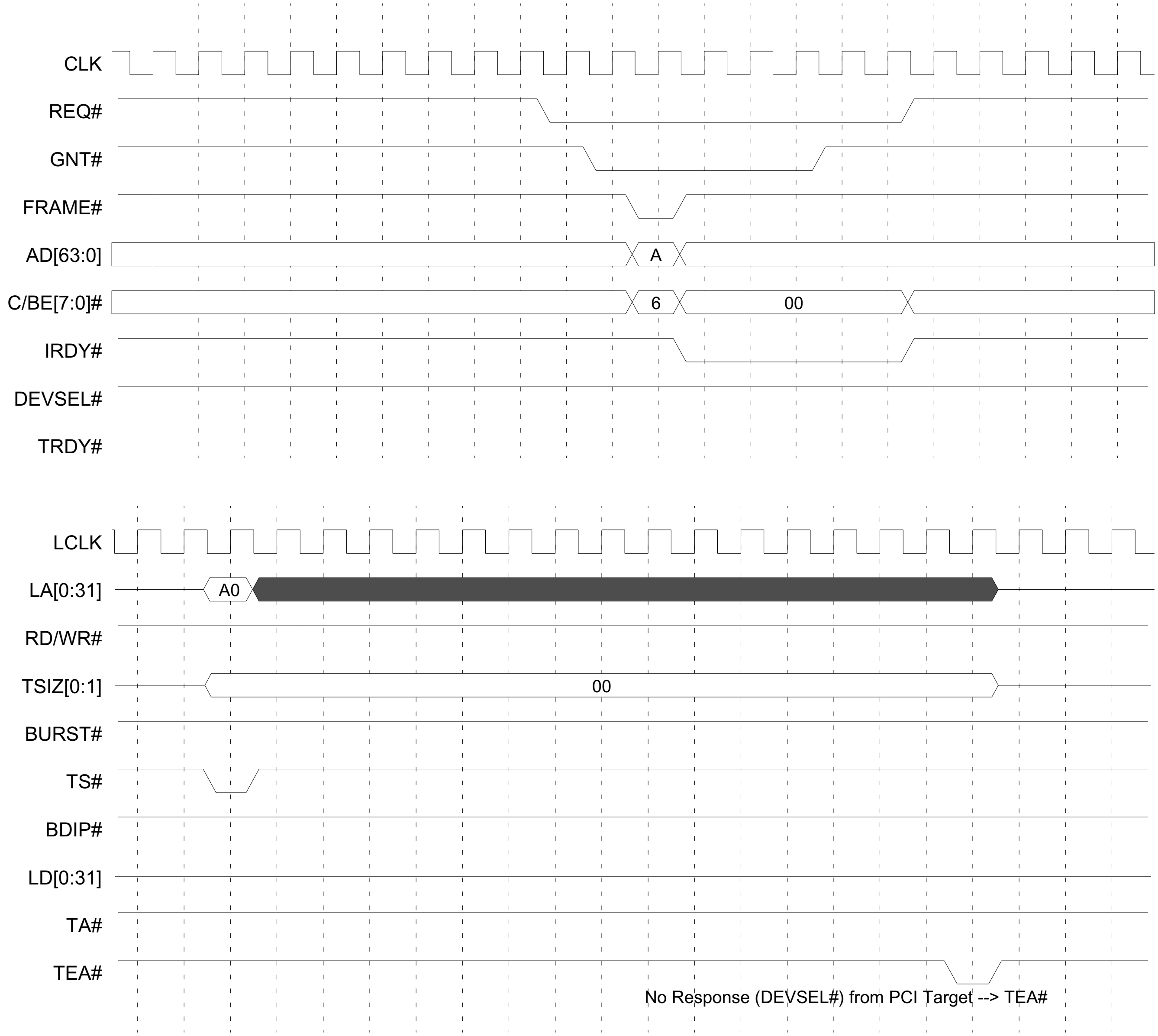
Timing Diagram 3-8. Direct Master Burst Read of 8 Lwords (64-Bit PCI Bus)



Notes: *Key register value is DMPBAM[15:0]=0007h.*

The PCI STOP# signal and M-mode RD/WR# signal are both equal to 1 for the duration of this simulation, but are not shown.

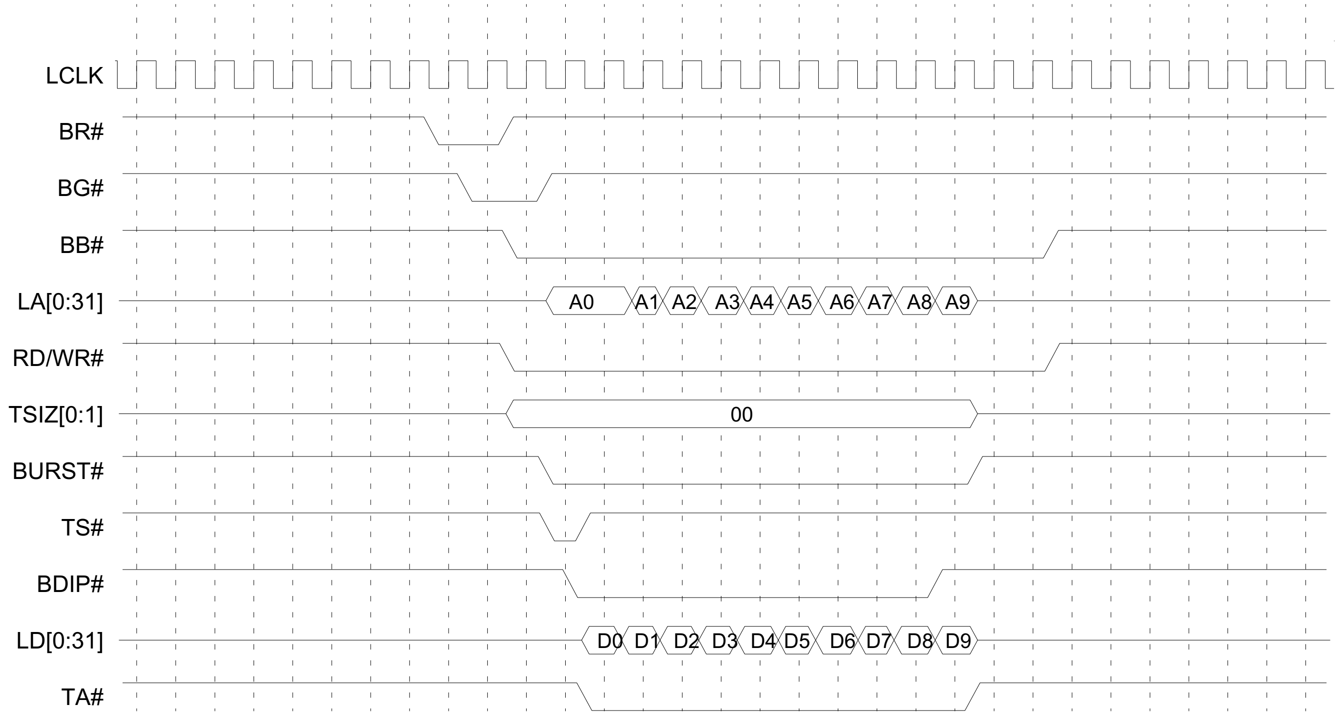
Timing Diagram 3-9. TEA# Assertion Caused by Direct Master Abort during Direct Master Single Cycle Read



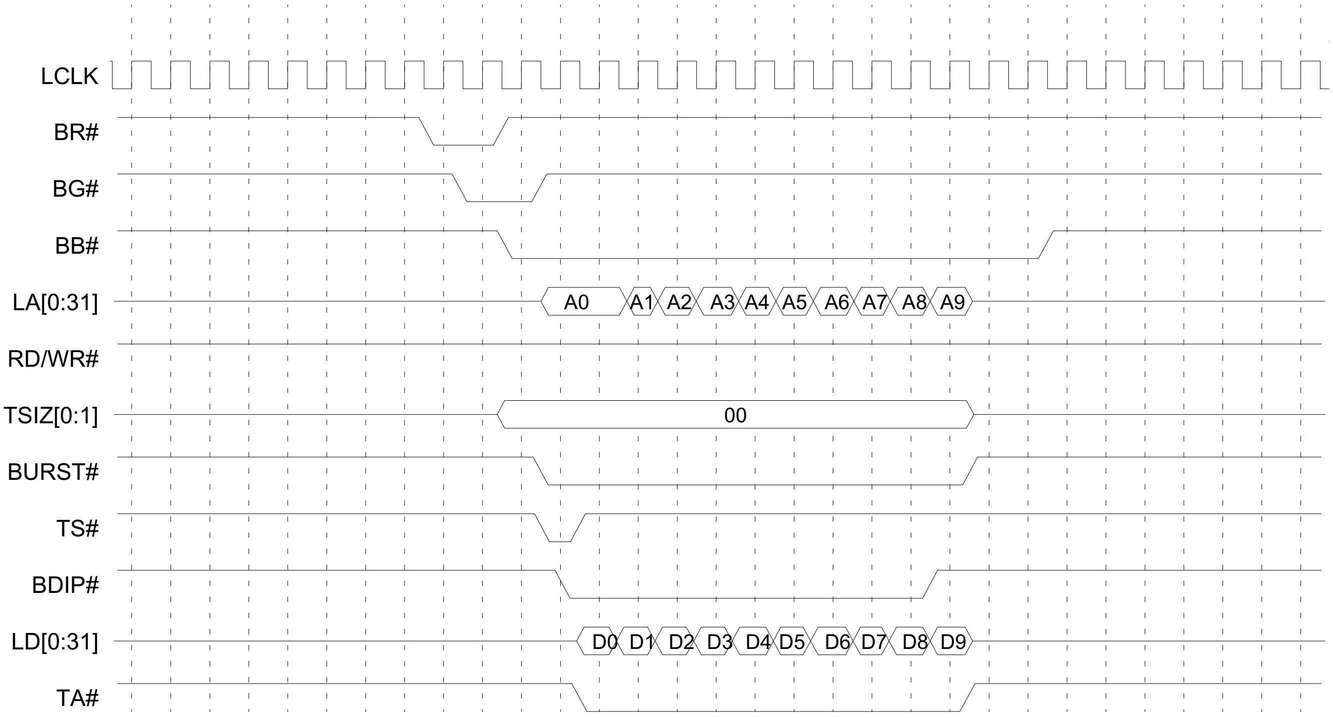
***Note:** In Timing Diagram 3-5, the PCI 9656 is configured to be a 32-bit PCI device [REQ64# de-asserted when RST# transitions high (not shown)].*

3.5.3 M Mode Direct Slave Timing Diagrams

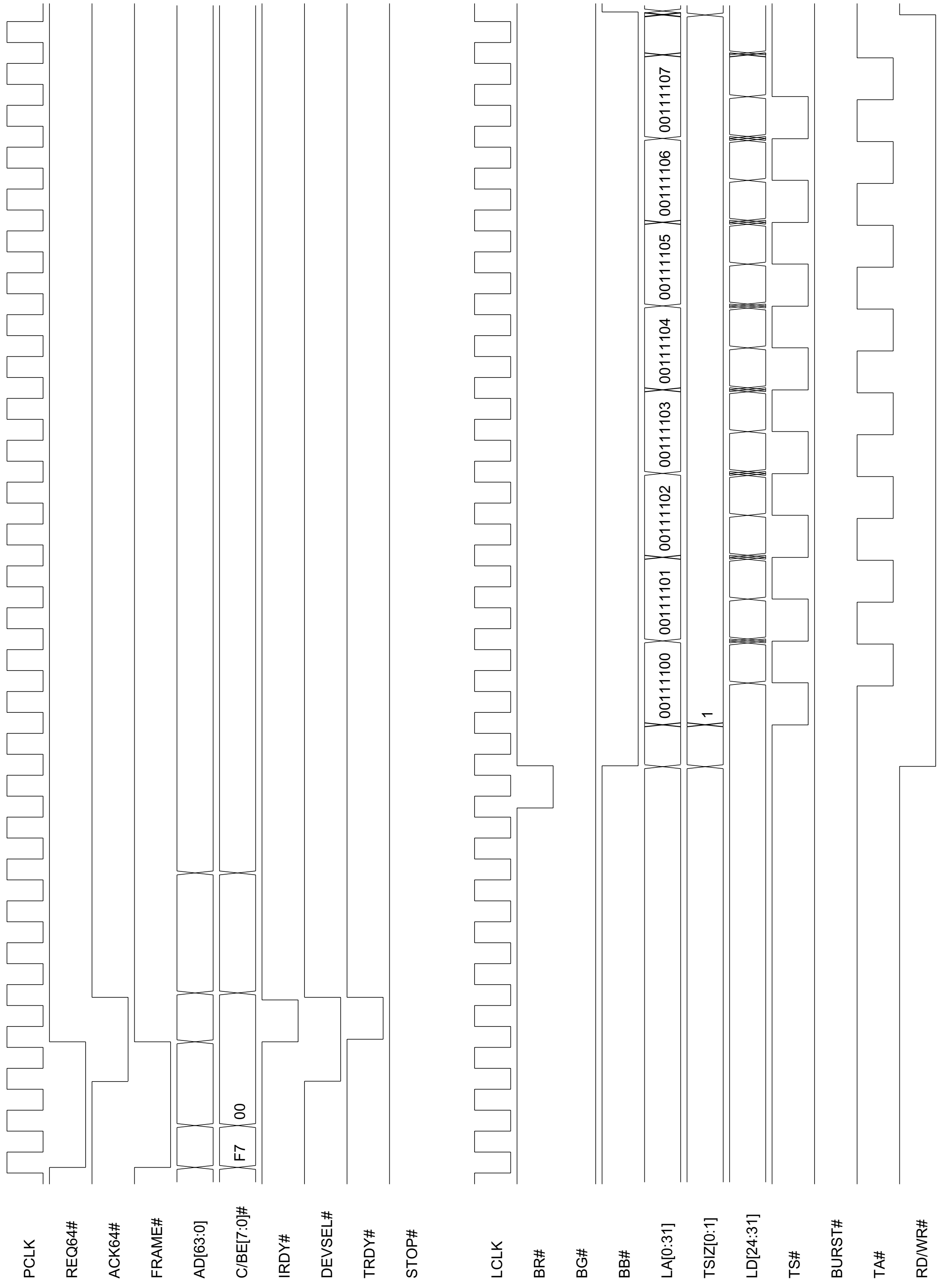
Timing Diagram 3-10. Direct Slave Burst Write of 10 Lwords, Zero Wait States, Continuous Burst Mode



Timing Diagram 3-11. Direct Slave Burst Read of 10 Lwords, Zero Wait States, Continuous Burst Mode

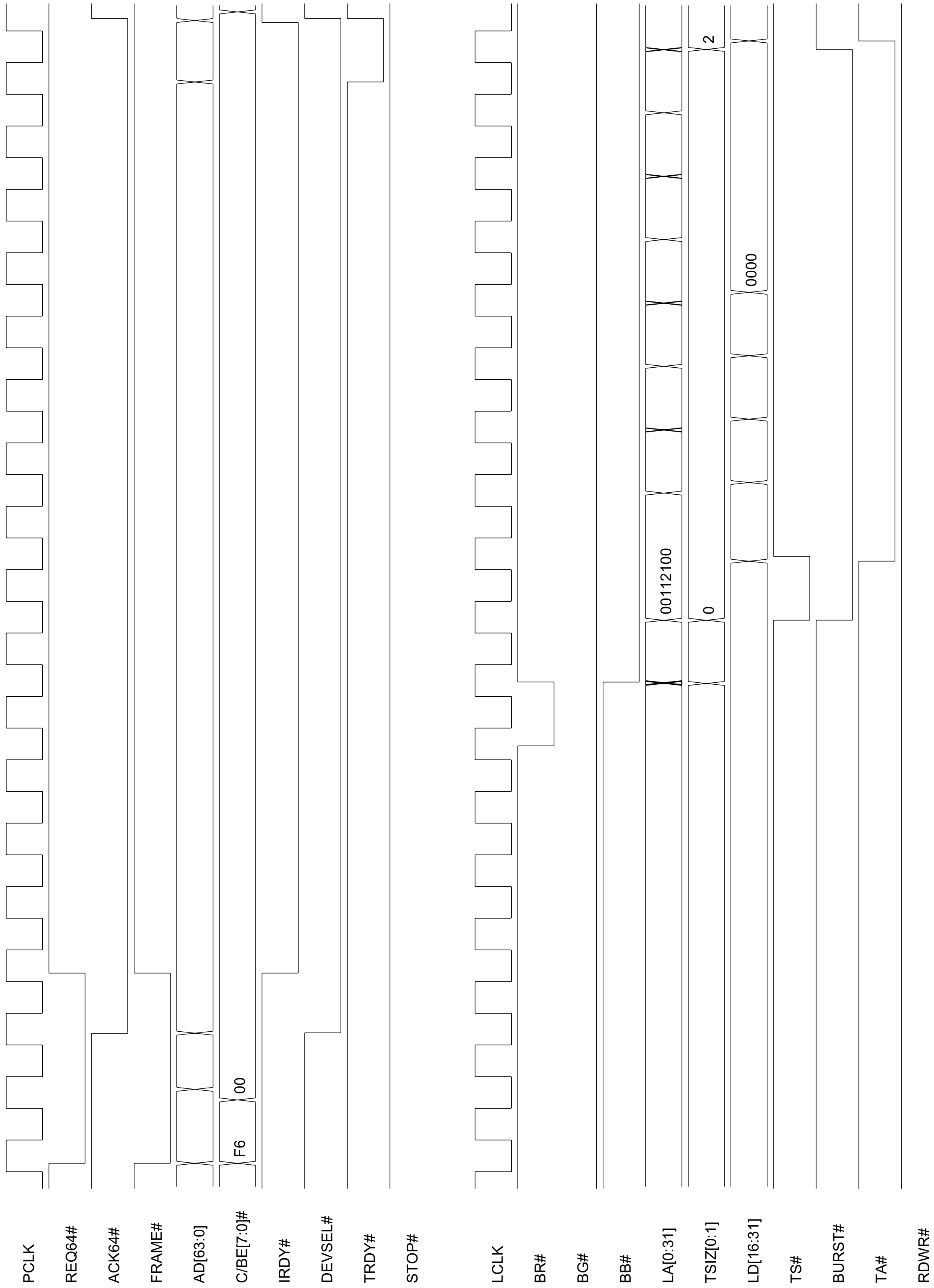


Timing Diagram 3-12. Direct Slave 64-Bit Single Cycle Write (8-Bit Local Bus)



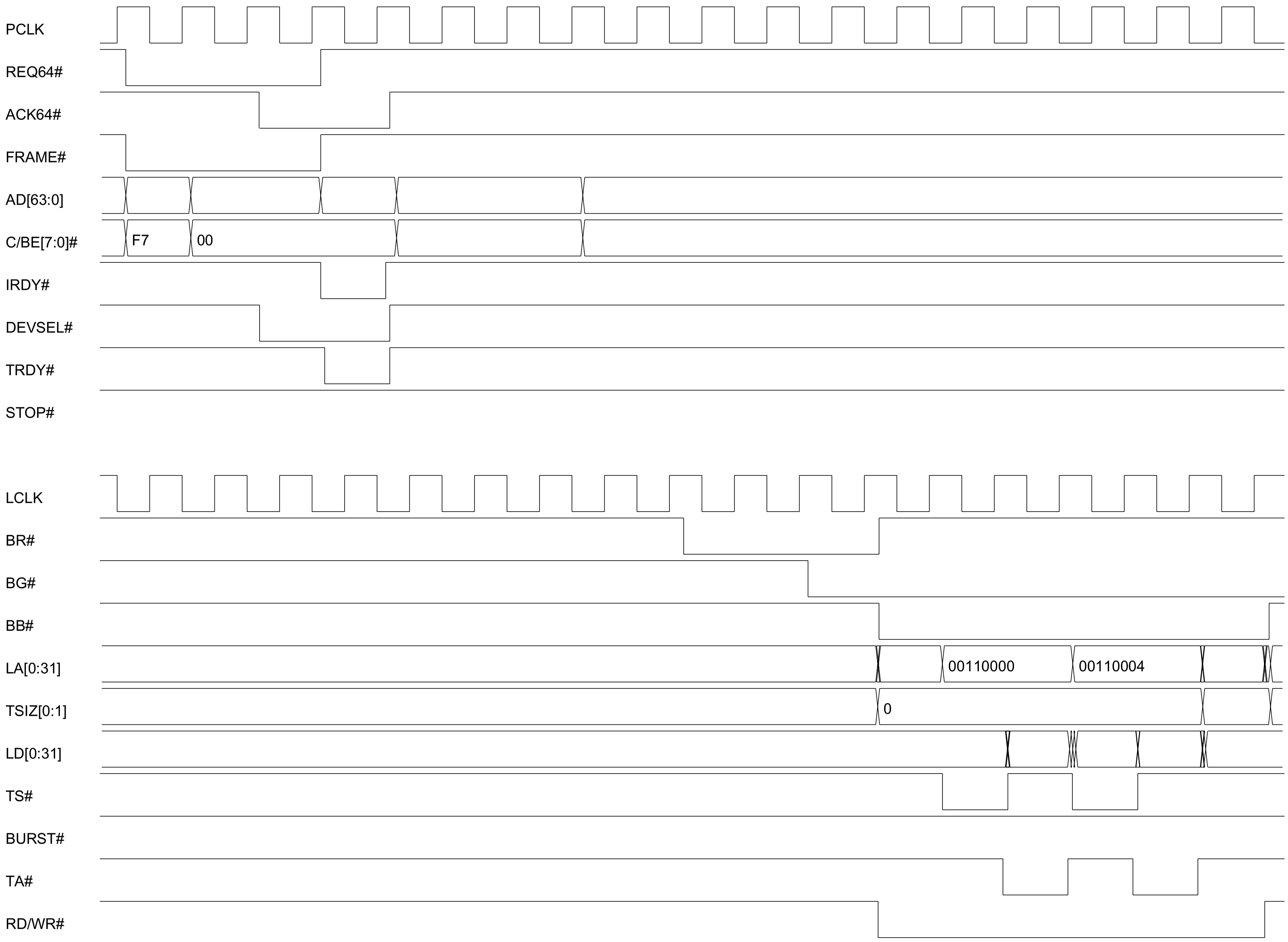
Notes: *Key bit/register values are MARBR[24]=1, BIGEND[4]=0, and LBRD1[15:0]=1540h.*
The BDIP# signal (not shown) is de-asserted for the duration of this timing diagram.

Timing Diagram 3-13. Direct Slave 64-Bit Single Cycle Write (16-Bit Local Bus)



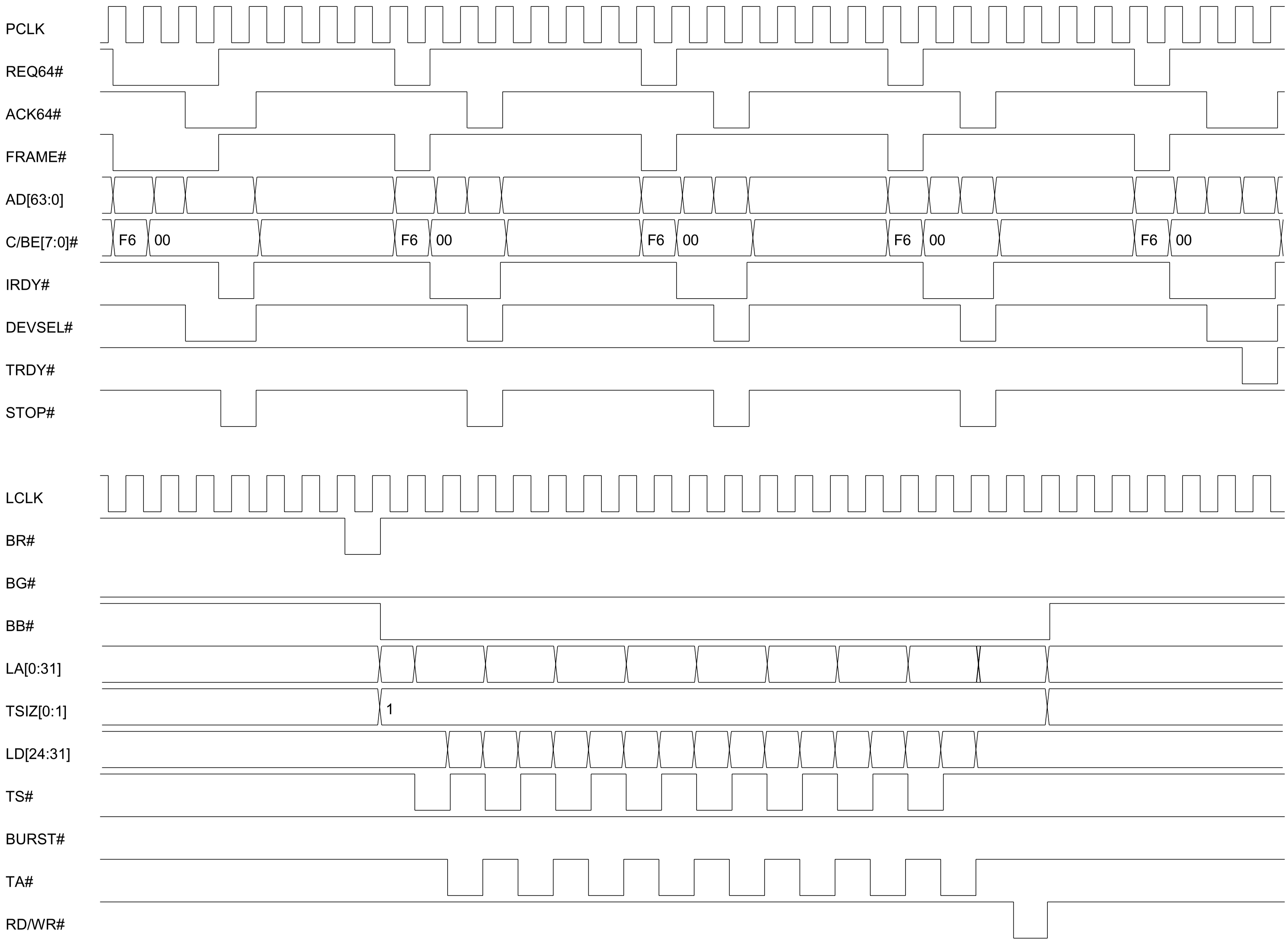
Notes: *Key bit/register values are MARBR[24]=0, BIGEND[4]=0, and LBRD1[15:0]=2541h.*
The BDIP# signal (not shown) is de-asserted for the duration of this timing diagram.

Timing Diagram 3-14. Direct Slave 64-Bit Single Cycle Write (32-Bit Local Bus)



Notes: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=1542h.*
The BDIP# signal (not shown) is de-asserted for the duration of this timing diagram.

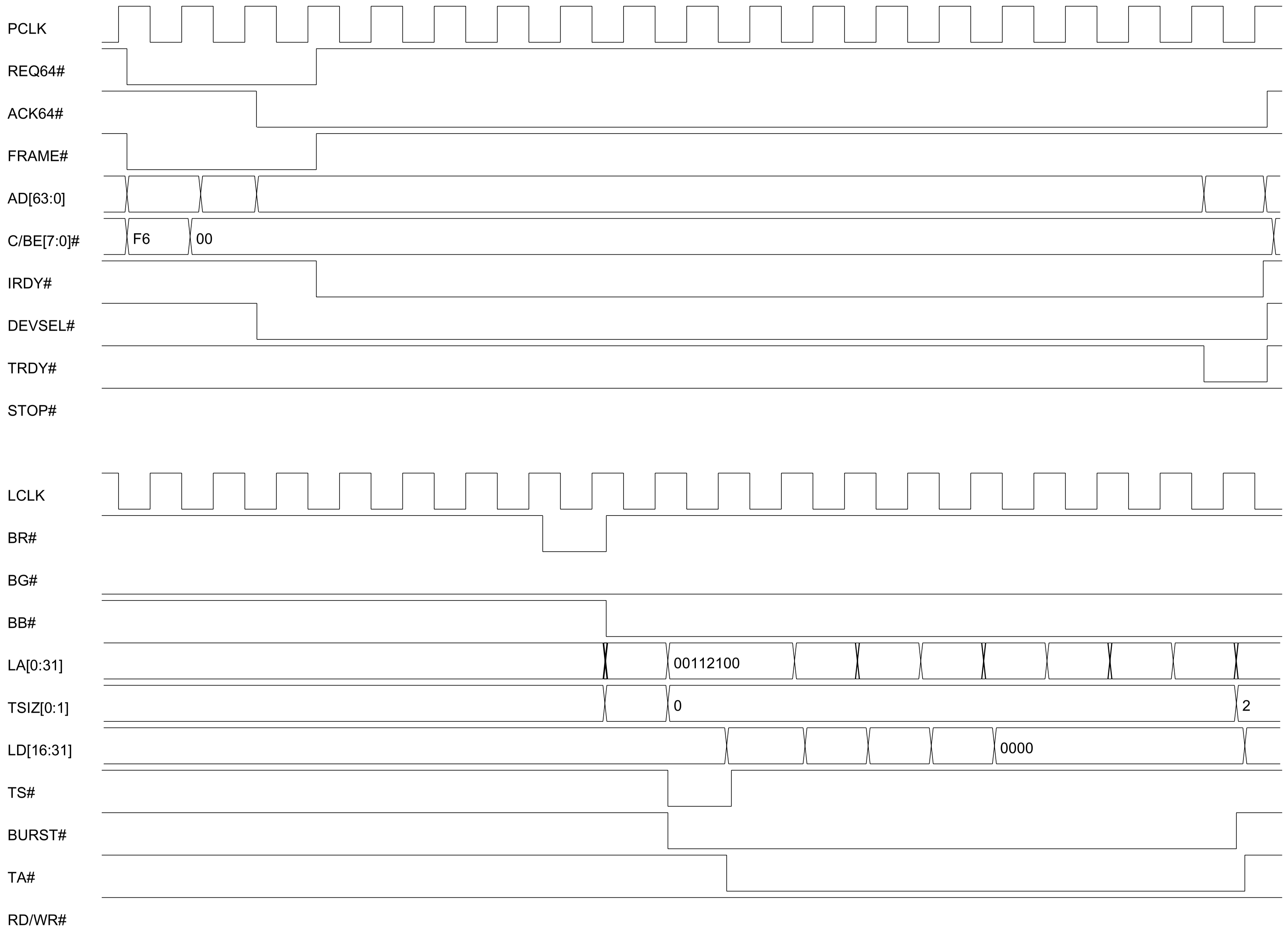
Timing Diagram 3-15. Direct Slave 64-Bit Single Cycle Read (8-Bit Local Bus)



Notes: *Key bit/register values are MARBR[24]=1, BIGEND[4]=0, and LBRD1[15:0]=1540h.*

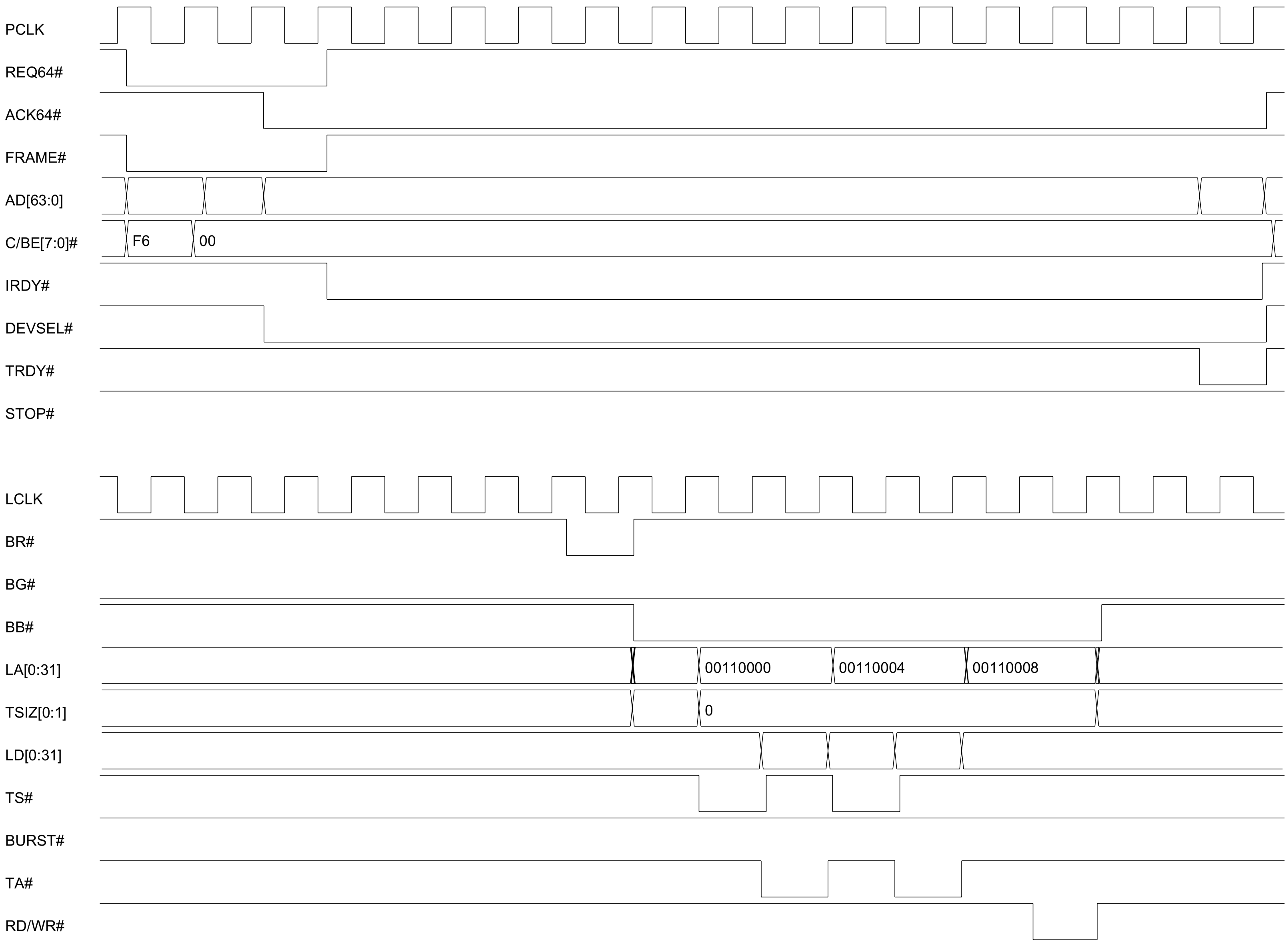
The BDIP# signal (not shown) is de-asserted for the duration of this timing diagram.

Timing Diagram 3-16. Direct Slave 64-Bit Single Cycle Read (16-Bit Local Bus)



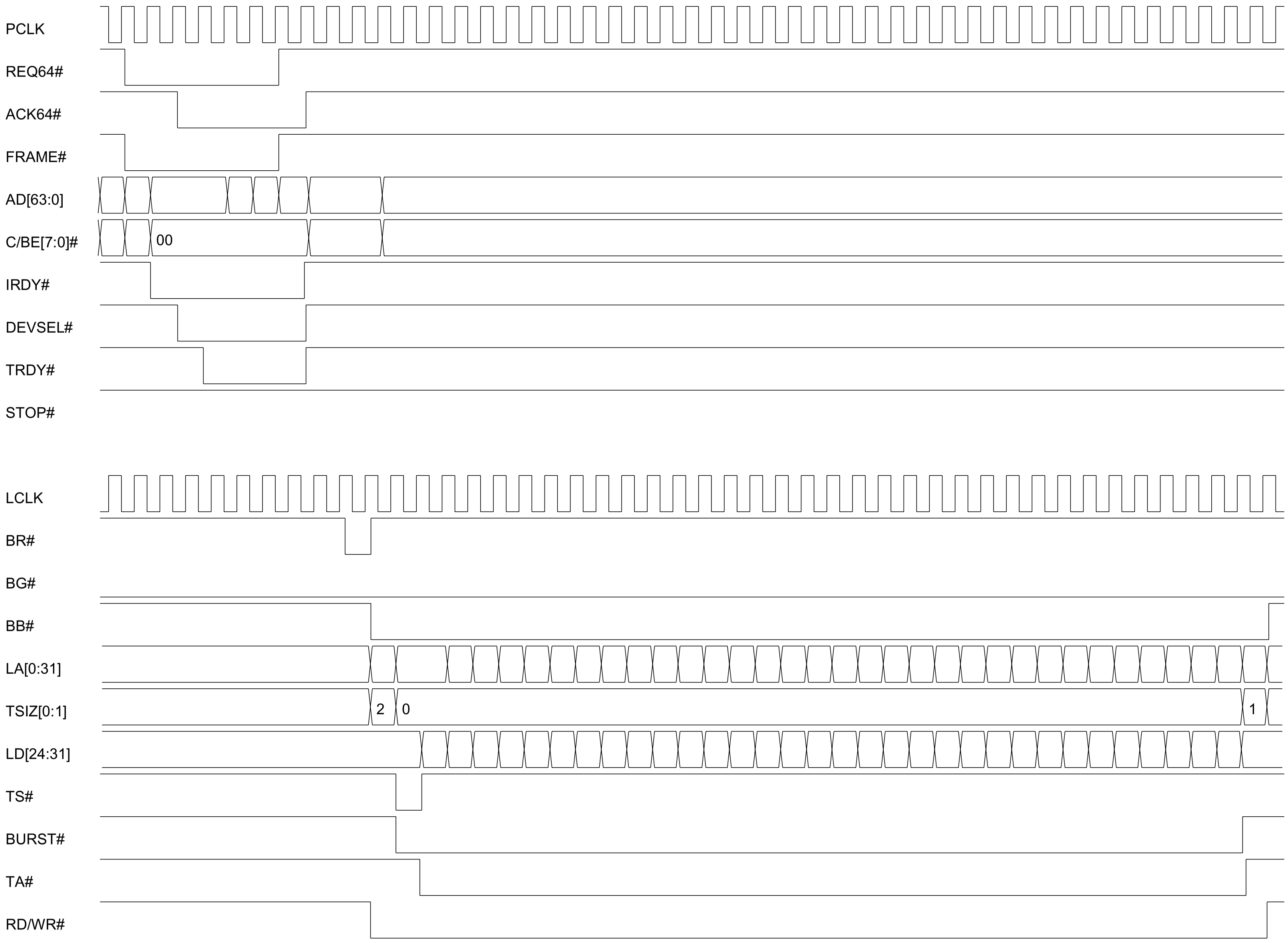
Notes: *Key bit/register values are MARBR[24]=0, BIGEND[4]=0, and LBRD1[15:0]=2541h. With prefetch enabled, LBRD1[9]=0 and LBRD1[14:11]=0100b, two additional Lwords of data are prefetched. The BDIP# signal (not shown) is asserted one clock after BURST# is asserted and de-asserted one clock before BURST# is de-asserted.*

Timing Diagram 3-17. Direct Slave 64-Bit Single Cycle Read (32-Bit Local Bus)



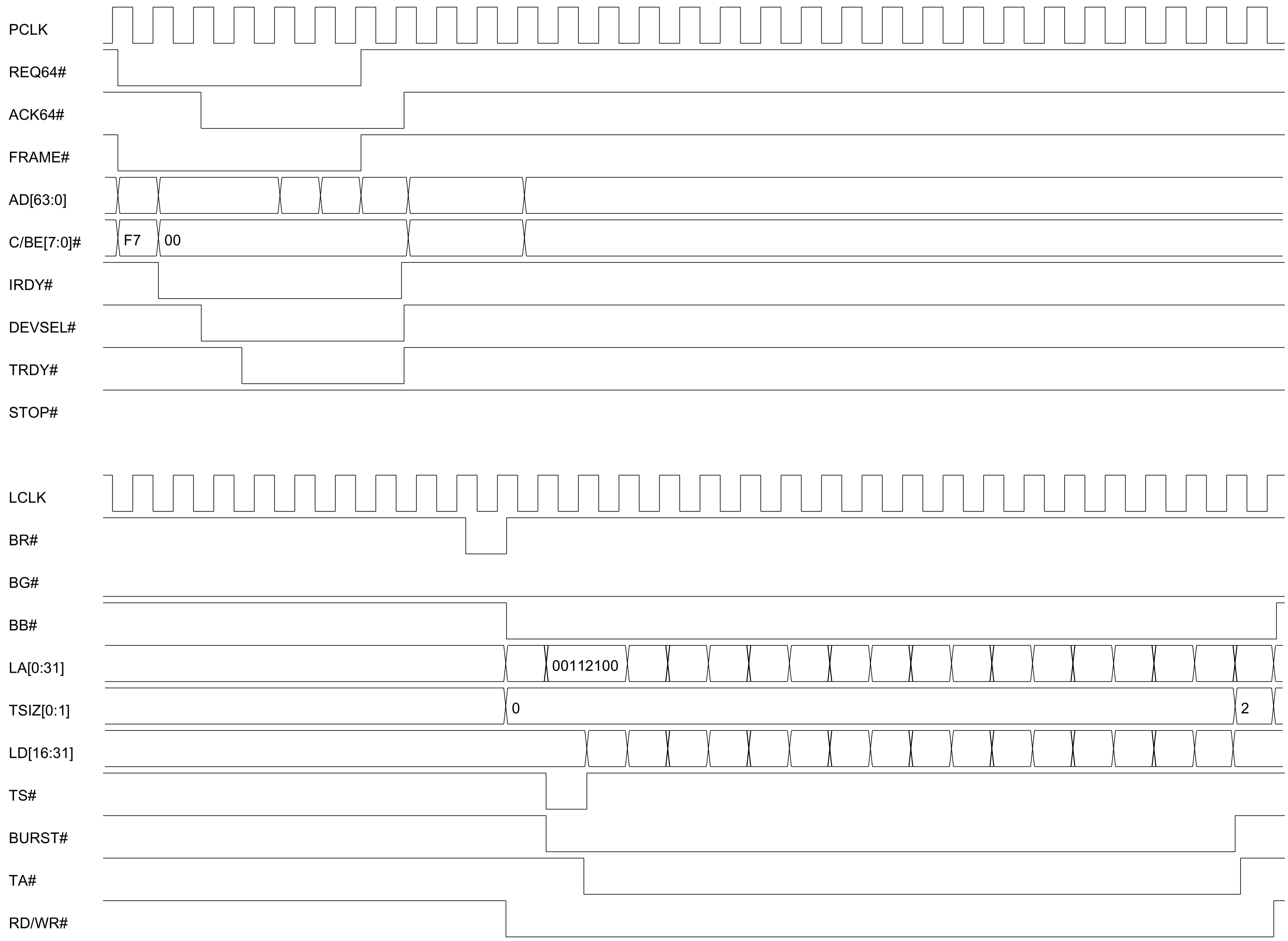
Notes: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=1542h.*
The BDIP# signal (not shown) is de-asserted for the duration of this timing diagram.

Timing Diagram 3-18. Direct Slave 64-Bit Burst Write of 4 Qwords (8-Bit Local Bus)



Notes: *Key bit/register values are MARBR[24]=0, BIGEND[4]=0, and LBRD1[15:0]=45C0h.*
The BDIP# signal (not shown) is asserted one clock after BURST# assertion and de-asserted one clock before BURST# de-assertion.

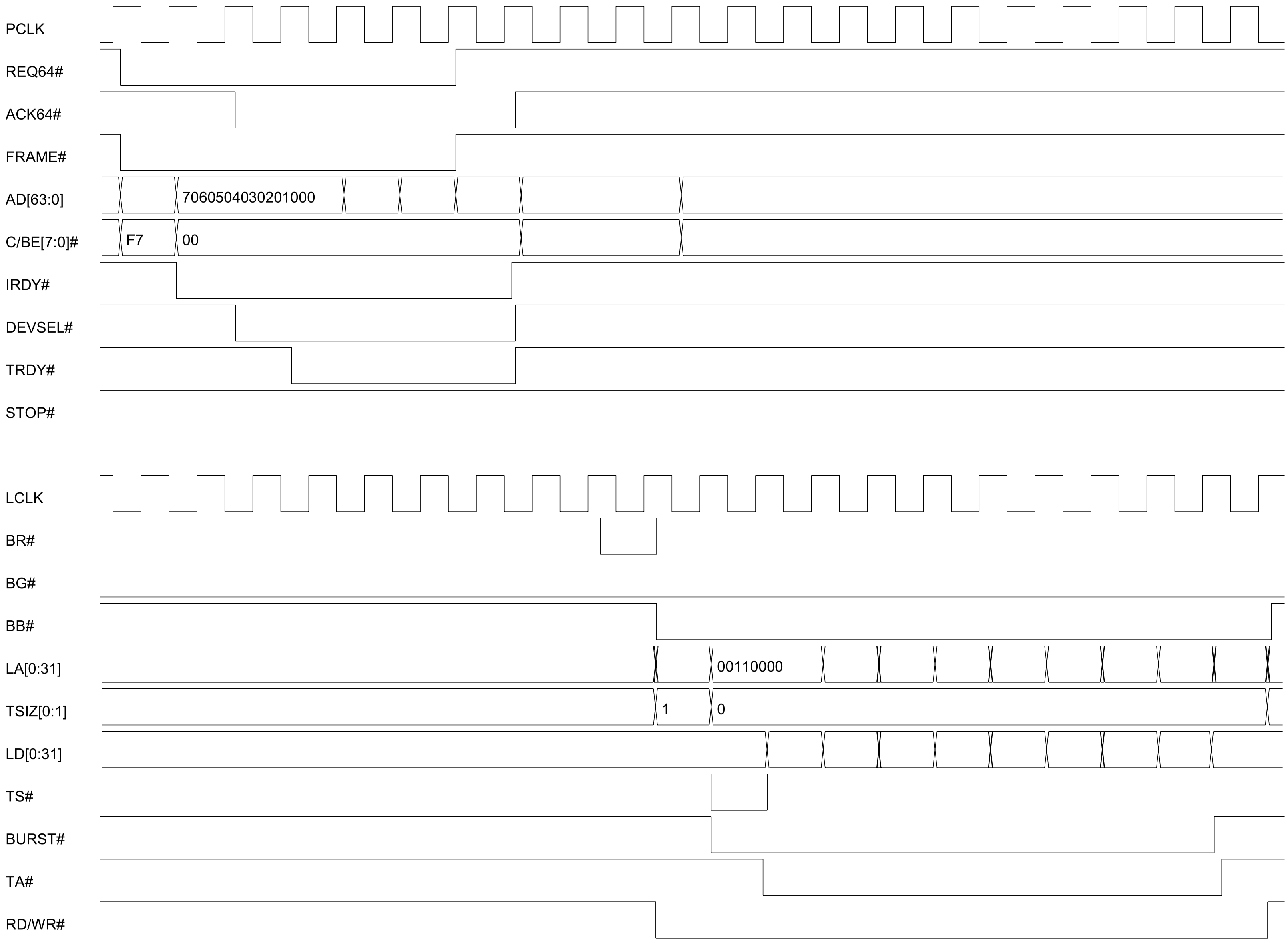
Timing Diagram 3-19. Direct Slave 64-Bit Burst Write of 4 Qwords (16-Bit Local Bus)



Notes: *Key bit/register values are MARBR[24]=1, BIGEND[4]=0, and LBRD1[15:0]=45C1h.*

The BDIP# signal (not shown) is asserted one clock after BURST# assertion and de-asserted one clock before BURST# de-assertion.

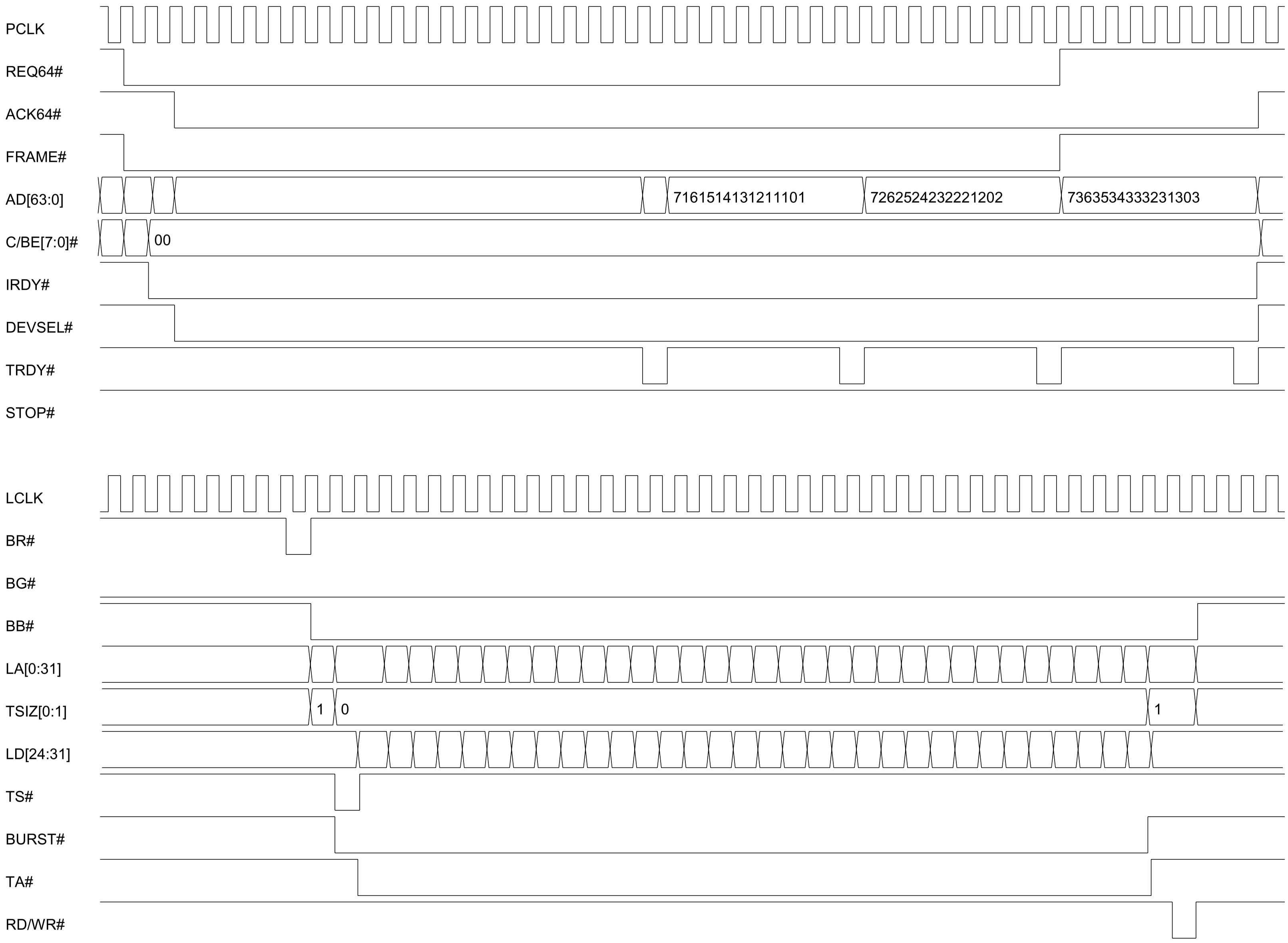
Timing Diagram 3-20. Direct Slave 64-Bit Burst Write of 4 Qwords (32-Bit Local Bus)



Notes: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.*

The BDIP# signal (not shown) is asserted one clock after BURST# assertion and de-asserted one clock before BURST# de-assertion.

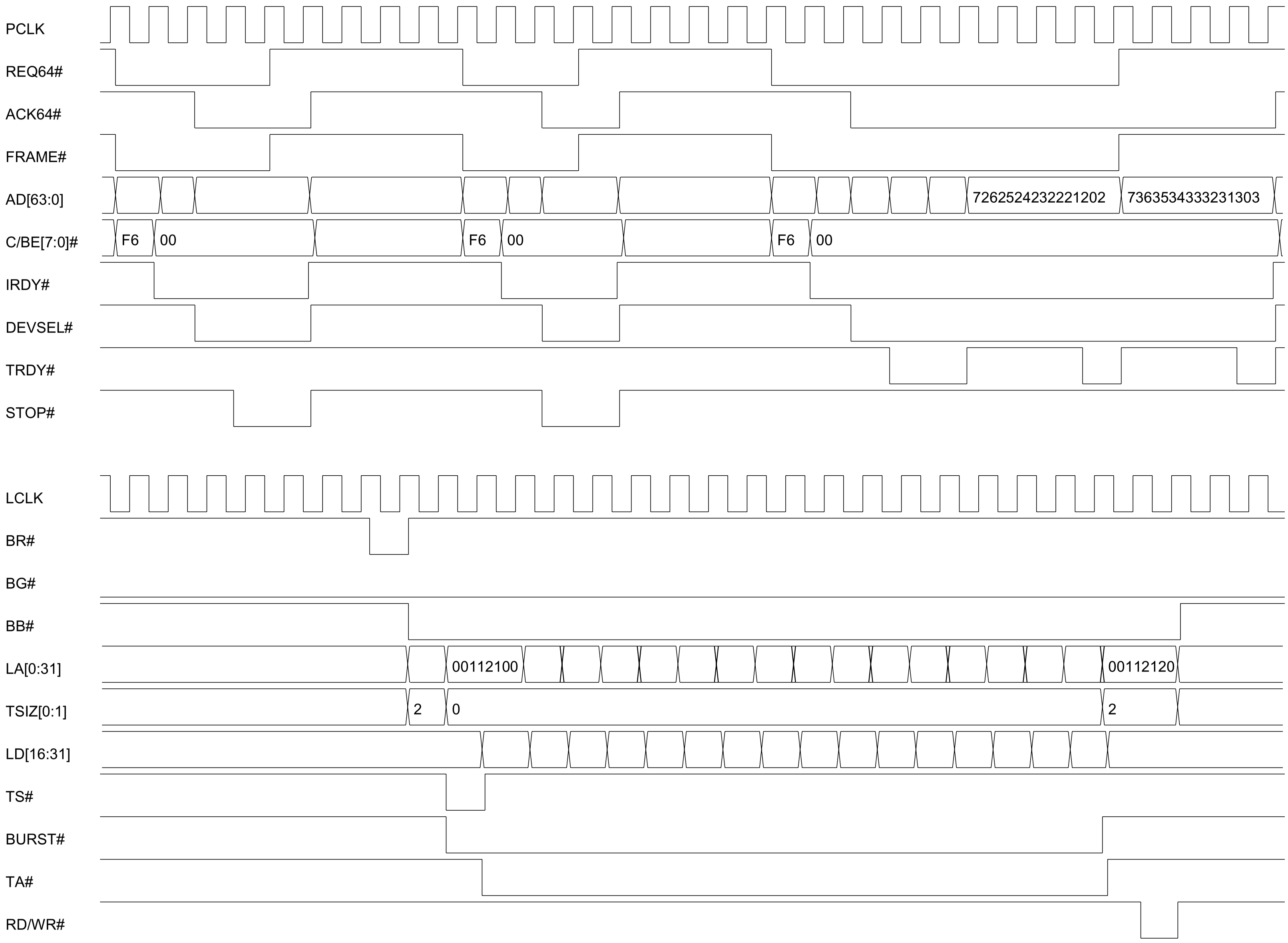
Timing Diagram 3-21. Direct Slave 64-Bit Burst Read of 4 Qwords (8-Bit Local Bus)



Notes: *Key bit/register values are MARBR[24]=0, BIGEND[4]=0, and LBRD1[15:0]=45C0h.*

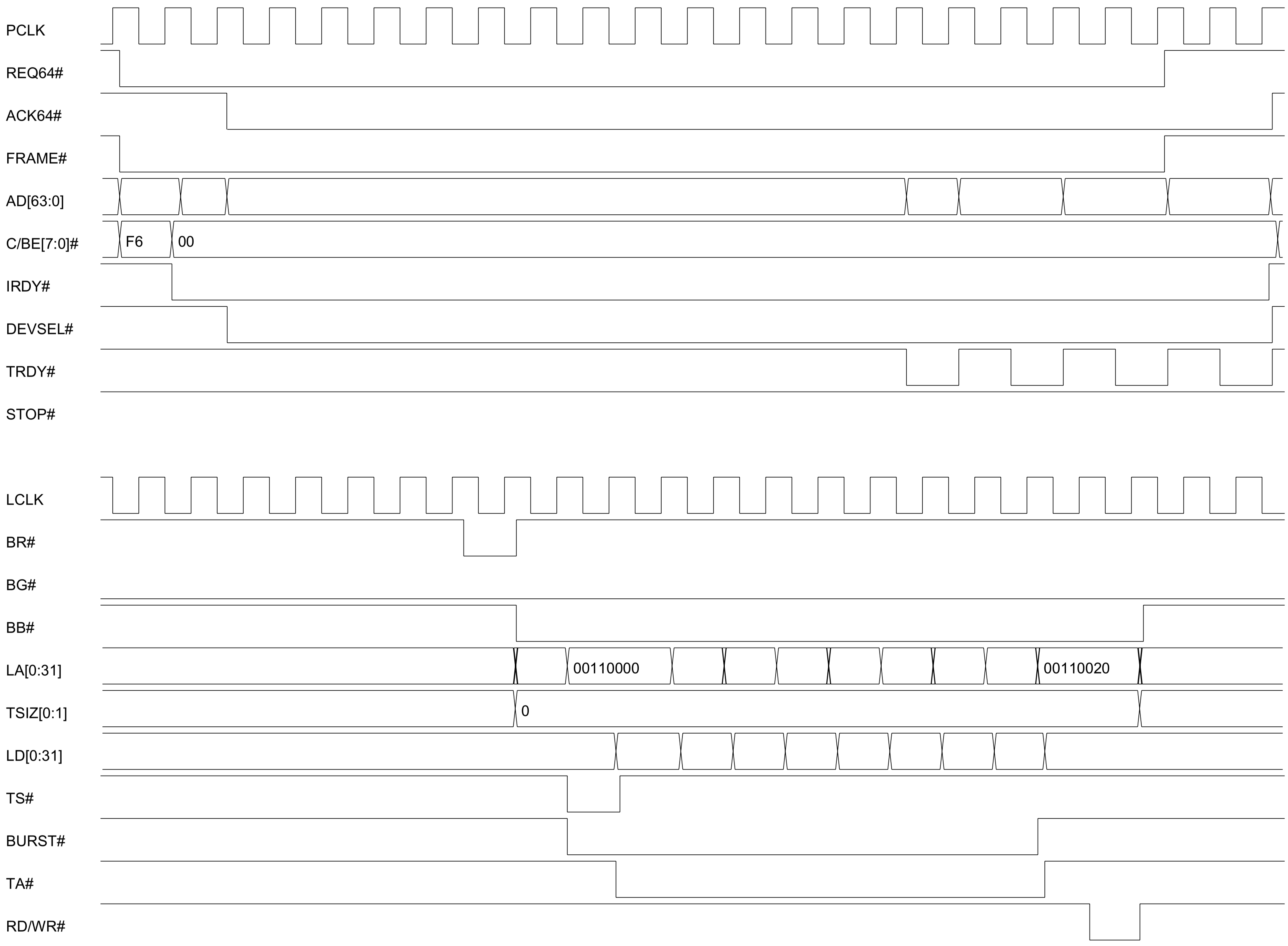
The BDIP# signal (not shown) is asserted one clock after BURST# assertion and de-asserted one clock before BURST# de-assertion.

Timing Diagram 3-22. Direct Slave 64-Bit Burst Read of 4 Qwords (16-Bit Local Bus)



Notes: *Key bit/register values are MARBR[24]=1, BIGEND[4]=0, and LBRD1[15:0]=45C1h.*
The BDIP# signal (not shown) is asserted one clock after BURST# assertion and de-asserted one clock before BURST# de-assertion.

Timing Diagram 3-23. Direct Slave 64-Bit Burst Read of 4 Qwords (32-Bit Local Bus)

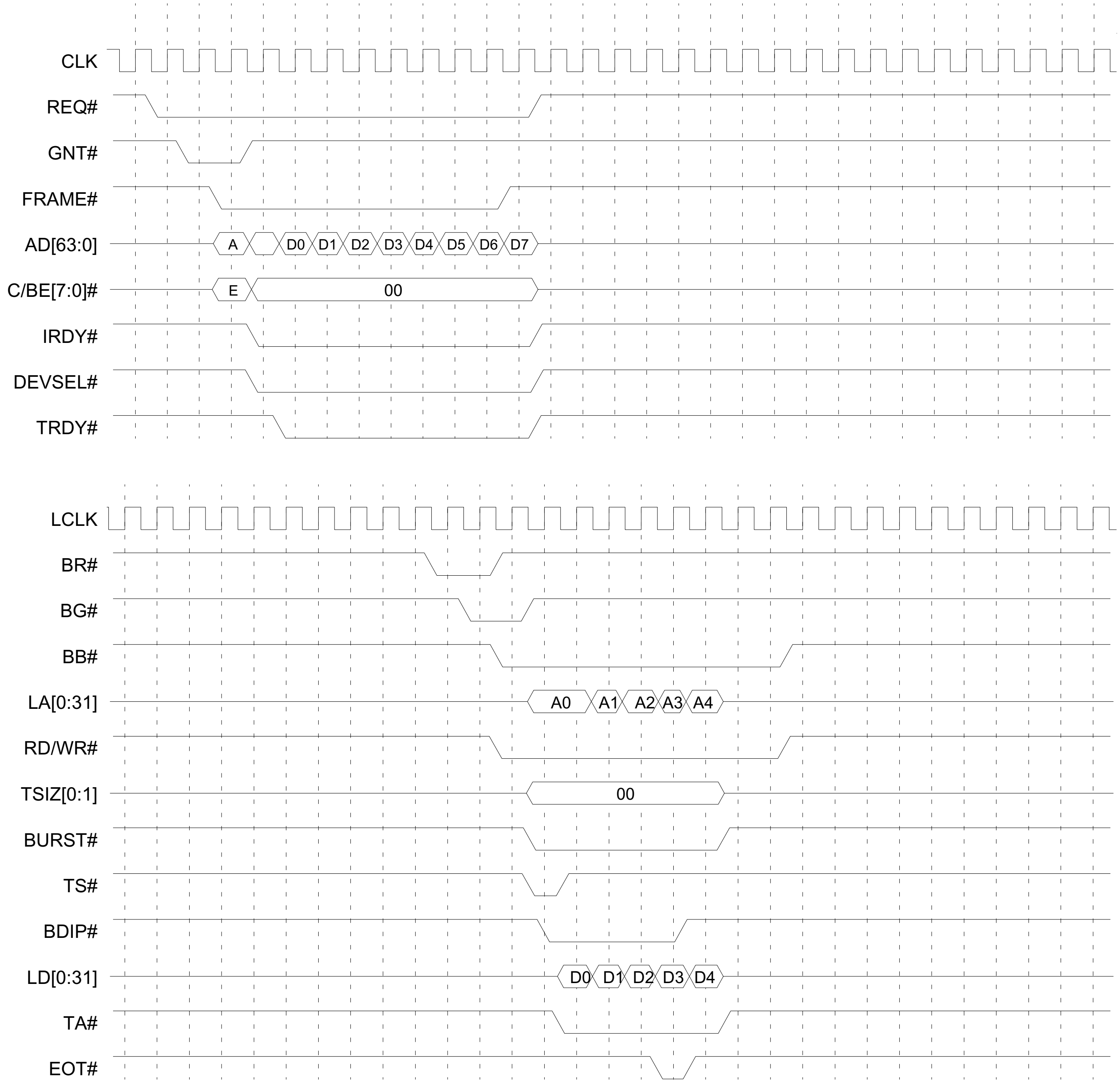


Notes: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.*

The BDIP# signal (not shown) is asserted one clock after BURST# assertion and de-asserted one clock before BURST# de-assertion.

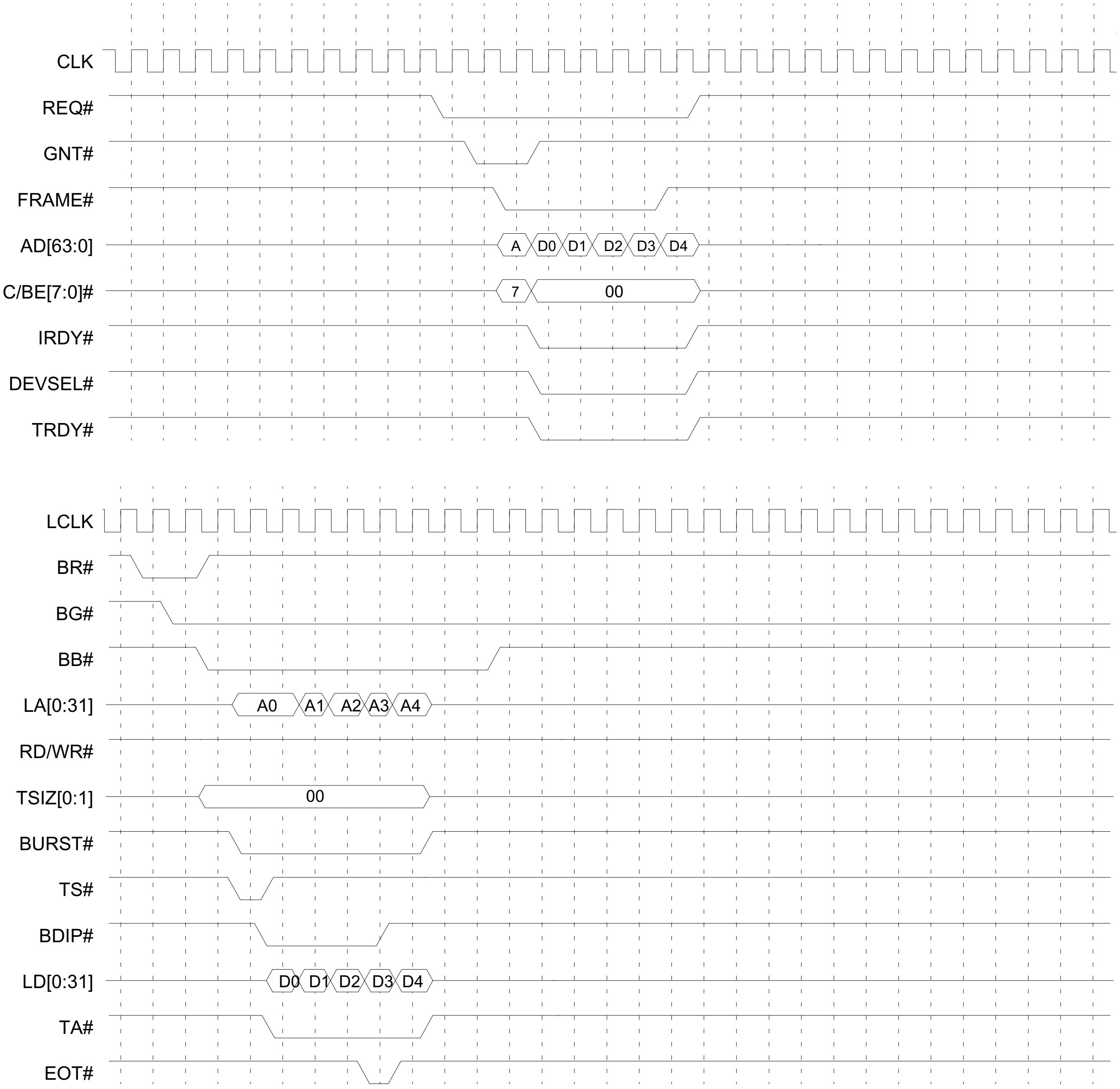
3.5.4 M Mode DMA Timing Diagrams

Timing Diagram 3-24. DMA PCI-to-Local, Continuous Burst Mode, with EOT# Assertion



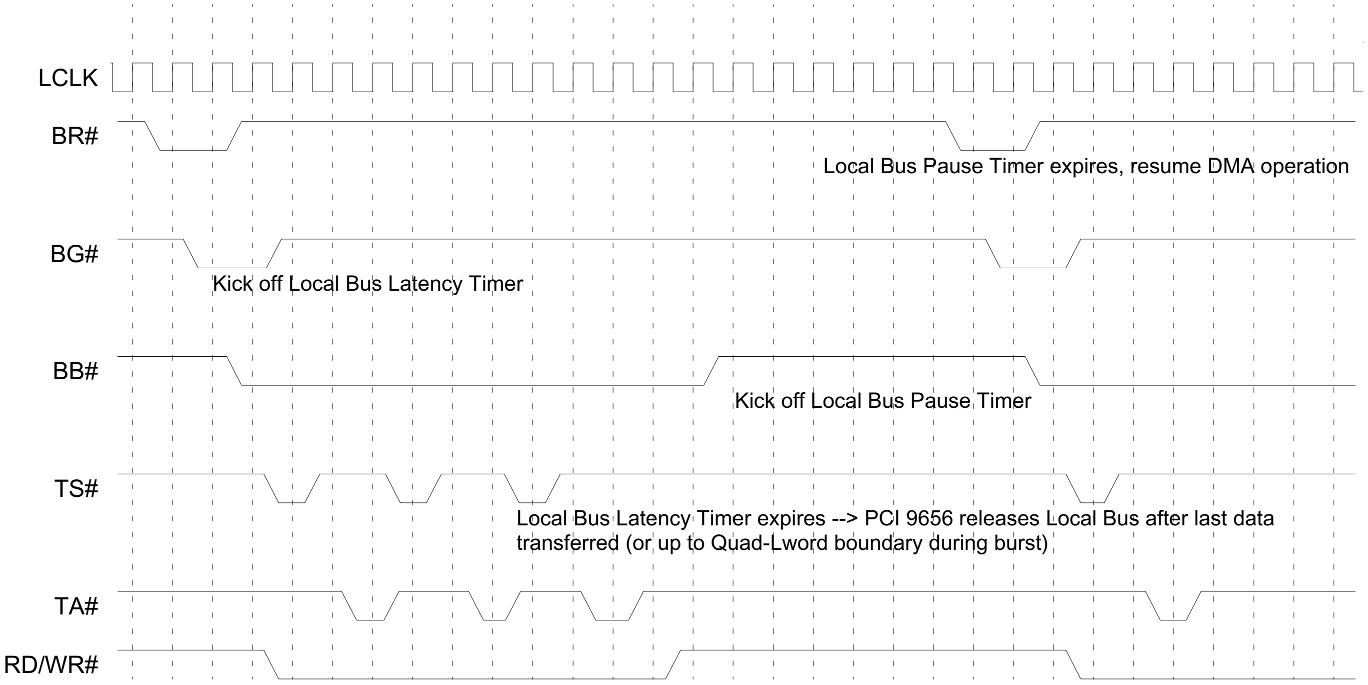
***Note:** For a PCI-to-Local DMA with EOT# assertion, all data read by the PCI 9656 from the PCI Bus is not necessarily written to the Local Bus – any data that is not written to the Local Bus due to EOT# assertion is flushed. The PCI 9656 is configured to be a 32-bit PCI device [REQ64# de-asserted when RST# transitions high (not shown)]. Therefore, ACK64# and REQ64# are always de-asserted and the PCI 9656 attempts only 32-bit transfers, as shown in the diagram. AD[63:32] and C/BE[7:4]# are driven to known values by the PCI 9656, but are irrelevant. For this example, the DMA transfer size count is assumed to be greater than or equal to the size of the transfer shown.*

Timing Diagram 3-25. DMA Local-to-PCI, Continuous Burst Mode, with EOT# Assertion

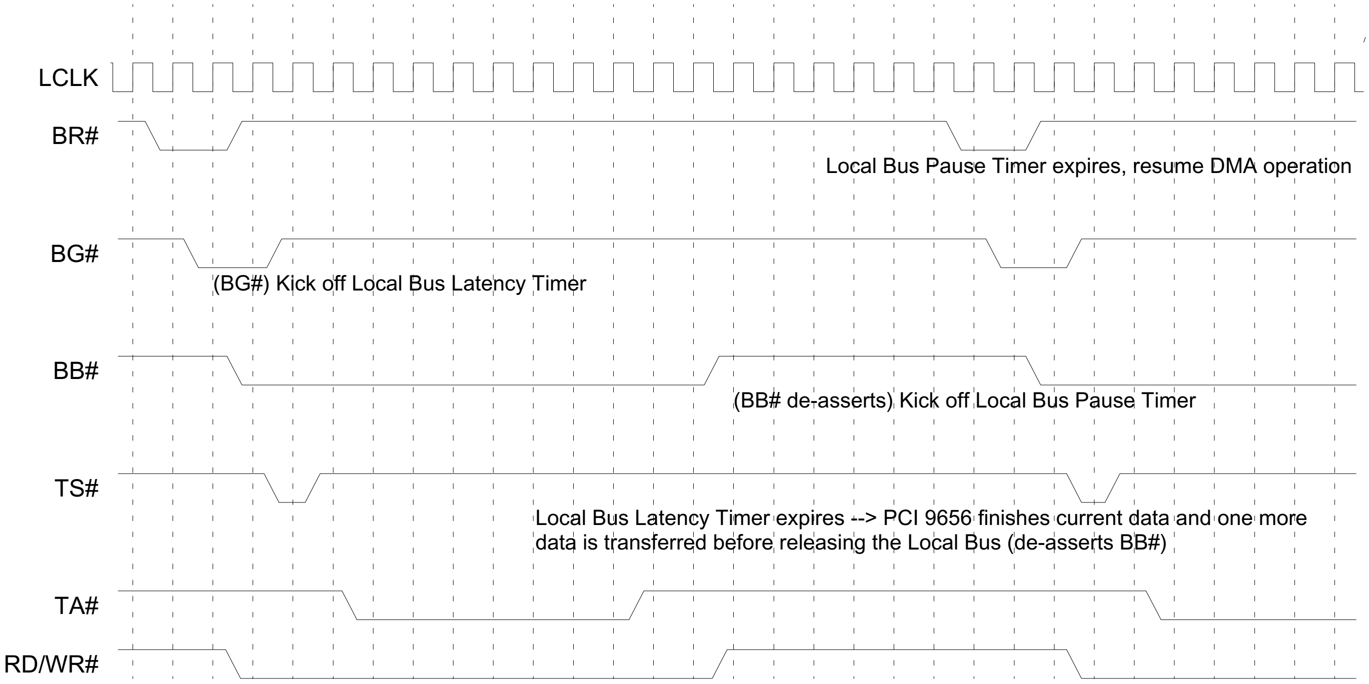


Note: *For a Local-to-PCI DMA with EOT# assertion, all data read by the PCI 9656 from the Local Bus is written to the PCI Bus. The PCI 9656 is configured to be a 32-bit PCI device [REQ64# de-asserted when RST# transitions high (not shown)]. Therefore, ACK64# and REQ64# are always de-asserted and the PCI 9656 attempts only 32-bit transfers, as shown in the diagram. AD[63:32] and C/BE[7:4]# are driven to known values by the PCI 9656, but are irrelevant. For this example, the DMA transfer size count is assumed to be greater than or equal to the size of the transfer shown.*

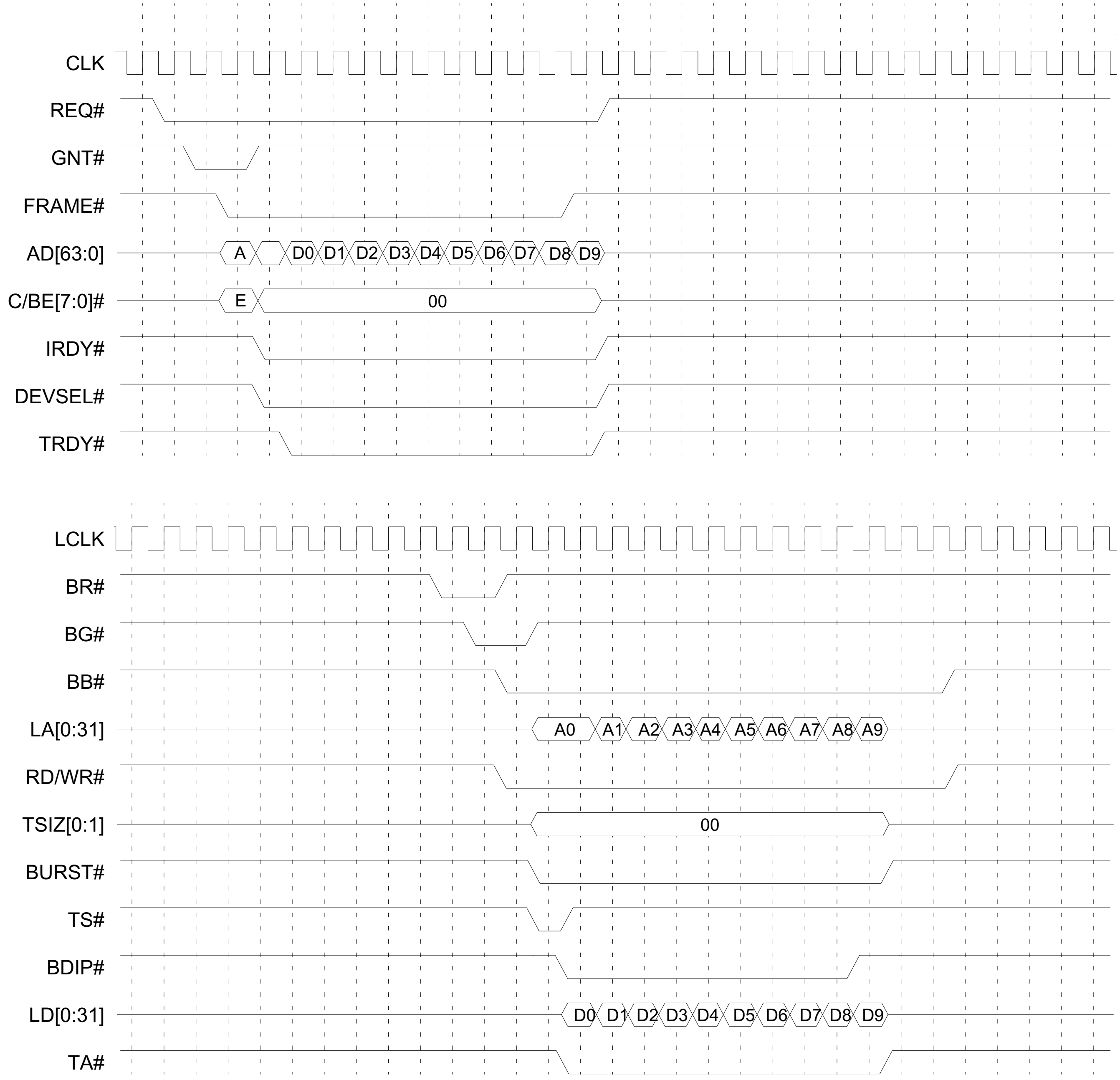
Timing Diagram 3-26. Local Bus Latency Timer (Eight Clocks) and Pause Timer (Four Clocks) in DMA Operation


LCLK
BR#
Local Bus Pause Timer expires, resume DMA operation
BG#
Kick off Local Bus Latency Timer
BB#
Kick off Local Bus Pause Timer
TS#
Local Bus Latency Timer expires --> PCI 9656 releases Local Bus after last data transferred (or up to Quad-Lword boundary during burst)
TA#
RD/WR#

Timing Diagram 3-27. Local Bus Latency Timer (Eight Clocks) and Pause Timer (Four Clocks) in DMA Operation, Continuous Burst Mode

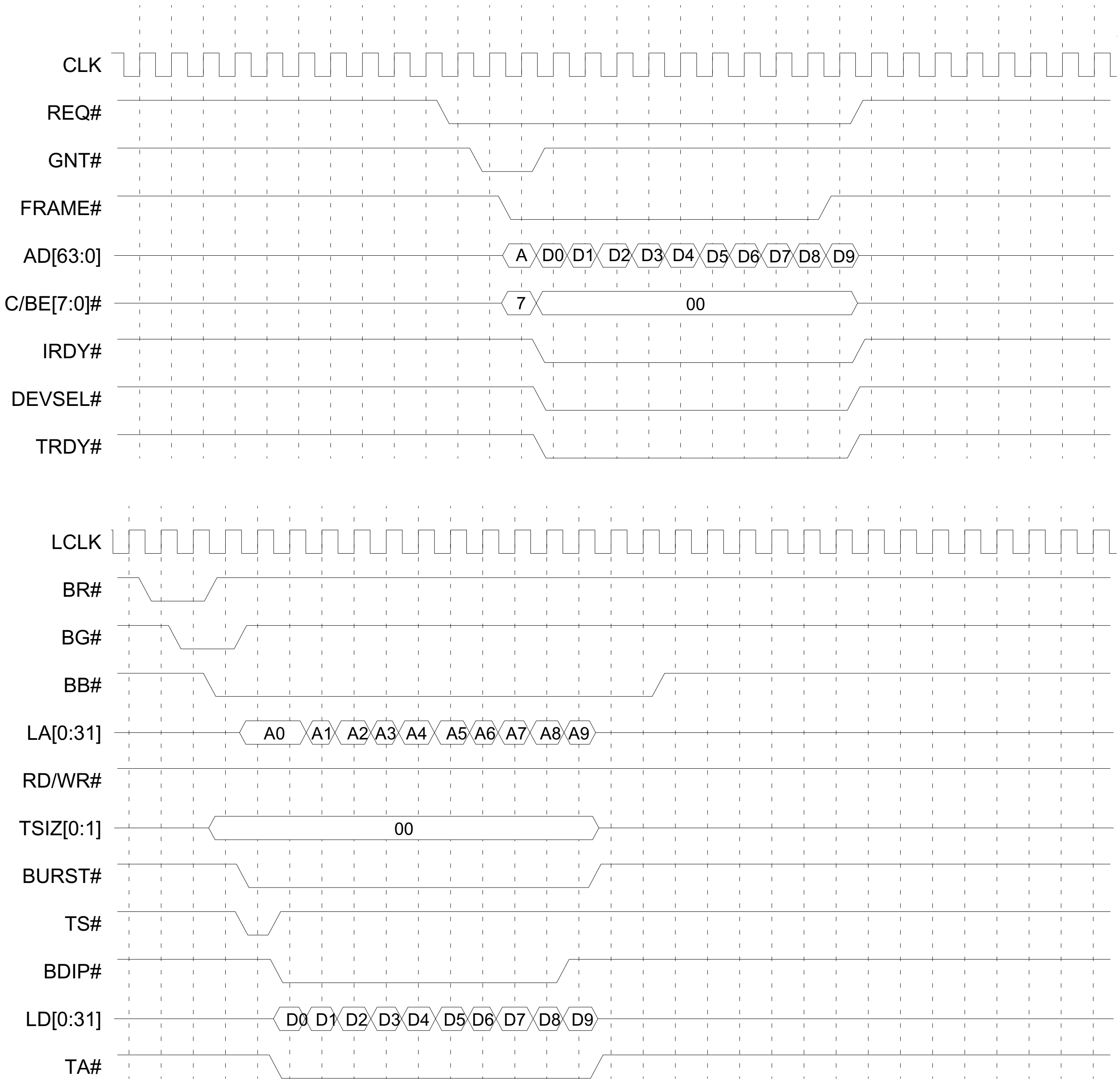

LCLK
BR#
Local Bus Pause Timer expires, resume DMA operation
BG#
(BG#) Kick off Local Bus Latency Timer
BB#
(BB# de-asserts) Kick off Local Bus Pause Timer
TS#
Local Bus Latency Timer expires --> PCI 9656 finishes current data and one more data is transferred before releasing the Local Bus (de-asserts BB#)
TA#
RD/WR#

Timing Diagram 3-28. DMA PCI-to-Local, Continuous Burst Mode, Transfer Size = 10 Lwords



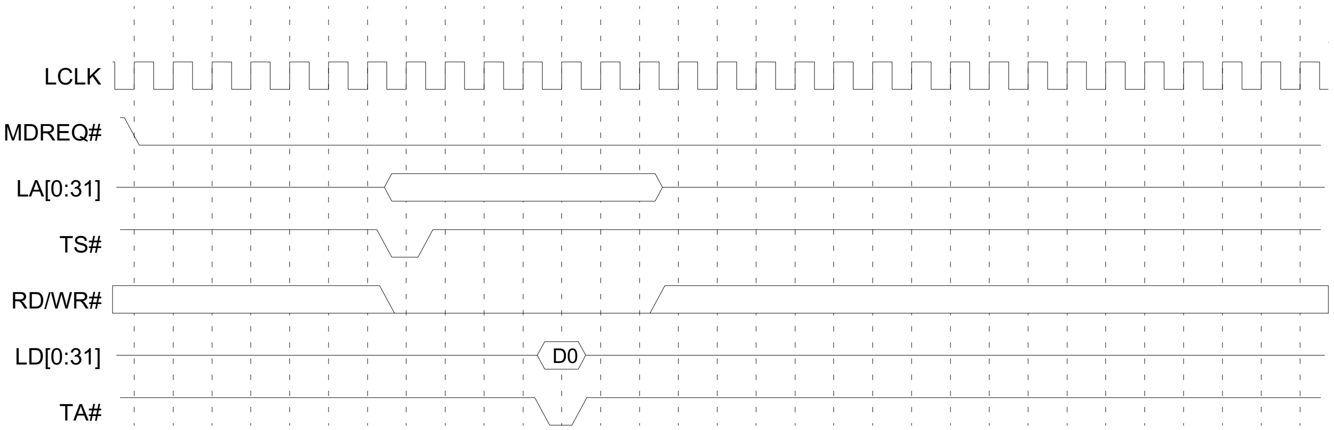
Note: *The PCI 9656 is configured to be a 32-bit PCI device [REQ64# de-asserted when RST# transitions high (not shown)]. Therefore, ACK64# and REQ64# are always de-asserted and the PCI 9656 attempts only 32-bit transfers, as shown in the diagram. AD[63:32] and C/BE[7:4]# are driven to known values by the PCI 9656, but are irrelevant.*

Timing Diagram 3-29. DMA Local-to-PCI, Continuous Burst Mode, Transfer Size = 10 Lwords



Note: *The PCI 9656 is configured to be a 32-bit PCI device [REQ64# de-asserted when RST# transitions high (not shown)]. Therefore, ACK64# and REQ64# are always de-asserted and the PCI 9656 attempts only 32-bit transfers, as shown in the diagram. AD[63:32] and C/BE[7:4]# are driven to known values by the PCI 9656, but are irrelevant.*

Timing Diagram 3-30. IDMA Single Write Cycle

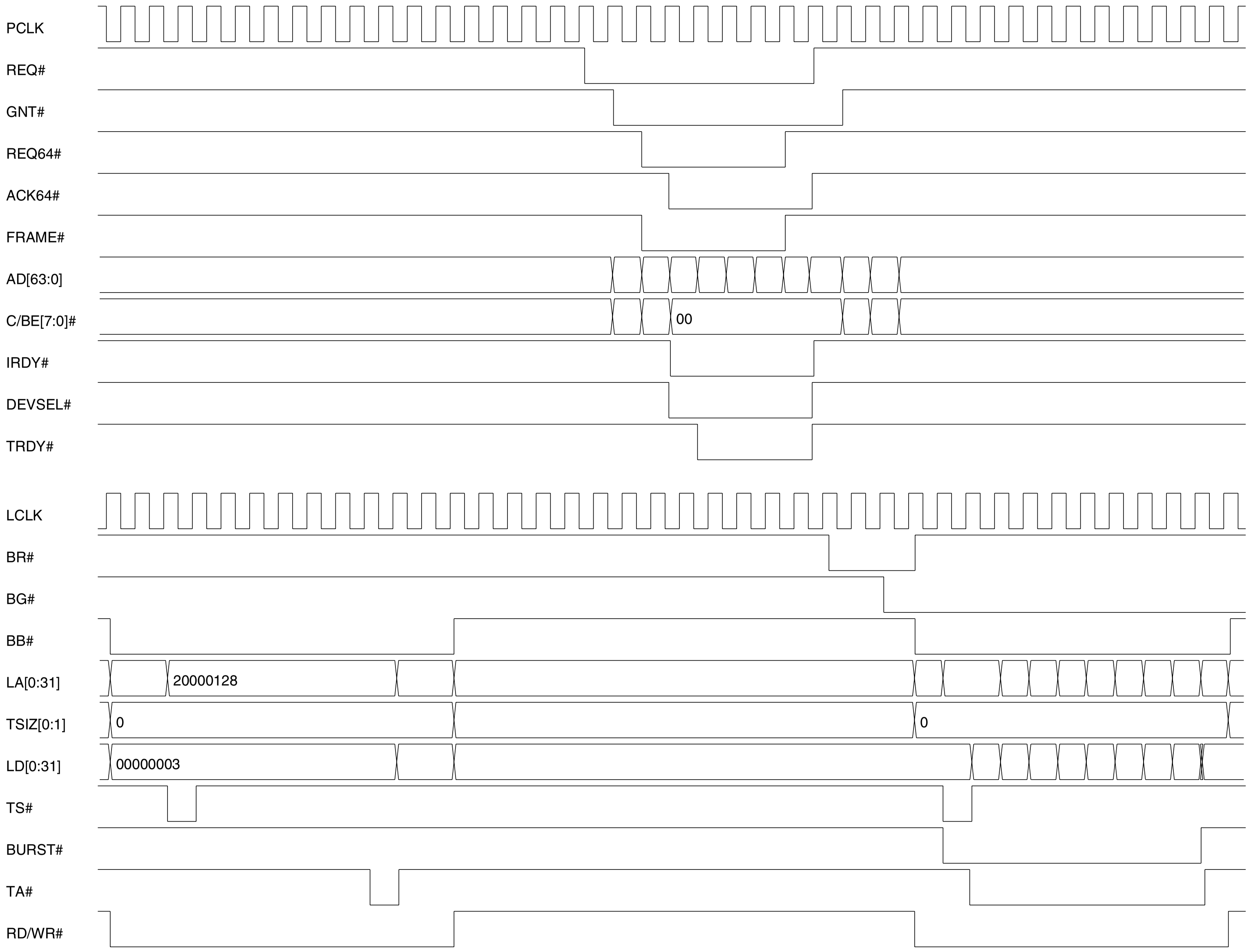


Notes: *The PCI 9656 treats the IDMA function from the MPC850 or MPC860 the same as a Direct Master cycle.*

The MPC850 or MPC860 starts IDMA cycle when the IDMA Enable bit is set in the MPC850 or MPC860 respective register.

User must clear the IDMA Enable bit when the MPC850 or MPC860 is done (monitor interrupt) with the IDMA cycle.

Timing Diagram 3-31. DMA PCI-to-Local (32-Bit Local Bus, 64-Bit PCI Bus)

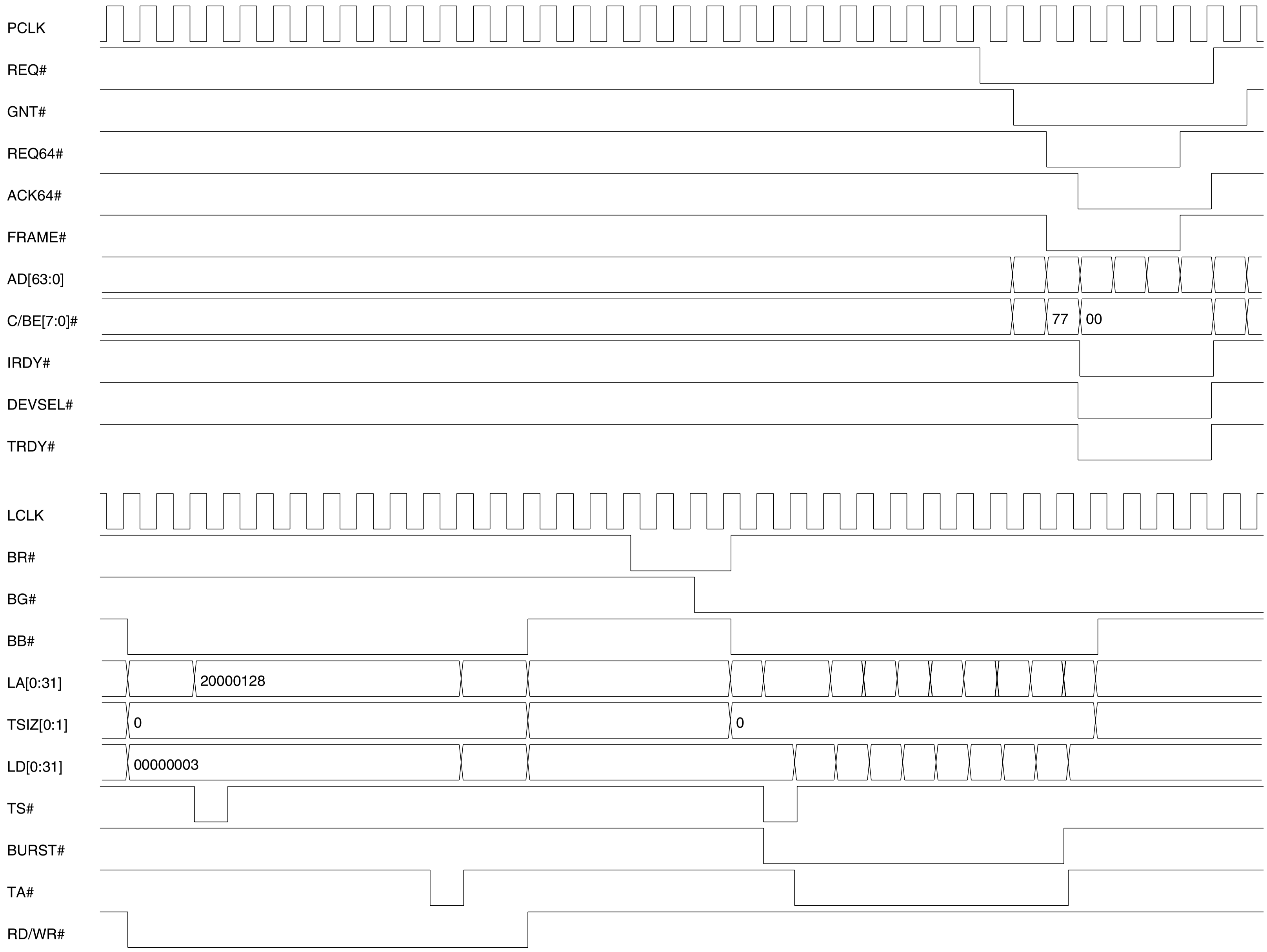


Notes: *The writing of the registers to set up this 32-byte DMA Block mode transfer using DMA Channel 0 is not shown.*

Writing the DMACSR0 register (LOC:128h) with 03h enables and starts this DMA transfer. The PCI STOP# signal (not shown) is not asserted during this transfer.

The BDIP# signal (not shown) is asserted one clock after BURST# assertion and de-asserted one clock before BURST# de-assertion.

Timing Diagram 3-32. DMA Local-to-PCI (32-Bit Local Bus, 64-Bit PCI Bus)

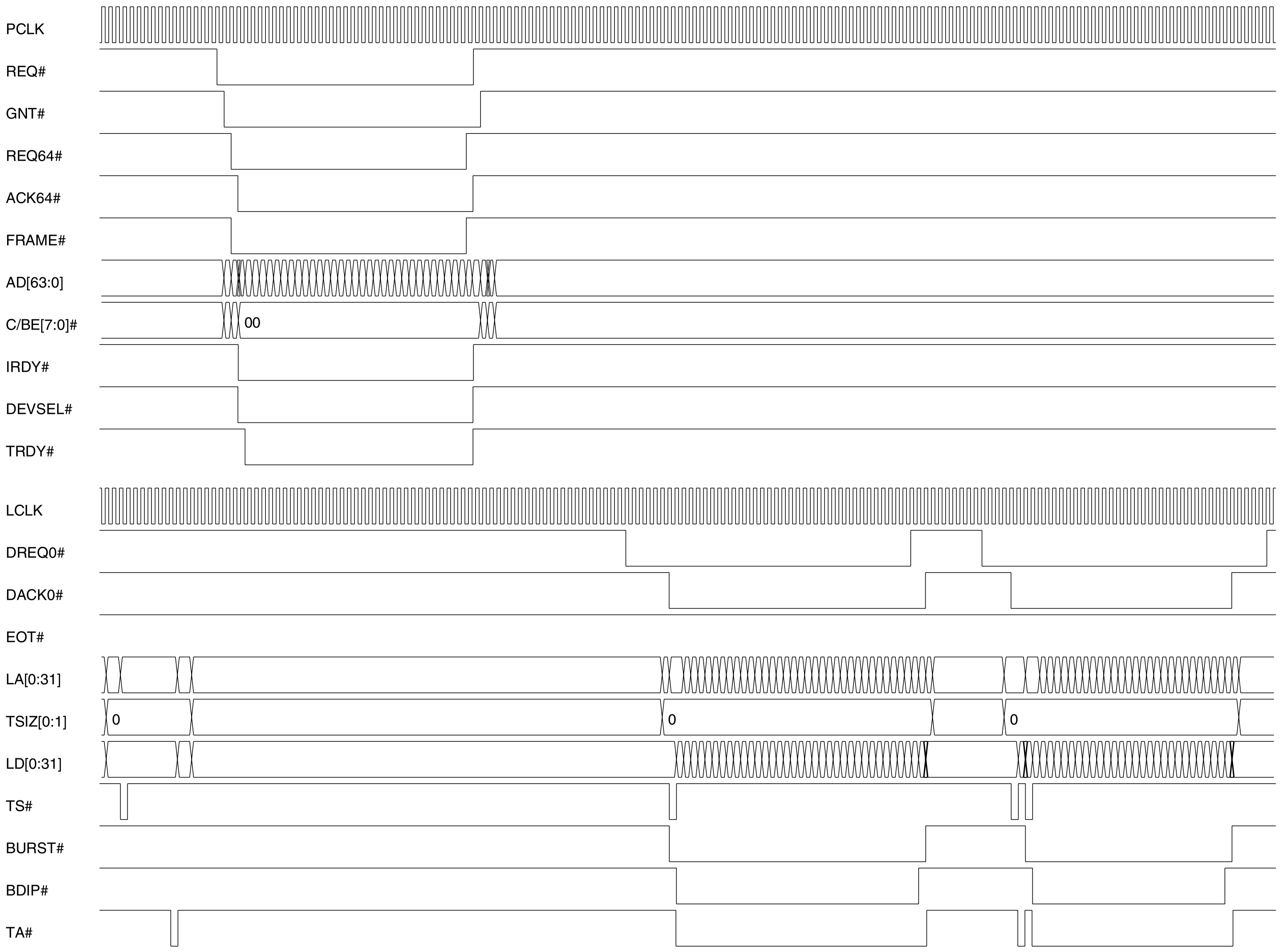


Notes: *The writing of the registers to set up this 32-byte DMA Block mode transfer using DMA Channel 0 is not shown.*

Writing the DMACSR0 register (LOC:128h) with 03h enables and starts this DMA transfer. The PCI STOP# signal (not shown) is not asserted during this transfer.

The BDIP# signal (not shown) is asserted one clock after BURST# assertion and de-asserted one clock before BURST# de-assertion.

Timing Diagram 3-33. DMA PCI-to-Local Demand Mode (32-Bit Local Bus, 64-Bit PCI Bus)



Notes: *The Channel 0 Demand mode DMA transfer starts when the DMACSR0[1:0] bits are written to 11b.*

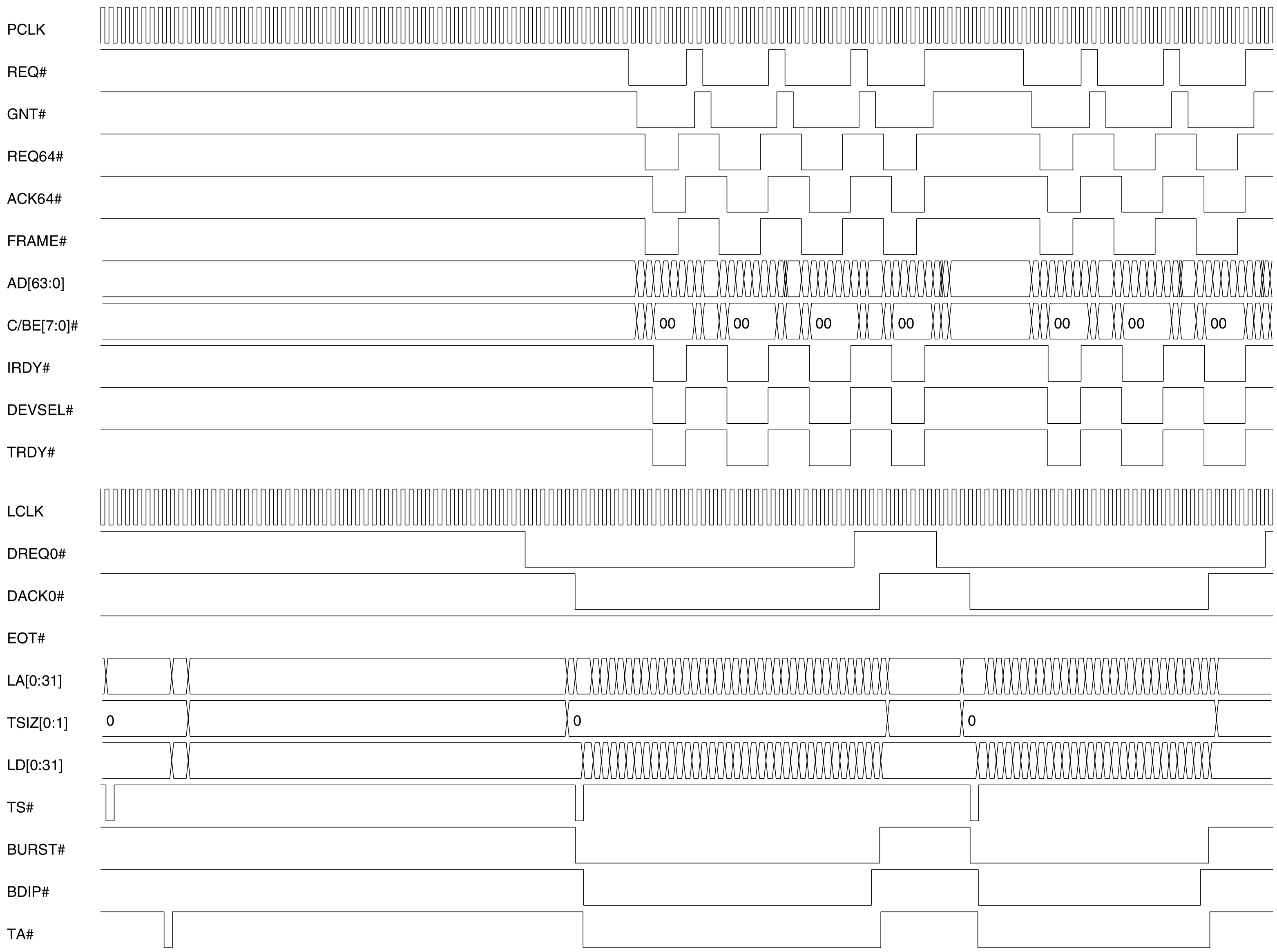
The PCI 9656 does not attempt to arbitrate for and write data to the Local Bus until it detects the DREQ0# signal asserted.

The transfer is temporarily suspended when DREQ0# is de-asserted, then resumed upon its re-assertion.

The PCI STOP# signal (not shown) is not asserted during this transfer.

The RD/WR# signal is 0 when the PCI 9656 is writing data to the Local Bus.

Timing Diagram 3-34. DMA Local-to-PCI Demand Mode (32-Bit Local Bus, 64-Bit PCI Bus)



Notes: *The Channel 0 Demand mode DMA transfer starts when the DMACSR0[1:0] bits are written to 11b and the PCI 9656 detects the DREQ0# signal asserted. Once started, the PCI 9656 arbitrates for and reads data from the Local Bus, then writes the data to the PCI Bus until the transfer is temporarily suspended when DREQ0# is de-asserted. The transfer resumes upon DREQ0# re-assertion. The PCI STOP# signal (not shown) is not asserted during this transfer.*

The RD/WR# signal is 1 when the PCI 9656 is reading data from the Local Bus.

4 C AND J MODES BUS OPERATION

4.1 PCI BUS CYCLES

The PCI 9656 is *PCI r2.2*-compliant. Refer to *PCI r2.2* for specific PCI Bus functions.

4.1.1 Direct Slave Command Codes

As a target, the PCI 9656 allows access to the PCI 9656 internal registers and the Local Bus, using the commands listed in Table 4-1.

All Direct Slave Read or Write accesses to the PCI 9656 can be Byte (8-bit), Word (16-bit), Lword (32-bit), or Qword (64-bit) accesses. All memory commands are aliased to basic memory commands. All I/O accesses to the PCI 9656 are decoded to an Lword boundary. The PCI Byte Enables (C/BE[7:0]#) 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. All Configuration register Read or Write accesses to the PCI 9656 can be Byte, Word, or Lword accesses, with Byte Enables used to determine which bytes are read or written.

Table 4-1. Direct Slave Command Codes

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

4.1.2 PCI Master Command Codes

The PCI 9656 becomes the PCI Bus Master to perform Direct Master or DMA transfers. The PCI command code used by the PCI 9656 during a Direct Master or DMA transfer is specified by the value(s) contained in the CNTRL[15:0] register bits. Except when in Memory Write and Invalidate (MWI) mode (PCICR[4]=1; DMPBAM[9]=1 or DMAMODE*x*[13]=1; PCICLSR[7:0] = cache line size of 8 or 16 Lwords). In MWI mode for a Direct Master or DMA transfer, if the starting address alignment is aligned with the cache line size and upon determining it can transfer at least one cache line of data, the PCI 9656 uses a PCI command code of Fh regardless of the value(s) contained in the CNTRL[15:0] register bits.

Note: *DMA can only perform Memory accesses. DMA* ***cannot*** *perform I/O or Configuration accesses.*

4.1.2.1 DMA Master Command Codes

The PCI 9656 DMA Controllers can assert the Memory cycles listed in Table 4-2.

Table 4-2. DMA Master Command Codes

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

Note: *During each command phase, the PCI 9656 drives C/BE[7:4]# to unspecified values.*

4.1.2.2 Direct Local-to-PCI Command Codes

For direct Local-to-PCI Bus accesses, the PCI 9656 asserts the cycles listed in Tables 4-3 through 4-5.

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

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

Note: *During each command phase, the PCI 9656 drives C/BE[7:4]# to unspecified values.*

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

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

Note: *During each command phase, the PCI 9656 drives C/BE[7:4]# to unspecified values.*

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

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

Note: *During each command phase, the PCI 9656 drives C/BE[7:4]# to unspecified values.*

4.1.3 PCI Arbitration

The PCI 9656 asserts REQ# to request the PCI Bus. The PCI 9656 can be programmed using the PCI Request Mode bit (MARBR[23]) to de-assert REQ# when the PCI 9656 asserts FRAME# during a Bus Master cycle, or to hold REQ# asserted for the entire Bus Master cycle. The PCI 9656 always de-asserts REQ# for a minimum of two PCI clocks between Bus Master ownership that includes a Target disconnect.

During a Direct Master Write cycle, the PCI 9656 PCI REQ# assertion can be delayed by programming the Direct Master Delayed Write Mode bits (DMPBAM [15:14]). DMPBAM can be programmed to wait 0, 4, 8, or 16 PCI Bus clocks after the PCI 9656 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 9656 Direct Master Write FIFO, which provides for better use of the PCI Bus.

4.1.4 PCI Bus Wait States

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

4.2 LOCAL BUS CYCLES

The PCI 9656 interfaces a PCI Host Bus to several Local Bus types, as listed in Table 4-6. It operates in one of three modes – M, C, and J, selected through the MODE[1:0] pins – corresponding to the three bus types.

Table 4-6. MODE Pin-to-Bus Mode Cross-Reference

MODE1	MODE0	Bus Mode	Bus Type
1	1	M	Motorola, 32-bit non-multiplexed
1	0	***Reserved***	–
0	0	C	Intel/Generic, 32-bit non-multiplexed
0	1	J	Intel/Generic, 32-bit multiplexed

4.2.1 Local Bus Arbitration and BREQi

The PCI 9656 asserts LHOLD to request the Local Bus for a Direct Slave or DMA transfer. The PCI 9656 owns and becomes the Local Bus Master when LHOLD and LHOLDA are both asserted. As the Local Bus Master, the PCI 9656 responds to BREQi assertion to relinquish Local Bus ownership during Direct Slave or DMA transfers if either of the following conditions are true:

- BREQi is asserted and enabled (MARBR[18]=1)
- Gating is enabled and the Local Bus Latency Timer is enabled and expires (MARBR[27, 16, 7:0])

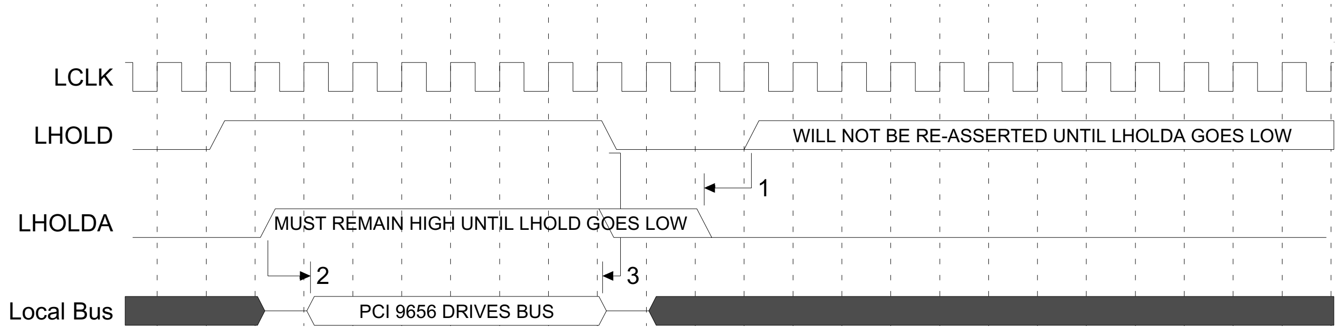
Before releasing the Local Bus, the PCI 9656 attempts to transfer up to a total of three additional Lwords of data. This includes the last transfer with BLAST# asserted. The actual number of additional transfers depends upon the Local Bus data width and address when BREQi is asserted. The minimum assertion duration of the BREQi signal is one Local Bus clock cycle. Typically, Local Bus logic asserts BREQi and leaves it asserted until either BLAST# is asserted (BLAST#=0) or LHOLD is de-asserted (LHOLD=0).

When the PCI 9656 releases the Local Bus, the external Local Bus Arbiter can grant the Local Bus to another Master. If the PCI 9656 needs to complete the disconnected transfer, it requests the Local Bus by re-asserting LHOLD upon detection of LHOLDA de-assertion and the Local Bus Pause Timer being zero (0), if enabled.

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

4.2.1.1 Local Bus Arbitration Timing Diagram

Timing Diagram 4-1. Local Bus Arbitration (LHOLD and LHOLDA)


LCLK
LHOLD
WILL NOT BE RE-ASSERTED UNTIL LHOLDA GOES LOW
1
LHOLDA
MUST REMAIN HIGH UNTIL LHOLD GOES LOW
2
3
Local Bus
PCI 9656 DRIVES BUS

Notes: *In response to the PCI 9656 assertion of LHOLD, the external Local Bus Arbiter asserts LHOLDA to grant the Local Bus to the PCI 9656. In Timing Diagram 4-1, the Local Bus Arbiter asserts LHOLDA immediately upon detection of LHOLD assertion; however, the Local Bus Arbiter has the option, if necessary, to assert LHOLDA significantly later.*

Timing Diagram 4-1 was created using the Timing Designer tool. It is accurate and adheres to its specified protocol(s). This diagram shows transfers and signal transitions, but should not be relied upon to show exactly, on a clock-for-clock basis, where PCI 9656-driven signal transitions will occur.

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 9656 performs a single PCI Bus cycle. Otherwise, the PCI 9656 performs a Burst PCI Bus cycle and BLAST# is used to end the cycle. As a Local Bus Target, the PCI 9656 allows access to the PCI 9656 internal registers and the PCI Bus.

Local Bus Direct Master accesses to the PCI 9656 must be for a 32-bit non-pipelined bus. Non-32-bit Direct Master accesses to the PCI 9656 require simple external logic (latch array to combine data into a 32-bit bus).

4.2.3 Direct Slave

The PCI Bus Master reads from and writes to the Local Bus (the PCI 9656 is a PCI Bus Target and a Local Bus Master).

4.2.4 Wait State Control

If READY# mode is disabled (LBRD0[6]=0 for Space 0, LBRD1[6]=0 for Space 1, LBRD0[22]=0 for Expansion ROM, and/or DMAMODE*x*[6]=0 for Channel *x*, where *x* is the DMA Channel number), the external READY# input signal has no effect on wait states for a Local access. Wait states between Data cycles are asserted internally by Wait State Counter(s) (LBRD0[5:2] for Space 0, LBRD1[5:2] for Space 1, LBRD0[21:18] for Expansion ROM, and/or DMAMODE*x*[5:2] for Channel *x*). The Wait State Counter(s) is initialized with its Configuration register value at the start of each data access.

If READY# mode is enabled (LBRD0[6]=1 for Space 0, LBRD1[6]=1 for Space 1, LBRD0[22]=1 for Expansion ROM, and/or DMAMODE*x*[6]=1 for Channel *x*), it has no effect until the Wait State Counter(s) reaches 0. READY# then controls the number of additional wait states.

BTERM# input is not sampled until the Wait State Counter(s) reaches 0. BTERM# assertion overrides READY# when BTERM# is enabled (LBRD0[7]=1 for Space 0, LBRD1[7]=1 for Space 1, LBRD0[23]=1 for Expansion ROM, and/or DMAMODE*x*[7]=1 for Channel *x*) and asserted.

Figure 4-1 illustrates the PCI 9656 wait states for C and J modes.

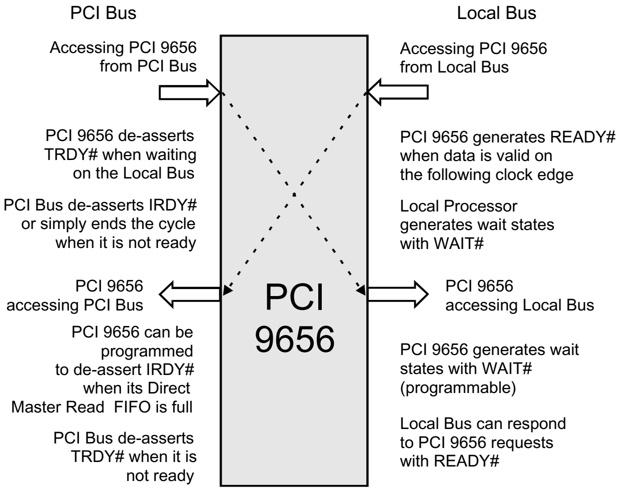

PCI Bus
Local Bus
Accessing PCI 9656 from PCI Bus
Accessing PCI 9656 from Local Bus
PCI 9656 de-asserts TRDY# when waiting on the Local Bus
PCI 9656 generates READY# when data is valid on the following clock edge
PCI Bus de-asserts IRDY# or simply ends the cycle when it is not ready
Local Processor generates wait states with WAIT#
PCI 9656 accessing PCI Bus
PCI 9656
PCI 9656 accessing Local Bus
PCI 9656 can be programmed to de-assert IRDY# when its Direct Master Read FIFO is full
PCI 9656 generates wait states with WAIT# (programmable)
PCI Bus de-asserts TRDY# when it is not ready
Local Bus can respond to PCI 9656 requests with READY#

Figure 4-1. Wait States

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

4.2.4.1 Local Bus Wait States

In Direct Master mode when accessing the PCI 9656 registers, the PCI 9656 acts as a Local Bus slave. The PCI 9656 asserts wait states by delaying the READY# signal. The Local Bus Master inserts wait states with the WAIT# signal. For writes to PCI 9656 registers, WAIT# should be asserted until data is valid.

In Direct Slave and DMA modes, the PCI 9656 acts as a Local Bus master. The PCI 9656 inserts Local Bus slave wait states with the WAIT# signal. The Local Bus Target asserts external wait states by delaying the READY# signal. The Internal Wait State Counter(s) can be programmed to insert wait states between the address and first data states, with the same number of wait states between subsequent data within bursts. (Refer to Table 4-7.)

For Direct Master writes, to insert wait states between the Address phase and the first data, the WAIT# signal must be asserted during the Address phase [ADS# (C mode) or ALE (J mode) assertion]. Thereafter, WAIT# can be toggled, as necessary, to insert wait states.

During Direct Master accesses, if the WAIT# signal is asserted during the Address phase (ADS# assertion) or after the ADS# assertion for a Direct Master read, the PCI 9656 latches the address but does not begin the PCI Bus read until WAIT# is de-asserted. This results in the PCI 9656 not having the data available on the Local Bus. In this case, WAIT# de-assertion by the Local Bus Master enables the PCI 9656 to being the Direct Master Read access on the Local Bus. Assertion of WAIT# by the Local Bus Master on the second Local Bus Clock cycle after the Address phase (ADS# de-assertion) allows the Direct Master Read access on the PCI Bus to begin.

Note: *Do not use READY# as a gating signal for WAIT# de-assertion. Otherwise, the PCI 9656 does not complete the transfer.*

Table 4-7. Internal Wait State Counters

Bits	Description
LBRD0[5:2]	Local Address Space 0 Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)
LBRD1[5:2]	Local Address Space 1 Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)
LBRD0[21:18]	Expansion ROM Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)
DMAMODE0[5:2]	DMA Channel 0 Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)
DMAMODE1[5:2]	DMA Channel 1 Internal Wait State Counter (address-to-data; data-to-data; 0 to 15 wait states)

4.2.5 Data Transfer Modes

The PCI 9656 supports C and J modes with three Data Transfer modes:

- Single Cycle
- Burst-4
- Continuous Burst

Single Cycle mode is the default Data Transfer mode, Burst-4 mode is C and J mode-compatible, and Continuous Burst mode provides the highest throughput.

Table 4-8 summarizes the register settings used to select Local Bus Data Transfer modes. It also indicates the data quantity transferred per Address Cycle (ADS#).

Notes: *The BI#/BTERM# pin is referred to as the BI# pin in M mode, and as the BTERM# pin in C and J modes.*

The term "Burst Forever" was formerly used to describe "Continuous Burst."

Table 4-8. Local Bus Data Transfer Modes

Mode	Burst Enable Bit(s)	BTERM# Input Enable Bit(s)	Result
Single Cycle	0	X	One ADS# per data.
Burst-4	1	0	One ADS# per four data (recommended for i960 and PPC401).
Continuous Burst	1	1	One ADS# per data burst or until BTERM# is asserted.

Notes: *Single cycle is the default Data Transfer mode.*
"X" is "Don't Care."

4.2.5.1 Single Cycle Mode

Single Cycle mode is the default Data Transfer mode. In Single Cycle mode, the PCI 9656 issues one ADS# per data cycle. The starting address for a single cycle Data transfer can be on any address.

For single cycle Data transfers, Burst mode is disabled (LBRD0[24]=0 for Space 0, LBRD1[8]=0 for Space 1, LBRD0[26]=0 for Expansion ROM, and/or DMAMODE*x*[8]=0 for Channel *x*). For a 32-bit Local Bus, if a starting address in a Direct Slave or DMA PCI-to-Local transfer is ***not*** aligned to an Lword boundary, the PCI 9656 performs a single cycle until the next Lword boundary.

4.2.5.1.1 Partial Data Accesses

Partial Data accesses (not all Byte Enables are asserted) are broken into single cycles. If there is remaining data that is ***not*** Lword-aligned during DMA PCI-to-Local transfers, it results in a single cycle Data transfer.

4.2.5.2 Burst-4 Mode

Burst-4 mode forces the PCI 9656 to perform Data transfers as bursts of four data (4 Lwords, four 16-bit words, or 4 bytes to a 32-, 16-, or 8-bit bus, respectively).

Burst-4 mode Data transfers are set up by enabling bursting and clearing the BTERM# Input Enable bit(s) (LBRD0[24, 7]=10b for Space 0, LBRD1[8:7]=10b for Space 1, LBRD0[26, 23]=10b for Expansion ROM, and/or DMAMODE*x*[8:7]=10b for Channel *x*, respectively).

Burst-4 mode bursting starts on any data boundary and bursts to the next four-data boundary. The first burst starts on any address and goes to the data boundary. The next bursts are four data. The first or last burst may be less than four data. The PCI 9656 can continue to burst by asserting ADS# and performing another burst. For a 32-bit Local Bus, if a starting address in a Direct Slave or DMA PCI-to-Local transfer is ***not*** aligned to an Lword boundary, the PCI 9656 performs a single cycle until the next Lword boundary. (Refer to Table 4-9.)

For DMA, if Burst-4 mode is implemented with Address Increment disabled (DMAMODE*x*[11]=1), the PCI 9656 defaults to Continuous Burst mode.

Table 4-9. Burst-4 Mode

Local Bus Data Width	Burst-4
32 bit	4 Lwords start/stop at a 4-Lword boundary
16 bit	4 words start/stop at a 4-word boundary
8 bit	4 bytes start/stop at a 4-byte boundary

Note: *The first or last burst may be less than four data.*

4.2.5.2.1 Partial Data (<4 Bytes) Accesses

Partial Data accesses occur when one or more of the PCI Byte Enables (C/BE[7:0]#) are ***not*** asserted. For a 32-bit Local Bus, they are broken in single cycles until the next Lword boundary.

4.2.5.3 Continuous Burst Mode

Continuous Burst mode enables the PCI 9656 to perform data bursts of longer than four data. However, special external interface devices are required that can accept bursts longer than four data.

Continuous Burst mode Data transfers are set up by enabling bursting and setting the BTERM# Input Enable bit(s) (LBRD0[24, 7]=11b for Space 0, LBRD1 [8:7]=11b for Space 1, LBRD0[26, 23]=11b for Expansion ROM, and/or DMAMODE*x*[8:7]=11b for Channel *x*, respectively).

The PCI 9656 asserts one ADS# cycle and continues to burst data. If a slave device requires a new Address cycle (ADS#), it can assert the BTERM# input. The PCI 9656 completes the current Data transfer and stops the burst. The PCI 9656 continues the transfer by asserting ADS# and beginning a new burst at the next address.

For DMA, if Burst-4 mode is implemented with Address Increment disabled (DMAMODE*x*[11]=1), the PCI 9656 defaults to Continuous Burst mode.

4.2.6 Recovery States (J Mode Only)

In J mode, the PCI 9656 inserts one recovery state between the last Data transfer and the next Address cycle.

Note: *The PCI 9656 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, when the PCI 9656 is the Local Bus Master, the PCI 9656 reads only bytes corresponding to Byte Enables requested by the PCI Master applicable to 32 bits. For a 64-bit PCI Bus single cycle read, the Local Bus performs multiple Read cycles (with all Byte Enables on). For all Burst Read cycles, the PCI 9656 passes all the bytes and can be programmed to:

- Prefetch
- Perform Direct Slave 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 PCI 9656 DMA Controller(s) are written.

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

Direct PCI access to an 8- or 16-bit Local Bus results in the PCI Bus Lword/Qword being broken into multiple Local Bus transfers. For each transfer, Byte Enables are encoded to provide Local Address bits LA[1:0]. In J mode, LAD[1:0] also provide these address bits during the Address phase.

4.2.10 Local Bus Data Parity

Generation or use of Local Bus data parity is optional. The Local Bus Parity Check is passive and provides only 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 9656 data bus (DP[3:0]). "Even data parity" is asserted for each lane during Local Bus reads from the PCI 9656 and during PCI 9656 Master writes to the Local Bus.

Even data parity is checked for Direct Slave reads, Direct Master writes, and DMA Local Bus reads. If an error is detected, the PCI 9656 sets the Direct Master Write/Direct Slave Read Local Data Parity Check Error Status bit (INTCSR[7]=1) and asserts an interrupt (LSERR#), if enabled (INTCSR[0, 6]=11b, respectively). This occurs in the Clock cycle following the data being checked.

For applications where READY# is disabled in the PCI 9656 registers, an external pull-down resistor is required for READY# to have INTCSR[7] and the LSERR# interrupt to be set and asserted, respectively.

4.3 BIG ENDIAN/LITTLE ENDIAN

4.3.1 PCI Bus Data Bits Mapping onto Local Bus

Tables 4-10 through 4-13 clarify PCI 9656 signal mapping between the PCI and Local Buses during Big/Little Endian conversion.

Notes:

1. *During 64-bit PCI transfers, the lower 32 bits of the PCI Bus (AD[31:0]) are always mapped first.*
2. *In each Local Bus pin entry, **n-m** denotes that that row's PCI pin maps to Local Bus pin **m** during Local Bus cycle **n** that results from the PCI cycle (PCI-to-Local Bus transfers) or in the PCI cycle (Local-to-PCI Bus transfers). Examples follow.*

   C Mode:

   In Table 4-10 (refer to shaded entry), a Local Bus pin of "2-LD5" for PCI pin AD21 during 16-bit Little Endian Local Bus transfers denotes that during a PCI-to-Local Bus transfer, the value of PCI pin AD21 during each 32-bit PCI transfer occurs on Local Bus pin LD5 of the second resulting 16-bit Local Bus transfer. During a Local-to-PCI Bus transfer, it denotes that the value of PCI pin AD21 results from the value of Local Bus pin LD5 during the second 16-bit Local Bus transfer.

   In Table 4-11 (refer to shaded entry), a Local Bus pin of "2-LD21" for PCI pin AD21 during 16-bit Little Endian Local Bus transfers denotes that during a PCI-to-Local Bus transfer, the value of PCI pin AD21 during each 32-bit PCI transfer occurs on Local Bus pin LD21 of the second resulting 16-bit Local Bus transfer. During a Local-to-PCI Bus transfer, it denotes that the value of PCI pin AD21 results from the value of Local Bus pin LD21 during the second 16-bit Local Bus transfer.

   J Mode:

   In Table 4-12 (refer to shaded entry), a Local Bus pin of "2-LAD5" for PCI pin AD21 during 16-bit Little Endian Local Bus transfers denotes that during a PCI-to-Local Bus transfer, the value of PCI pin AD21 during each 32-bit PCI transfer occurs on Local Bus pin LAD5 of the second resulting 16-bit Local Bus transfer. During a Local-to-PCI Bus transfer, it denotes that the value of PCI pin AD21 results from the value of Local Bus pin LAD5 during the second 16-bit Local Bus transfer.

   In Table 4-13 (refer to shaded entry), a Local Bus pin of "2-LAD21" for PCI pin AD21 during 16-bit Little Endian Local Bus transfers denotes that during a PCI-to-Local Bus transfer, the value of PCI pin AD21 during each 32-bit PCI transfer occurs on Local Bus pin LAD21 of the second resulting 16-bit Local Bus transfer. During a Local-to-PCI Bus transfer, it denotes that the value of PCI pin AD21 results from the value of Local Bus pin LAD21 during the second 16-bit Local Bus transfer.
3. *Mappings occur only during Data phases. Addresses always map to/from PCI AD[31:0], as indicated in the 32-bit Little Endian column after the address translation specified in the Configuration registers is performed.*
4. *Big/Little Endian modes are selected by register bits and pin signals, depending on the Data phase type (Direct Master read/write, Direct Slave read/write, DMA PCI-to-Local Bus/Local-to-PCI Bus, and Configuration register read/write). [For further details, refer to the BIGEND register, and to the BIGEND# pin descriptions in Table 12-11, "C Mode Local Bus Pins," and Table 12-12, "J Mode Local Bus Pins."]*

Table 4-10. PCI Bus Data Bits Mapping onto Local Bus C Mode Low Byte Lane

PCI Pins Mapped 2nd (64-Bit Transfers Only)	PCI Pins Mapped 1st	C Mode Local Bus Pin Byte Lane Mode = 0 (BIGEND[4]=0)					
		Little Endian			Big Endian		
		32-Bit	16-Bit	8-Bit	32-Bit	16-Bit	8-Bit
AD32	AD0	1-LD0	1-LD0	1-LD0	1-LD24	1-LD8	1-LD0
AD33	AD1	1-LD1	1-LD1	1-LD1	1-LD25	1-LD9	1-LD1
AD34	AD2	1-LD2	1-LD2	1-LD2	1-LD26	1-LD10	1-LD2
AD35	AD3	1-LD3	1-LD3	1-LD3	1-LD27	1-LD11	1-LD3
AD36	AD4	1-LD4	1-LD4	1-LD4	1-LD28	1-LD12	1-LD4
AD37	AD5	1-LD5	1-LD5	1-LD5	1-LD29	1-LD13	1-LD5
AD38	AD6	1-LD6	1-LD6	1-LD6	1-LD30	1-LD14	1-LD6
AD39	AD7	1-LD7	1-LD7	1-LD7	1-LD31	1-LD15	1-LD7
AD40	AD8	1-LD8	1-LD8	2-LD0	1-LD16	1-LD0	2-LD0
AD41	AD9	1-LD9	1-LD9	2-LD1	1-LD17	1-LD1	2-LD1
AD42	AD10	1-LD10	1-LD10	2-LD2	1-LD18	1-LD2	2-LD2
AD43	AD11	1-LD11	1-LD11	2-LD3	1-LD19	1-LD3	2-LD3
AD44	AD12	1-LD12	1-LD12	2-LD4	1-LD20	1-LD4	2-LD4
AD45	AD13	1-LD13	1-LD13	2-LD5	1-LD21	1-LD5	2-LD5
AD46	AD14	1-LD14	1-LD14	2-LD6	1-LD22	1-LD6	2-LD6
AD47	AD15	1-LD15	1-LD15	2-LD7	1-LD23	1-LD7	2-LD7
AD48	AD16	1-LD16	2-LD0	3-LD0	1-LD8	2-LD8	3-LD0
AD48	AD17	1-LD17	2-LD1	3-LD1	1-LD9	2-LD9	3-LD1
AD50	AD18	1-LD18	2-LD2	3-LD2	1-LD10	2-LD10	3-LD2
AD51	AD19	1-LD19	2-LD3	3-LD3	1-LD11	2-LD11	3-LD3
AD52	AD20	1-LD20	2-LD4	3-LD4	1-LD12	2-LD12	3-LD4
AD53	AD21	1-LD21	2-LD5	3-LD5	1-LD13	2-LD13	3-LD5
AD54	AD22	1-LD22	2-LD6	3-LD6	1-LD14	2-LD14	3-LD6
AD55	AD23	1-LD23	2-LD7	3-LD7	1-LD15	2-LD15	3-LD7
AD56	AD24	1-LD24	2-LD8	4-LD0	1-LD0	2-LD0	4-LD0
AD57	AD25	1-LD25	2-LD9	4-LD1	1-LD1	2-LD1	4-LD1
AD58	AD26	1-LD26	2-LD10	4-LD2	1-LD2	2-LD2	4-LD2
AD59	AD27	1-LD27	2-LD11	4-LD3	1-LD3	2-LD3	4-LD3
AD60	AD28	1-LD28	2-LD12	4-LD4	1-LD4	2-LD4	4-LD4
AD61	AD29	1-LD29	2-LD13	4-LD5	1-LD5	2-LD5	4-LD5
AD62	AD30	1-LD30	2-LD14	4-LD6	1-LD6	2-LD6	4-LD6
AD63	AD31	1-LD31	2-LD15	4-LD7	1-LD7	2-LD7	4-LD7

Table 4-11. PCI Bus Data Bits Mapping onto Local Bus C Mode High Byte Lane

PCI Pins Mapped 2nd (64-Bit Transfers Only)	PCI Pins Mapped 1st	C Mode Local Bus Pin Byte Lane Mode = 1 (BIGEND[4]=1)					
		Little Endian			Big Endian		
		32-Bit	16-Bit	8-Bit	32-Bit	16-Bit	8-Bit
AD32	AD0	1-LD0	1-LD16	1-LD24	1-LD24	1-LD24	1-LD24
AD33	AD1	1-LD1	1-LD17	1-LD25	1-LD25	1-LD25	1-LD25
AD34	AD2	1-LD2	1-LD18	1-LD26	1-LD26	1-LD26	1-LD26
AD35	AD3	1-LD3	1-LD19	1-LD27	1-LD27	1-LD27	1-LD27
AD36	AD4	1-LD4	1-LD20	1-LD28	1-LD28	1-LD28	1-LD28
AD37	AD5	1-LD5	1-LD21	1-LD29	1-LD29	1-LD29	1-LD29
AD38	AD6	1-LD6	1-LD22	1-LD30	1-LD30	1-LD30	1-LD30
AD39	AD7	1-LD7	1-LD23	1-LD31	1-LD31	1-LD31	1-LD31
AD40	AD8	1-LD8	1-LD24	2-LD24	1-LD16	1-LD16	2-LD24
AD41	AD9	1-LD9	1-LD25	2-LD25	1-LD17	1-LD17	2-LD25
AD42	AD10	1-LD10	1-LD26	2-LD26	1-LD18	1-LD18	2-LD26
AD43	AD11	1-LD11	1-LD27	2-LD27	1-LD19	1-LD19	2-LD27
AD44	AD12	1-LD12	1-LD28	2-LD28	1-LD20	1-LD20	2-LD28
AD45	AD13	1-LD13	1-LD29	2-LD29	1-LD21	1-LD21	2-LD29
AD46	AD14	1-LD14	1-LD30	2-LD30	1-LD22	1-LD22	2-LD30
AD47	AD15	1-LD15	1-LD31	2-LD31	1-LD23	1-LD23	2-LD31
AD48	AD16	1-LD16	2-LD16	3-LD24	1-LD8	2-LD24	3-LD24
AD48	AD17	1-LD17	2-LD17	3-LD25	1-LD9	2-LD25	3-LD25
AD50	AD18	1-LD18	2-LD18	3-LD26	1-LD10	2-LD26	3-LD26
AD51	AD19	1-LD19	2-LD19	3-LD27	1-LD11	2-LD27	3-LD27
AD52	AD20	1-LD20	2-LD20	3-LD28	1-LD12	2-LD28	3-LD28
AD53	AD21	1-LD21	2-LD21	3-LD29	1-LD13	2-LD29	3-LD29
AD54	AD22	1-LD22	2-LD22	3-LD30	1-LD14	2-LD30	3-LD30
AD55	AD23	1-LD23	2-LD23	3-LD31	1-LD15	2-LD31	3-LD31
AD56	AD24	1-LD24	2-LD24	4-LD24	1-LD0	2-LD16	4-LD24
AD57	AD25	1-LD25	2-LD25	4-LD25	1-LD1	2-LD17	4-LD25
AD58	AD26	1-LD26	2-LD26	4-LD26	1-LD2	2-LD18	4-LD26
AD59	AD27	1-LD27	2-LD27	4-LD27	1-LD3	2-LD19	4-LD27
AD60	AD28	1-LD28	2-LD28	4-LD28	1-LD4	2-LD20	4-LD28
AD61	AD29	1-LD29	2-LD29	4-LD29	1-LD5	2-LD21	4-LD29
AD62	AD30	1-LD30	2-LD30	4-LD30	1-LD6	2-LD22	4-LD30
AD63	AD31	1-LD31	2-LD31	4-LD31	1-LD7	2-LD23	4-LD31

Table 4-12. PCI Bus Data Bits Mapping onto Local Bus J Mode Low Byte Lane

PCI Pins Mapped 2nd (64-Bit Transfers Only)	PCI Pins Mapped 1st	J Mode Local Bus Pin Byte Lane Mode = 0 (BIGEND[4]=0)					
		Little Endian			Big Endian		
		32-Bit	16-Bit	8-Bit	32-Bit	16-Bit	8-Bit
AD32	AD0	1-LAD0	1-LAD0	1-LAD0	1-LAD24	1-LAD8	1-LAD0
AD33	AD1	1-LAD1	1-LAD1	1-LAD1	1-LAD25	1-LAD9	1-LAD1
AD34	AD2	1-LAD2	1-LAD2	1-LAD2	1-LAD26	1-LAD10	1-LAD2
AD35	AD3	1-LAD3	1-LAD3	1-LAD3	1-LAD27	1-LAD11	1-LAD3
AD36	AD4	1-LAD4	1-LAD4	1-LAD4	1-LAD28	1-LAD12	1-LAD4
AD37	AD5	1-LAD5	1-LAD5	1-LAD5	1-LAD29	1-LAD13	1-LAD5
AD38	AD6	1-LAD6	1-LAD6	1-LAD6	1-LAD30	1-LAD14	1-LAD6
AD39	AD7	1-LAD7	1-LAD7	1-LAD7	1-LAD31	1-LAD15	1-LAD7
AD40	AD8	1-LAD8	1-LAD8	2-LAD0	1-LAD16	1-LAD0	2-LAD0
AD41	AD9	1-LAD9	1-LAD9	2-LAD1	1-LAD17	1-LAD1	2-LAD1
AD42	AD10	1-LAD10	1-LAD10	2-LAD2	1-LAD18	1-LAD2	2-LAD2
AD43	AD11	1-LAD11	1-LAD11	2-LAD3	1-LAD19	1-LAD3	2-LAD3
AD44	AD12	1-LAD12	1-LAD12	2-LAD4	1-LAD20	1-LAD4	2-LAD4
AD45	AD13	1-LAD13	1-LAD13	2-LAD5	1-LAD21	1-LAD5	2-LAD5
AD46	AD14	1-LAD14	1-LAD14	2-LAD6	1-LAD22	1-LAD6	2-LAD6
AD47	AD15	1-LAD15	1-LAD15	2-LAD7	1-LAD23	1-LAD7	2-LAD7
AD48	AD16	1-LAD16	2-LAD0	3-LAD0	1-LAD8	2-LAD8	3-LAD0
AD48	AD17	1-LAD17	2-LAD1	3-LAD1	1-LAD9	2-LAD9	3-LAD1
AD50	AD18	1-LAD18	2-LAD2	3-LAD2	1-LAD10	2-LAD10	3-LAD2
AD51	AD19	1-LAD19	2-LAD3	3-LAD3	1-LAD11	2-LAD11	3-LAD3
AD52	AD20	1-LAD20	2-LAD4	3-LAD4	1-LAD12	2-LAD12	3-LAD4
AD53	AD21	1-LAD21	2-LAD5	3-LAD5	1-LAD13	2-LAD13	3-LAD5
AD54	AD22	1-LAD22	2-LAD6	3-LAD6	1-LAD14	2-LAD14	3-LAD6
AD55	AD23	1-LAD23	2-LAD7	3-LAD7	1-LAD15	2-LAD15	3-LAD7
AD56	AD24	1-LAD24	2-LAD8	4-LAD0	1-LAD0	2-LAD0	4-LAD0
AD57	AD25	1-LAD25	2-LAD9	4-LAD1	1-LAD1	2-LAD1	4-LAD1
AD58	AD26	1-LAD26	2-LAD10	4-LAD2	1-LAD2	2-LAD2	4-LAD2
AD59	AD27	1-LAD27	2-LAD11	4-LAD3	1-LAD3	2-LAD3	4-LAD3
AD60	AD28	1-LAD28	2-LAD12	4-LAD4	1-LAD4	2-LAD4	4-LAD4
AD61	AD29	1-LAD29	2-LAD13	4-LAD5	1-LAD5	2-LAD5	4-LAD5
AD62	AD30	1-LAD30	2-LAD14	4-LAD6	1-LAD6	2-LAD6	4-LAD6
AD63	AD31	1-LAD31	2-LAD15	4-LAD7	1-LAD7	2-LAD7	4-LAD7

Table 4-13. PCI Bus Data Bits Mapping onto Local Bus J Mode High Byte Lane

PCI Pins Mapped 2nd (64-Bit Transfers Only)	PCI Pins Mapped 1st	J Mode Local Bus Pin Byte Lane Mode = 1 (BIGEND[4]=1)					
		Little Endian			Big Endian		
		32-Bit	16-Bit	8-Bit	32-Bit	16-Bit	8-Bit
AD32	AD0	1-LAD0	1-LAD16	1-LAD24	1-LAD24	1-LAD24	1-LAD24
AD33	AD1	1-LAD1	1-LAD17	1-LAD25	1-LAD25	1-LAD25	1-LAD25
AD34	AD2	1-LAD2	1-LAD18	1-LAD26	1-LAD26	1-LAD26	1-LAD26
AD35	AD3	1-LAD3	1-LAD19	1-LAD27	1-LAD27	1-LAD27	1-LAD27
AD36	AD4	1-LAD4	1-LAD20	1-LAD28	1-LAD28	1-LAD28	1-LAD28
AD37	AD5	1-LAD5	1-LAD21	1-LAD29	1-LAD29	1-LAD29	1-LAD29
AD38	AD6	1-LAD6	1-LAD22	1-LAD30	1-LAD30	1-LAD30	1-LAD30
AD39	AD7	1-LAD7	1-LAD23	1-LAD31	1-LAD31	1-LAD31	1-LAD31
AD40	AD8	1-LAD8	1-LAD24	2-LAD24	1-LAD16	1-LAD16	2-LAD24
AD41	AD9	1-LAD9	1-LAD25	2-LAD25	1-LAD17	1-LAD17	2-LAD25
AD42	AD10	1-LAD10	1-LAD26	2-LAD26	1-LAD18	1-LAD18	2-LAD26
AD43	AD11	1-LAD11	1-LAD27	2-LAD27	1-LAD19	1-LAD19	2-LAD27
AD44	AD12	1-LAD12	1-LAD28	2-LAD28	1-LAD20	1-LAD20	2-LAD28
AD45	AD13	1-LAD13	1-LAD29	2-LAD29	1-LAD21	1-LAD21	2-LAD29
AD46	AD14	1-LAD14	1-LAD30	2-LAD30	1-LAD22	1-LAD22	2-LAD30
AD47	AD15	1-LAD15	1-LAD31	2-LAD31	1-LAD23	1-LAD23	2-LAD31
AD48	AD16	1-LAD16	2-LAD16	3-LAD24	1-LAD8	2-LAD24	3-LAD24
AD48	AD17	1-LAD17	2-LAD17	3-LAD25	1-LAD9	2-LAD25	3-LAD25
AD50	AD18	1-LAD18	2-LAD18	3-LAD26	1-LAD10	2-LAD26	3-LAD26
AD51	AD19	1-LAD19	2-LAD19	3-LAD27	1-LAD11	2-LAD27	3-LAD27
AD52	AD20	1-LAD20	2-LAD20	3-LAD28	1-LAD12	2-LAD28	3-LAD28
AD53	AD21	1-LAD21	2-LAD21	3-LAD29	1-LAD13	2-LAD29	3-LAD29
AD54	AD22	1-LAD22	2-LAD22	3-LAD30	1-LAD14	2-LAD30	3-LAD30
AD55	AD23	1-LAD23	2-LAD23	3-LAD31	1-LAD15	2-LAD31	3-LAD31
AD56	AD24	1-LAD24	2-LAD24	4-LAD24	1-LAD0	2-LAD16	4-LAD24
AD57	AD25	1-LAD25	2-LAD25	4-LAD25	1-LAD1	2-LAD17	4-LAD25
AD58	AD26	1-LAD26	2-LAD26	4-LAD26	1-LAD2	2-LAD18	4-LAD26
AD59	AD27	1-LAD27	2-LAD27	4-LAD27	1-LAD3	2-LAD19	4-LAD27
AD60	AD28	1-LAD28	2-LAD28	4-LAD28	1-LAD4	2-LAD20	4-LAD28
AD61	AD29	1-LAD29	2-LAD29	4-LAD29	1-LAD5	2-LAD21	4-LAD29
AD62	AD30	1-LAD30	2-LAD30	4-LAD30	1-LAD6	2-LAD22	4-LAD30
AD63	AD31	1-LAD31	2-LAD31	4-LAD31	1-LAD7	2-LAD23	4-LAD31

4.3.2 Local Bus Big/Little Endian Mode Accesses

For each of the following transfer types, the PCI 9656 Local Bus can be independently programmed, by way of the BIGEND register, to operate in Little or Big Endian mode:

- Local Bus accesses to the PCI 9656 Configuration registers
- Direct Slave PCI accesses to Local Address Space 0, Space 1, and Expansion ROM
- DMA Channel *x* 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.*
For Little Endian to Big Endian conversion, only byte lanes are swapped, ***not*** *individual bits.*

4.4 SERIAL EEPROM

This section describes the use of the serial EEPROM with the PCI 9656.

The PCI 9656 supports 2K bit or 4K bit serial EEPROMs. The PCI 9656 requires serial EEPROMs that support sequential reads. (The PLX PCI 9656 Toolbox includes the latest information on supported serial EEPROMs.)

The PCI 9656 supports two serial EEPROM load lengths – long serial EEPROM and extra long serial EEPROM (refer to Figure 4-2):

- **Long Load Mode** – Default. The PCI 9656 loads 34, 16-bit words from the serial EEPROM if the Extra Long Load from Serial EEPROM bit is cleared (LBRD0[25]=0)
- **Extra Long Load Mode** – The PCI 9656 loads 50, 16-bit words from the serial EEPROM if the Extra Long Load from Serial EEPROM bit is set (LBRD0[25]=1)

The serial EEPROM can be read or written from the PCI or Local Buses, through either the Serial EEPROM Control register bits (CNTRL[31, 27:24]) or VPD capability.

The 3.3V serial EEPROM clock (EESK) is derived from the PCI clock. The PCI 9656 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.

During serial EEPROM loading, the PCI 9656 responds to Direct Slave accesses with a Retry or allows a Master Abort (refer to Section 4.4.2); Local Processor accesses are delayed by holding READY# de-asserted.

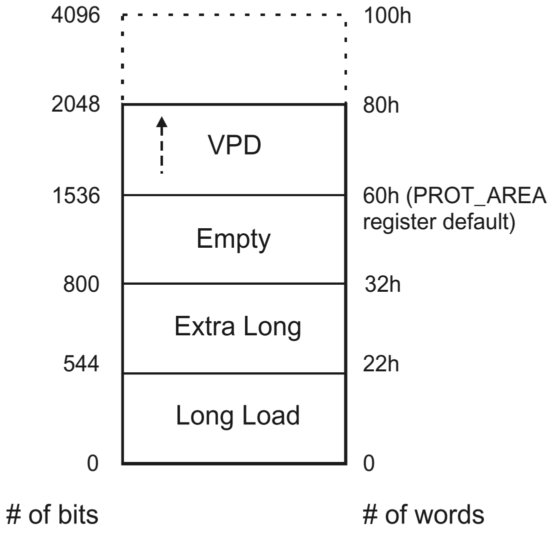


Figure 4-2. Serial EEPROM Memory Map

4.4.1 PCI 9656 Initialization from Serial EEPROM

After reset, the PCI 9656 attempts to read the serial EEPROM to determine its presence. A first-returned bit cleared to 0 indicates a serial EEPROM is present. The first word is then checked to verify that the serial EEPROM is programmed. If the first word (16 bits) is all ones (1), a blank serial EEPROM is present. If the first word (16 bits) is all zeros (0), no serial EEPROM is present. For both conditions, the PCI 9656 reverts to the default values. (Refer to Table 4-14.) The Serial EEPROM Present bit is set (CNTRL[28]=1) if the serial EEPROM is detected as present and is either programmed or blank.

Table 4-14. Serial EEPROM Guidelines

Local Processor	Serial EEPROM	System Boot Condition	Device which Sets LMISC1[2]
None	None	The PCI 9656 uses default values. The EEDI/EEDO pin must be pulled low – a 1KΩ resistor is required (rather than pulled high, which is typically done for this pin). If the PCI 9656 detects all zeros (0), it reverts to default values and sets the Local Init Status bit (LMISC1[2]=1) by itself.	PCI 9656
None	Programmed	Boot with serial EEPROM values. The Local Init Status bit must be set (LMISC1[2]=1) by the serial EEPROM.	Serial EEPROM
None	Blank	The PCI 9656 detects a blank device, reverts to default values, and sets the Local Init Status bit (LMISC1[2]=1) by itself.	PCI 9656
Present	None	The Local processor programs the PCI 9656 registers, then sets the Local Init Status bit (LMISC1[2]=1). A 1KΩ or greater value pull-up resistor or EEDI/EEDO is recommended, but not required. The EEDI/EEDO pin has an internal pull-up resistor. (Refer to Table 12-2, "Pins with Internal Pull-Up Resistors.") ***Note:*** *Some systems may avoid configuring devices that do not complete PCI Configuration accesses in less than 2^{15} PCI clocks after RST# is de-asserted. In addition, some systems may hang if Direct Slave reads and writes take too long (during initialization, the PCI Host also performs Direct Slave accesses). The value of the Direct Slave Retry Delay Clocks (LBRD0[31:28]) may resolve this.*	Local CPU
Present	Programmed	Load serial EEPROM, but the Local processor can reprogram the PCI 9656. Either the Local processor or the serial EEPROM must set the Local Init Status bit (LMISC1[2]=1).	Serial EEPROM or Local CPU
Present	Blank	The PCI 9656 detects a blank serial EEPROM and reverts to default values. ***Notes:*** *In some systems, the Local processor may be overly late to reconfigure the PCI 9656 registers before the BIOS configures them.* *The serial EEPROM can be programmed through the PCI 9656 after the system boots in this condition.*	PCI 9656

Note: *If the serial EEPROM is missing and a Local processor is present with blank Flash, the condition None/None (as listed in Table 4-14) applies, until the processor's Flash is programmed.*

4.4.2 Local Initialization and PCI Bus Behavior

The PCI 9656 issues a Retry or allows a Master Abort to all PCI accesses until the Local Init Status bit is set (LMISC1[2]=1). This bit can be programmed three ways:

1. By the Local processor, through the Local Configuration register.
2. By the serial EEPROM, during a serial EEPROM load, if the Local processor does not set this bit.
3. If the Local processor and the serial EEPROM are missing, or the serial EEPROM is blank (with or without a Local processor), the PCI 9656 uses defaults and sets this bit. (Refer to Table 4-14.)

PCI r2.2, Section 3.5.1.1, states:

"If the target is accessed during initialization-time, it is allowed to do any of the following:

1. Ignore the request (except if it is a boot device). This results in a Master Abort.
2. Claim the access and hold in wait sates until it can complete the request, not to exceed the end of initialization-time.
3. Claim the access and terminate with [PCI] Retry."

The PCI 9656 supports Option 1 *(Initially Not Respond)*, and Option 3 *(Initially Retry)*, above. For CompactPCI Hot Swap live insertion systems, the preferred method for the silicon is not to respond to PCI Configuration accesses during initialization. For legacy systems, Retries are usually preferred for compatibility. However, it is ultimately the designer's choice which option to use.

The PCI 9656 determines the option, as follows. The USERi pin is sampled at the rising edge of RST# to determine the selected PCI Bus response mode during Local Bus initialization. If USERi is low (through an external 1KΩ pull-down resistor), the PCI 9656 does not respond to PCI activity until the device's Local Bus initialization is complete. This results in a Master Abort (the preferred method for CompactPCI Hot Swap systems). If USERi is high (through an external 1KΩ to 4.7KΩ pull-up resistor), the PCI 9656 responds to PCI accesses with PCI Retry cycles until the device's Local Bus initialization is complete. Local Bus initialization is complete when the Local Init Status bit is set (LMISC1[2]=1).

During run time, USERi can be used as a general-purpose input. [Refer to Table 12-11, "C Mode Local Bus Pins" (C mode), or Table 12-12, "J Mode Local Bus Pins" (J mode).]

Refer to Section 9, "CompactPCI Hot Swap," for specifics on using this feature in *PICMG 2.1 R2.0*-compliant systems.

4.4.2.1 Long Serial EEPROM Load

If the Extra Long Load from Serial EEPROM bit is ***not*** set (LBRD0[25]=0), the registers listed in Table 4-15 are loaded from the serial EEPROM after a reset is de-asserted. The serial EEPROM is organized in words (16 bits). The PCI 9656 first loads the Most Significant Word bits (MSW[31:16]), starting from the Most Significant Bit (MSB[31]). The PCI 9656 then loads the Least Significant Word bits (LSW[15:0]), starting again from the Most Significant Bit (MSB[15]). Therefore, the PCI 9656 loads the Device ID, Vendor ID, Class Code, and so forth.

The serial EEPROM values can be programmed using a serial EEPROM programmer. The values can also be programmed using the PCI 9656 VPD function [refer to Section 10 "PCI Vital Product Data (VPD)"] or through the Serial EEPROM Control register (CNTRL).

Using the CNTRL register or VPD, the designer can perform reads and writes from/to the serial EEPROM, 32 bits at a time. Program serial EEPROM values in the order listed in Table 4-15. 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

If the Extra Long Load from Serial EEPROM bit is set (LBRD0[25]=1), the registers listed in Tables 4-15 and 4-16 are loaded from the serial EEPROM, with the Table 4-15 values loaded first.

Program serial EEPROM values in the order listed in Table 4-16. The 50 16-bit words listed in Tables 4-15 and 4-16 should be stored sequentially in the serial EEPROM, with the Table 4-15 values loaded first.

Table 4-15. Long Serial EEPROM Load Registers

Serial EEPROM Byte Offset	Description	Register Bits Affected
0h	PCI Device ID	PCIIDR[31:16]
2h	PCI Vendor ID	PCIIDR[15:0]
4h	PCI Class Code	PCICCR[23:8]
6h	PCI Class Code / Revision ID	PCICCR[7:0] / PCIREV[7:0]
8h	PCI Maximum Latency / PCI Minimum Grant	PCIMLR[7:0] / PCIMGR[7:0]
Ah	PCI Interrupt Pin / PCI Interrupt Line	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 Direct Slave Local Address Space 0 Range	LAS0RR[31:16]
16h	LSW of Direct Slave Local Address Space 0 Range	LAS0RR[15:0]
18h	MSW of Direct Slave Local Address Space 0 Local Base Address (Remap)	LAS0BA[31:16]
1Ah	LSW of Direct Slave Local Address Space 0 Local Base Address (Remap)	LAS0BA[15:2, 0], ***Reserved*** [1]
1Ch	MSW of Mode/DMA Arbitration	MARBR[31,29:16] or DMAARB[31, 29:16], ***Reserved*** [30]
1Eh	LSW of Mode/DMA Arbitration	MARBR[15:0] or DMAARB[15:0]
20h	Local Miscellaneous Control 2 / Serial EEPROM Write-Protected Address Boundary	LMISC2[5:0], ***Reserved*** [7:6] / PROT_AREA[6:0], ***Reserved*** [7]
22h	Local Miscellaneous Control 1 / Local Bus Big/Little Endian Descriptor	LMISC1[7:0] / BIGEND[7:0]
24h	MSW of Direct Slave Expansion ROM Range	EROMRR[31:16]
26h	LSW of Direct Slave Expansion ROM Range	EROMRR[15:11, 0], ***Reserved*** [10:1]
28h	MSW of Direct Slave Expansion ROM Local Base Address (Remap) and BREQo Control	EROMBA[31:16]
2Ah	LSW of Direct Slave Expansion ROM Local Base Address (Remap) and BREQo Control	EROMBA[15:11, 5:0], ***Reserved*** [10:6]
2Ch	MSW of Local Address Space 0/Expansion ROM Bus Region Descriptor	LBRD0[31:16]
2Eh	LSW of Local Address Space 0/Expansion ROM Bus Region Descriptor	LBRD0[15:0]
30h	MSW of Local Range for Direct Master-to-PCI	DMRR[31:16]
32h	LSW of Local Range for Direct Master-to-PCI *(**Reserved**)*	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 *(**Reserved**)*	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 *(**Reserved**)*	DMLBAI[15:0]
3Ch	MSW of PCI Base Address (Remap) for Direct Master-to-PCI Memory	DMPBAM[31:16]
3Eh	LSW of PCI Base Address (Remap) for Direct Master-to-PCI Memory	DMPBAM[15:0]
40h	MSW of PCI Configuration Address for Direct Master-to-PCI I/O Configuration	DMCFGA[31, 23:16], ***Reserved*** [30:24]
42h	LSW of PCI Configuration Address for Direct Master-to-PCI I/O Configuration	DMCFGA[15:0]

Table 4-16. Extra Long Serial EEPROM Load Registers

Serial EEPROM Byte Offset	Description	Register Bits Affected
44h	PCI Subsystem ID	PCISID[15:0]
46h	PCI Subsystem Vendor ID	PCISVID[15:0]
48h	MSW of Direct Slave Local Address Space 1 Range (1 MB)	LAS1RR[31:16]
4Ah	LSW of Direct Slave Local Address Space 1 Range (1 MB)	LAS1RR[15:0]
4Ch	MSW of Direct Slave Local Address Space 1 Local Base Address (Remap)	LAS1BA[31:16]
4Eh	LSW of Direct Slave Local Address Space 1 Local Base Address (Remap)	LAS1BA[15:2, 0], ***Reserved*** [1]
50h	MSW of Local Address Space 1 Bus Region Descriptor	LBRD1[31:16]
52h	LSW of Local Address Space 1 Bus Region Descriptor *(**Reserved**)*	LBRD1[15:0]
54h	Hot Swap Control/Status *(**Reserved**)*	***Reserved***
56h	Hot Swap Next Capability Pointer / Hot Swap Control	HS_NEXT[7:0] / HS_CNTL[7:0]
58h	***Reserved***	***Reserved***
5Ah	PCI Arbiter Control	PCIARB[3:0], ***Reserved*** [15:4]
5Ch	Power Management Capabilities	PMC[15:9, 2:0]
5Eh	Power Management Next Capability Pointer ***(Reserved)*** / Power Management Capability ID ***(Reserved)***	***Reserved***
60h	Power Management Data / PMCSR Bridge Support Extension ***(Reserved)***	PMDATA[7:0] / ***Reserved***
62h	Power Management Control/Status	PMCSR[14:8]

4.4.3 Serial EEPROM Access

The CNTRL[31, 27:24] register bits are programmed to control the EESK, EECS, and EEDI/EEDO settings. Bit 24 is used to generate EESK (clock), bit 25 controls the chip select, bit 26 controls the EEDI output logic level, and bit 31 enables the EEDO Input buffer. Bit 27, when read, returns the value of EEDI.

Setting bits [31, 25:24] to 1 causes the output to go high. A pull-up resistor is required on the EEDI/EEDO pin for the pin to go high when bit 31 is set. When reading the EEPROM, bit 31 must be set (CNTRL[31]=1).

To perform the read, the basic approach is to set the EECS and EEDO bits (bits [25, 31]=11b, 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 the EECS, EEDI, EEDO, and EESK Input Enable bits.
2. Set EECS high.
3. Toggle EESK high, then low.
4. Set the EEDI bit high (Start bit).
5. Toggle EESK high, then low.
6. Repeat step 5.
7. Clear EEDI.
8. Toggle EESK high, then low.
9. Toggle EESK high, then low 8 times (clock in Serial EEPROM Address 0).
10. Set EEDO Input Enable to 1 (CNTRL[31]=1) to float the EEDO input for reading.
11. Toggle EESK high, then low 16 times (clock in 1 word from serial EEPROM).
12. After each clock pulse, read bit 27 and save.
13. Clear the EECS bit.
14. Toggle EESK high, then low. The read is complete.

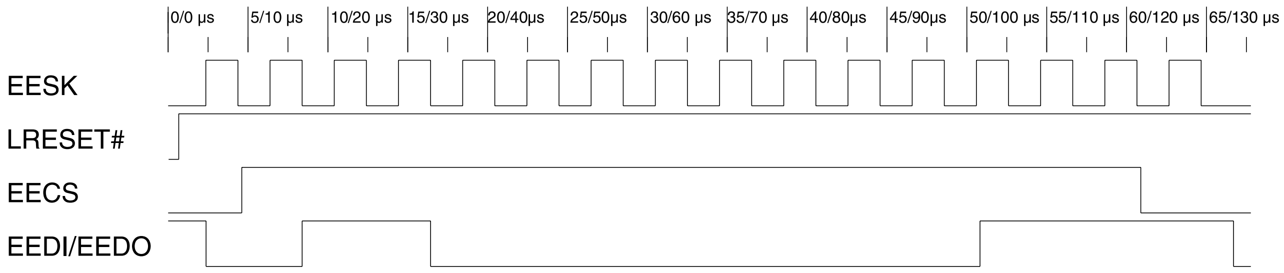
The serial EEPROM uses word addressing with 8-bit sequential address values. Due to PCI 9656 EEDI/EEDO signal multiplexing and to avoid contention between the least-significant address bit and the serial EEPROM's Start bit, Read access to an odd-numbered word address (bit 0 is set to 1) must be performed by using the even-numbered word address (bit 0 is cleared to 0), along with the serial EEPROM's sequential read functionality, clocking in 32 (or more) data bits while keeping EECS continually asserted during the Read access.

During a serial EEPROM Write operation, the serial EEPROM's programming cycle is triggered by EECS de-assertion. When EECS is re-asserted, the serial EEPROM drives its Ready status on its DO pin connected to the PCI 9656 EEDI/EEDO pin. (Refer to manufacturer's data sheet for Ready functionality.) Ready status (DO=1) indicates internal write completion. If the PCI 9656 is driving the EEDI (EEDI/EEDO pin) signal to a logic low when EECS is re-asserted after a Write operation, contention exists on the multiplexed EEDI/EEDO bus. The contention is released when Ready functionality is cleared, by clocking in a Start bit (causing the serial EEPROM to float its DO output). To remove contention following a Write cycle, re-assert EECS after waiting the Minimum CS Low Time, and, after Ready status goes high or the Write Cycle Time expires (typically 10 to 15 µs), set the EEDI output to logic high, then clock the Start bit into the serial EEPROM by toggling EESK high, then low. This Start bit can be used as part of the next serial EEPROM instruction, or, the cycle can be terminated by de-asserting EECS. Contention can also be avoided by setting the EEDI output high after clocking out the LSB of the data after each Write sequence.

The serial EEPROM can also be read or written, using the VPD function. [Refer to Section 10, "PCI Vital Product Data (VPD)."]

4.4.4 Serial EEPROM Initialization Timing Diagram

Timing Diagram 4-2. Serial EEPROM Start-Up (Serial EEPROM Absence Detected)



Notes: *The time scales provided are based on a 66 MHz/33 MHz PCI clock.*
The EEDI/EEDO signal is not sampled by the serial EEPROM until after EECS is asserted.
Upon LRESET# de-assertion, the PCI 9656 issues a Read command of Serial EEPROM Address 0.

For this timing diagram, no serial EEPROM is present; however, the Local processor is present to perform initialization. Therefore, the PCI 9656 does not detect a serial EEPROM Start bit because the EEDI/EEDO signal is sampled high (due to the internal pull-up resistor). Thereafter, the EESK, EECS and EEDI/EEDO signal pins are driven to 0.

If the PCI 9656 detects a serial EEPROM Start bit and the first 16 bits (which correspond to PCIIDR[31:16]) are not all zeros (0) or all ones (1), the PCI 9656 continues to assert EECS and generate the serial EEPROM clock (EESK) to load its registers with the programmed serial EEPROM values.
The first bit downloaded after the Start bit is PCIIDR[31].

Figure 4-3 illustrates the approximate amount of time required to detect the Start Bit and how much time is required for either a Long or Extra Long Serial EEPROM Load. While the PCI 9656 is downloading serial EEPROM data, PCI accesses to the PCI 9656 are Retried and Local Master accesses are accepted, but do not complete (READY# de-asserted) until the serial EEPROM download completes.

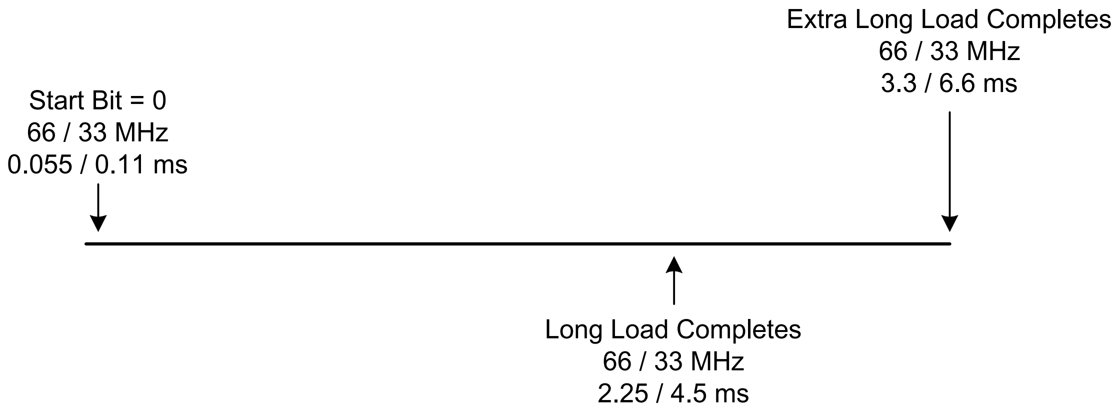


Figure 4-3. Serial EEPROM Timeline

4.5 INTERNAL REGISTER ACCESS

The PCI 9656 internal registers provide maximum flexibility in bus-interface control and performance. These registers are accessible from the PCI and Local Buses (refer to Figure 4-4) and include the following:

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

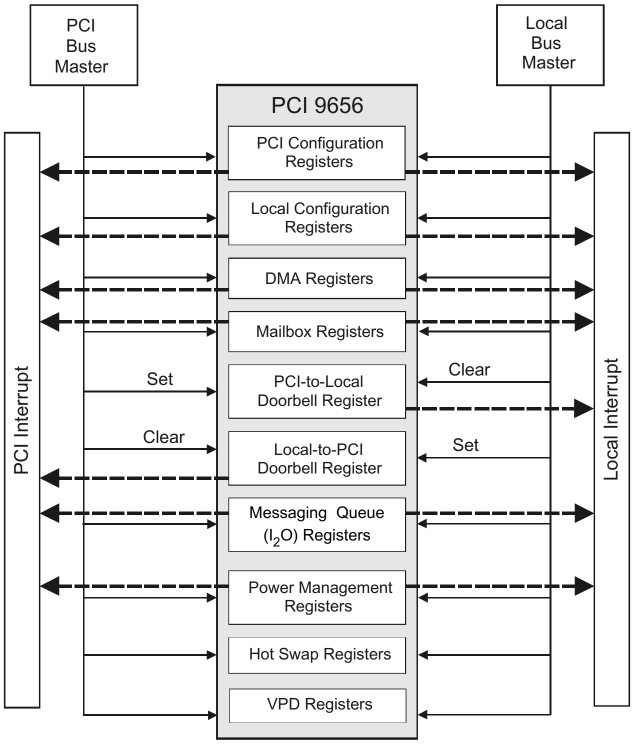

PCI
Bus
Master
Local
Bus
Master
PCI 9656
PCI Configuration Registers
Local Configuration Registers
DMA Registers
Mailbox Registers
PCI-to-Local Doorbell Register
Local-to-PCI Doorbell Register
Messaging Queue (I_2O) Registers
Power Management Registers
Hot Swap Registers
VPD Registers
Set
Clear
Clear
Set
PCI Interrupt
Local Interrupt

Figure 4-4. PCI 9656 Internal Register Access

4.5.1 PCI Bus Access to Internal Registers

The PCI 9656 PCI Configuration registers can be accessed from the PCI Bus with a Type 0 Configuration cycle.

All other PCI 9656 internal registers can be accessed by a Memory cycle, with the PCI Bus address that matches the Memory Base Address specified in the PCI Base Address for Memory Accesses to Local, Runtime, DMA, and Messaging Queue Registers register (PCIBAR0[31:9]). The PCI 9656 decodes 512 bytes for Memory-Mapped Configuration register access. These registers can also be accessed by an I/O cycle, with the PCI Bus address matching the I/O Base Address specified in the PCI Base Address for I/O Accesses to Local, Runtime, DMA, and Messaging Queue Registers register (PCIBAR1[31:8]). The PCI 9656 decodes the first 256 bytes for I/O-Mapped Configuration register access.

All PCI Read or Write accesses to the PCI 9656 registers can be Byte, Word, or Lword accesses. All PCI Memory accesses to the PCI 9656 registers can be Burst or Non-Burst accesses. The PCI 9656 responds with a PCI disconnect for all Burst I/O accesses (PCIBAR1[31:8]) to the PCI 9656 internal registers.

4.5.1.1 New Capabilities Function Support

The New Capabilities Function Support includes PCI Power Management, Hot Swap, and VPD features, as listed in Table 4-17.

If a New Capabilities Function is not used and passed over in the Capability Pointer's linked list, the unused New Capabilities Function registers are set to default values.

Table 4-17. New Capabilities Function Support Features

New Capability Function	PCI Register Offset Location
First (Power Management)	40h, if the New Capabilities Function Support bit is enabled (PCISR[4]=1; default). (Refer to Section 8, "PCI Power Management," for details about this feature.)
Second (Hot Swap)	48h, which is pointed to from PMNEXT[7:0]. (Refer to Section 9, "CompactPCI Hot Swap," for details about this feature.)
Third (VPD)	4Ch, which is pointed to from HS_NEXT[7:0]. Because PVPD_NEXT[7:0] defaults to 0h, this indicates that VPD is the last New Capability Function Support feature of the PCI 9656. [Refer to Section 10, "PCI Vital Product Data (VPD)," for details about this feature.]

4.5.2 Local Bus Access to Internal Registers

The Local processor can access all PCI 9656 internal registers through an external chip select. The PCI 9656 responds to a Local Bus access when the PCI 9656 Configuration Chip Select input (CCS#) is asserted low during the Address phase.

Local Read or Write accesses to the PCI 9656 internal registers can be Byte, Word, or Lword accesses. Local accesses to the PCI 9656 internal registers can be Burst or Non-Burst accesses.

The PCI 9656 READY# signal indicates that the Data transfer is complete.

THIS PAGE INTENTIONALLY LEFT BLANK.

5 C AND J MODES FUNCTIONAL DESCRIPTION

The functional operation described in this section can be modified through the PCI 9656 programmable internal registers.

5.1 RESET OPERATION

5.1.1 Adapter Mode

The PCI 9656 operates in Adapter mode when the HOSTEN# pin is strapped high.

5.1.1.1 PCI Bus RST# Input

The PCI Bus RST# input pin is a PCI Host reset. It causes all PCI Bus outputs to float, resets the entire PCI 9656 and causes Local LRESET# to assert. LEDon# is asserted during PCI Bus RST# assertion for CompactPCI Hot Swap systems. Most Local Bus output signals are set to float while LRESET# remains asserted, with the exceptions listed in Table 5-1.

Table 5-1. Local Bus Output Signals that Do Not Float when RST# Is Asserted

Signal	Value when RST# Is Asserted
EECS	0
EESK	0
LEDon#	0
LRESET#	0
PME#	X (undefined)
USERo	X (tri-stated)

Note: *When HOSTEN#=1 and during a hard reset (RST# = 0) USERo is tri-stated. On the second rising edge of LCLK after RST# is de-asserted, LRESET# de-asserts. On the next falling edge of LCLK, the USERo pin is driven to 0 (from tri-state). On the next rising edge of LCLK, USERo is driven to the value specified in the General-Purpose Output bit (CNTRL[16]=1, default).*

5.1.1.2 JTAG Reset TRST# Input

The TRST# input pin is the asynchronous JTAG logic reset. TRST# assertion causes the PCI 9656 Test Access Port (TAP) controller to initialize. In addition, when the TAP controller is initialized, it selects the PCI 9656 normal logic path (core-to-I/O). It is recommended that the designer take the following into consideration when implementing the asynchronous JTAG logic reset on a board:

- If JTAG functionality is required, the TRST# input signal should use a low-to-high transition once during the PCI 9656 boot-up, along with the RST# signal.
- If JTAG functionality is not required, the TRST# signal should always be directly connected to ground.

5.1.1.3 Software Reset

When the Software Reset bit is set (CNTRL[30]=1), the following functional blocks are reset:

- All Local Bus logic
- Local Configuration and Messaging Queue registers
- PCI and Local Bus DMA logic
- FIFOs

The PCI Bus logic, PCI Configuration DMA and Runtime registers, and the Local Init Status bit (LMISC1[2]) are not reset.

When the Software Reset bit is set (CNTRL[30]=1), the PCI 9656 responds to PCI Configuration, Runtime, and DMA registers’ accesses, initiated from the PCI Bus. Because the Local Bus is in reset, accesses are ***not*** allowed by the Local Bus. The PCI 9656 remains in this reset condition until the PCI Host clears the bit (CNTRL[30]=0). 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, although the Local processor does not use LRESET# to reset.*

5.1.1.4 Power Management Reset

When the Power Management reset is asserted (transition from D_3 to any other power state), the PCI 9656 resets as if a PCI reset was asserted. (Refer to Section 8, “PCI Power Management.”)

5.1.2 Host Mode

The PCI 9656 operates in Host mode when the HOSTEN# pin is strapped low.

5.1.2.1 Local Reset

The PCI Bus RST# output is driven when Local LRESET# is asserted or the Software Reset bit is set (CNTRL[30]=1). Most PCI and Local Bus output signals are set to float, with the exceptions listed in Table 5-2.

Table 5-2. PCI and Local Bus Output Signals that Do Not Float when LRESET# Is Asserted

Signal	Value when LRESET# Is Asserted
AD[63:0]	0
C/BE[7:0]#	0
EECS	0
EESK	0
LEDon#	0
PAR	0
PAR64	0
PME#	X (undefined)
REQ64#	0
RST#	0

5.1.2.2 Software Reset

When the Software Reset bit is set (CNTRL[30]=1), the following functional blocks are reset:

- PCI Master logic
- PCI Slave logic
- PCI and Local Bus DMA logic
- PCI 9656 PCI Configuration, Mailbox, and DMA registers
- FIFOs
- PCI interrupts/lock
- Internal PCI Arbiter
- Power Management interrupts

The Local Bus logic, Local Configuration, Runtime, and Messaging Queue registers are not reset. A software reset can be cleared only from a host on the Local Bus, and the PCI 9656 remains in this reset condition until a Local Host clears the bit (CNTRL[30]=0). When the Software Reset bit is set (CNTRL[30]=1), the PCI 9656 responds to the Local Configuration, Runtime, and DMA registers’ accesses initiated from the Local Bus.

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

When HOSTEN#=1 and the Soft Reset bit is set (CNTRL[30]=1), the LRESET# and USERo signals are asserted low (0). When the Soft Reset bit is cleared (CNTRL[30]=0), USERo reverts to the value specified in the General-Purpose Output bit (CNTRL[16]), one LCLK after the LRST# signal is de-asserted (driven to 1).

5.1.2.3 Power Management Reset

Power Management reset is not applicable for Host mode.

5.2 PCI 9656 INITIALIZATION

The PCI 9656 Configuration registers can be programmed by an optional serial EEPROM and/or by a Local processor, as listed in Table 4-14, “Serial EEPROM Guidelines.” The serial EEPROM can be reloaded by setting the Reload Configuration Registers bit (CNTRL[29]=1).

The PCI 9656 Retries all PCI cycles or allows Master Aborts 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 5-3 lists the response of the PCI 9656 to full and empty FIFOs.

Table 5-3. Response to FIFO Full or Empty

Mode	Direction	FIFO	PCI Bus	Local Bus
Direct Master Write	Local-to-PCI	Full	Normal	De-asserts READY#
		Empty	De-asserts REQ# (releases the PCI Bus)	Normal
Direct Master Read	PCI-to-Local	Full	De-asserts REQ# or throttles IRDY#[1]	Normal
		Empty	Normal	De-asserts READY#
Direct Slave Write	PCI-to-Local	Full	Disconnects or throttles TRDY#[2]	Normal
		Empty	Normal	De-asserts LHOLD or retains the Local Bus, asserts BLAST#[3]
Direct Slave Read	Local-to-PCI	Full	Normal	De-asserts LHOLD or retains the Local Bus, asserts BLAST#[3]
		Empty	Throttles TRDY#[4]	Normal
DMA	Local-to-PCI	Full	Normal	Asserts BLAST#, de-asserts LHOLD[5]
		Empty	De-asserts REQ#	Normal
	PCI-to-Local	Full	De-asserts REQ#	Normal
		Empty	Normal	Asserts BLAST#, de-asserts LHOLD[5]

[1] Throttle IRDY# depends on the Direct Master PCI Read Mode bit being set (DMPBAM[4]=1).

[2] Throttle TRDY# depends on the Direct Slave PCI Write Mode bit being set (LBRD0[27]=1).

[3] LHOLD de-assertion depends upon the Local Bus Direct Slave Release Bus Mode bit being set (MARBR[21]=1).

[4] If PCI Compliance is enabled (MARBR[24]=1), PCI Retry is issued instead of throttling TRDY#.

[5] BLAST# de-assertion depends upon the DMA Channel x Fast/Slow Terminate Mode Select bit setting (DMAMODEx[15]).

5.4 DIRECT DATA TRANSFER MODES

In C and J modes, the PCI 9656 supports three Direct Data Transfer modes:

- **Direct Master** – Local CPU accesses PCI memory or I/O
- **Direct Slave** – PCI Master accesses Local memory or I/O
- **DMA** – PCI 9656 DMA Controller(s) reads/writes PCI memory to/from Local memory

All three Direct Data Transfer modes generate dummy cycles. The PCI 9656 generates dummy cycles on the PCI and Local Buses. The PCI Bus C/BE[7:0]# pins and the Local Bus LBE[3:0]# pins must be monitored so that dummy cycles can be recognized on either bus.

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

The PCI 9656 supports a direct access of the PCI Bus by the Local processor or an intelligent controller. Master mode must be enabled in the PCI Command register (PCICR[2]=1). The following registers define Local-to-PCI accesses (refer to Figure 5-1):

- 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)

Important Note: The PCI 9656 generates dummy cycles on the PCI Bus. (Refer to Sections 5.4 and 5.4.1.3.1.)

5.4.1.1 Direct Master Memory and I/O Decode

The Range register and the Local Base Address specify the Local Address bits to use for decoding a Local-to-PCI access (Direct Master). The Memory or I/O space range must be a power of 2 and the Range register value must be two’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 9656. All Direct Master cycles are then decoded as PCI Memory, I/O, or Type 0 or Type 1 Configuration. Moreover, a Direct Master Memory or I/O cycle is remapped according to the Remap register value, which must be a multiple of the Direct Master range value (not the Range register value).

The PCI 9656 can accept Memory cycles only from the Local processor. The Local Base Address and range determine whether PCI Memory or PCI I/O transactions occur.

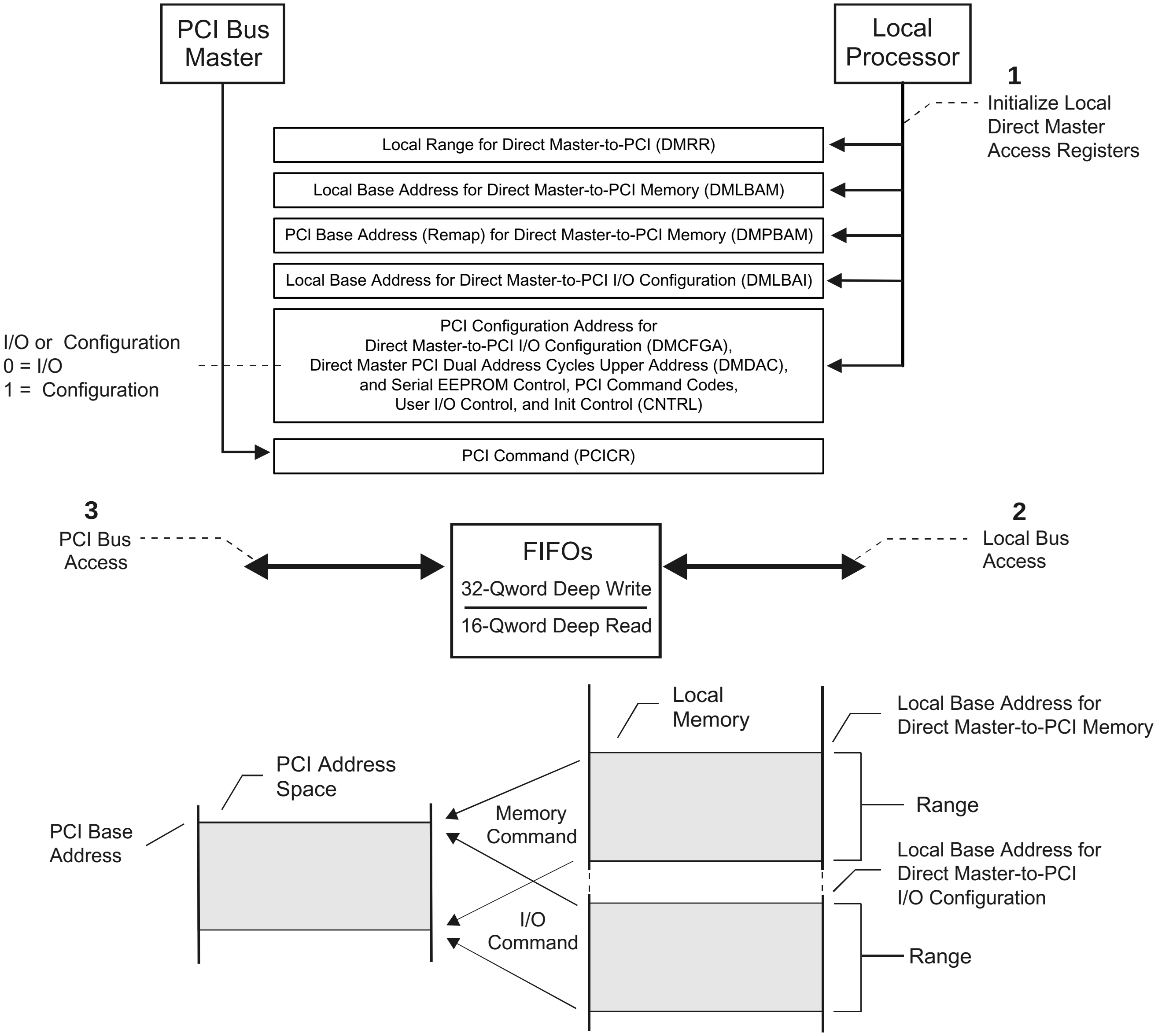

PCI Bus Master
Local Processor
1
Initialize Local Direct Master Access Registers
Local Range for Direct Master-to-PCI (DMRR)
Local Base Address for Direct Master-to-PCI Memory (DMLBAM)
PCI Base Address (Remap) for Direct Master-to-PCI Memory (DMPBAM)
Local Base Address for Direct Master-to-PCI I/O Configuration (DMLBAI)
PCI Configuration Address for Direct Master-to-PCI I/O Configuration (DMCFGA), Direct Master PCI Dual Address Cycles Upper Address (DMDAC), and Serial EEPROM Control, PCI Command Codes, User I/O Control, and Init Control (CNTRL)
I/O or Configuration
0 = I/O
1 = Configuration
PCI Command (PCICR)
3
PCI Bus Access
FIFOs
32-Qword Deep Write
16-Qword Deep Read
2
Local Bus Access
Local Memory
Local Base Address for Direct Master-to-PCI Memory
PCI Address Space
PCI Base Address
Memory Command
Range
Local Base Address for Direct Master-to-PCI I/O Configuration
I/O Command
Range

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

5.4.1.2 Direct Master FIFOs

For Direct Master Memory access to the PCI Bus, the PCI 9656 has a 64-Lword/32-Qword (256-byte) Write FIFO and a 32-Lword/16-Qword (128-byte) Read FIFO. The FIFOs enable the Local Bus to operate independently 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 Figures 5-2 and 5-3.

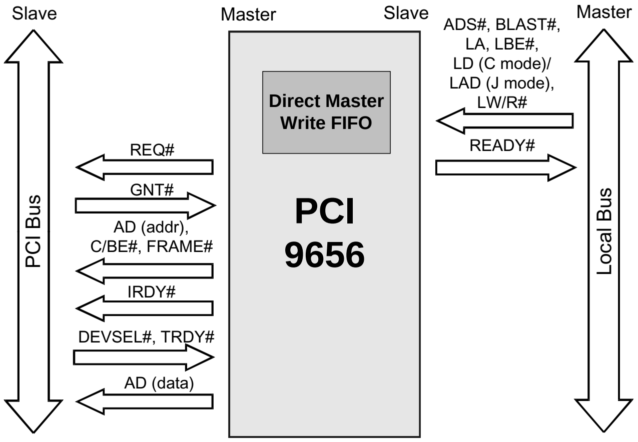


Figure 5-2. Direct Master Write

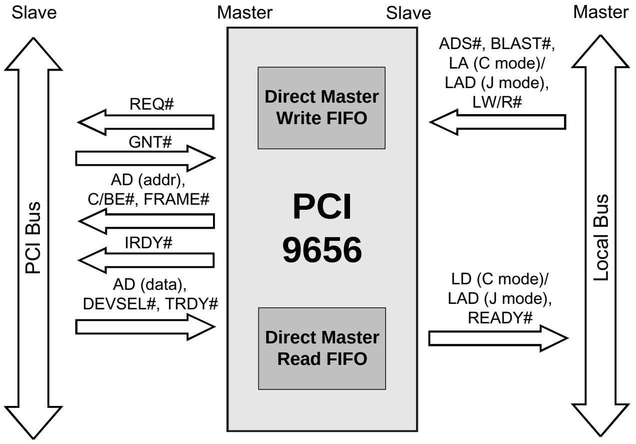


Figure 5-3. Direct Master Read

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 9656 and PCI Bus.

The PCI 9656 converts the Local Read/Write access to PCI Bus accesses. The Local Address space starts from the Direct Master Local Base Address up to the range. Remap (PCI Base Address) defines the PCI starting address.

A Local Bus Processor single cycle causes a single cycle PCI transaction. A Local processor Burst cycle causes a PCI Burst transaction. The PCI 9656 supports continuous Burst transfers.

Transactions are initiated by the Local Bus Processor (a Local Bus Master) when the Memory address on the Local Bus matches the Memory space decoded for Direct Master operations.

5.4.1.3.1 Direct Master Writes

For 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 9656 becomes the PCI Bus Master, arbitrates for the PCI Bus, and writes data to the PCI Slave device. The PCI 9656 continues to accept writes and returns READY# until the Direct Master Write FIFO is full. It then holds READY# de-asserted until space becomes available in the Direct Master Write FIFO. A programmable Direct Master “almost full” status output is provided (DMPAF signal). (Refer to DMPBAM[10, 8:5].)

Local Bus processor single cycle Write transactions result in PCI 9656 transfers of 1 Lword of data onto a 32-bit PCI Bus. A Qword-aligned single cycle write to a 64-bit PCI Bus results in a 64-bit PCI Write cycle, with PCI Byte Enables (C/BE[7:0]#) asserted as F0h. If the single cycle write is ***not*** Qword-aligned (*for example*, X4h or XCh), the PCI Bus Write cycle is 32 bits (REQ64# de-asserted).

A Local processor, with no burst limitations and a Burst cycle Write transaction of 2 Lwords, causes the PCI 9656 to perform two single writes on a 32-bit PCI Bus. The same Local transfer to a 64-bit PCI Bus results in the PCI 9656 transferring 1 Qword, with all PCI Byte Enables (C/BE[7:0]#=00h) asserted. Three (or more) Lword Burst cycles of any type result in the PCI 9656 bursting data onto the PCI Bus.

Direct Master writes on the Local Bus may generate Dummy Cycle writes on the PCI Bus (dummy cycles are C/BE[7:0]#=FFh for the 64-bit PCI Bus, and C/BE[7:0]#=XFh for the 32-bit PCI Bus).

- **64-bit PCI Bus** – No dummy cycles, although one-half of the Qword on the PCI Bus may have its C/BE[3:0]#=Fh or C/BE[7:4]#=Fh.
- **32-bit PCI Bus, with a Local Bus burst greater than two** – If the Local starting address is Qword-aligned, and the burst count is odd, the last PCI Write cycle is a dummy cycle, and the PCI starting address is Qword-aligned.

  If the Local starting address is Lword-aligned, but not Qword-aligned, and the burst count is even, the first and last PCI Write cycles are dummy cycles and the PCI starting address is Qword-aligned.

  If the Local starting address is Lword-aligned, but not Qword-aligned, and the burst count is odd, the first PCI Write cycle is a dummy cycle and the PCI starting address is Qword-aligned.

Other 32-bit PCI Bus cases do not generate dummy cycles.

5.4.1.3.2 Direct Master Reads

For a Local Bus read, the PCI 9656 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. The PCI 9656 asserts Local Bus READY# to indicate that the requested data is on the Local Bus.

The PCI 9656 holds READY# de-asserted while receiving data from the PCI Bus. Programmable prefetch modes are available if prefetch is enabled – prefetch, 4, 8, 16, or continuous data – 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 9656 does not prefetch PCI data for single cycle (Local BLAST# input is asserted during the clock following ADS#) Direct Master reads unless Read Ahead mode is enabled and prefetch length is set to continuous (DMPBAM[2, 11]=10b, respectively). (Refer to Section 5.4.1.6 for details on Read Ahead mode.) For single cycle Direct Master reads of a 32-bit PCI Bus (Read Ahead mode disabled, DMPBAM[2]=0), the PCI 9656 reads 1 Lword from the PCI Bus. For single cycle Direct Master reads of a 64-bit PCI Bus (Read Ahead mode disabled) starting at a Qword boundary, the PCI 9656 reads 1 Qword from the PCI Bus, with C/BE[7:0]#=F0h. If the starting address is not a Qword boundary (*for example*, X4h or XCh), the PCI 9656 performs a 32-bit PCI read (REQ64# de-asserted).

For single cycle Direct Master reads, the PCI 9656 passes the Local Byte Enables (LBE[3:0]#) to the PCI Byte Enables.

For Burst Cycle reads, the PCI 9656 reads entire Lwords or Qwords (all PCI Byte Enables are asserted), dependent upon the PCI Bus data width.

If the Direct Master Prefetch Limit bit is enabled (DMPBAM[11]=1), the PCI 9656 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 (DMPBAM[11]=0), the prefetch crosses the 4-KB boundaries.

If the 4-KB Prefetch Limit bit is enabled, and the PCI 9656 started a Direct Master read to a 64-bit PCI Bus PCI Address FF8h (Qword-aligned, 1 Qword before the 4-KB boundary), without ACK64# acknowledgment from the PCI Slave, the PCI 9656 does ***not*** perform a Burst prefetch of 2 Lwords. The PCI 9656 instead performs a prefetch of two single cycle Lwords to prevent crossing the PCI 4-KB boundary limit. If the PCI Slave responds with ACK64#, the PCI 9656 performs a single cycle read of 1 Qword and terminates to prevent crossing the 4-KB limit boundary. The cycle then restarts at the new boundary.

5.4.1.4 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 Configuration Enable bit is set (DMCFGA[31]=1), the PCI cycle becomes a PCI Configuration cycle. (Refer to Section 5.4.1.7.1 and the DMCFGA register for details.)

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. When the I/O Remap Select bit is cleared (DMPBAM[13]=0), DMPBAM[31:16] are used as the remap addresses for the PCI address.

For writes, data is loaded into the Direct Master Write FIFO and READY# is returned to the Local Bus. For reads, the PCI 9656 holds READY# de-asserted while receiving an Lword from the PCI Bus.

Local Burst accesses are broken into single PCI I/O (Address/Data) cycles. The PCI 9656 does not prefetch Read data for I/O and Configuration reads.

For Direct Master I/O and Configuration cycles, the PCI 9656 derives the PCI Byte Enables from the Local Byte Enables.

5.4.1.5 Direct Master Delayed Write Mode

The PCI 9656 supports Direct Master Delayed Write mode transactions, in which posted Write data accumulates in the Direct Master Write FIFO before the PCI 9656 requests the PCI Bus. Direct Master Delayed Write mode is programmable to delay REQ# assertion for the amount of PCI clocks selected 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 9656 only uses the Delay Counter and accumulates data in the Direct Master Write FIFO for Burst transactions on the Local Bus. A single cycle write on the Local Bus immediately starts a PCI cycle (ignores delay setting).

5.4.1.6 Direct Master Read Ahead Mode

The PCI 9656 only supports Direct Master Read Ahead mode (DMPBAM[2]=1) when the Direct Master Read Prefetch Size Control bits are set to continuous prefetch (DMPBAM[12, 3]=00b). Other Direct Master Read Prefetch Size Control bit settings force Direct Master Read Ahead mode off. Direct Master Read Ahead mode allows prefetched data to be read from the PCI 9656 internal FIFO instead of the PCI Bus. The address must be subsequent to the previous address and 32-bit-aligned (next address = current address + 4). Direct Master Read Ahead mode functions can be used with or without Direct Master Delayed Read mode.

A Local Bus single cycle Direct Master transaction, with Direct Master Read Ahead mode enabled results in the PCI 9656 processing continuous PCI Bus Read Burst data with all bytes enabled (C/BE[7:0]#=00h), if the Direct Master Read Prefetch Size Control bits are set to continuous prefetch (DMPBAM[12, 3]=00b). (Refer to Table 5-4.)

Caution: *To prevent constant locking of the PCI Bus, do* ***not*** *simultaneously enable Direct Master Read Ahead and Direct Master PCI Read modes (DMPBAM[2, 4]≠11b, respectively).*

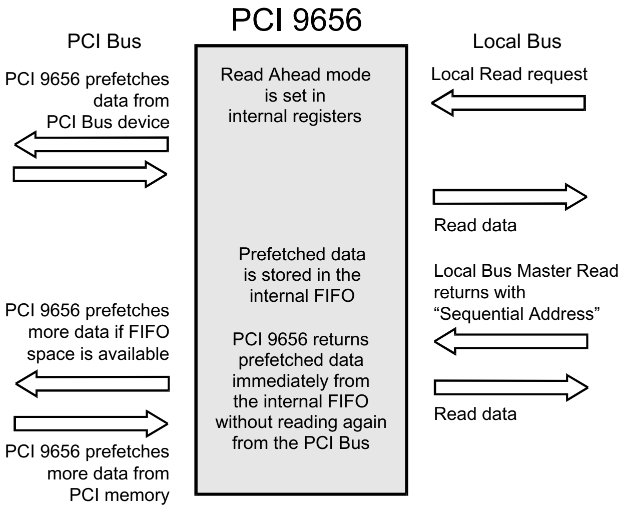

PCI 9656
PCI Bus
Local Bus
PCI 9656 prefetches data from PCI Bus device
Read Ahead mode is set in internal registers
Local Read request
Read data
Prefetched data is stored in the internal FIFO
Local Bus Master Read returns with “Sequential Address”
PCI 9656 prefetches more data if FIFO space is available
PCI 9656 returns prefetched data immediately from the internal FIFO without reading again from the PCI Bus
Read data
PCI 9656 prefetches more data from PCI memory

Figure 5-4. Direct Master Read Ahead Mode

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

Table 5-4. Example of PCI Bus Behavior with Direct Master Read Ahead Mode Enabled

Direct Master Local Bus Cycle to the PCI 9656	Direct Master PCI Bus Behavior with Direct Master Read Prefetch Size Control Bits Not Set to Continuous Prefetch (DMPBAM[12, 3]≠00b)	Direct Master PCI Bus Behavior with Direct Master Read Prefetch Size Control Bits Set to Continuous Prefetch (DMPBAM[12, 3]=00b)
Single cycle read followed by single cycle read	Two single cycle reads	Burst cycle with Read Ahead
Single cycle read followed by consecutive address Burst Read cycle	One single cycle followed by Burst cycle	Burst cycle followed by restart of Burst cycle with Read Ahead
Burst cycle read of four data followed by consecutive address Burst cycle read of four data	Burst cycle read followed by restart of Burst cycle read	Burst read followed by a second burst read with a contiguous address
Burst cycle read of four data followed by consecutive address single cycle read	Burst cycle read followed by restart of a single cycle read	Burst read followed by a second burst read with a contiguous address

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

If the Configuration Enable bit is set (DMCFGA[31]=1), a Configuration access is made to the PCI Bus. In addition to enabling this bit, the user must provide all register information. (Refer to the DMCFGA register.) The Register Number and Device Number bits (DMCFGA[7:2, 15:11], respectively) must be modified and a new Direct Master Read/Write cycle must be performed at the Direct Master I/O base address for each Configuration register. Before performing Non-Configuration cycles, the Configuration Enable bit must be cleared (DMCFGA[31]=0).

If the PCI Configuration Address register selects a Type 0 command, DMCFGA[10:0] are copied to address bits [10:0]. DMCFGA[15:11] (Device Number) are translated into a single bit being set in PCI Address bits [31:11]. The PCI Address bits [31:11] can be used as a device select. For a Type 1 command, DMCFGA[23:0] are copied to PCI address bits [23:0]. The PCI Address bits [31:24] are cleared 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 9656 holds READY# de-asserted while receiving an Lword from the PCI Bus.

5.4.1.7.1 Direct Master Configuration Cycle Example

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

1. The PCI 9656 must be configured to allow Direct Master access to the PCI Bus. The PCI 9656 must also be enabled to master the PCI Bus (PCICR[2]=1).

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

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 two’s complement of 00100000h:

   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 user must know which PCI device and PCI Configuration register the PCI Configuration cycle is accessing. This example assumes the Target PCI device IDSEL signal is connected to AD[21] (logical device #10=0Ah). Also, access PCIBAR0 (the fourth register, counting from 0; use Table 11-2, “PCI Configuration Register Address Mapping,” for reference). Set DMCFGA[31, 23:0], as listed in Table 5-5.

After these registers are configured, a single Local Master Memory cycle to the I/O base address is necessary to generate a PCI Configuration Read or Write cycle. The PCI 9656 takes the Local Bus Master Memory cycle and checks for the Configuration Enable bit (DMCFGA[31]). If set to 1, the PCI 9656 converts the current cycle to a PCI Configuration cycle, using the DMCFGA register and the Write/Read signal (LW/R#).

Table 5-5. Direct Master Configuration Cycle Example – DMCFGA[31, 23:0] Settings

Bits	Description	Value
1:0	Type 0 Configuration.	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.	0h
31	Configuration Enable.	1

5.4.1.8 Direct Master PCI Dual Address Cycles

The PCI 9656 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 64-bit address to devices that support 64-bit addressing when the address is not in the lower 4-GB Address space. The PCI 9656 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]#) "Dh" and the second PCI address will be a Hi-Addr with the command (C/BE[3:0]#) "6h" or "7h", depending upon it being a PCI Read or a PCI Write cycle. (Refer to Figure 5-5.) When the DMDAC register contains a value of 0h (feature enabled when DMDAC register value is ***not*** 0h), the PCI 9656 performs a Single Address Cycle (SAC) on the PCI Bus.

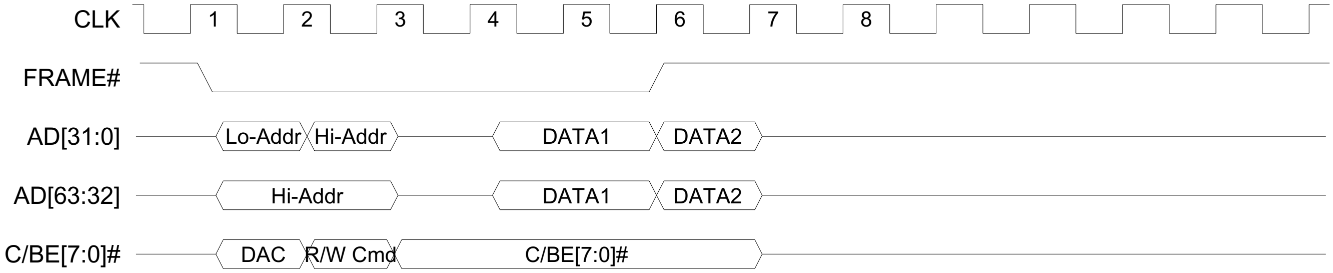


Figure 5-5. PCI Dual Address Cycle Timing

5.4.1.9 PCI Master/Target Abort

The following describes how the PCI 9656 logic handles a PCI Master/Target Abort.

As a PCI master, if the PCI 9656 attempts an 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 (PCISR[13]=0) and continue by processing the next task.

If a PCI Master/Target Abort or 256 consecutive Master Retry timeout is encountered during a transfer, the PCI 9656 asserts LSERR#, if enabled [INTCSR [0]=1, which can be used as a Non-Maskable Interrupt (NMI)]. (Refer to Section 6.1.13 for specific LSERR# behavior.)

When Direct Master writes are posted and a PCI Master/Target Abort occurs, this causes LSERR# assertion, if enabled (INTCSR[0]=1). If a Local Bus Master is waiting for READY#, it is asserted along with BTERM# only for Direct Master Read operations. The Local Bus Master's interrupt handler can take the appropriate application-specific action. It can then clear the Received Master or Target Abort bit (PCISR[13 or 12]=1, respectively; writing 1 clears the bit to 0) to de-assert the LSERR# interrupt and re-enable Direct Master transfers.

If a PCI Master/Target Abort or Retry Timeout is encountered during a Direct Master transfer with multiple Direct Master operations posted in the Direct Master Write FIFO (*for example*, an address, followed by data, followed by another address, followed by data), the PCI 9656 flushes the entire Direct Master Write FIFO if Direct Master Write FIFO Flush during PCI Master Abort is disabled (LMISC1[3]=0). If the bit is enabled (LMISC1[3]=1), the PCI 9656 flushes only the aborted Direct Master operation in the Direct Master Write FIFO and skips to the next available address in the FIFO entry. The PCI 9656 does not arbitrate on the PCI Bus until the Received Master/Target Abort bits are cleared (PCISR[13:12]=00b, respectively).

If a Local Bus Master is attempting a Burst read from a defective PCI device (Master/Target Abort), it receives READY# and BTERM# until the Local Bus completes a transfer. In addition, the PCI 9656 asserts LSERR# if INTCSR[1:0]=11b (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 a Direct Master transfer, the PCI 9656 stores the PCI Abort Address into PABTADR[31:0] and sets the Received Master/Target Abort bits (PCISR [13:12]=11b, respectively). The PCI 9656 disables all PCI Master capabilities when PCISR[13:12]=11b. Thus, in this condition, the PCI 9656 does not arbitrate for the PCI Bus to execute new or pending Direct Master or DMA transfers. The Received Master/ Target Abort bits should be read and then cleared before starting new Direct Master or DMA transfers. The PCI Abort Address register bits (PABTADR[31:0]) contents are updated for each PCI Master/Target Abort. (For details about DMA PCI Master/Target Abort, refer to Section 5.4.4.16.)

The PCI 9656 PCI Master/Target Abort logic can be used by 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).

5.4.1.10 Direct Master Memory Write and Invalidate

The PCI 9656 can be programmed to perform Memory Write and Invalidate (MWI) cycles to the PCI Bus for Direct Master transfers, as well as DMA transfers. (Refer to Section 5.4.4.4.) The PCI 9656 supports MWI 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 9656 performs Write transfers (using the command code programmed in CNTRL [7:4]) rather than MWI transfers.

Direct Master MWI transfers are enabled when the MWI Mode and the MWI Enable bits are set (DMPBAM[9]=1 and PCICR[4]=1, respectively).

In MWI mode, if the start address of the Direct Master transfer is on a cache line boundary, the PCI 9656 waits until the number of Lwords required for the specified cache line size are written from the Local Bus before starting a PCI MWI 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 9656 starts a PCI Write access using the PCI command codes from the CNTRL[15:12] bits, and the PCI 9656 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 9656 completes the remainder of that cache line, using normal writes.

If the Direct Master write is less than the cache line size, the PCI 9656 waits until the next Direct Master write begins before starting the PCI write. This PCI write retains the MWI command, regardless of the number of Lwords in the Direct Master Write FIFO, which may result in the PCI Bus completing an MWI cycle with less than a cache line of data. For this reason, Burst writes should be equal to, or multiples of, the cache line size.

5.4.1.11 Direct Master Write FIFO Programmable Almost Full, DMPAF Flag

The PCI 9656 supports the Direct Master Write FIFO Programmable Full Flag (DMPBAM[10, 8:5]. The DMPAF signal can be used as a hardware indicator Direct Master Write FIFO Full Flag. An alternative method (software polling a register bit) is also provided to check the Direct Master Write FIFO Full Status Flag (MARBR[30]). To enable DMPAF signal functionality and disable EOT#, clear the DMA EOT# Enable bit(s) (DMAMODE*x*[14]=0; default configuration).

DMPAF output assertion relies on the programmable value in DMPBAM[10, 8:5] to determine when to signal that the Direct Master Write FIFO is almost full. After DMPAF assertion, the PCI 9656 de-asserts DMPAF upon the last word of the transfer entering into the internal PCI Data Out Holding latch. The DMPAF signal indicates the Direct Master Write FIFO status, ***not*** transfer status completion.

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

For Direct Slave writes, the PCI Bus writes data to the Local Bus. Direct Slave is the “Command from the PCI Host” that has the highest priority.

For Direct Slave reads, the PCI Bus Master reads data from the Local Bus Slave. Direct Slave or Direct Master preempts DMA; however, Direct Slave does ***not*** preempt Direct Master. (Refer to Section 5.4.2.10.)

The PCI 9656 supports Burst Memory-Mapped Transfer accesses and I/O-Mapped, Single-Transfer PCI-to-Local Bus accesses through a 32-Lword/16-Qword (128-byte) Direct Slave Read FIFO and a 64-Lword/32-Qword (256-byte) Direct Slave Write FIFO. The PCI Base Address registers are provided to set the adapter location in PCI Memory and I/O space. In addition, Local mapping registers allow address translation from the PCI Address Space to the Local Address Space.

Three Local Address Spaces are available:

- Space 0
- Space 1
- Expansion ROM (Expansion ROM is intended to support a bootable Host ROM device)

Direct Slave supports on-the-fly Endian conversion for Space 0, Space 1, and Expansion ROM space.

Each Local space can be programmed to operate with an 8-, 16-, or 32-bit Local Bus data width. The PCI 9656 includes an internal wait state generator and READY# input signal that can be used to externally assert wait states. READY# can be selectively enabled or disabled for each Local Address Space (LBRD0[6] for Space 0, LBRD1[6] for Space 1, and/or LBRD0[22] for Expansion ROM).

Independent of the PCI Bus, the Local Bus can perform:

- Bursts as long as data is available (Continuous Burst mode)
- Bursts of 4 Lwords, words, or bytes at a time (Burst-4 mode, recommended)
- Continuous single cycles (default)
- Dummy cycles, which are generated on the Local Bus (refer to Sections 5.4 and 5.4.2.1)

The Direct Slave Retry Delay Clock bits (LBRD0 [31:28] can be used to program the period of time in which the PCI 9656 holds TRDY# de-asserted. The PCI 9656 issues a Retry to the PCI Bus Transaction Master when the programmed time period expires. This occurs when the PCI 9656 cannot gain control of the Local Bus and return TRDY# within the programmed time period.

5.4.2.1 Direct Slave Writes

For 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 9656 arbitrates for the Local Bus, becomes the Local Bus Master, and writes data to a Local slave device. The PCI 9656 continues to accept writes and asserts TRDY# until the Direct Slave Write FIFO is full. It then holds TRDY# de-asserted until space becomes available in the Direct Slave Write FIFO, or asserts STOP# and Retries the PCI Bus Master, dependent upon the LBRD0[27] register bit setting.

The PCI 9656 transfers dummy data by way of a dummy cycle, with LBE[3:0]#=Fh at the end of each transfer when the following conditions are true:

- **64-bit PCI Bus with BTERM# input enabled** – If the last Qword on the PCI Bus has Byte Enables of F0h, the last Lword on the Local Bus has an LBE[3:0]# of Fh (dummy cycle).
- **64-bit PCI Bus with BTERM# input disabled** – If the last Qword on the PCI Bus has Byte Enables of F0h and the Lword with Byte Enables = Fh corresponds to the last Lword of a Local Bus burst of four, the last Lword on the Local Bus has an LBE[3:0]# of Fh.
- **64-bit PCI Bus with BTERM# input enabled or disabled** – If any Qword on the PCI Bus has Byte Enables of F0h that are not for the first or last Lword, the Local burst is broken and the last Lword of the Local burst is a dummy cycle. After the dummy cycle, the burst is restarted with a new ADS#.
- **32-bit PCI Bus with BTERM# input enabled** – If the number of Lwords written on the PCI Bus is odd, and adequate for the Local Bus to burst data, the last Lword on the Local Bus has an LBE[3:0]# of Fh (dummy cycle).
- **32-bit PCI Bus with BTERM# input enabled or disabled** – If any Lword on the PCI Bus has Byte Enables of Fh that are not for the first or last Lword, the Local burst is broken and the last Lword of the Local burst is a dummy cycle. After the dummy cycle, the burst is restarted with a new ADS#.
- If the PCI data is of insufficient length to cause the Local Bus to burst data, there are no dummy cycles.

A 32-bit PCI Bus Master single cycle Write transaction results in a PCI 9656 transfer of 1 Lword of data onto the Local Bus. A 64-bit PCI Bus Master Qword data single cycle write results in a PCI 9656 Burst transfer of 2 Lwords onto the Local Bus, if the Burst bit(s) is enabled (LBRD0[24]=1 for Space 0, LBRD1[8]=1 for Space 1, and/or LBRD0[26]=1 for Expansion ROM).

The PCI 9656 can be programmed to retain the PCI Bus by generating a wait state(s) and de-asserting TRDY#, if the Direct Slave Write FIFO becomes full. The PCI 9656 can also be programmed to retain the Local Bus and continue asserting LHOLD, if the Direct Slave Write FIFO becomes empty. If the Local Bus Latency Timer is enabled (MARBR[16]=1) and expires (MARBR[7:0]), the Local Bus is dropped.

A Burst Write cycle from the PCI Bus through the PCI 9656 performs a Local processor Burst transaction, if the following conditions are true:

- The address is Qword-aligned (64-bit PCI Bus) or Lword-aligned (32-bit PCI Bus),
- The Direct Slave Write FIFO contains at least 2 Lwords, and
- All PCI Byte Enables are set.

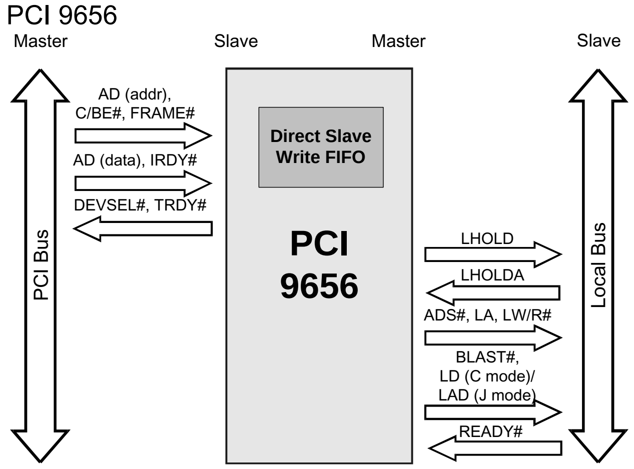


Figure 5-6. Direct Slave Write

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

5.4.2.2 Direct Slave Reads

The PCI 9656 holds TRDY# de-asserted while receiving an Lword from the Local Bus, unless the PCI Compliance Enable bit is set (MARBR[24]=1). (Refer to Section 5.4.2.4.) Programmable Prefetch modes are available, if prefetch is enabled – prefetch, 1 to 16, or continuous data – until the Direct Slave read ends. The Local prefetch continues until several clocks after FRAME# is de-asserted or the PCI 9656 issues a Retry or disconnect. Unused data is flushed from the FIFO unless Direct Slave Read Ahead mode is enabled (MARBR[28]=1).

For the highest data transfer rate, the PCI 9656 can be programmed to prefetch data during a PCI Burst read on the Local Bus. When Prefetch is enabled (LBRD0[8]=0 for Space 0, LBRD1[9]=0 for Space 1, and/or LBRD0[9]=0 for Expansion ROM), prefetch can be from 1 to 16 Lwords or until the PCI Bus stops requesting. When the PCI 9656 prefetches, it drops the Local Bus after reaching the Prefetch Counter limit. In Continuous Prefetch mode, the PCI 9656 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 9656 disconnects from the Local Bus after each Read transfer.

The PCI 9656 64-bit PCI Bus Direct Slave unaligned Qword Data Prefetch Read transfers are special cases that result in prefetching one additional Lword (32 bits) of Local Bus data than specified in the Prefetch Counter(s) (LBRD0[14:11] and/or LBRD1[14:11]) to sustain zero wait state 64-bit PCI Data transfers. For 64-bit Qword-aligned and all 32-bit PCI Bus Direct Slave Prefetch Read transfers, the PCI 9656 prefetches the amount specified in the Prefetch Counter(s). If a Direct Slave Prefetch Burst Read transfer is performed to a Local Bus device that cannot burst, and READY# is asserted for only one clock period, the PCI 9656 retains the Read data in the Direct Slave Read FIFO without ever completing the Read transfer to the PCI Bus. This occurs because the PCI 9656 is built with 64-bit FIFO architecture, and a prefetch mechanism is required to prefetch further data for transferring to occur. Under such conditions, it is necessary to disable a Prefetch or Burst feature.

In addition to Prefetch mode, the PCI 9656 supports Direct Slave Read Ahead mode (MARBR[28]=1). (Refer to Section 5.4.2.5.)

Only 32-bit PCI Bus single cycle Direct Slave Read transactions result in the PCI 9656 passing requested PCI Byte Enables (C/BE[3:0]#) to a Local Bus Target device by way of LBE[3:0]# assertion back to a PCI Bus master. This transaction results in the PCI 9656 reading 1 Lword or partial Lword data. For all other types of Read transactions (64-bit PCI Bus Burst transfers or Unaligned), the PCI 9656 reads Local Bus data with all bytes asserted (LBE[3:0]#=0h).

The PCI 9656 disconnects after one transfer for all Direct Slave I/O accesses.

The PCI 9656 can be programmed to retain the Local Bus and continue asserting LHOLD if the Direct Slave Read FIFO becomes full. If the Local Bus Latency Timer is enabled (MARBR[16]=1) and expires (MARBR[7:0]), the Local Bus is dropped.

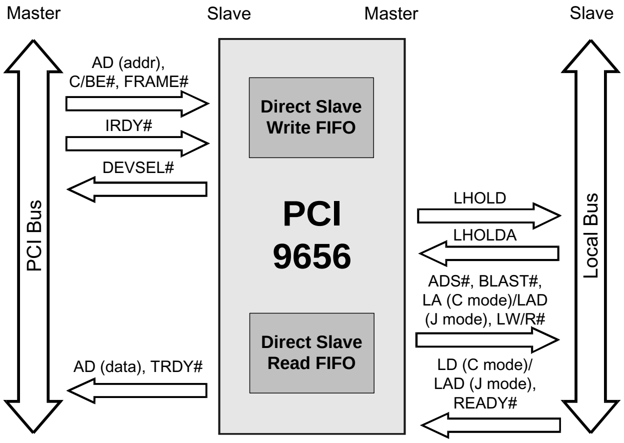


Figure 5-7. Direct Slave Read

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

5.4.2.3 Direct Slave Lock

The PCI 9656 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 PCI LOCK# Input Enable bit (MARBR[22]) for PCI-to-Local accesses.

5.4.2.4 PCI Compliance Enable

The PCI 9656 can be programmed through the PCI Compliance Enable bit to perform all PCI Read/Write transactions in compliance with *PCI r2.2* (MARBR [24]=1). The following sections describe the behavior of the PCI 9656 when MARBR[24]=1.

5.4.2.4.1 Direct Slave Delayed Read Mode

PCI Bus single cycle aligned or unaligned 32-bit Direct Slave 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[3:0]#) unless the PCI Read No Flush Mode bit is enabled (MARBR[28]=1). (Refer to Section 5.4.2.5.) This causes the PCI 9656 to Retry all PCI Bus Read requests that follow, until the original PCI Byte Enables are matched.

The PCI Bus single cycle aligned or unaligned 64-bit Direct Slave Delayed Read transactions always result in 2-Lword Prefetch transfers on the Local Bus, with all Local Byte Enables (LBE[3:0]#=0h) asserted to successfully complete Read Data transfers to a 64-bit PCI Bus. This causes the PCI 9656 to always return requested data to a 64-bit PCI master, although the PCI Byte Enables do not match the original request.

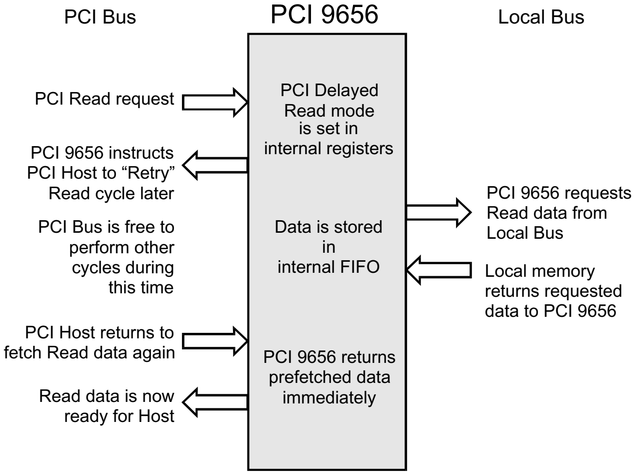

PCI Bus
PCI 9656
Local Bus
PCI Read request
PCI 9656 instructs PCI Host to "Retry" Read cycle later
PCI Bus is free to perform other cycles during this time
PCI Host returns to fetch Read data again
Read data is now ready for Host
PCI Delayed Read mode is set in internal registers
Data is stored in internal FIFO
PCI 9656 returns prefetched data immediately
PCI 9656 requests Read data from Local Bus
Local memory returns requested data to PCI 9656

Figure 5-8. Direct Slave Delayed Read Mode

Note: *Figure 5-8 represents a sequence of Bus cycles.*

5.4.2.4.2 2^{15} PCI Clock Timeout

If the PCI Master does not complete the originally requested Direct Slave Delayed Read transfer, the PCI 9656 flushes the Direct Slave Read FIFO after 2^{15} PCI clocks and grants an access to a new Direct Slave Read access. All other Direct Slave Read accesses before the 2^{15} PCI clock timeout are Retried by the PCI 9656.

5.4.2.4.3 *PCI r2.2* 16- and 8-Clock Rule

The PCI 9656 guarantees that if the first Direct Slave Write data cannot be accepted by the PCI 9656 and/or the first Direct Slave Read data cannot be returned by the PCI 9656 within 16 PCI clocks from the beginning of the Direct Slave cycle (FRAME# asserted), the PCI 9656 issues a Retry (STOP# asserted) to the PCI Bus.

During successful Direct Slave Read and/or Write accesses, the subsequent data after the first access must be accepted for writes or returned for reads within 8 PCI clocks (TRDY# asserted). Otherwise, the PCI 9656 issues a PCI disconnect (STOP# asserted) to the PCI Master. In addition, the PCI 9656 supports the following *PCI r2.2* functions:

- No write while a Delayed Read is pending (PCI Retries for writes) (MARBR[25])
- Write and flush pending Delayed Read (MARBR[26])

5.4.2.5 Direct Slave Read Ahead Mode

The PCI 9656 also supports Direct Slave Read Ahead mode (MARBR[28]=1), where prefetched data can be read from the internal FIFO of the PCI 9656 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 and 64-bit-aligned (next address = current address + 8) for 64-bit Direct Slave transfers. Direct Slave Read Ahead mode functions with or without Direct Slave Delayed Read mode.

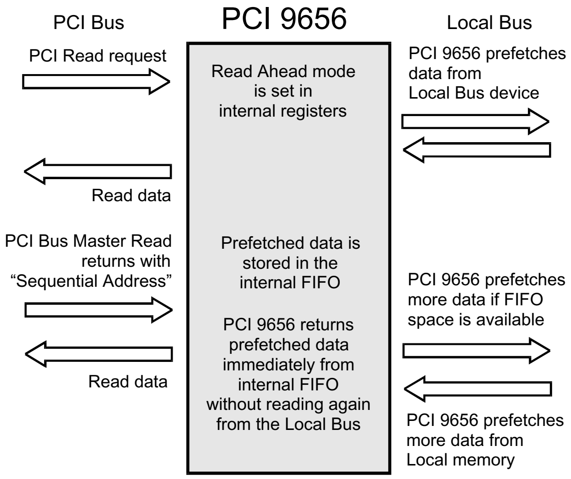

PCI Bus
PCI 9656
Local Bus
PCI Read request
Read Ahead mode is set in internal registers
PCI 9656 prefetches data from Local Bus device
Read data
PCI Bus Master Read returns with "Sequential Address"
Prefetched data is stored in the internal FIFO
PCI 9656 prefetches more data if FIFO space is available
PCI 9656 returns prefetched data immediately from internal FIFO without reading again from the Local Bus
Read data
PCI 9656 prefetches more data from Local memory

Figure 5-9. Direct Slave Read Ahead Mode

Note: *Figure 5-9 represents a sequence of Bus cycles.*

5.4.2.6 Direct Slave Delayed Write Mode

The PCI 9656 supports Direct Slave Delayed Write mode transactions, in which posted Write data accumulates in the Direct Slave Write FIFO before the PCI 9656 requests ownership of the Local Bus (LHOLD assertion). The Direct Slave Delayed Write mode is programmable to delay the LHOLD assertion for the amount of Local clocks specified in 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.

5.4.2.7 Direct Slave Local Bus READY# Timeout Mode

Direct Slave Local Bus READY# Timeout mode is enabled/disabled by LMISC2[0].

If Direct Slave Local Bus READY# Timeout mode is disabled (LMISC2[0]=0), *and* the PCI 9656 is mastering the Local Bus during a Direct Slave Read or Write Data transfer, *and* no Local Bus device asserts READY# in response to the Local Bus address generated by the PCI 9656, the PCI 9656 hangs on the Local Bus waiting for READY# assertion. To recover, reset the PCI 9656.

If Direct Slave Local Bus READY# Timeout mode is enabled (LMISC2[0]=1), *and* the PCI 9656 is mastering the Local Bus during a Direct Slave Read or Write Data transfer, the PCI 9656 uses the READY# Timeout Select bit (LMISC2[1]) to determine when to timeout while waiting for a Local Bus device to assert READY# in response to the Local Bus address generated by the PCI 9656. When LMISC2[1]=0, the PCI 9656 waits 32 Local Bus clocks for READY# assertion before timing out. When LMISC2[1]=1, the PCI 9656 waits 1,024 Local Bus clocks for READY# assertion before timing out.

If a Direct Slave Local Bus READY# Timeout occurs during a Direct Slave read, the PCI 9656 issues a Target Abort to the PCI Bus Master. If the PCI Bus Master is currently mastering the PCI 9656 (*that is*, the PCI 9656 is not awaiting a PCI Bus Master Retry of the Direct Slave read), the Target Abort is immediately issued. If the PCI Bus Master is not currently mastering the PCI 9656 (*that is*, the PCI 9656 is awaiting a PCI Bus Master Retry of the Direct Slave read), the PCI 9656 issues a Target Abort the next time the PCI Bus Master repeats the Direct Slave read.

If a Direct Slave Local Bus READY# Timeout occurs during a Direct Slave write *and* the PCI Bus Master is currently mastering the PCI 9656 (*that is*, the PCI Bus Master has not posted the Direct Slave write to the PCI 9656), the PCI 9656 immediately issues a Target Abort to the PCI Bus Master. If the PCI Bus Master is not currently mastering the PCI 9656 (*that is*, the PCI Bus Master has posted the Direct Slave write to the PCI 9656), the PCI 9656 does ***not*** issue to the PCI Bus Master any indication that the timeout occurred.

Caution: *Only use the PCI 9656 Direct Slave Local Bus READY# Timeout feature during design debug. This feature allows illegal Local Bus addresses to be easily detected and debugged. Once the design is debugged and legal addresses are guaranteed, disable Direct Slave Local Bus READY# Timeout mode to avoid unreported timeouts during Direct Slave writes.*

5.4.2.8 Direct Slave Transfer Error

The PCI 9656 supports Local Bus error conditions that use LSERR# as an input. A Local Bus device may assert LSERR#, either before or simultaneously with READY#. In either case, the PCI 9656 tries to complete the current transaction by transferring data and then asserting ADS# for every address that follows, waiting for another READY# or LSERR# to be issued (used to flush the Direct Slave FIFOs). To prevent bursting, hold LSERR# asserted until the Direct Slave transfer completes. After sensing LSERR# is asserted, the PCI 9656 asserts PCI SERR# and sets an error flag, using the Signaled System Error (SERR# Status) bit (PCISR[14]=1). When set, this indicates a catastrophic error occurred on the Local Bus. SERR# may be masked off by resetting the LSERR# Input Interrupt Mask bit (LMISC1[5]=0).

5.4.2.9 Direct Slave PCI-to-Local Address Mapping

Note: *Not applicable in I_2O mode.*

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:

- Direct Slave Local Address Space Range (LAS0RR, LAS1RR, and/or EROMRR)
- Direct Slave Local Address Space Local Base Address (Remap) (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 5.4.2.9.1. These Local Bus characteristics can be modified at any time before actual data transactions.

5.4.2.9.1 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.

5.4.2.9.2 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 9656 returns zeros (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.)

5.4.2.9.3 Direct Slave PCI Initialization Example

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

1. Local initialization software or the serial EEPROM contents set 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 (LAS*x*BA[0]=1, where *x* is the Local Address Space number)]
2. PCI Initialization software writes all ones (1) to the PCI Base Address, then reads it back again.
   - The PCI 9656 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).

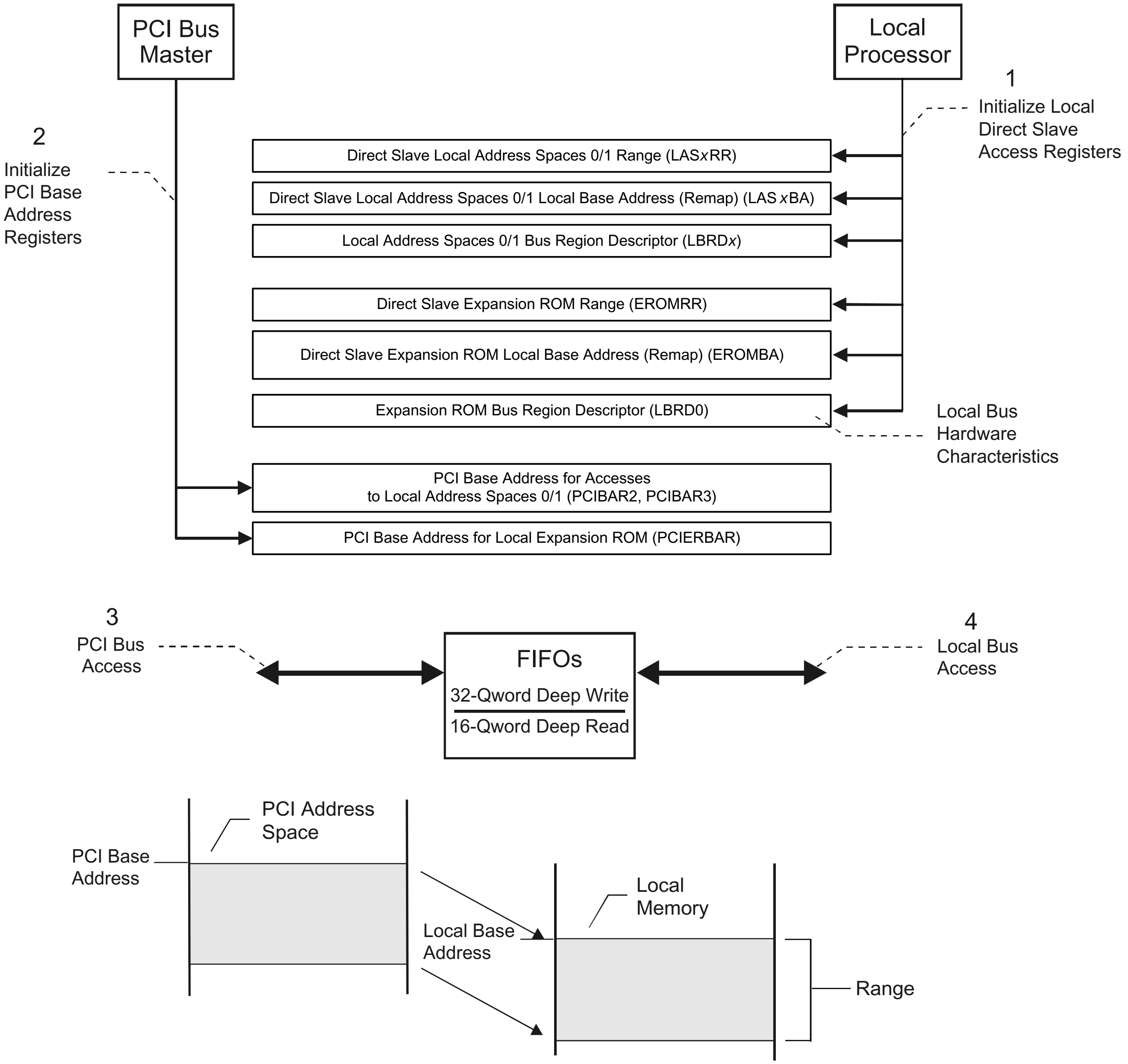

PCI Bus Master
Local Processor
1
Initialize Local Direct Slave Access Registers
2
Initialize PCI Base Address Registers
Direct Slave Local Address Spaces 0/1 Range (LASxRR)
Direct Slave Local Address Spaces 0/1 Local Base Address (Remap) (LASxBA)
Local Address Spaces 0/1 Bus Region Descriptor (LBRDx)
Direct Slave Expansion ROM Range (EROMRR)
Direct Slave Expansion ROM Local Base Address (Remap) (EROMBA)
Expansion ROM Bus Region Descriptor (LBRD0)
Local Bus Hardware Characteristics
PCI Base Address for Accesses to Local Address Spaces 0/1 (PCIBAR2, PCIBAR3)
PCI Base Address for Local Expansion ROM (PCIERBAR)
3
PCI Bus Access
FIFOs
32-Qword Deep Write
16-Qword Deep Read
4
Local Bus Access
PCI Address Space
PCI Base Address
Local Base Address
Local Memory
Range

Figure 5-10. Local Bus Direct Slave Access

5.4.2.9.4 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 data width, by encoding the Local Byte Enables (LBE[3:0]#) as follows:

32-Bit Bus – The four Byte Enables indicate which of the four bytes are valid 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)

5.4.2.9.5 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 data width, by encoding the Local Byte Enables (LBE[3:0]#) as follows:

32-Bit Bus – The four Byte Enables indicate which of the four bytes are valid 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#, LA1, and BLE#, respectively:

- LBE3# Byte High Enable (BHE#) – LAD[15:8]
- LBE2# ***not used***
- LBE1# Address bit 1 (LA1)
- LBE0# Byte Low Enable (BLE#) – LAD[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)

5.4.2.10 Direct Slave Priority

Direct Slave accesses have a higher priority than DMA accesses, thereby preempting DMA transfers. During a DMA transfer, if the PCI 9656 detects a pending Direct Slave access, it releases the Local Bus. The PCI 9656 resumes the DMA operation after the Direct Slave access completes.

When the PCI 9656 DMA Controller(s) owns the Local Bus, its LHOLD output and LHOLDA input are asserted. When a Direct Slave access occurs, the PCI 9656 releases the Local Bus by de-asserting LHOLD and floating the Local Bus outputs. After the PCI 9656 senses that LHOLDA is de-asserted, it requests the Local Bus for a Direct Slave transfer by asserting LHOLD. When the PCI 9656 receives LHOLDA, it performs the Direct Slave transfer. After completing a Direct Slave transfer, the PCI 9656 releases the Local Bus by de-asserting LHOLD and floating the Local Bus outputs. After the PCI 9656 senses that LHOLDA is de-asserted and the Local Bus Pause Timer Counter is zero (MARBR[15:8]=0h) or disabled (MARBR[17]=0), it requests the Local Bus to complete the DMA transfer by re-asserting LHOLD. When the PCI 9656 receives LHOLDA and LHOLD=1, it drives the bus and continues the DMA transfer.

5.4.3 Deadlock Conditions

A deadlock can occur when a PCI Bus Master must access the PCI 9656 Local Bus at the same time a Master on the PCI 9656 Local Bus must access the PCI Bus.

There are two types of deadlock:

- **Partial Deadlock** – Occurs when the PCI device “A” Local Master tries to access the PCI 9656 Local Bus concurrently with the PCI 9656 Local Master trying to access the PCI device “B” Local Bus.
- **Full Deadlock** – Occurs when the PCI device “A” Local Master tries to access the PCI 9656 Local Bus concurrently with the PCI 9656 Local Master trying to access the PCI device “A” Local Bus and neither Local Master relinquishes Local Bus ownership.

This applies only to Direct Master and Direct Slave accesses through the PCI 9656. Deadlock does ***not*** occur in transfers through the PCI 9656 DMA channels or internal registers (*such as*, Mailboxes).

For partial deadlock, the PCI access to the Local Bus times out (Direct Slave Retry Delay Clocks, LBRD0 [31:28]) and the PCI 9656 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 external PCI Bus Arbiter to grant the PCI Bus to the PCI 9656 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 timeout is undesirable, or a full deadlock condition exists.

When a full deadlock occurs, it is necessary to back off the Local Bus Master to prevent a system hang.

5.4.3.1 Backoff

The backoff sequence is used to break a deadlocked condition. The PCI 9656 BREQo signal indicates that a deadlock condition has occurred.

The PCI 9656 starts the Backoff Timer (refer to the EROMBA register for further details) 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 9656 Direct Master Write FIFO cannot accept another Write cycle.

If Local Bus Backoff is enabled (EROMBA[4]=1), the Backoff Timer expires, and the PCI 9656 has not received LHOLDA, the PCI 9656 asserts BREQo. External bus logic can use BREQo to perform backoff.

The Backoff cycle is device/bus architecture dependent. External logic (an arbiter) can assert the necessary signals 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 9656 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 9656 detects LHOLDA, it proceeds with the PCI Master-to-Local Bus access. When this access completes and the PCI 9656 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 9656 Direct Master Write FIFO retains all accepted data (*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 data the PCI 9656 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). The PCI 9656 is ***not*** allowed to perform a new read.

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.

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 Local Bus Arbiter not grant the bus to any Local Bus Master except the PCI 9656. Status output from the Local Bus Arbiter can be connected to 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.

5.4.4 DMA Operation

The PCI 9656 supports two independent DMA channels capable of transferring data from the PCI-to-Local Bus and Local-to-PCI Bus.

Each channel consists of a DMA Controller and a dedicated bi-directional FIFO. Both channels support DMA Block, Scatter/Gather, and Demand Mode transfers, with or without End of Transfer (EOT#). Master mode must be enabled (PCICR[2]=1) before the PCI 9656 can become a PCI Bus master. In addition, both DMA channels can be programmed to:

- Operate with 8-, 16-, or 32-bit Local Bus data widths
- Use zero to 15 internal wait states (Local Bus)
- Enable/disable external wait states (Local Bus)
- Enable/disable Local Bus burst capability
- Set Local Bus mode (refer to Table 5-6)
- Hold Local address constant (Local Slave is FIFO) or incremented
- Perform PCI Memory Write and Invalidate (command code = Fh) or normal PCI Memory Write (command code = 7h)
- Stop/pause Local transfer with or without BLAST# (DMA Fast/Slow Terminate mode)
- Operate in DMA Clear Count mode

The PCI 9656 supports PCI Dual Address Cycles (DAC) with the upper 32-bit register(s) (DMADAC*x*). The PCI 9656 also generates dummy cycles on the PCI and Local Buses. (Refer to Section 5.4.)

The Local Bus Latency Timer (MARBR[7:0]) determines the number of Local clocks the PCI 9656 can burst data before relinquishing Local Bus ownership. The Local Pause Timer (MARBR[15:8]) sets how soon (after the Latency Timer times out) the DMA channel can request the Local Bus again.

Table 5-6. DMA Local Burst Mode

Burst Enable Bit(s)	BTERM# Input Enable Bit(s)	Result
Disabled (0)	X	Single cycle
Enabled (1)	Disabled (0)	Burst up to four Data cycles
Enabled (1)	Enabled (1)	Continuous Burst (terminate when BTERM# is asserted or transfer is completed)

Notes: *Single cycle is the default DMA Local Burst mode.*
"X" is "Don't Care."

5.4.4.1 DMA PCI Dual Address Cycles

The PCI 9656 supports PCI Dual Address Cycles (DAC) when it is a PCI Bus Master, using the DMADAC*x* register(s) for DMA Block transactions. DMA Scatter/Gather can use the DAC function by way of the DMADAC*x* register(s) or DMAMODE*x*[18]. The DAC command is used to transfer a 64-bit address to devices that support 64-bit addressing when the address is above the 4-GB Address space. The PCI 9656 performs a DAC within two PCI clock periods, where the first PCI address is a Lo-Addr, with the command (C/BE[3:0]#) “Dh”, and the second PCI address is a Hi-Addr, with the command (C/BE[3:0]#) “6h” or “7h”, depending upon whether it is a PCI Read or PCI Write cycle. (Refer to Figure 5-11.)

5.4.4.2 DMA Block Mode

The Host processor or the Local processor sets the PCI and Local starting addresses, transfer byte count, and transfer direction. The Host or Local processor then sets the DMA Channel Start and Enable bits (DMACSR*x*[1:0]=11b) to initiate a transfer. The PCI 9656 requests the PCI and Local Buses and transfers data. Once the transfer completes, the PCI 9656 sets the Channel Done bit(s) (DMACSR*x*[4]=1) and asserts an interrupt(s) (INTCSR[16], INTCSR[8], DMAMODE*x*[17], and/or DMAMODE*x*[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 5-12.)

During DMA transfers, the PCI 9656 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 9656 releases the PCI Bus, if one of the following conditions occur (refer to Figures 5-13 and 5-14):

- 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 according to the PCI Maximum Latency (PCIMLR) register value – and PCI GNT# is de-asserted
- PCI Target asserts STOP#

The PCI 9656 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 (MARBR[16]=1) and expires (MARBR[7:0])
- BREQi is asserted
- Direct Slave request is pending

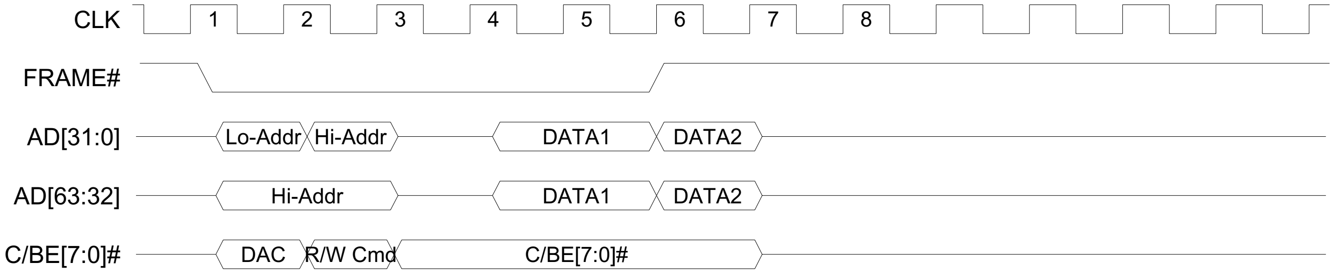


Figure 5-11. PCI Dual Address Cycle Timing

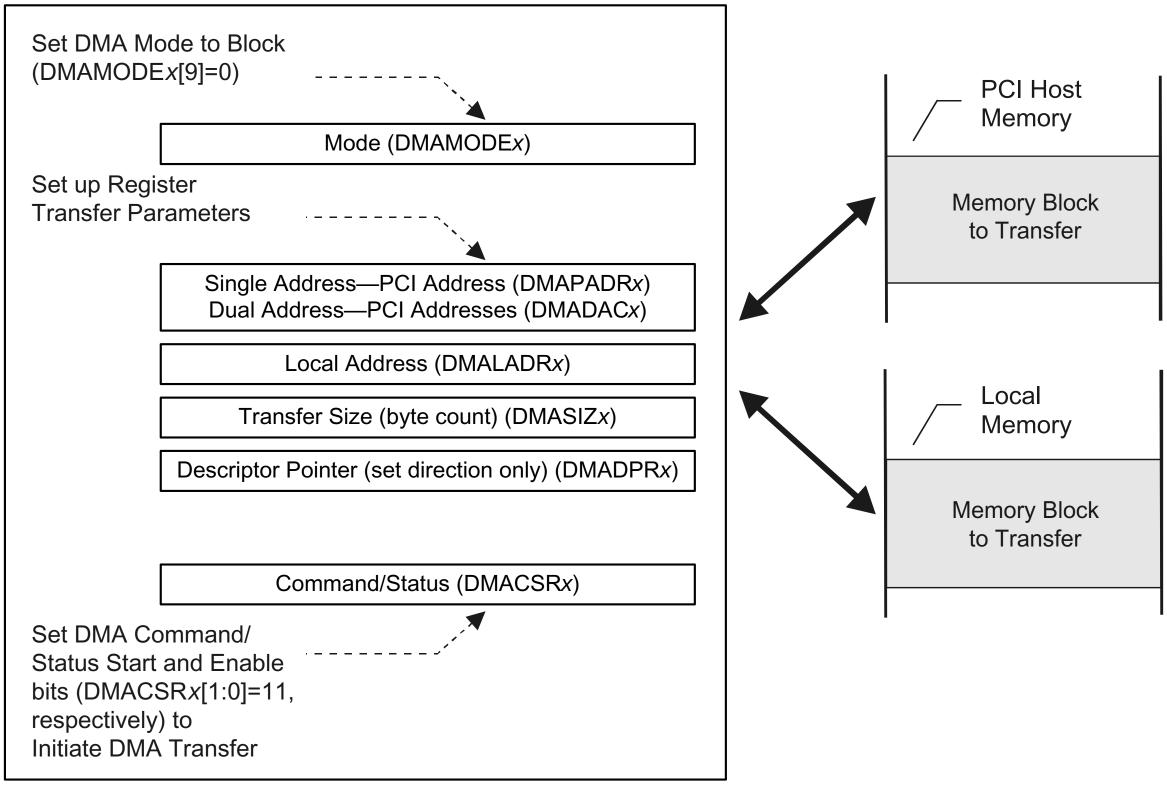


Figure 5-12. DMA Block Mode Initialization (Single Address or Dual Address PCI)

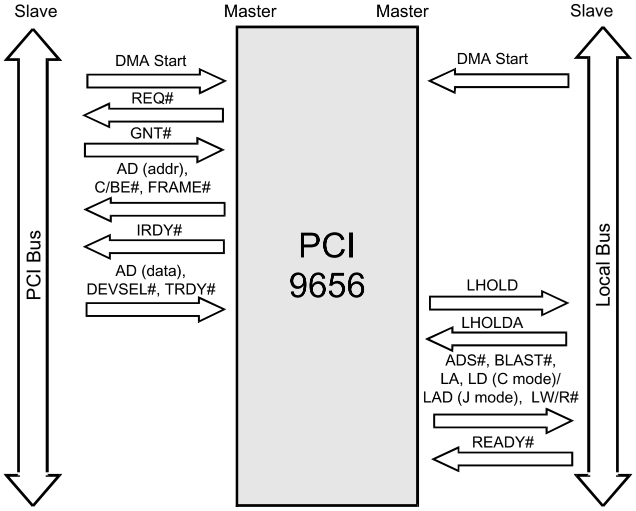


Figure 5-13. DMA, PCI-to-Local Bus

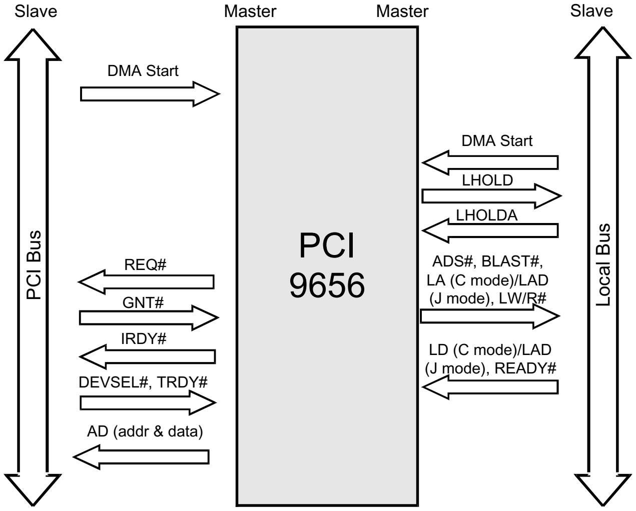


Figure 5-14. DMA, Local-to-PCI Bus

Note: *Figures 5-13 and 5-14 represent a sequence of Bus cycles.*

5.4.4.2.1 DMA Block Mode PCI Dual Address Cycles

The PCI 9656 supports the DAC feature in DMA Block mode. When the DMADAC*x* register(s) contains a value of 0h, the PCI 9656 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 5-11.)

5.4.4.3 DMA Scatter/Gather Mode

In DMA Scatter/Gather mode, the Host processor or Local processor sets up descriptor blocks in PCI or Local memory composed of PCI and Local addresses, transfer count, transfer direction, and address of the next descriptor block. (Refer to Figures 5-15 and 5-16.) The Host or Local processor then:

- Enables the DMA Scatter/Gather mode bit(s) (DMAMODE*x*[9]=1)
- Sets up the address of initial descriptor block in the PCI 9656 Descriptor Pointer register(s) (DMADPR*x*)
- Initiates the transfer by setting a control bit(s) (DMACSR*x*[1:0]=11b)

The PCI 9656 supports zero wait state Descriptor Block bursts from the Local Bus when the Local Burst Enable bit(s) is enabled (DMAMODE*x*[8]=1).

The PCI 9656 loads the first descriptor block and initiates the Data transfer. The PCI 9656 continues to load descriptor blocks and transfer data until it detects the End of Chain bit(s) is set (DMADPR*x*[1]=1). When the End of Chain bit(s) is detected, the PCI 9656 completes the current descriptor block, sets the DMA Done bit(s) (DMACSR*x*[4]=1), and asserts a PCI or Local interrupt (INTA# or LINTo#, respectively), if the interrupt is enabled (DMAMODE*x*[10]=1).

The PCI 9656 can also be programmed to assert PCI or Local interrupts after each descriptor is loaded, then the corresponding Data transfer is finished.

The DMA Controller(s) can be programmed to clear the transfer size at completion of each DMA transfer, using the DMA Clear Count Mode bit(s) (DMAMODE*x* [16]=1).

Notes: *In DMA Scatter/Gather 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 (DMAMODEx[18]=1). Otherwise, the DMADACx register values are used. The Descriptor Pointer register(s) (DMADPRx) contains descriptor location (bit 0), end of chain (bit 1), interrupt after terminal count (bit 2), direction of transfer (bit 3), and next descriptor address (bits [31:4]) bits.*

DMA descriptors stored on the Local Bus are accessed using the same bus data width as programmed for the Data transfer.

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

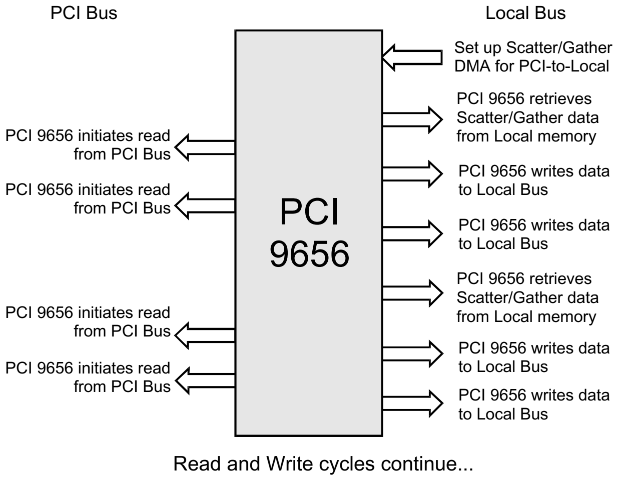

PCI Bus
Local Bus
Set up Scatter/Gather DMA for PCI-to-Local
PCI 9656 retrieves Scatter/Gather data from Local memory
PCI 9656 initiates read from PCI Bus
PCI 9656 writes data to Local Bus
PCI 9656 initiates read from PCI Bus
PCI 9656 writes data to Local Bus
PCI 9656
PCI 9656 retrieves Scatter/Gather data from Local memory
PCI 9656 initiates read from PCI Bus
PCI 9656 writes data to Local Bus
PCI 9656 initiates read from PCI Bus
PCI 9656 writes data to Local Bus
Read and Write cycles continue...

Figure 5-15. DMA Scatter/Gather Mode from PCI-to-Local Bus (Control Access from the Local Bus)

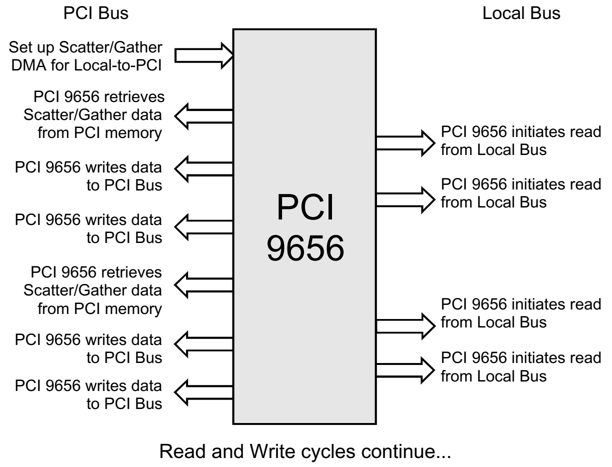

PCI Bus
Local Bus
Set up Scatter/Gather DMA for Local-to-PCI
PCI 9656 retrieves Scatter/Gather data from PCI memory
PCI 9656 initiates read from Local Bus
PCI 9656 writes data to PCI Bus
PCI 9656 initiates read from Local Bus
PCI 9656 writes data to PCI Bus
PCI 9656
PCI 9656 retrieves Scatter/Gather data from PCI memory
PCI 9656 initiates read from Local Bus
PCI 9656 writes data to PCI Bus
PCI 9656 initiates read from Local Bus
PCI 9656 writes data to PCI Bus
Read and Write cycles continue...

Figure 5-16. DMA Scatter/Gather 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.*

5.4.4.3.1 DMA Scatter/Gather PCI Dual Address Cycle

The PCI 9656 supports the PCI DAC feature in DMA Scatter/Gather mode for Data transfers only. The descriptor blocks should reside below the 4-GB Address space. The PCI 9656 offers three different options of how PCI DAC DMA Scatter/Gather is used. Assuming the descriptor blocks are located on the PCI Bus:

- DMADAC*x* contain(s) a non-zero value. DMAMODE*x*[18] is cleared to 0. The PCI 9656 performs a Single Address Cycle (SAC) 4-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. (Refer to Figure 5-17.) The DMA descriptor block starting addresses should be 16-byte-aligned.
- DMADAC*x* contain(s) a value of 0h. DMAMODE*x*[18] is set to 1. The PCI 9656 performs a SAC 5-Lword descriptor block load from PCI memory and DMA transfer with PCI DAC on the PCI Bus. (Refer to Figure 5-18.) The DMA descriptor block starting addresses should be 32-byte-aligned.
- DMADAC*x* contain(s) a non-zero value. DMAMODE*x*[18] is set to 1. The PCI 9656 performs a SAC 5-Lword descriptor block load from PCI memory and DMA transfer with DAC on the PCI Bus. The fifth descriptor overwrites the value of the DMADAC*x* register(s). (Refer to Figure 5-18.) The DMA descriptor block starting addresses should be 32-byte-aligned.

5.4.4.3.2 DMA Clear Count Mode

The PCI 9656 supports DMA Clear Count mode (Write-Back feature, DMAMODE*x*[16]). The PCI 9656 clears each transfer size descriptor to zero (0) by writing to its location on the PCI and/or Local Bus memory at the end of each transfer chain. This feature is available for DMA descriptors located on the PCI and Local Buses.

The DMA Clear Count mode can also be used in conjunction with the EOT feature (DMAMODE*x*[14]). EOT# assertion during DMA Data transfers causes the PCI 9656 to write back the amount of data bytes not transferred to the destination bus.

When encountering a PCI Master/Target Abort on a DMA Data transfer, the PCI 9656 writes random values when the descriptor is on the Local Bus. No write occurs if the descriptor is on the PCI Bus.

Note: *DMA Clear Count mode works only if there is more than one descriptor in the descriptor chain, because the first descriptor is written directly into the PCI 9656 DMA Configuration registers and the remainder are loaded from PCI or Local memory.*

5.4.4.3.3 DMA Ring Management (Valid Mode)

In DMA Scatter/Gather mode, when the Ring Management Valid Mode Enable bit(s) is cleared to 0 (DMAMODE*x*[20]=0), the Valid bit(s) [bit(s) 31 of transfer count] is ignored. When the Ring Management Valid Mode Enable bit(s) is set to 1 (DMAMODE*x*[20]=1), the DMA descriptor proceeds only when the Ring Management Valid bit(s) is set (DMASIZ*x*[31]=1). If the Valid bit(s) is set, the transfer count is 0, and the descriptor is not the last descriptor, then the DMA Controller(s) moves on to the next descriptor in the chain.

When the Ring Management Valid Stop Control bit(s) is cleared to 0 (DMAMODE*x*[21]=0), the DMA Scatter/Gather controller continuously polls the descriptor with the Valid bit(s) cleared to 0 (invalid descriptor) until the Valid bit(s) is read as 1, which initiates the DMA transfer. When the Valid bit(s) is read as 1, the DMA transfer begins. When the Ring Management Valid Stop Control bit(s) is set to 1 (DMAMODE*x*[21]=1), the DMA Scatter/Gather controller pauses if a Valid bit(s) with a value of 0 is detected. In this case, the processor must restart the DMA Controller(s) by setting the DMA Channel Start bit(s) (DMACSR*x*[1]=1). The DMA Clear Count mode bit(s) must be enabled (DMAMODE*x*[16]=1) for the Ring Management Valid bit(s) (DMASIZ*x*[31]) to be cleared at the completion of each descriptor. (Refer to Figure 5-19 and/or Figure 5-20 for the DMA Ring Management descriptor load sequence for SAC and DAC PCI addresses.)

Notes: *In DMA Ring Management Scatter/Gather mode, the descriptor includes the transfer size with a valid descriptor bit, PCI and Local Address Space, and next descriptor pointer. It also includes a DAC value if the DAC Chain Load bit(s) is enabled (DMAMODEx[18]=1). Otherwise, the DMADACx register value is used.*

In Ring Management mode, the transfer size is loaded before the PCI address.

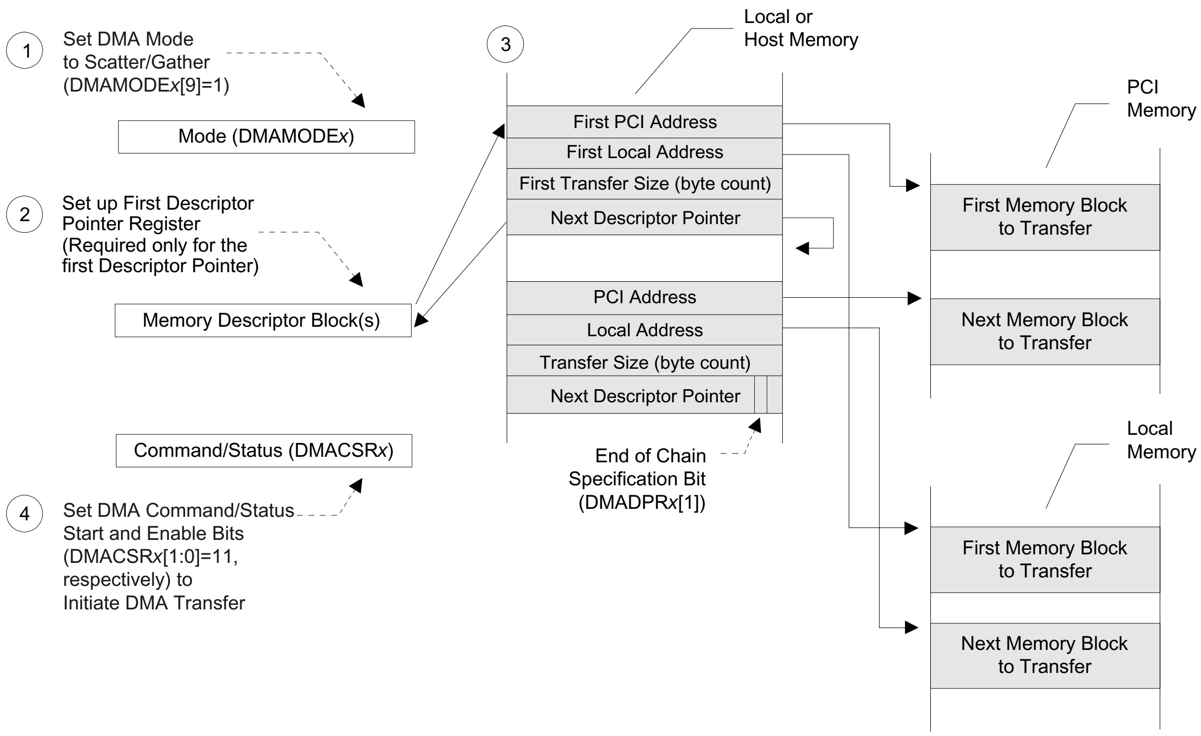


Figure 5-17. DMA Scatter/Gather Mode Descriptor Initialization [PCI SAC/DAC PCI Address (DMADACx) Register Dependent]

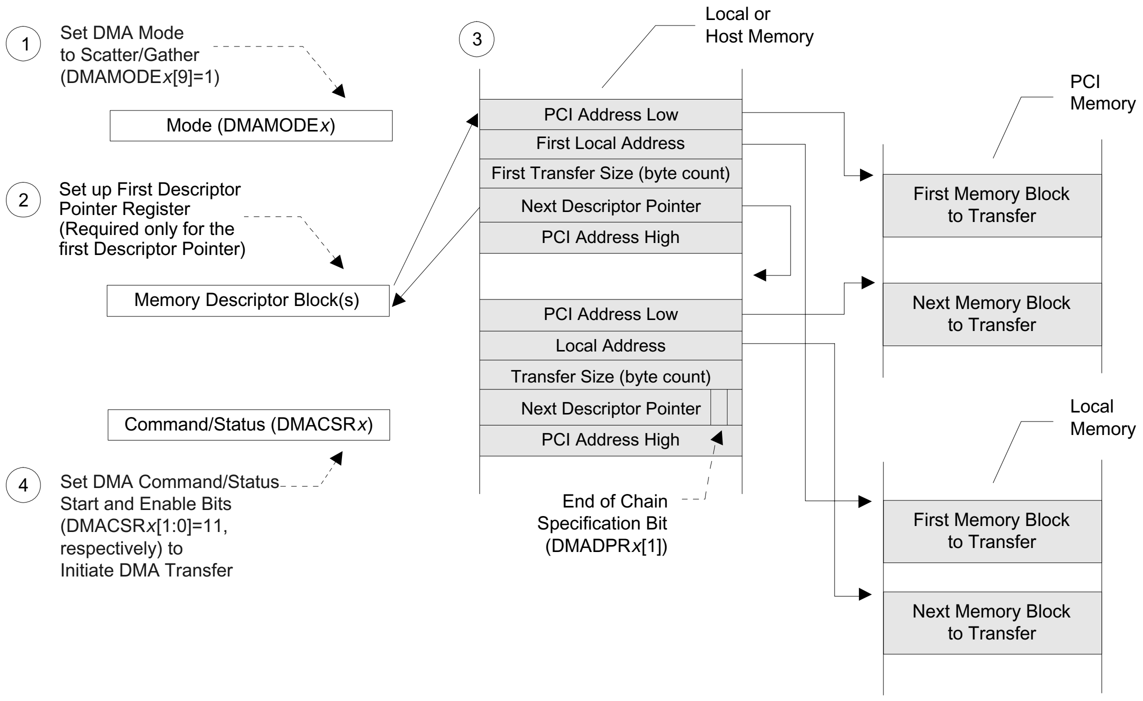


Figure 5-18. DMA Scatter/Gather Mode Descriptor Initialization [DAC PCI Address (DMAMODEx[18]) Descriptor Dependent (PCI Address High Added)]

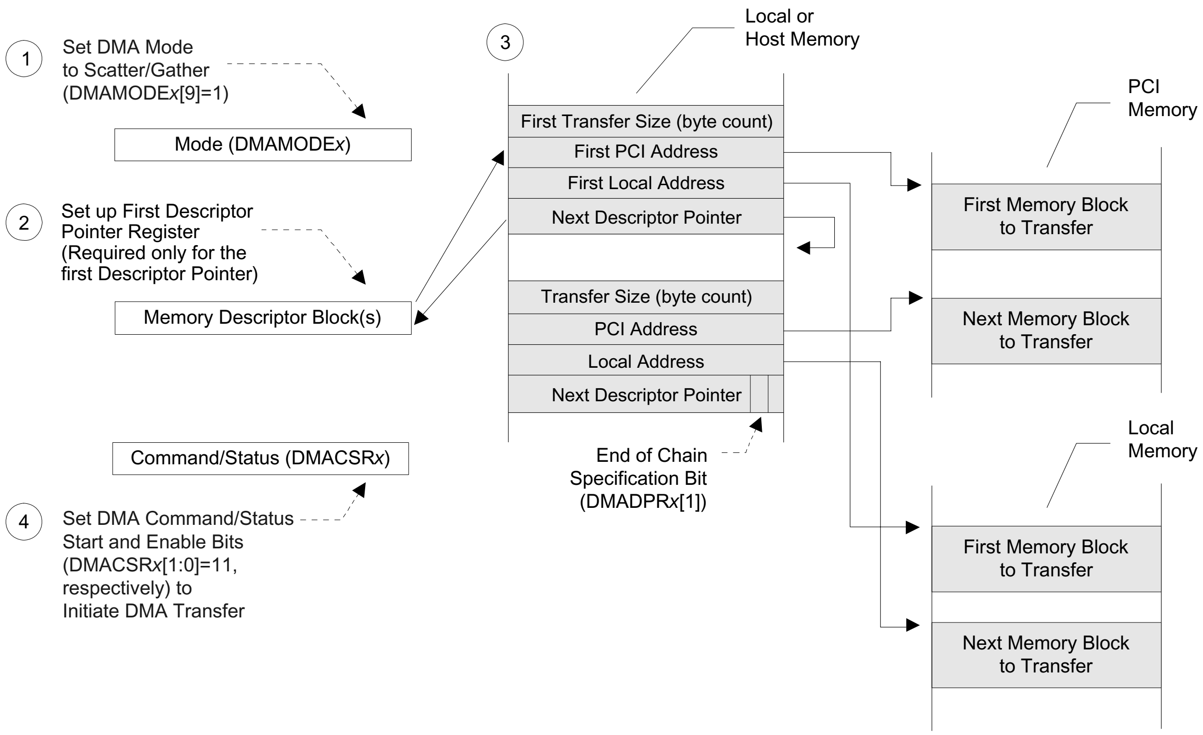


Figure 5-19. DMA Ring Management Scatter/Gather Mode Descriptor Initialization [PCI SAC/DAC PCI Address (DMADACx) Register Dependent]

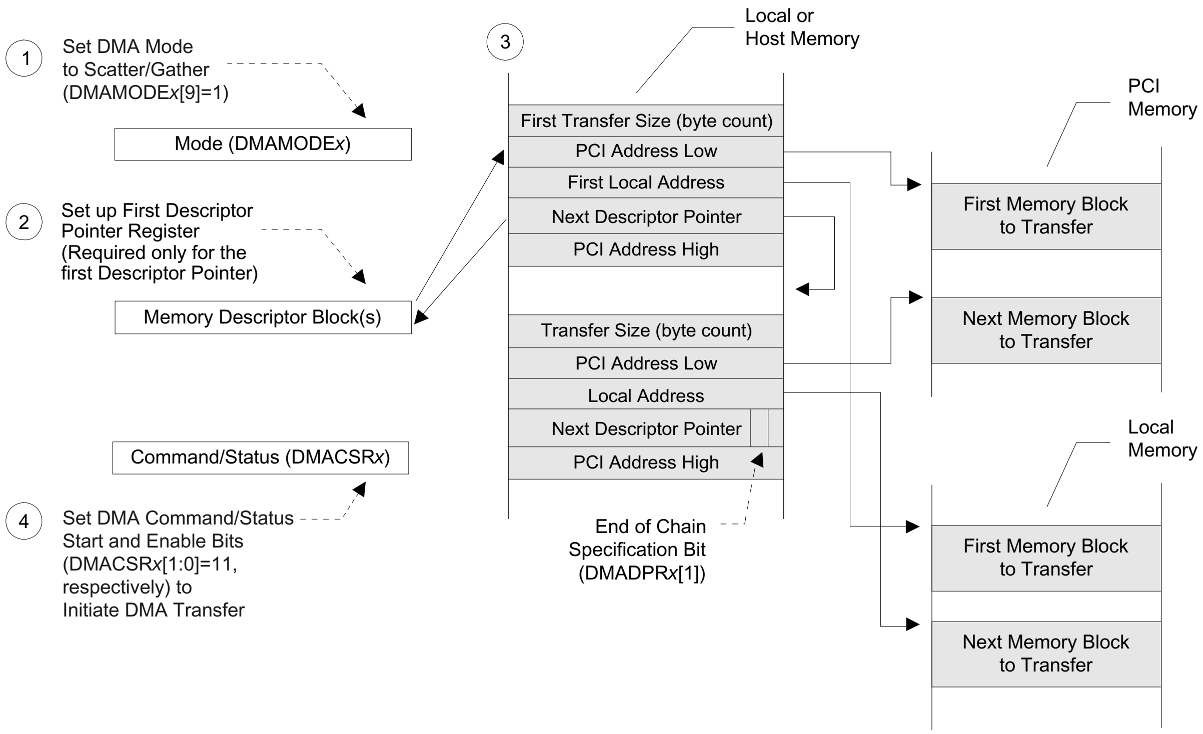


Figure 5-20. DMA Ring Management Scatter/Gather Mode Descriptor Initialization [DAC PCI Address (DMAMODEx[18]) Descriptor Dependent (PCI Address High Added)]

5.4.4.4 DMA Memory Write and Invalidate

The PCI 9656 can be programmed to perform Memory Write and Invalidate (MWI) cycles to the PCI Bus for DMA transfers, as well as Direct Master transfers. (Refer to Section 5.4.1.10.) The PCI 9656 supports MWI 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 9656 performs Write transfers (using the command code programmed in CNTRL[7:4]) rather than MWI transfers.

DMA MWI transfers are enabled when the Memory Write and Invalidate Mode for DMA Transfers and the Memory Write and Invalidate Enable bits are set (DMAMODE*x*[13]=1 and PCICR[4]=1, respectively).

In MWI mode, the PCI 9656 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 9656 completes the remainder of that cache line, using normal writes before resuming MWI transfers. If an MWI cycle is in progress, the PCI 9656 continues to burst if another cache line is read from the Local Bus before the cycle completes. Otherwise, the PCI 9656 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 9656 completes the DMA transfer, using normal writes.

An EOT# assertion, or DREQ*x*# de-assertion in DMA Demand mode occurring before the cache line is read from the Local Bus, results in a normal PCI Memory write of the data read into a DMA FIFO. If the DMA Data transfer starts from a non-cache line boundary, it first performs normal writes until it reaches the cache line boundary. If the DMA Data transfer possesses more than one line of data in the DMA FIFO, it starts the MWI transfer.

5.4.4.5 DMA Abort

DMA transfers may be aborted by software commands, or by issuing the EOT# signal. (Refer to Section 5.4.4.11 for further details about EOT#.) To abort a DMA transfer, follow these steps:

1. Clear the DMA Channel Enable bit(s) (DMACSR*x*[0]=0).
2. Abort the DMA transfer by setting the DMA Channel Abort bit(s) along with the corresponding DMA Channel Start bit(s) (DMACSR*x*[2:1]=11b, respectively).
3. Wait until the DMA Channel Done bit(s) is set (DMACSR*x*[4]=1).

Note: *One to two Data transfers occur after the Abort bit(s) is set. Software may simultaneously set DMACSRx[2:0]. Setting software commands to abort a DMA transfer when no DMA cycles are in progress causes the next DMA transfer to abort.*

5.4.4.6 DMA Channel Priority

The DMA Channel Priority bits (MARBR[20:19]) can be used to specify the priorities listed in Table 5-7.

Table 5-7. DMA Channel Priority Bit Specifications

MARBR[20:19]	Channel Priority
00b	Rotating
01b	DMA Channel 0
10b	DMA Channel 1
11b	***Reserved***

5.4.4.7 DMA Channel *x* Interrupts

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

The DMA Channel Done bit(s) (DMACSR*x*[4]) can be used to determine whether an interrupt is one of the following:

- DMA Done interrupt
- Transfer complete for current descriptor interrupt

Setting DMAMODE*x*[10]=1 enables a DMA Channel Done interrupt. In DMA Scatter/Gather mode, the Descriptor Pointer register Interrupt after Terminal Count bit(s) (DMADPR*x*[2], loaded from Local memory) specifies whether to assert an interrupt at the end of the transfer for the current descriptor.

Setting DMACSR*x*[3]=1 clears a DMA Channel interrupt.

5.4.4.8 DMA Data Transfers

The PCI 9656 DMA Controllers can be programmed to transfer data from the Local-to-PCI Bus or from the PCI-to-Local Bus, as illustrated in Figures 5-21 and 5-22, respectively.

5.4.4.8.1 Local-to-PCI Bus DMA Transfer

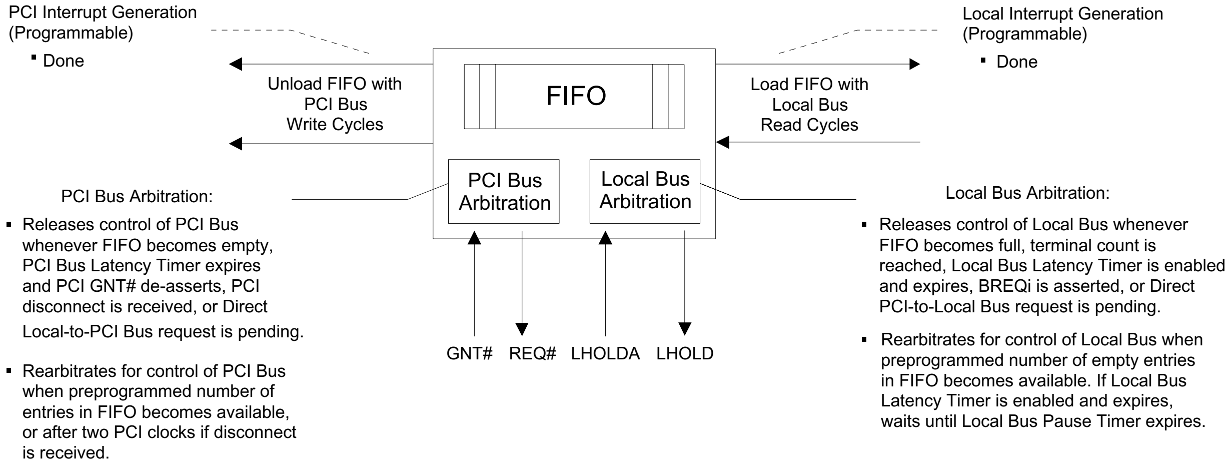

PCI Interrupt Generation
(Programmable)
▪ Done
Unload FIFO with
PCI Bus
Write Cycles
FIFO
Load FIFO with
Local Bus
Read Cycles
Local Interrupt Generation
(Programmable)
▪ Done
PCI Bus Arbitration:
PCI Bus
Arbitration
Local Bus
Arbitration
Local Bus Arbitration:
▪ Releases control of PCI Bus whenever FIFO becomes empty, PCI Bus Latency Timer expires and PCI GNT# de-asserts, PCI disconnect is received, or Direct Local-to-PCI Bus request is pending.
▪ Rearbitrates for control of PCI Bus when preprogrammed number of entries in FIFO becomes available, or after two PCI clocks if disconnect is received.
GNT#
REQ#
LHOLDA
LHOLD
▪ Releases control of Local Bus whenever FIFO becomes full, terminal count is reached, Local Bus Latency Timer is enabled and expires, BREQi is asserted, or Direct PCI-to-Local Bus request is pending.
▪ Rearbitrates for control of Local Bus when preprogrammed number of empty entries in FIFO becomes available. If Local Bus Latency Timer is enabled and expires, waits until Local Bus Pause Timer expires.

Figure 5-21. Local-to-PCI Bus DMA Data Transfer Operation

5.4.4.8.2 PCI-to-Local Bus DMA Transfer

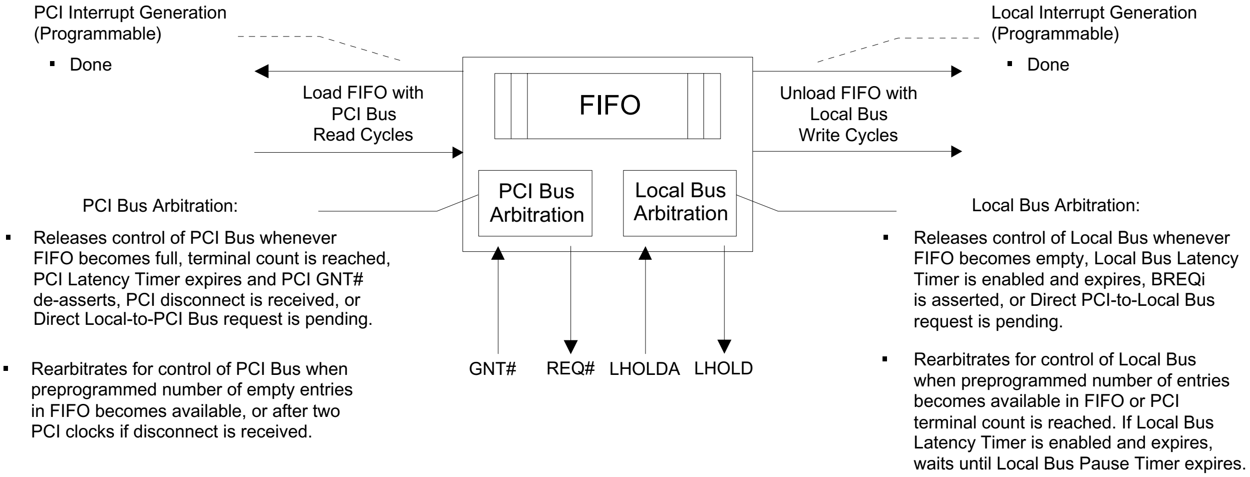

PCI Interrupt Generation
(Programmable)
▪ Done
Load FIFO with
PCI Bus
Read Cycles
FIFO
Unload FIFO with
Local Bus
Write Cycles
Local Interrupt Generation
(Programmable)
▪ Done
PCI Bus Arbitration:
PCI Bus
Arbitration
Local Bus
Arbitration
Local Bus Arbitration:
▪ Releases control of PCI Bus whenever FIFO becomes full, terminal count is reached, PCI Latency Timer expires and PCI GNT# de-asserts, PCI disconnect is received, or Direct Local-to-PCI Bus request is pending.
▪ Rearbitrates for control of PCI Bus when preprogrammed number of empty entries in FIFO becomes available, or after two PCI clocks if disconnect is received.
GNT#
REQ#
LHOLDA
LHOLD
▪ Releases control of Local Bus whenever FIFO becomes empty, Local Bus Latency Timer is enabled and expires, BREQi is asserted, or Direct PCI-to-Local Bus request is pending.
▪ Rearbitrates for control of Local Bus when preprogrammed number of entries becomes available in FIFO or PCI terminal count is reached. If Local Bus Latency Timer is enabled and expires, waits until Local Bus Pause Timer expires.

Figure 5-22. PCI-to-Local Bus DMA Data Transfer Operation

5.4.4.9 DMA Unaligned Transfers

For unaligned Local-to-PCI transfers, the PCI 9656 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/Qwords 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 least significant address bits for the start of a transfer. For the last transfer, Byte Enables specify the bytes to be written.

5.4.4.10 DMA Demand Mode, Channel *x*

DMA Demand mode allows the transfer of DMA data to be controlled by the DREQ*x*# input pin(s). When DREQ*x*# is asserted, the PCI 9656 starts a DMA transfer, based upon the values programmed into the DMA registers or by using internally stored values when resuming the transfer after it was paused in response to a DREQ*x*# de-assertion. The PCI 9656 asserts DACK*x*#, to indicate that the DMA transfer is in-progress on the Local Bus. DACK*x*# asserts with ADS#, and remains asserted until one clock before the PCI 9656 de-asserts LHOLD to release the Local Bus. Data is only written to, or read from, the Local Bus while DACK*x*# is asserted. DACK*x*# de-assertion indicates that the DMA transfer has completed or is being paused in response to a DREQ*x*# de-assertion, or because the PCI 9656 needs to release the Local Bus as described in Section 5.4.4.2.

While processing in DMA Demand mode, the amount of data transferred during a Local Bus access depends upon the length and timing of the DREQ*x*# assertion. If DREQ*x*# is not asserted a sufficient length of time that allows all data to be transferred to complete the DMA transfer, the PCI 9656 transfers data in Lword-sized chunks until it releases the Local Bus. For an 8-bit Local Bus device, the PCI 9656 releases the Local Bus after transferring the last byte of an Lword. For a 16-bit Local Bus device, the PCI 9656 releases the bus after transferring the last word of an Lword, unless it is the last byte or word of the transfer.

Each DMA channel is capable of two types of transfers, PCI-to-Local or Local-to-PCI, as selected by the DMA Channel *x* Direction of Transfer bit (DMADPR*x*[3]). In either case, the PCI 9656 becomes a Master on the PCI and Local Buses, to execute the DMA transfer. If the DMA Local-to-PCI FIFO becomes full, DACK*x*# is de-asserted and the DREQ*x*# assertion is ignored. Subsequent transfers do not occur until DACK*x*# is re-asserted, indicating that additional room is now available in the FIFO.

The following can be used to modify the functionality of the DMA Demand mode transfer:

- DMA Channel *x* Fast/Slow Terminate Mode Select bit (DMAMODE*x*[15]) setting
- PCI Bus starting or resuming address
- Local Bus starting or resuming address
- Local Bus widths

The functional changes that these items cause are explained, in detail, in the sections that follow.

DREQ*x*# can be asserted from one clock to more clocks than is necessary, to transfer all the DMA data. This, combined with the other factors that modify the PCI 9656 functionality, create a wide variety of responses to the assertion of DREQ*x*#. The simplest way to use DMA Demand mode is to use Slow Terminate Mode (DMAMODE*x*[15]=0) to assert the DREQ*x*# pin, wait for DACK*x*# to assert, allow some data to transfer before de-asserting DREQ*x*#, then continue to accept or provide data until the last Lword is transferred with BLAST# asserted. If the DREQ*x*# assertion for a DMA transfer will never resume for ongoing transfers, a DMA Abort procedure or the EOT# Input signal/pin assertion can be applied, to terminate the DMA transfer and/or flush the DMA FIFO.

Note: *If bursting is enabled (DMAMODEx[8]=1) when DMA Constant Address mode is also enabled (DMAMODEx [11]=1), the PCI 9656 enables Continuous Burst mode, regardless of the Continuous Burst bit setting (DMADMODEx[7]=x).*

5.4.4.10.1 Fast/Slow Terminate Mode

In Slow Terminate mode (DMAMODEx[15]=0), the PCI 9656 adheres to the C and J mode protocol and asserts BLAST# output, to indicate that the current Data transfer is the last Lword to be transferred before releasing the Local Bus while processing in DMA Demand mode.

In Fast Terminate Mode (DMAMODE*x*[15]=1) the PCI 9656 is ***not*** required to adhere to the C and J mode protocol regarding BLAST# output assertion (to indicate that the current data transfer is the last Lword to be transferred before releasing the Local Bus). However, the PCI 9656 still asserts BLAST# in Fast Terminate mode if DREQ*x*# input is de-asserted and the external READY#=0 (asserted), on the transfer before the Local Address is to be LA[2:0]=000b (C mode) or LAD[2:0]=000b (J mode), or if the internal Wait State Counter(s) decrements to 0 for the current Lword.

Slow Terminate mode is the default setting. Running in Slow Terminate mode is recommended, unless the user needs the PCI 9656 to immediately release the Local Bus, and is willing to account for a disconnection with or without BLAST# assertion on the last Data transfer. Running in Fast Terminate mode is usually combined with EOT# assertion, to terminate/complete the DMA transfer and/or flush the DMA FIFO.

5.4.4.10.2 PCI-to-Local DMA Demand Mode

For a PCI-to-Local Bus DMA Demand Mode transfer, independent of the DREQ*x*# assertion the PCI 9656 starts reading data from the PCI Bus into the DMA FIFO, upon the setting of the DMA Channel *x* Start bit (DMACSR*x*[1]=1). The PCI 9656 continued to read data from the PCI Bus, and attempts to keep the FIFO full, until the entire DMA transfer is complete. The PCI 9656 waits for at least 2 Lwords of data to be read into the DMA FIFO and DREQ*x*# to be asserted before attempting to arbitrate for, and write data to, the Local Bus. The PCI 9656's behavior in response to DREQ*x*# assertion depends upon the following:

- When, and for how long, DREQ*x*# is asserted
- Where DREQ*x*# is de-asserted with relation to DACK*x*#
- Data availability
- PCI and Local Bus starting or resumption addresses
- DMA Channel *x* Fast/Slow Terminate Mode Select bit (DMAMODE*x*[15]) setting
- Transfer/protocol dependencies

At least 2 Lwords of data must be available in the FIFO, for the PCI 9656 to respond to DREQ*x*# assertion for one or more Local clock periods. The detailed response descriptions that follow assume that 2 Lwords have already been read into the FIFO, or that DREQ*x*# is being asserted for a sufficient length of time to read 2 Lwords into the FIFO.

EOT# assertion causes the PCI 9656 to terminate the ongoing DMA transfer and flush the FIFO. However, before the PCI 9656 releases the Local Bus, it may or may not need to write additional data. Details of the terminating Local Bus behavior, when EOT# is asserted, are also provided below.

Slow Terminate mode (DMAMODE*x*[15]=0)

- DREQ*x*# assertion for one Local clock period when at least 2 Lwords of data are available in the FIFO, or assertion up to, but not including, the first Clock cycle with DACK*x*# asserted, results in a one Lword transfer on the Local Bus with a BLAST# assertion.
- DREQ*x*# de-assertion on the clock period of DACK*x*# assertion, or any clock period after DACK*x*# assertion, results in one to two additional Lword transfers on the Local Bus. BLAST# is asserted with the last Lword.
- DREQ*x*# assertion (when 2 or more Lwords are available in the FIFO), then de-assertion followed by a re-assertion, up to and including the first cycle of DACK*x*# assertion, causes the PCI 9656 to hold LHOLD asserted and continue to transfer additional Lwords as long as DREQ# remains asserted and data is available. Otherwise, the PCI 9656 asserts BLAST#, transfers one Lword release, and re-arbitrates for the Local Bus when it detects that DREQ*x*# is asserted again. For each transfer, BLAST# is asserted with the last Lword.
- If EOT# is asserted and DREQ*x*# is de-asserted on the same clock, the PCI 9656 transfers one additional Lword with BLAST# asserted, before terminating the ongoing DMA Data transfer and flushing the DMA FIFO.

Fast Terminate mode (DMAMODE*x*[15]=1)

- DREQ*x*# assertion for one Local clock period when at least 2 Lwords of data are available in the FIFO, or assertion up to, but not including, the first Clock cycle with DACK*x*# asserted, results in a one Lword transfer on the Local Bus with a BLAST# assertion.
- DREQ*x*# de-assertion on the same clock as DACK*x*# assertion, or any other Clock cycle thereafter in which DACK*x*# is asserted, results in one additional Lword transferred to the Local Bus without BLAST# assertion.
- DREQ*x*# assertion (when 2 or more Lwords are available in the FIFO), then de-assertion followed by a re-assertion, up to and including the first cycle of DACK*x*# assertion, causes the PCI 9656 to hold LHOLD asserted and continue to transfer additional Lwords, as long as DREQ*x*# remains asserted and data is available. Otherwise, the PCI 9656 asserts BLAST#, transfers one Lword release, and re-arbitrates for the Local Bus when it detects that DREQ*x*# is asserted again. Upon the following DREQ*x*# de-assertion, one additional Lword is transferred on the Local Bus without a BLAST# assertion.
- If DREQ*x*# is de-asserted during the transfer, the PCI 9656 can assert BLAST# if either of the following conditions are met:
  - Data is the last Lword for the entire DMA transfer
  - Local Bus Latency Timer expires
- If EOT# is asserted and DREQ*x*# is de-asserted on the same clock that the PCI 9656 transfers valid data, the PCI 9656 does not assert BLAST# nor provide valid data on subsequent Clock cycles. The PCI 9656 releases the Local Bus 2 clocks after EOT# is asserted, terminates the ongoing DMA Data transfer, and flushes the DMA FIFO.

5.4.4.10.3 Local-to-PCI DMA Demand Mode

For a Local-to-PCI Bus DMA Demand Mode transfer, the PCI 9656 does not arbitrate on the Local Bus to read data into the DMA FIFO until the DMA Channel *x* Start bit (DMACSR*x*[1]=1) and the DREQ*x*# signal/pin is asserted. As long as DREQ*x*# is asserted, the PCI 9656 continues to read data from the Local Bus and attempts to keep the FIFO full, until the entire DMA transfer is complete. The PCI 9656 waits for at least 1 Lword of data to be read into the DMA FIFO before attempting to arbitrate for, and write data to, the PCI Bus. The PCI 9656 behavior in response to DREQ*x*# assertion depends upon the following:

- When, and for how long, DREQ*x*# is asserted
- Where DREQ*x*# is de-asserted with relation to DACK*x*#
- Data availability
- PCI and Local Bus starting or resumption addresses
- DMA Channel *x* Fast/Slow Terminate Mode Select bit (DMAMODE*x*[15]) setting
- Transfer/protocol dependencies

For transfers that are reading data from a 32-bit Local Bus device, the PCI 9656 requires 1 or 2 Lwords to be read (explanation follows) and placed into the DMA FIFO before it arbitrates for, and writes data to, the PCI Bus. As a result, if 2 Lwords must be read, one Lword of data will remain in the DMA FIFO until the second Lword is read. Upon reading the required amount of data, the PCI 9656 then arbitrates for, and writes the data to, the PCI Bus.

For non-Lword aligned transfers that read data from 8- or 16-bit Local Bus devices, the PCI 9656 requires a minimum amount of bytes or words to be read from the Local Bus device before it arbitrates for, and writes data to, the PCI Bus. The number of bytes or words that it must read before transferring the data to the PCI Bus is determined by the PCI starting address. As a result, there could be seven or fewer bytes remaining in the DMA FIFO at any given time.

Therefore, once a DMA Demand Mode transfer is started (DREQ*x*# asserted), it may be necessary for the PCI 9656 to read more bytes, words or Lwords from the Local Bus than are required to complete the DMA transfer. If bytes, words, or an Lword remain in the DMA FIFO due to a DREQ*x*# de-assertion, when DREQ*x*# is re-asserted (and additional data is read), that data is transferred to the PCI Bus. If the DREQ*x*# assertion never resumes for ongoing transfers, a DMA Abort procedure or EOT# assertion can be applied to terminate/complete the ongoing DMA transfer.

It is recommended that DREQ*x*# be asserted until DACK*x*# is asserted and at least 4 Lwords of data are read into the DMA FIFO. However, the following information is provided, for allowing smaller transfers inside an ongoing DMA transfer.

Because there are 2 Lwords per DMA FIFO location, the PCI 9656 places data into the DMA FIFO in 2-Lword quantities. Therefore, when DREQ*x*# is de-asserted the PCI 9656 continues to read data up to the next Qword-aligned boundary [X0h or X8h address boundary (Local Address LA[2:0]=000b (C mode) or LAD[2:0]=000b (J mode))].

Detailed descriptions of the PCI 9656's response to DREQ*x*# assertion when the DMA transfer first starts or resumes reading data from Local Bus are provided below.

EOT# assertion causes the PCI 9656 to terminate the ongoing DMA transfer. However, before the PCI 9656 releases the Local Bus, it may or may not need to read additional data. Details of the terminating Local Bus behavior when EOT# is asserted are also provided below. All data that is read into the DMA FIFO is written to the PCI Bus.

Slow Terminate mode (DMAMODE*x*[15]=0)

- If the PCI destination address is X0h/X8h, and DREQ*x*# is asserted for one Local Bus Clock cycle, or asserted then de-asserted on any cycle before READY# is asserted, the PCI 9656 requires 2 Lwords to be read into the DMA FIFO.
- If the PCI destination address is X0h/X8h, and DREQ*x*# is asserted and remains asserted until it is de-asserted on the first or second cycle that data is transferred (READY# asserted), the PCI 9656 requires one additional Lword to be read into the DMA FIFO.
- If the PCI destination address is X0h/X8h, and DREQ*x*# is asserted and held asserted until it is de-asserted on a cycle that is transferring the 3, 5, 7… (Odd) Lword with READY# asserted, the PCI 9656 requires one additional Lword to be read into the DMA FIFO.
- If the PCI destination address is X0h/X8h, and DREQ*x*# is asserted and held asserted until it is de-asserted on a cycle that is transferring the 4, 6, 8… (Even) Lword with READY# asserted, the PCI 9656 requires two additional Lwords to be read into the DMA FIFO.
- If the PCI destination address is X4h/XCh, and DREQ*x*# is asserted for one Local Bus Clock cycle, or is asserted and then de-asserted before or on the same cycle that the PCI 9656 asserts ADS#, the PCI 9656 requires 1 Lword to be read into the DMA FIFO.
- If the PCI destination address is X4h/XCh, and DREQ*x*# is asserted and remains asserted on the cycle that the PCI 9656 asserts ADS# and then de-asserts before any data is transferred (READY# not asserted), the PCI 9656 requires 2 Lwords to be read into the DMA FIFO.
- If the PCI destination address is X4h/XCh, and DREQ*x*# is asserted and held asserted until it is de-asserted on a cycle that is transferring the first or an even 2, 4, 6, 8… Lword with READY# asserted, the PCI 9656 requires one additional Lword to be read into the DMA FIFO.
- If the PCI destination address is X4h/XCh, and DREQ*x*# is asserted and held asserted until it is de-asserted on a cycle that is transferring the 3, 5, 7… (Odd) Lword with READY# asserted, the PCI 9656 requires two additional Lwords to be read into the DMA FIFO.
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is reading data (READY# asserted), the PCI 9656 asserts BLAST# and must read one additional Lword of data before releasing the Local Bus. Otherwise, upon the re-assertion of READY#, the PCI 9656 must read one Lword before asserting BLAST# and reading a second Lword.

Fast Terminate mode (DMAMODE*x*[15]=1)

- If DREQ*x*# is asserted for one Local clock period or is asserted and remains asserted until it is de-asserted before or during a Data transfer, the PCI 9656 does not assert BLAST#; however, it requires one to two additional Lwords to be read into the DMA FIFO.
- If DREQ*x*# is de-asserted during the Data transfer, the PCI 9656 can assert BLAST# if any of the following conditions are met:
  - Starting PCI address is X4h/XCh
  - Data is the last data for the entire DMA transfer
  - Local Bus Latency Timer expires
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is reading data (READY# asserted), the PCI 9656 reads one additional Lword of data (during the cycle that EOT# is asserted), then immediately releases the Local Bus. Otherwise, the PCI 9656 waits for the re-assertion of READY# to read 1 Lword of data before releasing the Local Bus.

5.4.4.11 End of Transfer (EOT#) Input

EOT# is a one-clock wide pulse that ends a DMA transfer. It should be asserted only when the PCI 9656 owns the Local Bus. Depending on whether Fast or Slow Terminate mode is selected, the DMA transfer ends without a BLAST# assertion (Fast Terminate mode) or with BLAST# asserted (Slow Terminate mode). The DMAMODE*x*[15:14] bits enable and control how the PCI 9656 responds to an EOT# input assertion.

In Fast Terminate mode, if EOT# is asserted while the PCI 9656 receives an external READY# signal assertion, the DMA Controller(s) ends the DMA transfer and releases the Local Bus. In Slow Terminate mode, if EOT# is asserted while the PCI 9656 receives an external READY# signal assertion, the DMA Controller(s) completes the current Lword and one additional Lword. If the DMA FIFO is full or empty after the Data phase in which EOT# is asserted, the second Lword is not transferred.

If the PCI 9656 does not require that BLAST# output be driven for the last data transferred when a DMA transfer is terminated using EOT#, the Fast Terminate mode setting (DMAMODE*x*[15]=1) may be used. If EOT# is asserted in Fast Terminate mode, the DMA Controller(s) terminates the DMA transfer and releases the Local Bus upon receiving an external READY# signal assertion. Or, the internal Wait State Counter(s) decrements to 0 for the current Lword when EOT# is asserted. If the data is the last data in the transfer, BLAST# is asserted on the last transfer.

If the PCI 9656 requires that BLAST# output be asserted on the last data transfer when a DMA transfer is terminated using EOT#, then the Slow Terminate mode setting (DMAMODE*x*[15]=0) must be used. If EOT# is asserted in Slow Terminate mode, the DMA Controller(s) transfers one or two additional Lwords, depending upon the Local Bus data width and the address when EOT# is asserted.

The DMA Controller(s) terminates a transfer on an Lword boundary after EOT# is asserted. For an 8-bit bus, the PCI 9656 terminates after transferring the last byte for the Lword. For a 16-bit bus, the PCI 9656 terminates after transferring the last word for the Lword.

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. EOT# has no effect when loading the descriptor from the PCI Bus.

5.4.4.12 DMA Arbitration

The PCI 9656 DMA Controller(s) releases control of the Local Bus (de-asserts LHOLD) when the following conditions occur:

- Local Bus Latency Timer is enabled (MARBR[16]=1) 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 9656 releases the Local Bus)
- Direct Slave access is pending
- EOT# input is received (if enabled)

The DMA Controller(s) 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(s) de-asserts PCI REQ# for a minimum of two PCI clocks.

5.4.4.13 Local Bus DMA Priority

The PCI 9656 supports programmable Local Bus arbitration priority for DMA Channel 0 and Channel 1, when both channels are active (priority set with MARBR[20:19]). DMA Block and Scatter/Gather modes have priority over DMA Demand mode. DMA transfer direction does ***not*** influence DMA channel priority.

There are three types of priorities:

- **Channel 0 Priority** – DMA Channel 0 completes the transfer on the Local Bus before Channel 1. If Channel 1 is performing a Data transfer, with Channel 0 set as highest priority and started, Channel 1 continues its transfer until the Local Bus Latency Timer (MARBR[7:0]) expires, preempted by a Direct Slave Data transfer, or another termination occurs (EOT# assertion, DREQ*x*# de-assertion, or BREQi assertion). Channel 0 then owns the Local Bus until transfer completion, before Channel 1 can continue the interrupted transfer, unless Channel 1 previously completed its transfer.
- **Channel 1 Priority** – DMA Channel 1 completes its transfer on the Local Bus before Channel 0. If Channel 0 is performing a Data transfer, with Channel 1 set as highest priority and started, Channel 0 continues its transfer until the Local Bus Latency Timer expires, preempted by a Direct Slave Data transfer, or another termination occurs (EOT# assertion, DREQ*x*# de-assertion, or BREQi assertion). Channel 1 then owns the Local Bus until transfer completion, before Channel 0 can continue the interrupted transfer, unless Channel 0 previously completed its transfer.
- **Rotational Priority** – Depends on the transfer direction, however, if the starting bus is the same for both DMA channels, in the freshly started DMA, Channel 0 always starts first. Rotational priority does not start unless the ongoing DMA channel Data transfer is interrupted by the Local Bus Latency Timer expiration, preempted by a Direct Slave Data transfer, or another termination occurs (EOT# assertion, DREQ*x*# de-assertion, or BREQi assertion). The other DMA channel then owns the Local Bus until the previously described interrupts or terminations occur. Rotational priority occurs each time a DMA channel loses Local Bus ownership, unless one of the DMA channels previously completed its transfer.

5.4.4.14 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 (MARBR[16]=1) and expires (MARBR[7:0]), the PCI 9656 completes the current Lword transfer and releases LHOLD. After its programmable Pause Timer expires, the PCI 9656 re-asserts LHOLD. The PCI 9656 continues to transfer when it receives LHOLDA. The PCI Bus transfer continues until the FIFO is empty for a PCI-to-Local transfer or full for a Local-to-PCI transfer.

The DMA transfer can be paused by writing 0 to the Channel Enable bit(s) (DMACSR*x*[0]=0). To acknowledge the disable, the PCI 9656 receives at least one data from the bus before it stops. However, this is ***not*** recommended during a burst.

The DMA Local Bus Latency Timer starts after the Local Bus is granted to the PCI 9656, and the Local Bus Pause Timer starts after the external Local Bus Arbiter de-asserts LHOLDA.

5.4.4.15 DMA FIFO Programmable Threshold

The PCI 9656 supports programmable DMA FIFO threshold (DMATHR). The DMATHR register can be programmed with any of four different FIFO Full/Empty conditions, for DMA Channel 0 and/or Channel 1, as listed in Table 5-8. (Refer to the DMATHR register and Table 11-7, "DMA Threshold Nybble Values," for further details.)

5.4.4.16 DMA PCI Master/Target Abort

During PCI Bus mastering and upon encountering non-existing or "bad" devices, the PCI 9656 PCI Master/Target Abort logic enables the PCI 9656 to successfully recover during transfers to and from other PCI devices. As a PCI master, if the PCI 9656 attempts an 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 or Target Abort bit (PCISR[13 or 12]=0, respectively) and continue by processing the next task. Not clearing the bit prevents the PCI 9656 from ever becoming a PCI Bus master again.

If a PCI Master/Target Abort or 256 consecutive Master Retry timeout is encountered during a transfer, the PCI 9656 asserts LSERR#, if enabled (INTCSR [0]=1, which can be used as an NMI). The PCI 9656 also flushes all PCI Bus Master FIFOs (DMA and Direct Master).

The PCI 9656 DMA channels function independently when a PCI Master/Target Abort occurs. If one DMA channel encounters a PCI Master/Target Abort, the FIFO on the aborted channel is flushed. PCI Master capabilities for both DMA channels are disabled until the Received Master/Target Abort bits are cleared (PCISR[13:12]=00b, respectively). However, if the second DMA channel begins a new operation while the first DMA channel is receiving a PCI Master/Target Abort, the FIFO contents of the second DMA channel are not affected, and its new or pending Direct Master operations successfully complete. (For details about PCI Master/Target Abort, refer to Section 5.4.1.9.)

Note: *In applications where both DMA channels are used and one of the DMA channels encounters a PCI Master/Target Abort, the Received Master/Target Abort bits should be cleared (PCISR[13:12]=00b, respectively) and the non-aborted DMA channel should be allowed to complete its Data transfer before the aborted DMA channel's registers are re-initialized and the transfer restarted.*

If a PCI Master/Target Abort is encountered during a DMA transfer, the PCI 9656 stores the PCI Abort Address into PABTADR[31:0]. PABTADR contents are updated for each PCI Master/Target Abort. The Received Master/Target Abort bits should be read and cleared before starting a new DMA or Direct Master, Type 0 or Type 1 Configuration transfer.

Table 5-9 outlines special PCI Master/Target Abort conditions for DMA mode Data transfers.

Table 5-8. DMATHR FIFO Threshold

DMA Transfer	Condition	Description
PCI-to-Local	DMA Channel *x* FIFO Almost Full	Number of full Qword entries (plus 1, times 2) in the FIFO before requesting the Local Bus for writes.
	DMA Channel *x* FIFO Almost Empty	Number of empty Qword entries (plus 1, times 2) in the FIFO before requesting the PCI Bus for reads.
Local-to-PCI	DMA Channel *x* FIFO Almost Full	Number of full Qword entries (plus 1, times 2) in the FIFO before requesting the PCI Bus for writes.
	DMA Channel *x* FIFO Almost Empty	Number of empty Qword entries (plus 1, times 2) in the FIFO before requesting the Local Bus for reads.

Table 5-9. DMA Mode Data Transfer Special PCI/Master/Target Abort Conditions

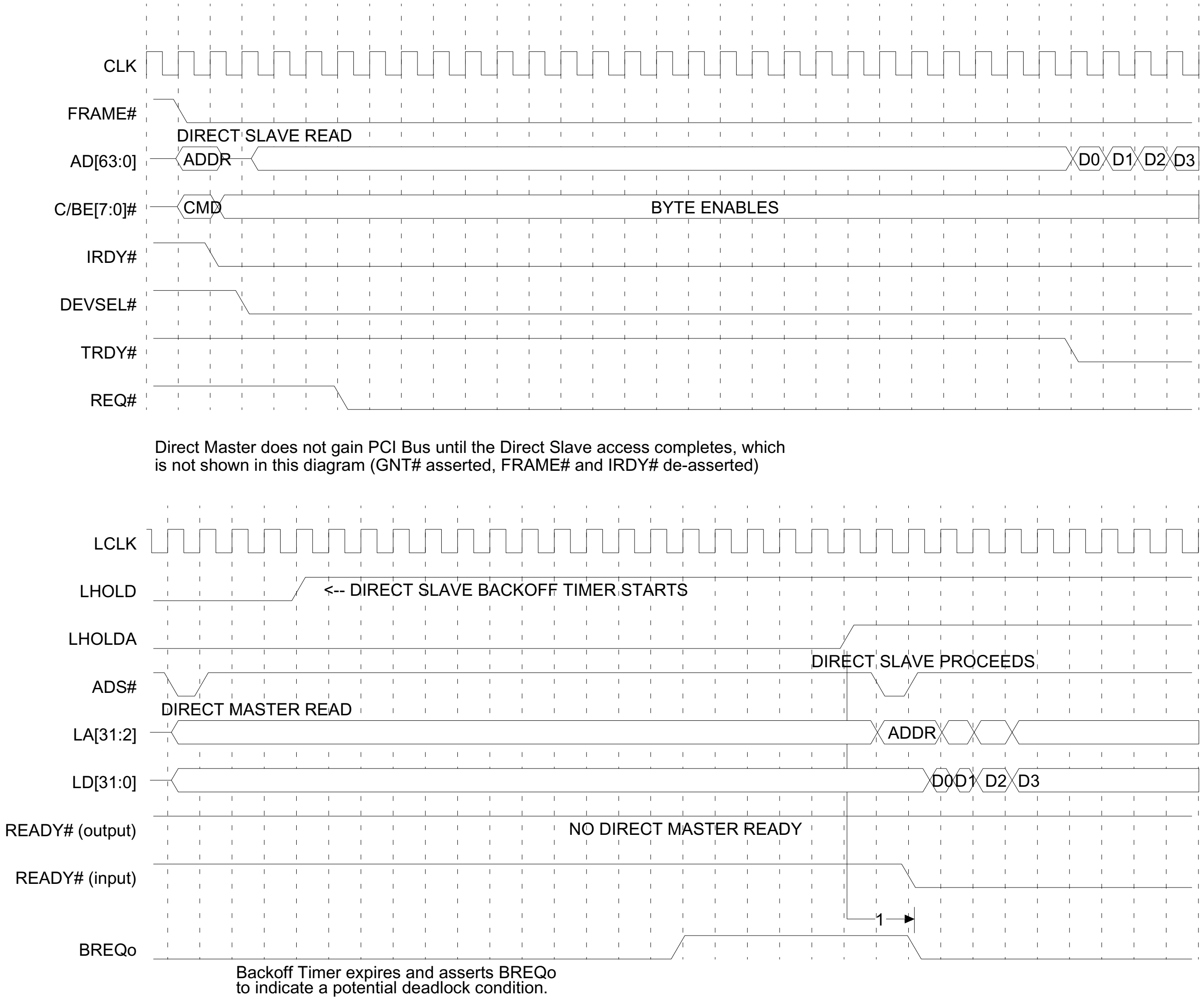
DMA Mode Data Transfer Type	PCI Master/Target Abort Condition
DMA Local-to-PCI in Slow Terminate Mode (DMAMODE*x*[15]=0	After encountering a PCI Master/Target Abort, the PCI 9656 immediately terminates data transfer on the PCI Bus and continues reading additional data from the Local Bus into the DMA Local-to-PCI FIFO until its full or transfer count is reached. (Refer to the DMATHR register.) The PCI 9656 then sets the DMA Channel Done bit (DMACSR*x*[4]=1) and the Received Master/Target Abort bits can be cleared (PCISR[13:12]=00b, respectively).
DMA Local-to-PCI in Fast Terminate Mode (DMAMODE*x*[15]=1)	After encountering a PCI Master/Target Abort, the PCI 9656 immediately terminates data transfer on the PCI and Local Buses and sets the DMA Channel Done bit (DMACSR*x*[4]=1).
DMA PCI-to-Local Independent of Fast/Slow Terminate Mode (DMAMODE*x*[15]=0 or 1)	After encountering a PCI Master/Target Abort on the PCI Bus, the PCI 9656 immediately terminates a Burst Data transfer on the Local Bus and single cycles data from the DMA PCI-to-Local FIFO until it is empty.
DMA Local-to-PCI, with EOT# assertion on the Local Bus	Upon encountering a PCI Master/Target Abort on the PCI Bus with EOT# assertion on the Local Bus, the PCI 9656 terminates data transfer on the PCI and Local Buses and sets the DMA Channel Done bit (DMACSR*x*[4]=1).
DMA PCI-to-Local, with EOT# assertion on the Local Bus	Upon encountering a PCI Master/Target Abort on the PCI Bus with EOT# assertion on the Local Bus, the PCI 9656 flushes the DMA PCI-to-Local FIFO before encountering the PCI Master/Target Abort.
DMA PCI-to-Local Scatter/Gather, Ring Management with EOT# assertion on the Local Bus	With the EOT# function enabled (DMAMODE*x*[14]=1), EOT# End Link enabled (DMAMODE*x*[19]=1), and the DMA Descriptor Links located on the Local Bus, the PCI 9656 starts the DMA Descriptor load and then transfers data after the DMA Channel Enable and Start bits are set (DMACSR*x*[0:1]=11b, respectively). Upon receiving the PCI Target Abort on the PCI Bus and EOT# assertion on the Local Bus during a Scatter/Gather DMA PCI-to-Local data transfer, the PCI 9656 pauses the transfer in response to the PCI Target Abort. The PCI 9656 then flushes the DMA FIFO and sets the DMA Channel Done bit (DMACSR*x*[4]=1) after sampling EOT# asserted. No further DMA Descriptor load is performed after the Received Target Abort bit is cleared (PCISR[12]=0), and the Scatter/Gather DMA transfer is terminated.

5.5 C AND J MODES FUNCTIONAL TIMING DIAGRAMS

General notes about timing designer (graphical) waveforms:
The graphical Timing Diagrams were created using the Timing Designer tool. They are accurate and adhere to their specified protocol(s). These diagrams show transfers and signal transitions, but should not be relied upon to show exactly, on a clock-for-clock basis, where PCI 9656-driven signal transitions will occur.

General notes about captured waveforms:
The captured Timing Diagrams were captured from a simulation signal display tool that displays the results of stimulus run on the netlist. Using the netlist illustrates realistic delay on signals driven by the PCI 9656. Signals driven by the test environment to the PCI 9656 are quite fast. Leading zeros for buses such as AD[63:0], LD[31:0] (C mode), or LAD[31:0] (J mode), may or may not be shown. If no value for a bus is shown while the PCI 9656 is executing a transfer, it is because the entire value cannot be displayed in the available space or is irrelevant. When the PCI 9656 is not executing a transfer on that bus, the value is not shown because it is either irrelevant or unknown (any or all signals may be 1, 0, or Z).

Timing Diagram 5-1. BREQo and Deadlock


CLK
FRAME#
DIRECT SLAVE READ
AD[63:0]
ADDR
D0
D1
D2
D3
C/BE[7:0]#
CMD
BYTE ENABLES
IRDY#
DEVSEL#
TRDY#
REQ#
Direct Master does not gain PCI Bus until the Direct Slave access completes, which is not shown in this diagram (GNT# asserted, FRAME# and IRDY# de-asserted)
LCLK
LHOLD
<-- DIRECT SLAVE BACKOFF TIMER STARTS
LHOLDA
DIRECT SLAVE PROCEEDS
ADS#
DIRECT MASTER READ
LA[31:2]
ADDR
LD[31:0]
D0
D1
D2
D3
READY# (output)
NO DIRECT MASTER READY
READY# (input)
1
BREQo
Backoff Timer expires and asserts BREQo to indicate a potential deadlock condition.

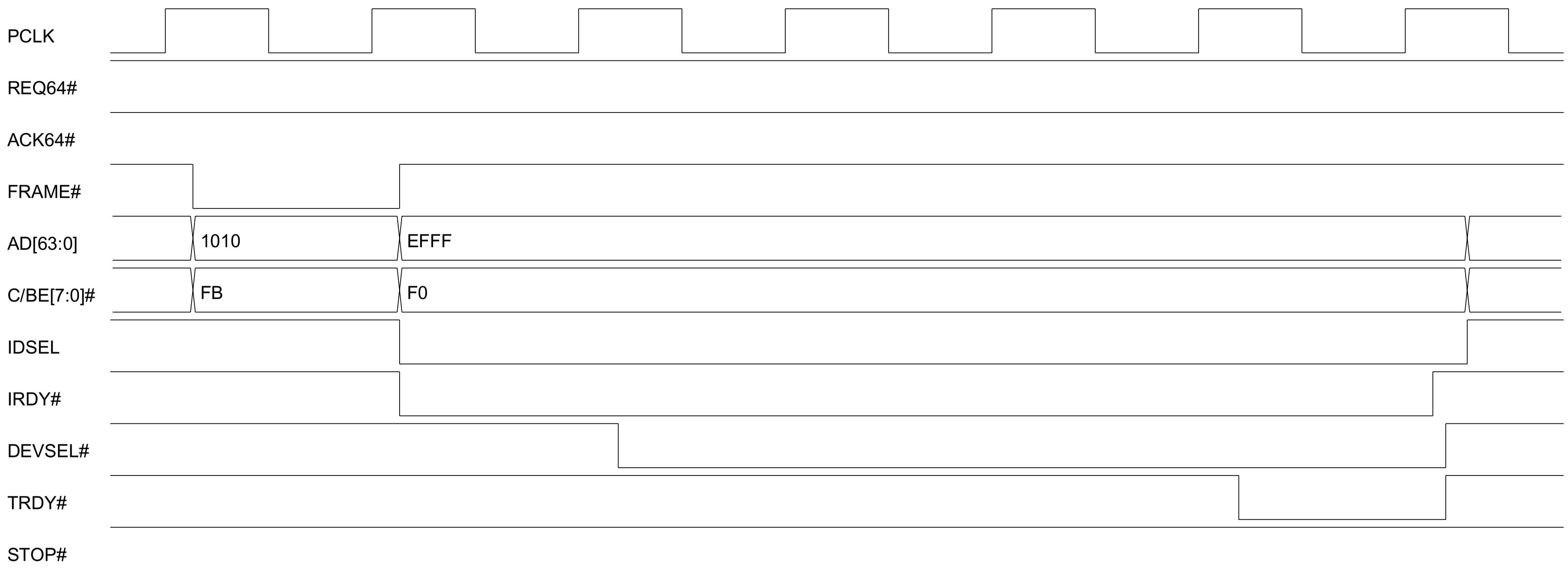
__Notes:__ For partial deadlock, Direct Slave Retry Delay Clocks bits (LBRD0[31:28]) can be used to issue Retrys to the Direct Master attempting the Direct Slave access.

Timing Diagram 5-1 contains C mode signals/names. The overall behavior in J mode is the same. Arrow 1 indicates that LHOLDA assertion causes the PCI 9656 to de-assert BREQo. The READY# (output and input) signals are the same pin. The PCI 9656 READY# pin is an input while processing Direct Slave reads and an output when processing Direct Master reads.

In this diagram, the PCI 9656 is configured to be a 32-bit PCI device [REQ64# is de-asserted when RST# transitions high (not shown)].

5.5.1 Configuration Timing Diagrams

Timing Diagram 5-2. PCI Configuration Write to PCI Configuration Register

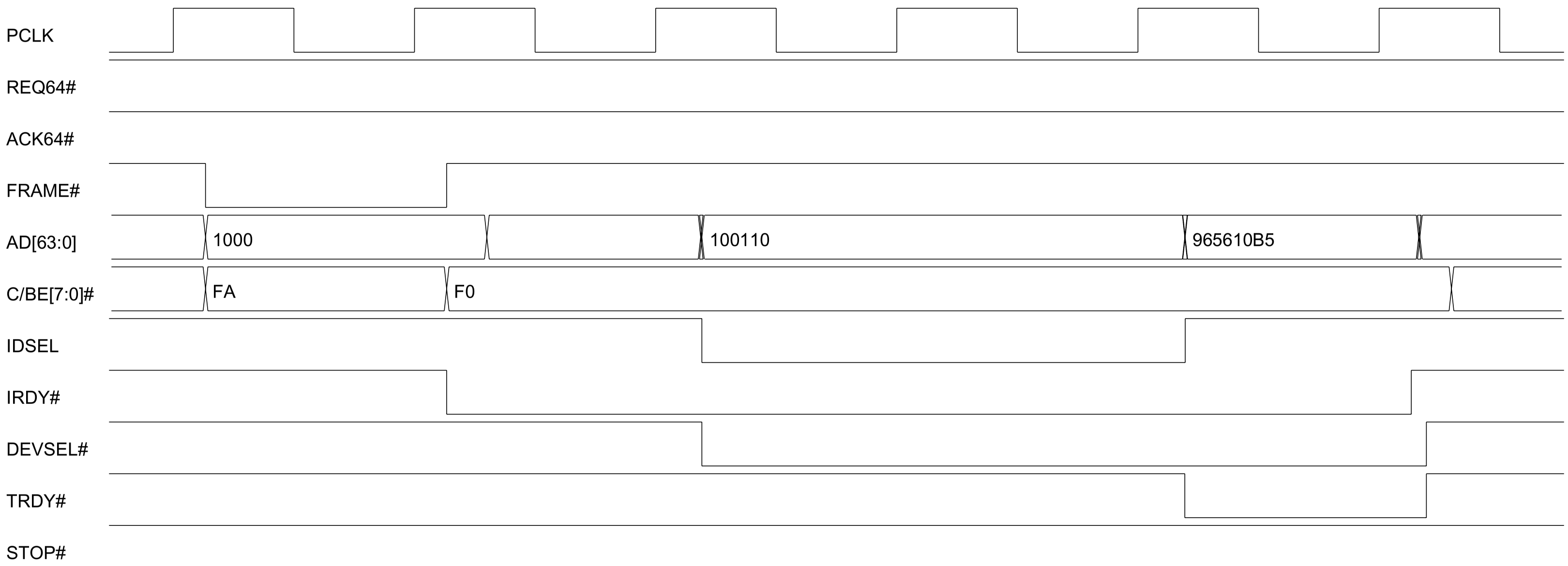


Notes: *PCI Configuration write to PCIBAR0 (PCI:10h) register.*

IDSEL = AD[12].

Leading zeros for AD[63:0] bus are not shown.

Timing Diagram 5-3. PCI Configuration Read of PCI Configuration Register

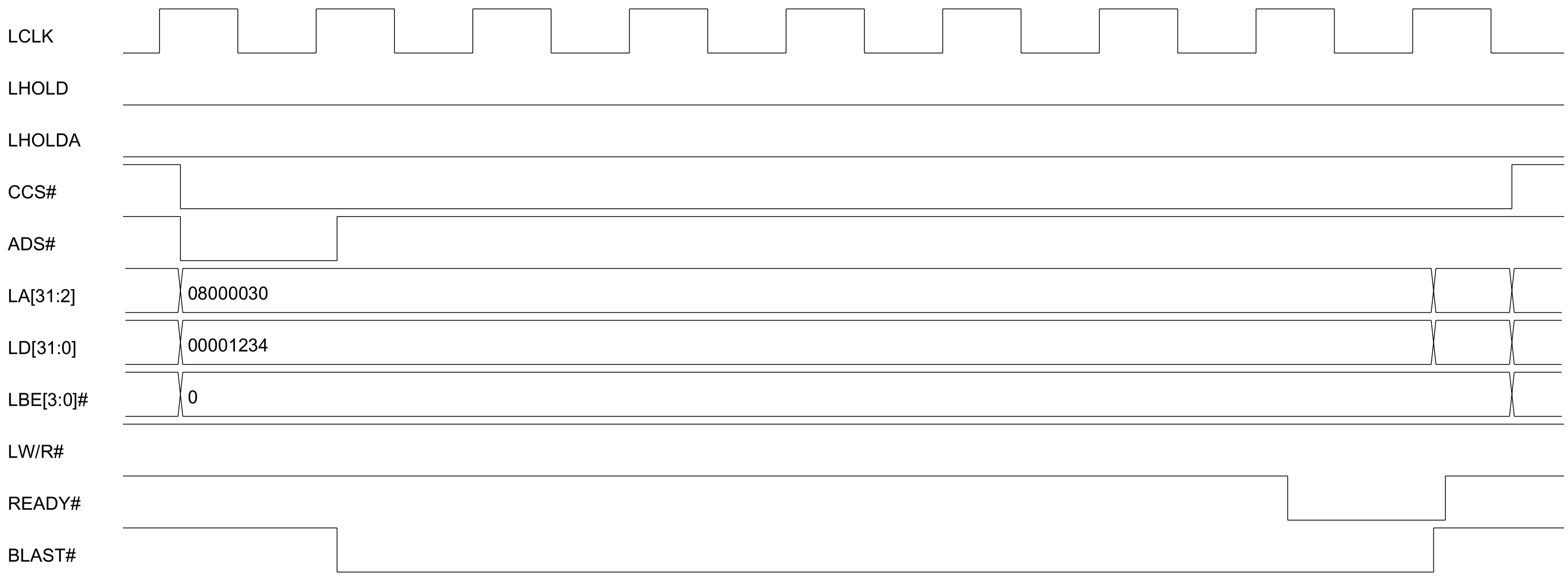


Notes: *PCI Configuration read of PCIIDR (PCI:00h) register.*

IDSEL = AD[12].

Leading zeros for AD[63:0] bus are not shown.

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



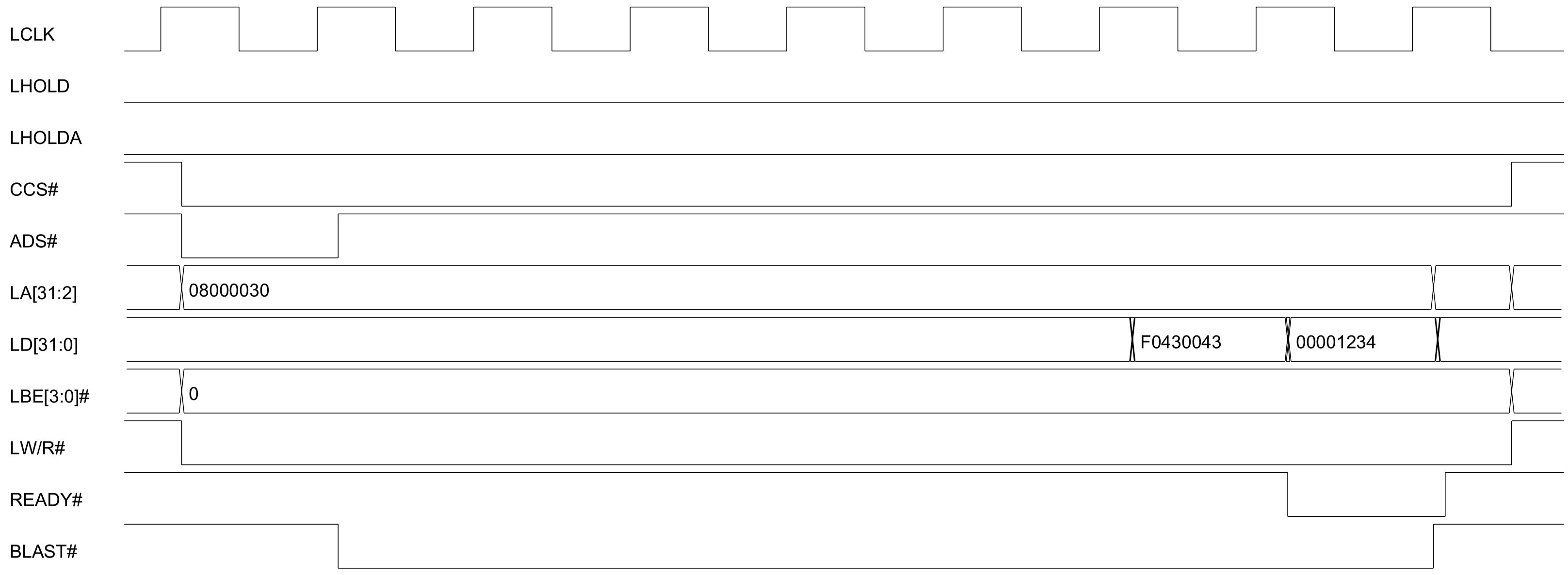
Notes: *Local Configuration write to MBOX0 (LOC:C0h) register.*

CCS# is asserted for multiple clock cycles, but it is required to be asserted only during the ADS# clock cycle.

Only the LA[8:2] signals are used to decode the register.

LA[31:2] is an Lword address (Master accesses to the PCI 9656 are always 32-bit data quantities).

Timing Diagram 5-5. Local Read of Configuration Register (C Mode)



Notes: *Local Configuration read of MBOX0 (LOC:C0h) register.*

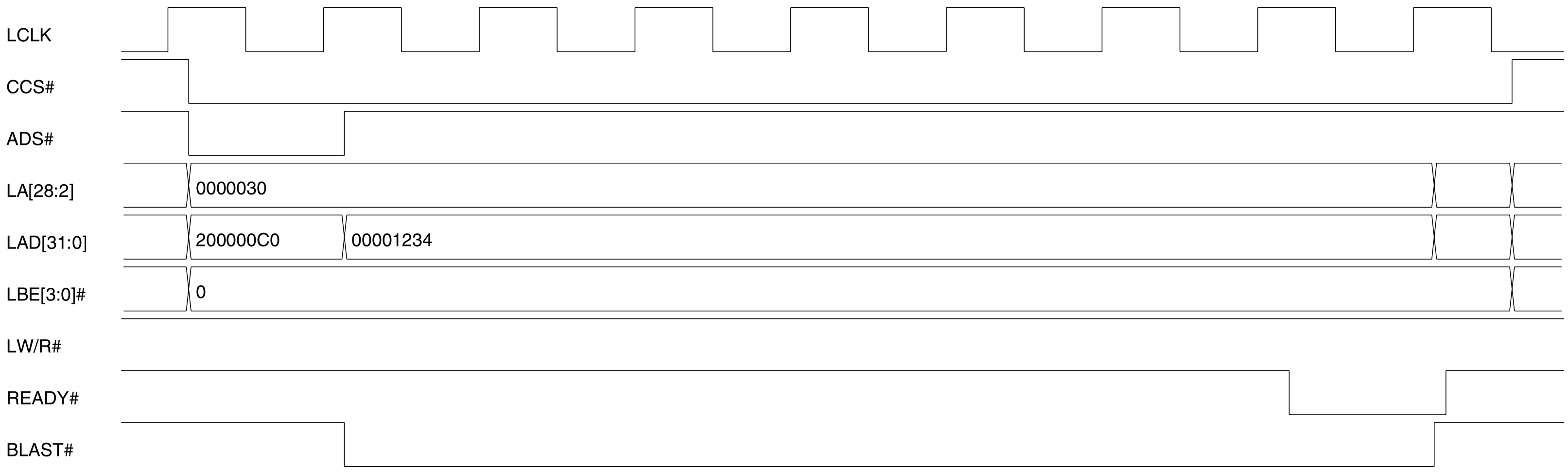
CCS# is asserted for multiple clock cycles, but it is required to be asserted only during the ADS# clock cycle.

Only the LA[8:2] signals are used to decode the register.

LA[31:2] is the Lword address (Master accesses to the PCI 9656 are always 32-bit data quantities).

The PCI 9656 starts driving the LD[31:0] bus with meaningless data (F0430043), one cycle before it asserts the READY# signal with the data read from the register.

Timing Diagram 5-6. Local Write to Configuration Register (J Mode)



Notes: *Local Configuration write to MBOX0 (LOC:C0h) register.*

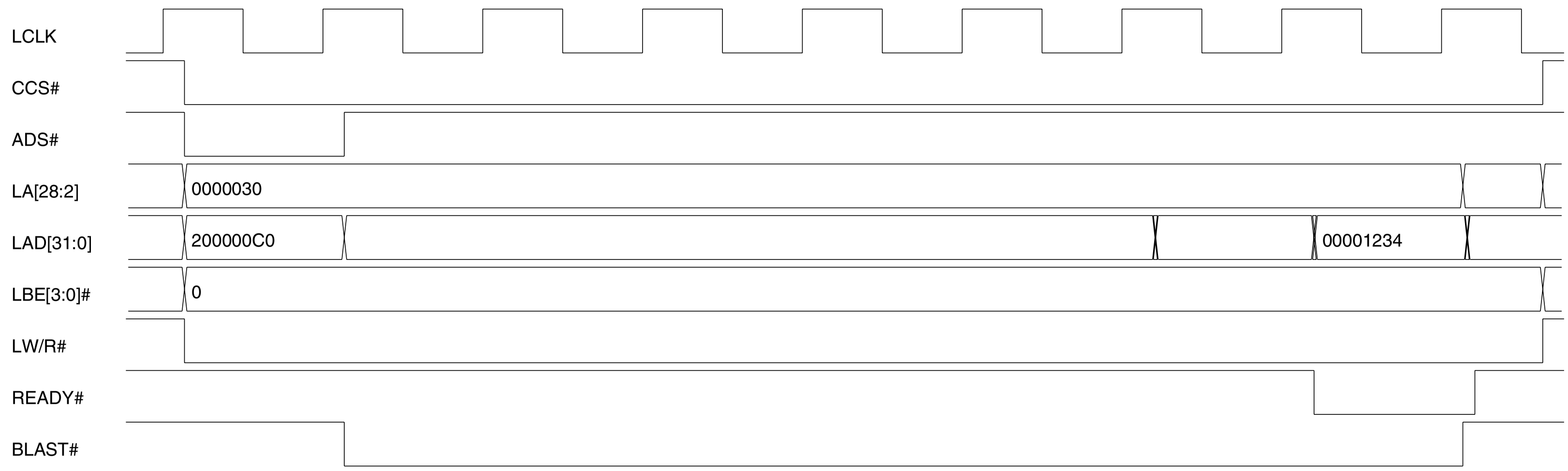
CCS# is asserted for multiple clock cycles, but it is only required to be asserted during the ADS# clock cycle (or while ALE is asserted, which is not shown).

Only the LAD[8:2] signals are used to decode the MBOX0 register.

LA[28:2] is shown, but is not used by the PCI 9656 during Local Master accesses to the PCI 9656.

Local Master accesses to the PCI 9656 are always 32-bit data quantities.

Timing Diagram 5-7. Local Read of Configuration Register (J Mode)



Notes: *Local Configuration read to MBOX0 (LOC:C0h) register.*

CCS# is asserted for multiple clock cycles, but it is only required to be asserted during the ADS# clock cycle (or while ALE is asserted, which is not shown).

Only the LAD[8:2] signals are used to decode the MBOX0 register.

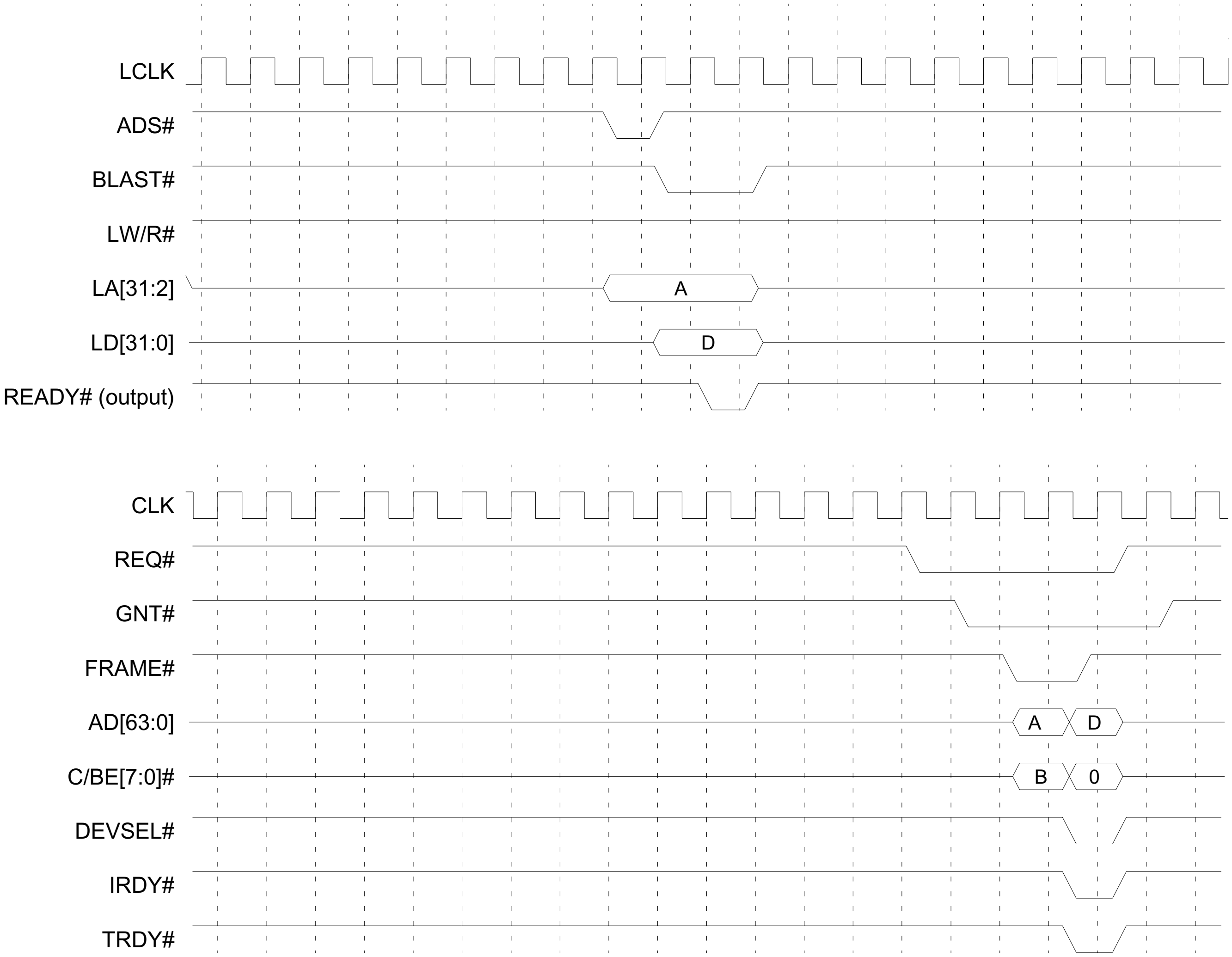
LA[28:2] is shown, but is not used by the PCI 9656 when a Local Master accesses the PCI 9656.

Local Master accesses to the PCI 9656 are always 32-bit data quantities.

5.6 C MODE FUNCTIONAL TIMING DIAGRAMS

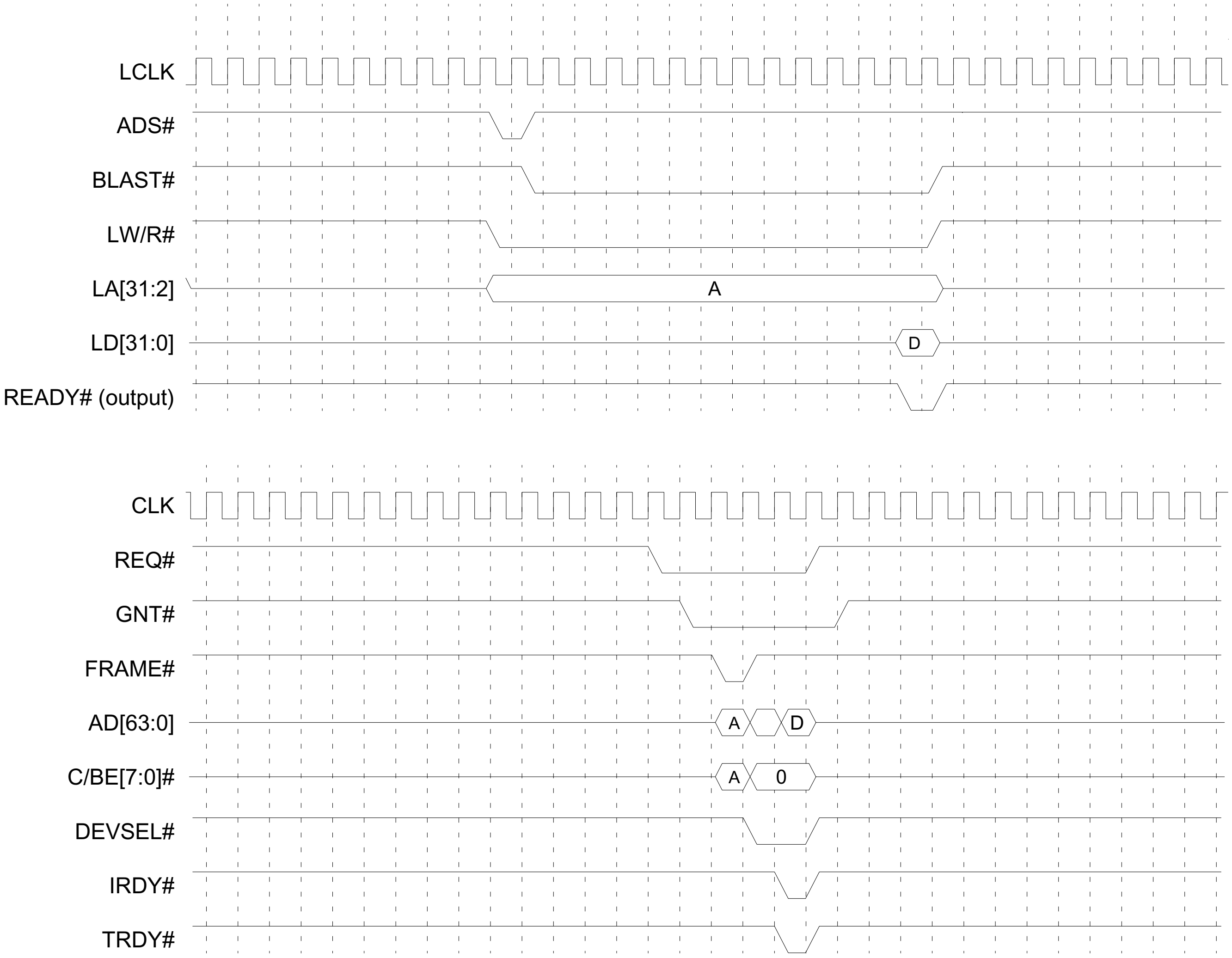
5.6.1 C Mode Direct Master Timing Diagrams

Timing Diagram 5-8. Direct Master Configuration Write – Type 0 or Type 1



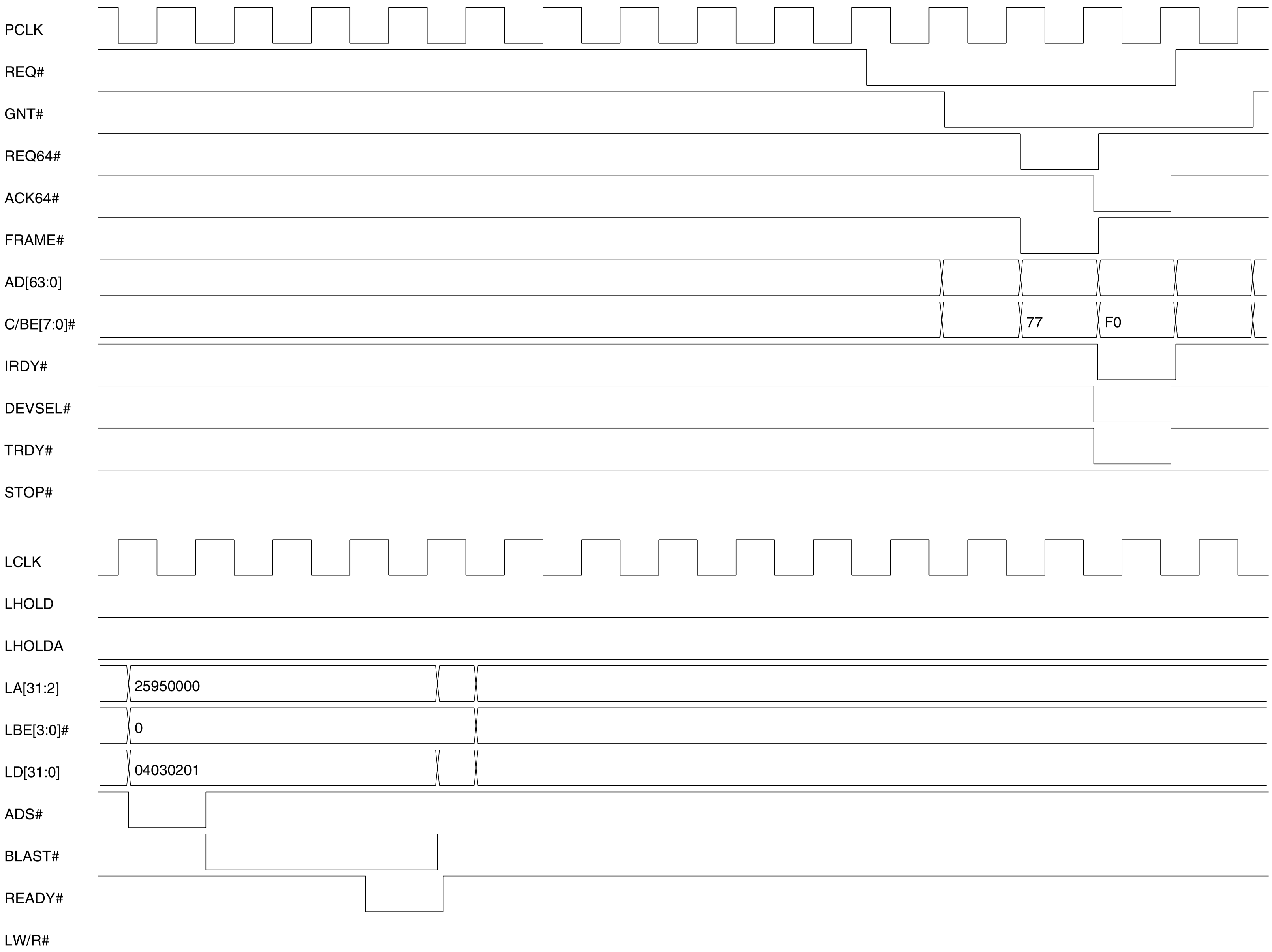
Note: *This diagram illustrates a Direct Master Configuration write that causes the PCI 9656 to generate a 32-bit PCI Type 0 or Type 1 Configuration Write cycle. Refer to Section 5.4.1.7 for further details. AD[63:32] and C/BE[7:4]# are driven to known values by the PCI 9656, but are irrelevant.*

Timing Diagram 5-9. Direct Master Configuration Read – Type 0 or Type 1



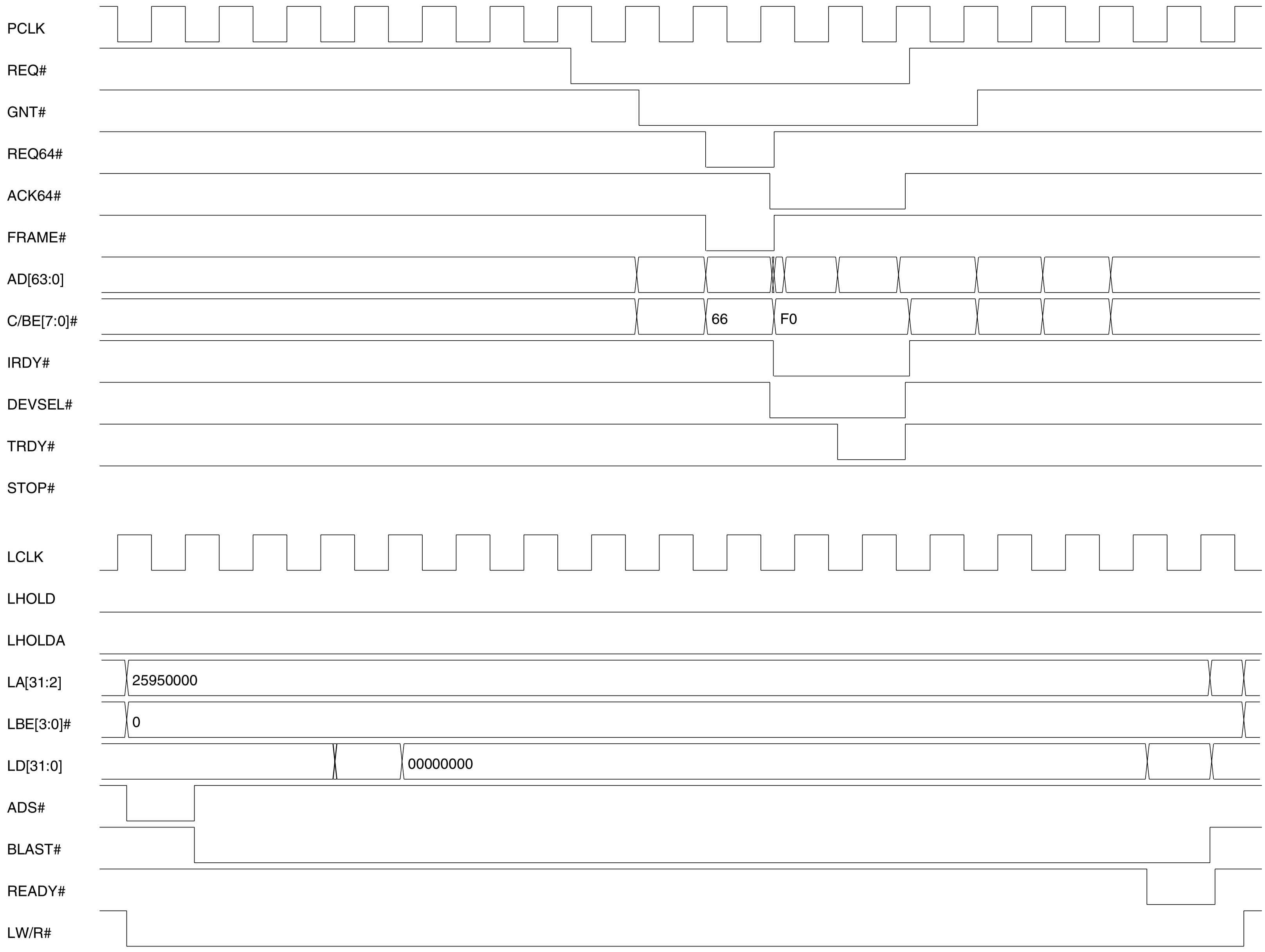
***Note:** This diagram illustrates a Direct Master Configuration read that causes the PCI 9656 to generate a 32-bit PCI Type 0 or Type 1 Configuration Read cycle. Refer to Section 5.4.1.7 for further details. AD[63:32] and C/BE[7:4]# are driven to known values by the PCI 9656, but are irrelevant.*

Timing Diagram 5-10. Direct Master Single Cycle Write (64-Bit PCI Bus)



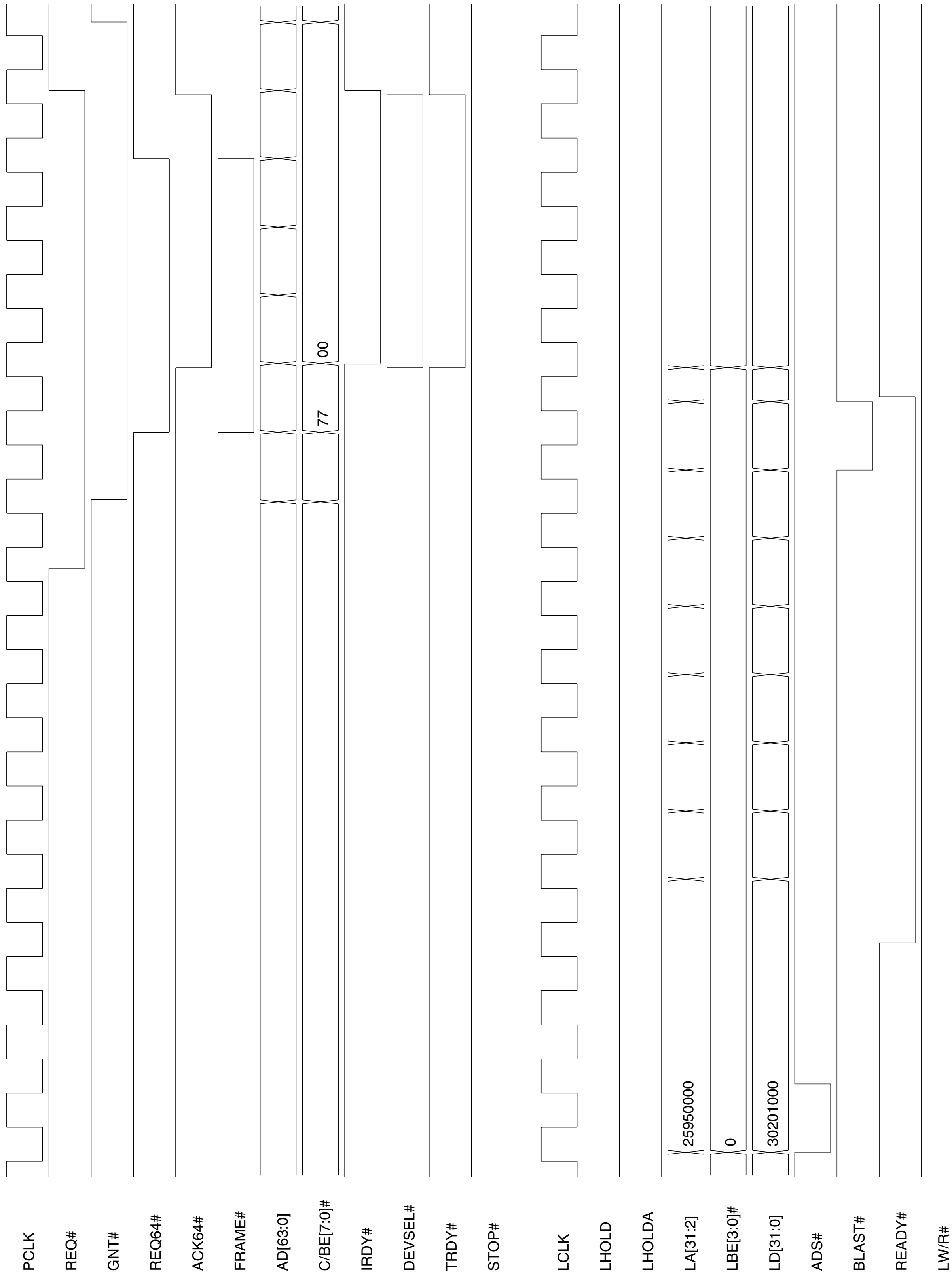
Note: *Key register value is DMPBAM[15:0]=00E3h.*

Timing Diagram 5-11. Direct Master Single Cycle Read (64-Bit PCI Bus)



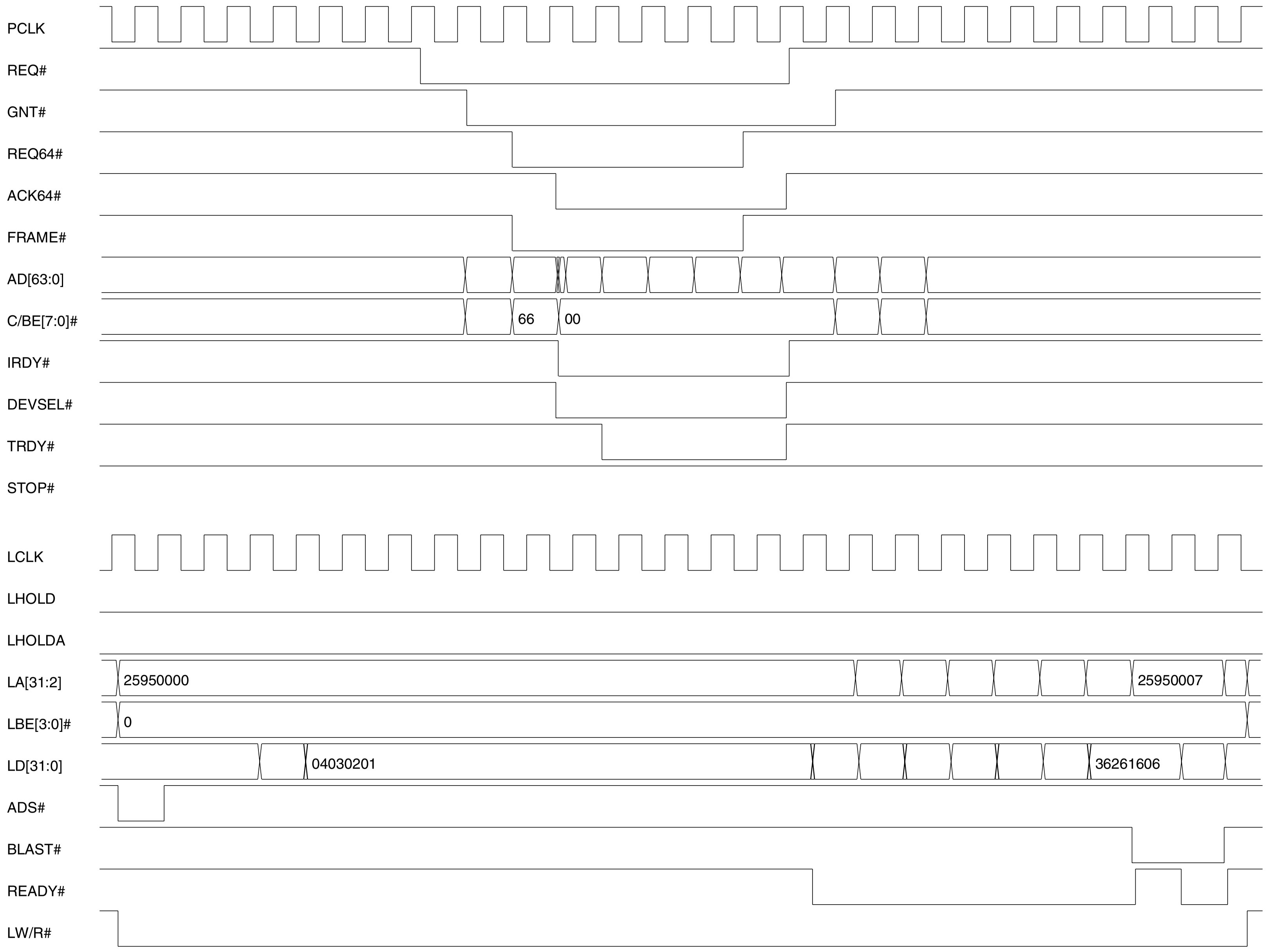
Note: *Key register value is DMPBAM[15:0]=00E3h.*

Timing Diagram 5-12. Direct Master Burst Write of 8 Lwords (64-Bit PCI Bus)



Note: *Key register value is DMPBAM[15:0]=000Bh.*

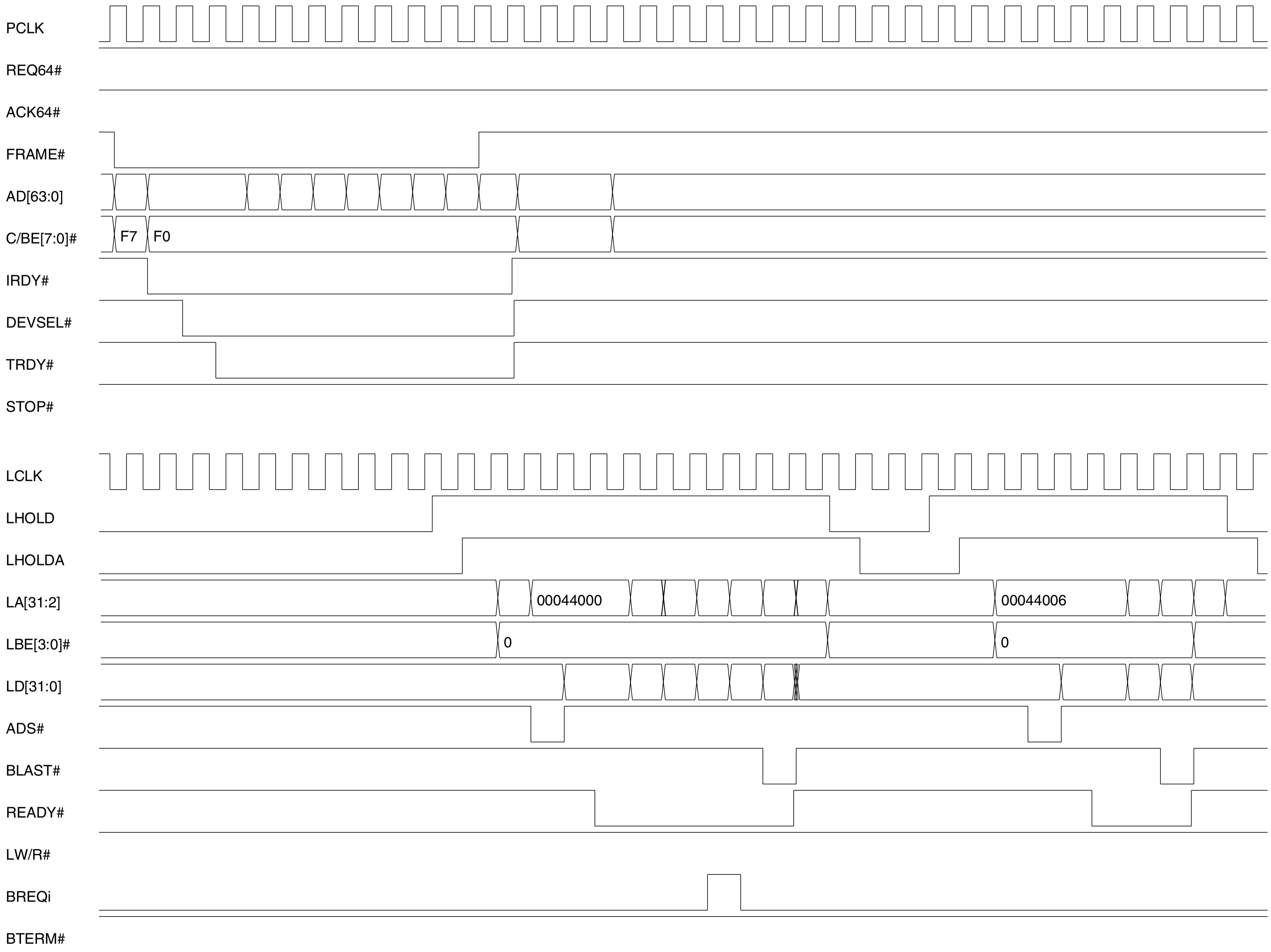
Timing Diagram 5-13. Direct Master Burst Read of 8 Lwords (64-Bit PCI Bus)



Note: *Key register value is DMPBAM[15:0]=000Bh.*

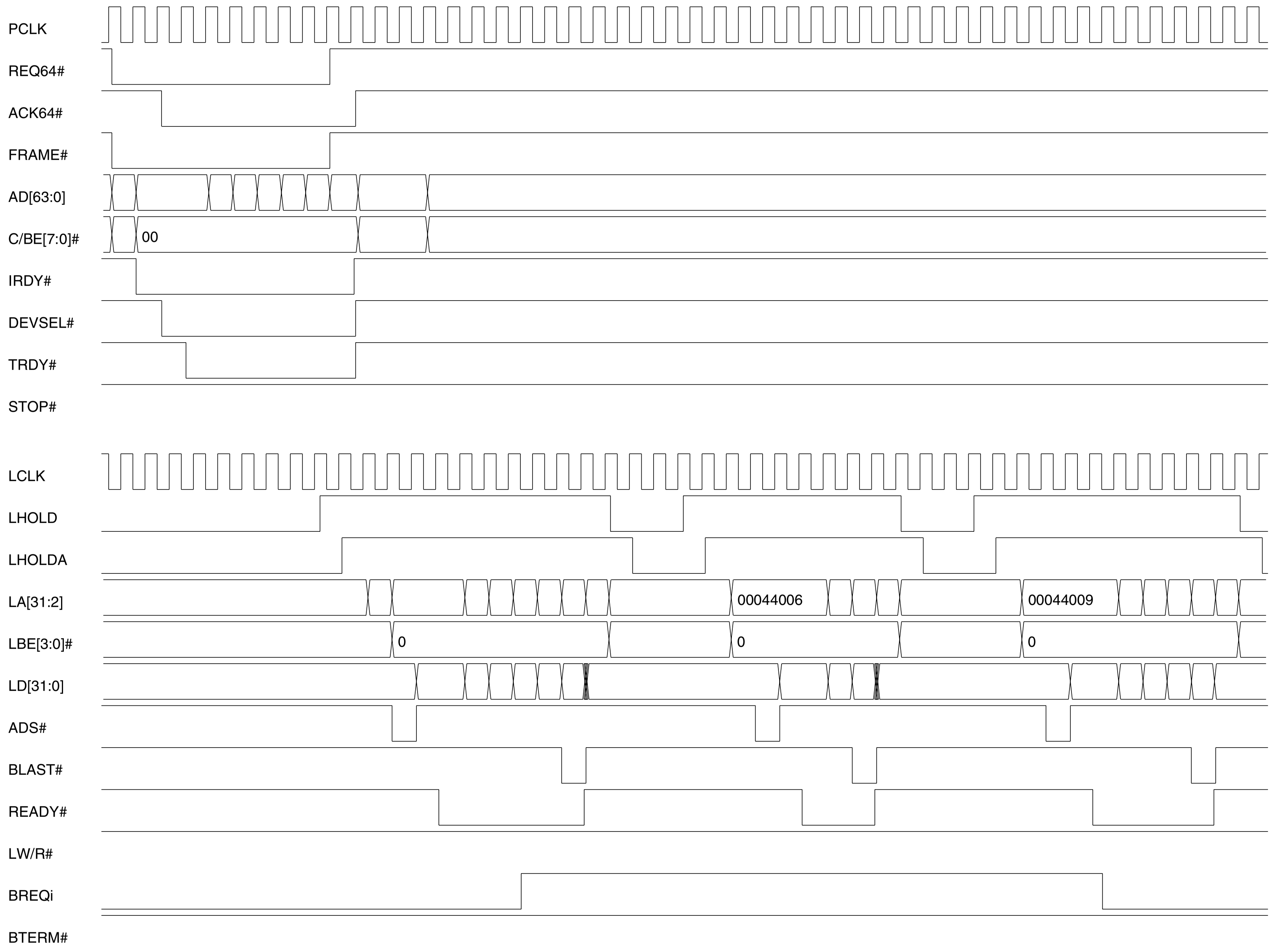
5.6.2 C Mode Direct Slave Timing Diagrams

Timing Diagram 5-14. Direct Slave Burst Write Suspended by BREQi (32-Bit PCI Bus)



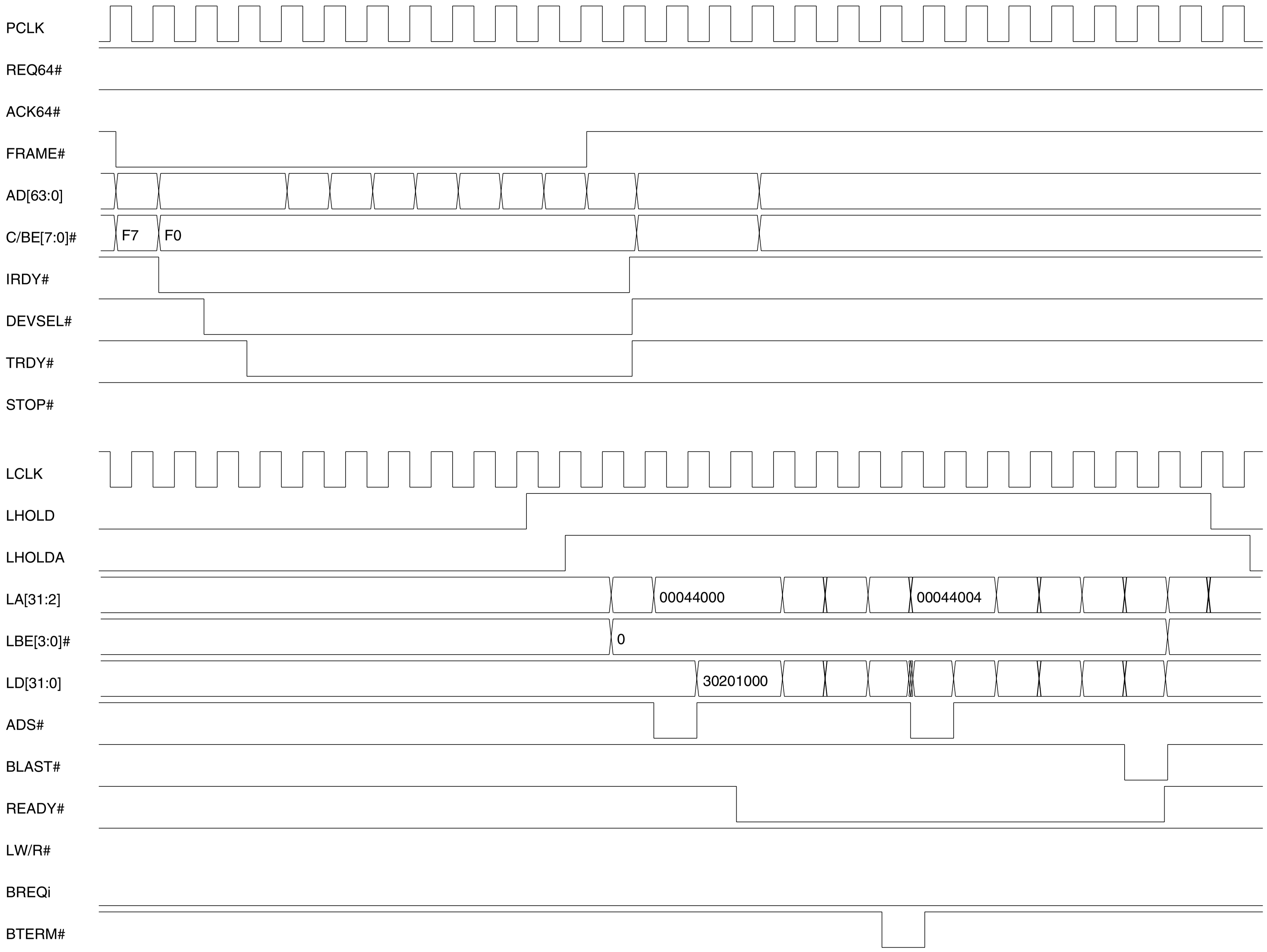
Notes: *This 9-Lword Direct Slave Burst Write transfer is suspended by a one-clock cycle BREQi assertion.*
The remaining three Lwords are transferred once the PCI 9656 is granted the Local Bus again.
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.

Timing Diagram 5-15. Direct Slave Burst Write Suspended by BREQi (64-Bit PCI Bus)



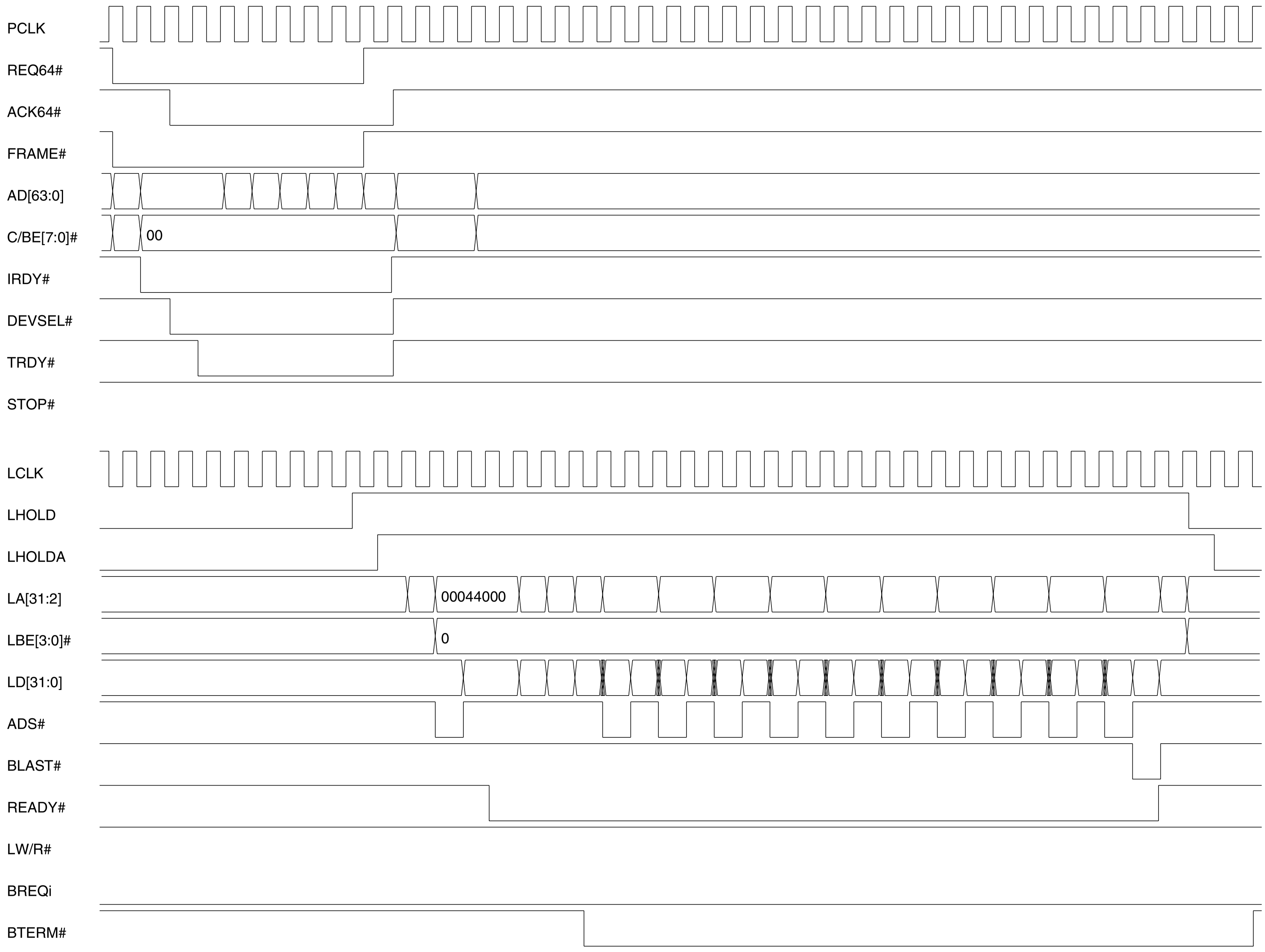
Notes: *This 14-Lword Direct Slave Burst Write transfer is suspended twice by a multi-clock cycle BREQi assertion.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.

Timing Diagram 5-16. Direct Slave Burst Write Interrupted by BTERM# (32-Bit PCI Bus)



Notes: *This 9-Lword Direct Slave Burst Write transfer is interrupted by a one-clock cycle BTERM# assertion. The remaining five Lwords are transferred after the PCI 9656 generates a new Address phase on the Local Bus. Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.*

Timing Diagram 5-17. Direct Slave Burst Write Interrupted by BTERM# (64-Bit PCI Bus)

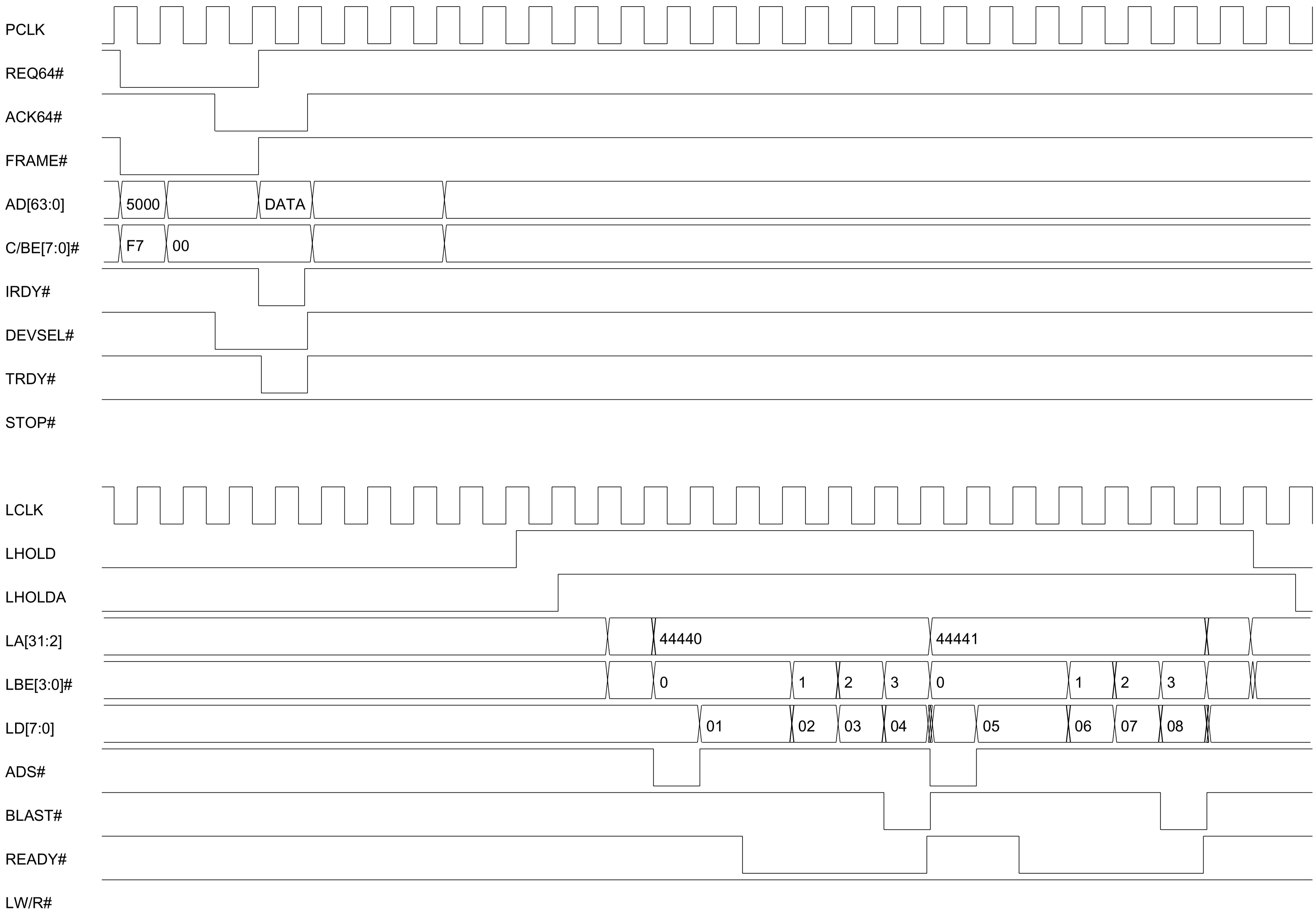


Notes: *This 14-Lword Direct Slave Burst Write transfer is interrupted multiple times by a multi-clock cycle BTERM# assertion.*

The PCI 9656 generates a new Address phase for each Lword of data transferred, as long as BTERM# remains asserted.

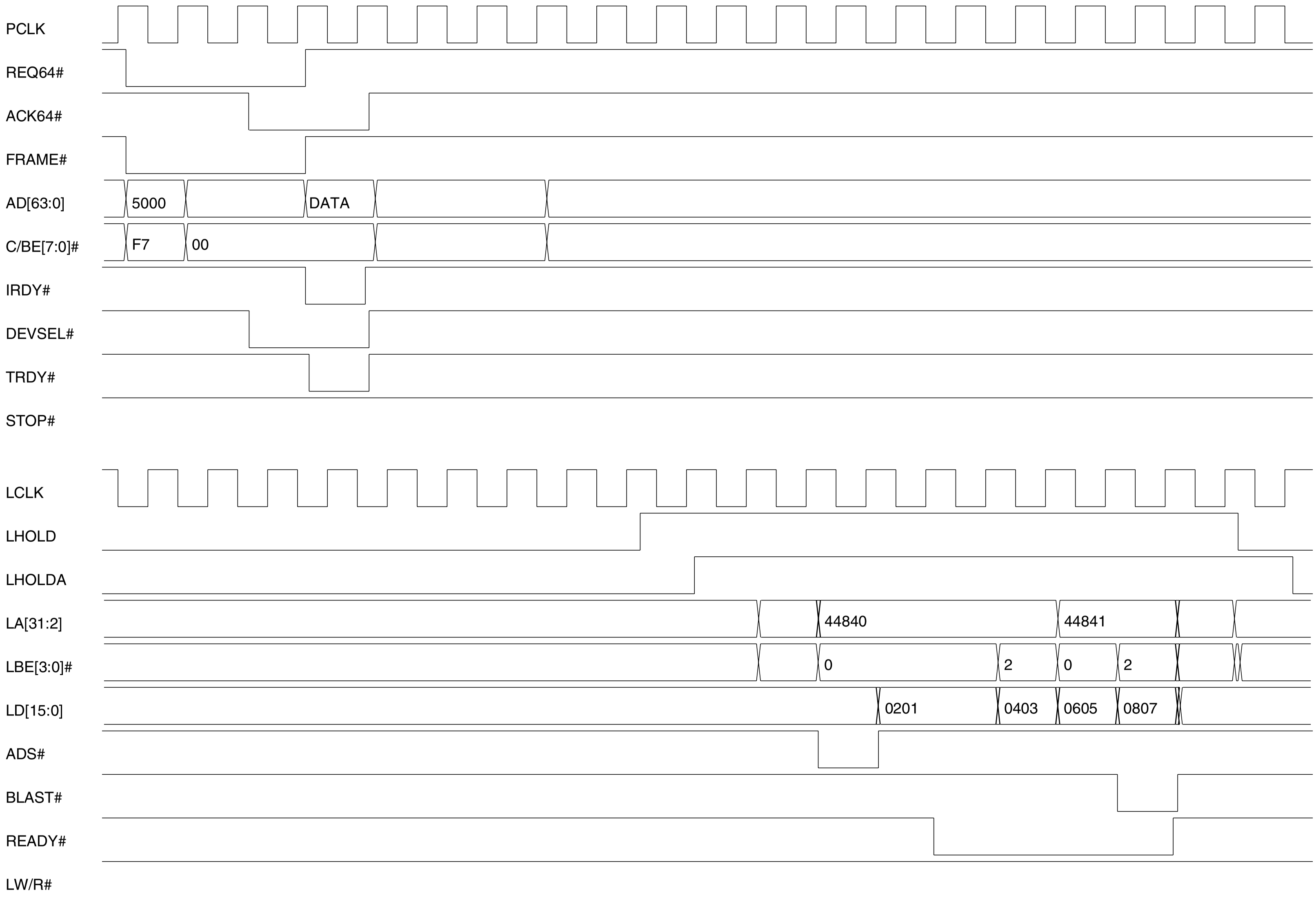
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.

Timing Diagram 5-18. Direct Slave 64-Bit Single Cycle Write (8-Bit Local Bus)



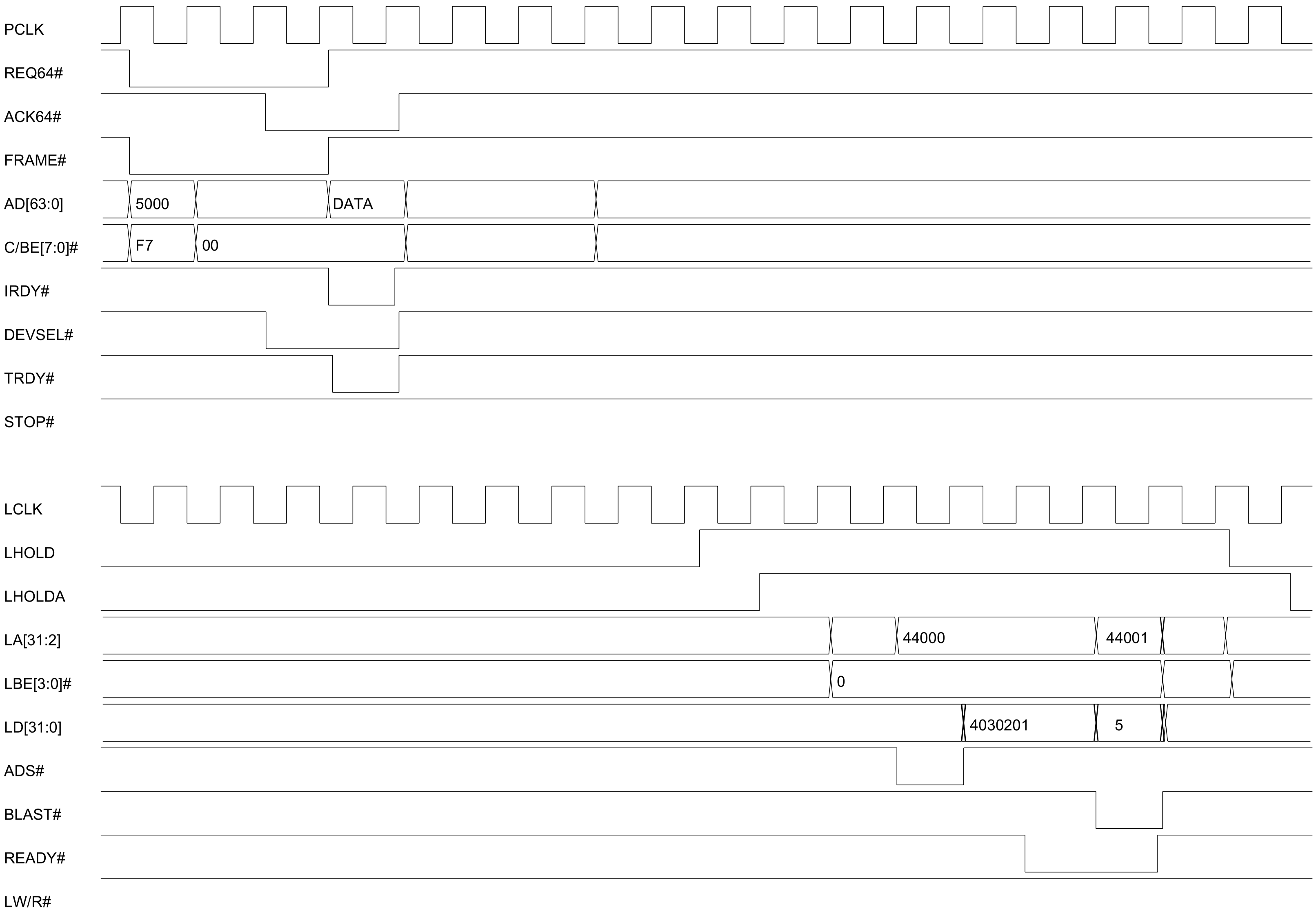
Notes: *Hex value for AD[63:0] DATA = 0807_0605_0403_0201h.*
Key bit/register values are MARBR[24]=1 and LBRD1[15:0]=1540h.

Timing Diagram 5-19. Direct Slave 64-Bit Single Cycle Write (16-Bit Local Bus)



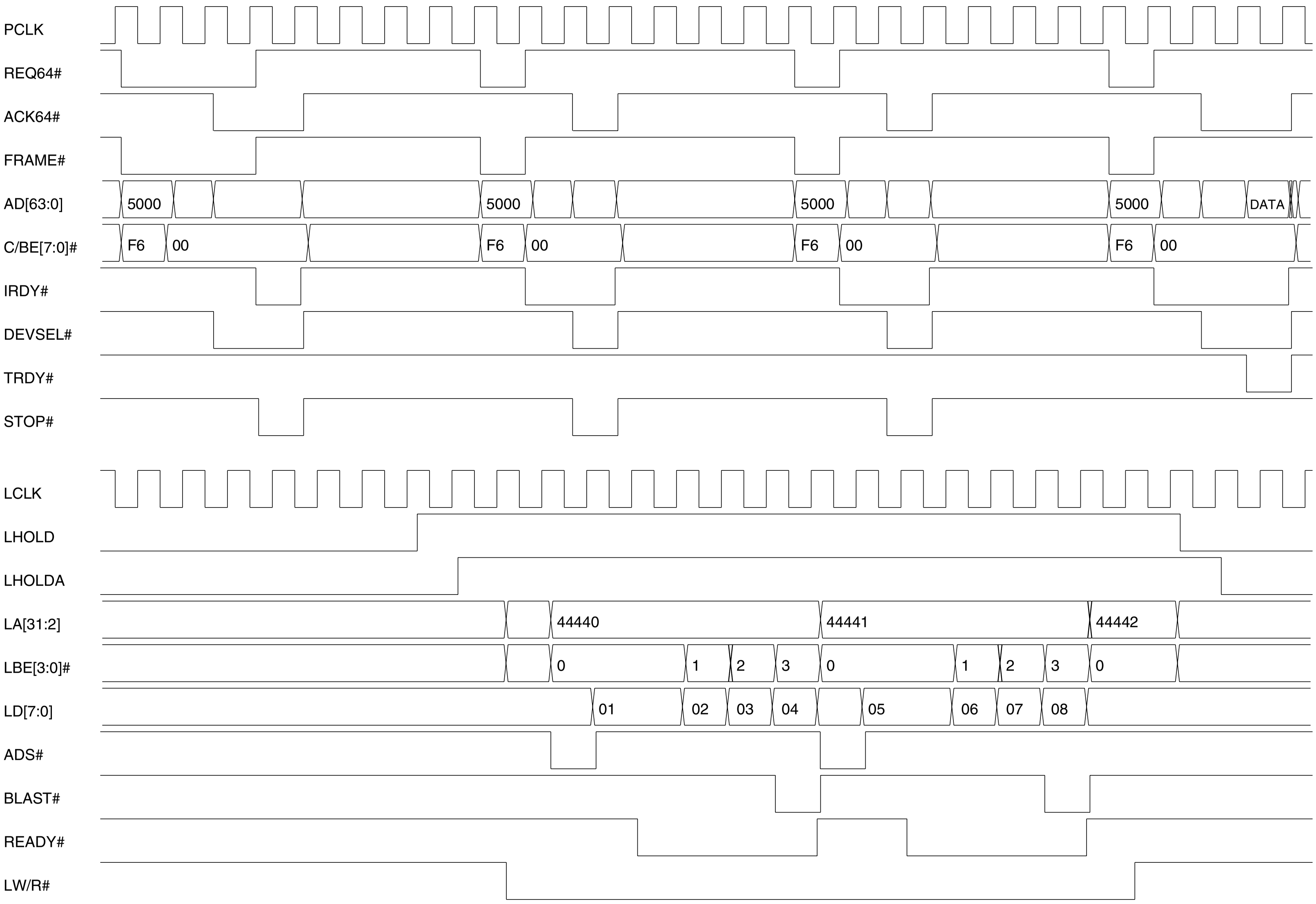
Notes: *Hex value for AD[63:0] DATA = 0807_0605_0403_0201h.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=2541h.

Timing Diagram 5-20. Direct Slave 64-Bit Single Cycle Write (32-Bit Local Bus)



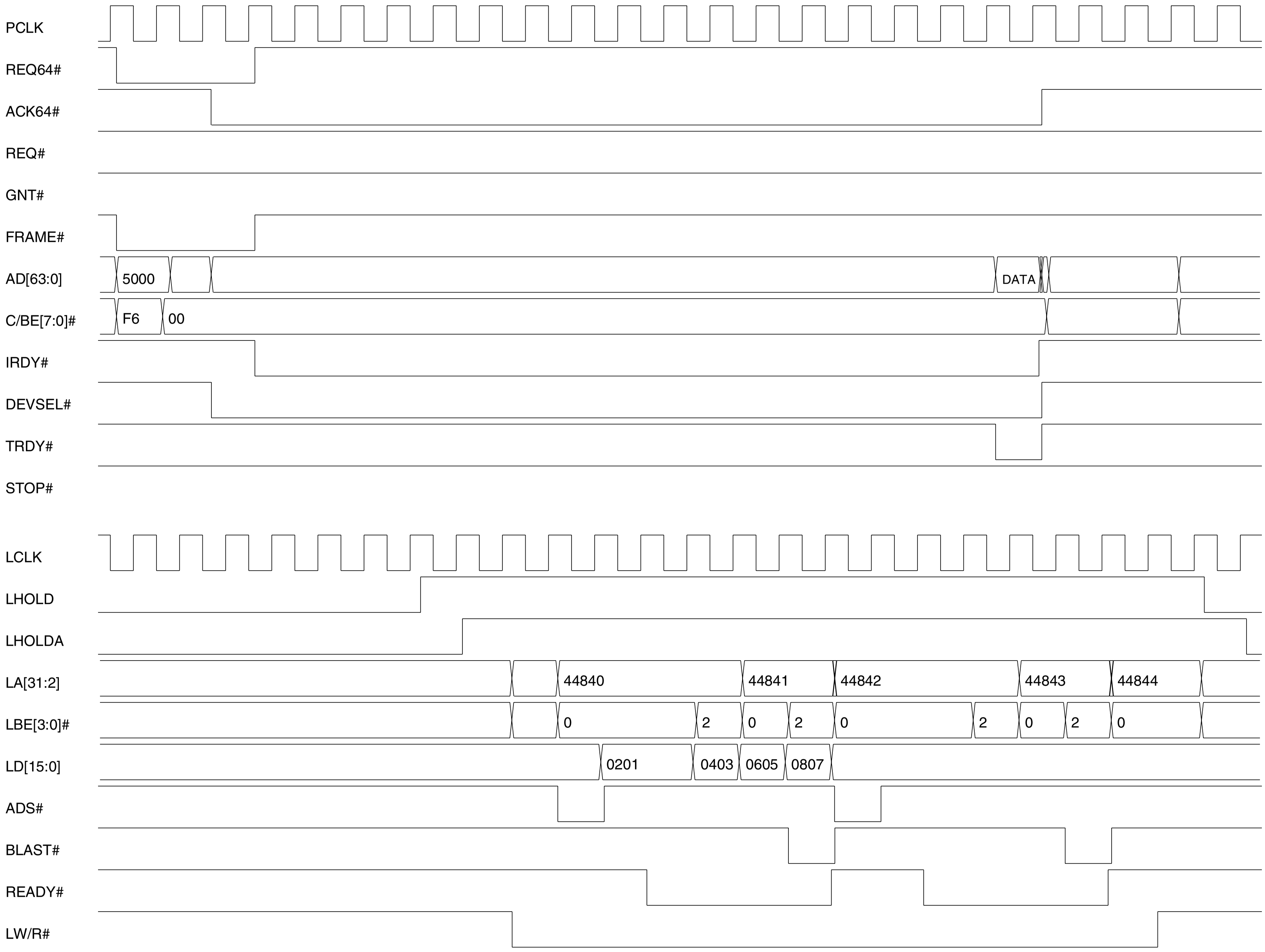
Notes: *Hex value for AD[63:0] DATA = 0000_0005_0403_0201h.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=1542h.

Timing Diagram 5-21. Direct Slave 64-Bit Single Cycle Read (8-Bit Local Bus)



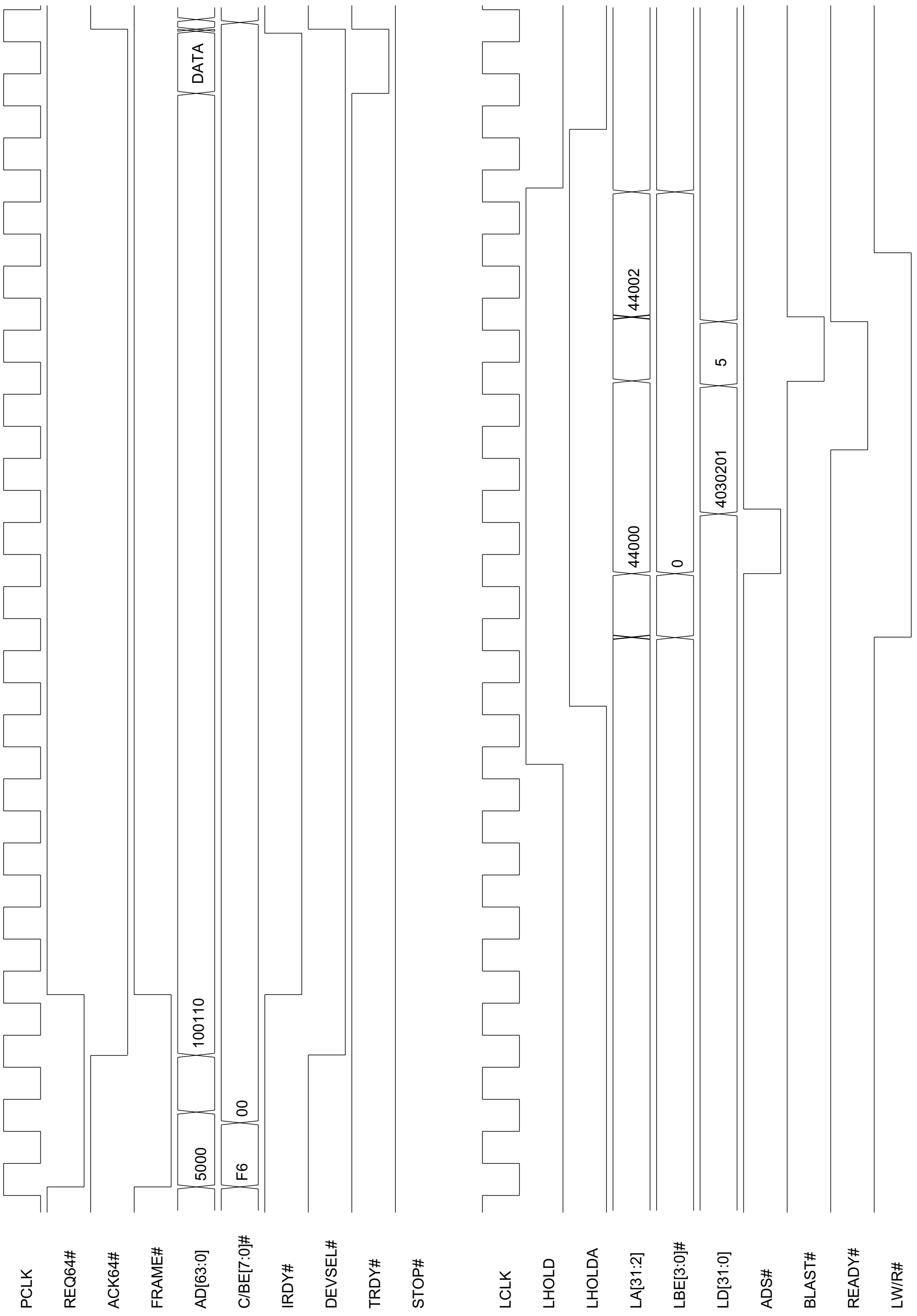
Notes: *Hex value for AD[63:0] DATA = 0807_0605_0403_0201h.*
Key bit/register values are MARBR[24]=1 and LBRD1[15:0]=1540h.

Timing Diagram 5-22. Direct Slave 64-Bit Single Cycle Read (16-Bit Local Bus)



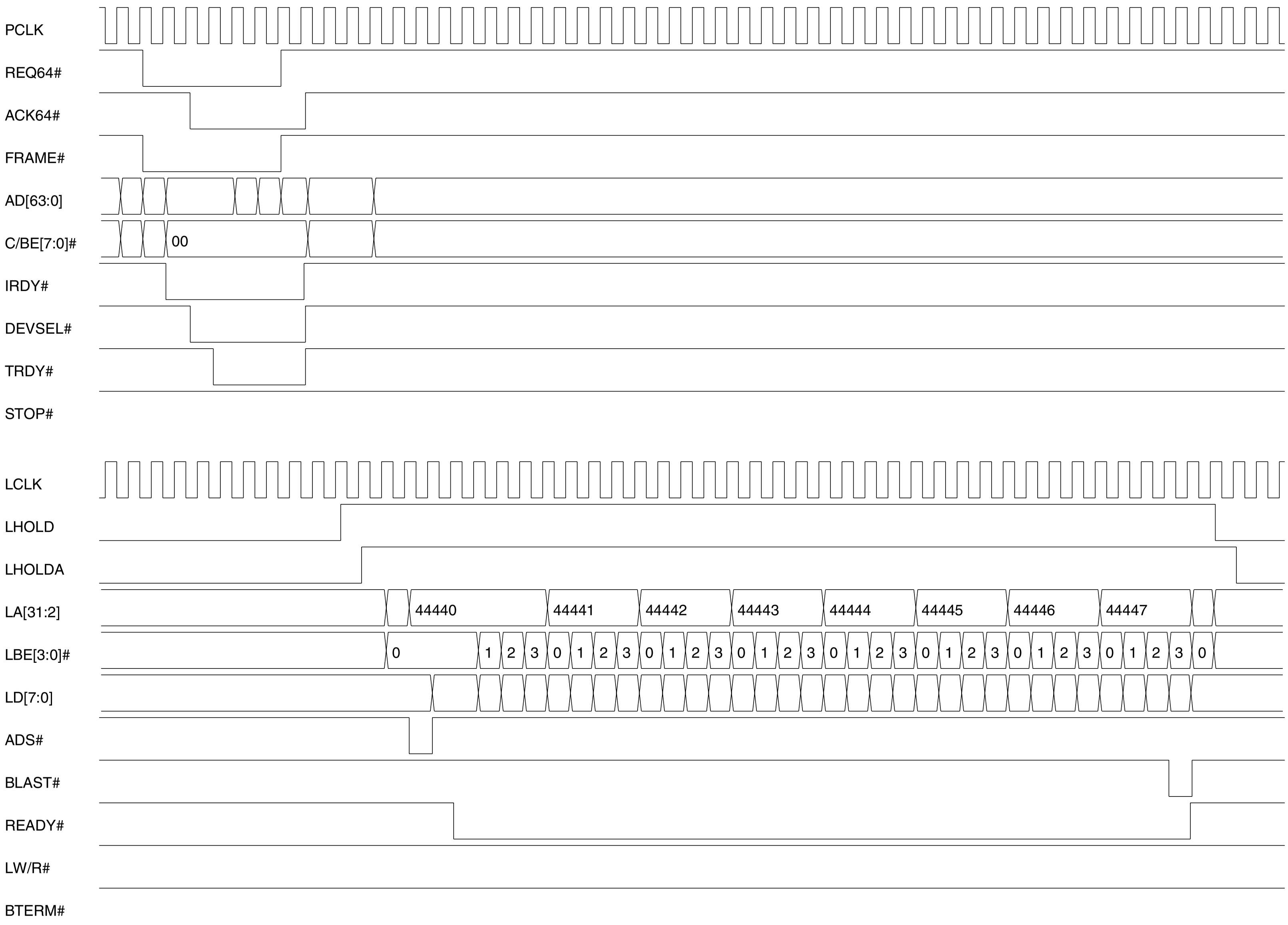
Notes: *Hex value for AD[63:0] DATA = 0807_0605_0403_0201h. Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=2541h.*
Because prefetch is enabled LBRD1[9]=0 and LBRD1[14:11]=0100b, an additional 2 Lwords of data are prefetched.
The LD[15:0] value for the additional 2 Lwords is not shown.

Timing Diagram 5-23. Direct Slave 64-Bit Single Cycle Read (32-Bit Local Bus)



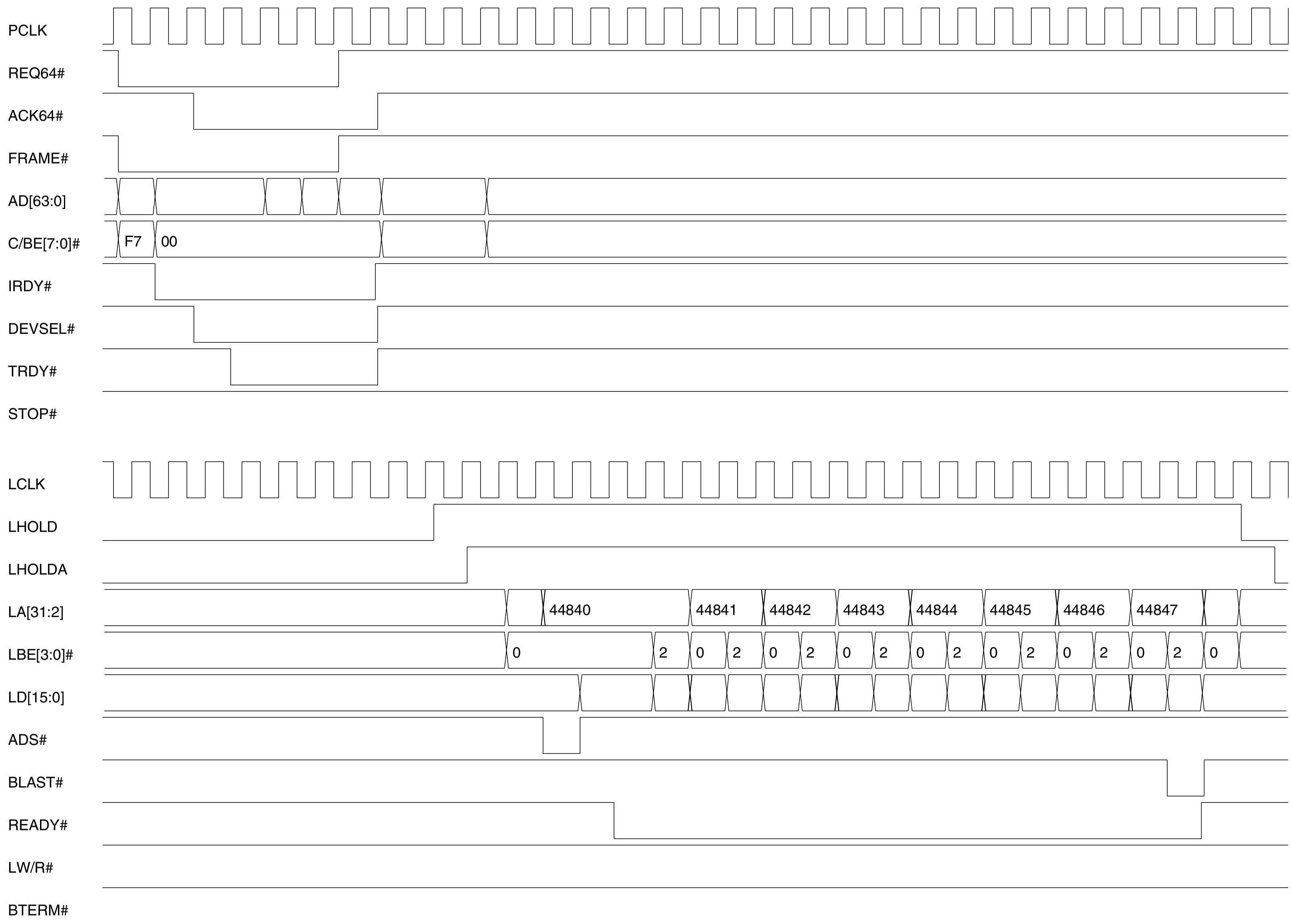
Notes: *Hex value for AD[63:0] DATA = 0000_0005_0403_0201h.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=1542h.

Timing Diagram 5-24. Direct Slave 64-Bit Burst Write of 4 Qwords (8-Bit Local Bus)



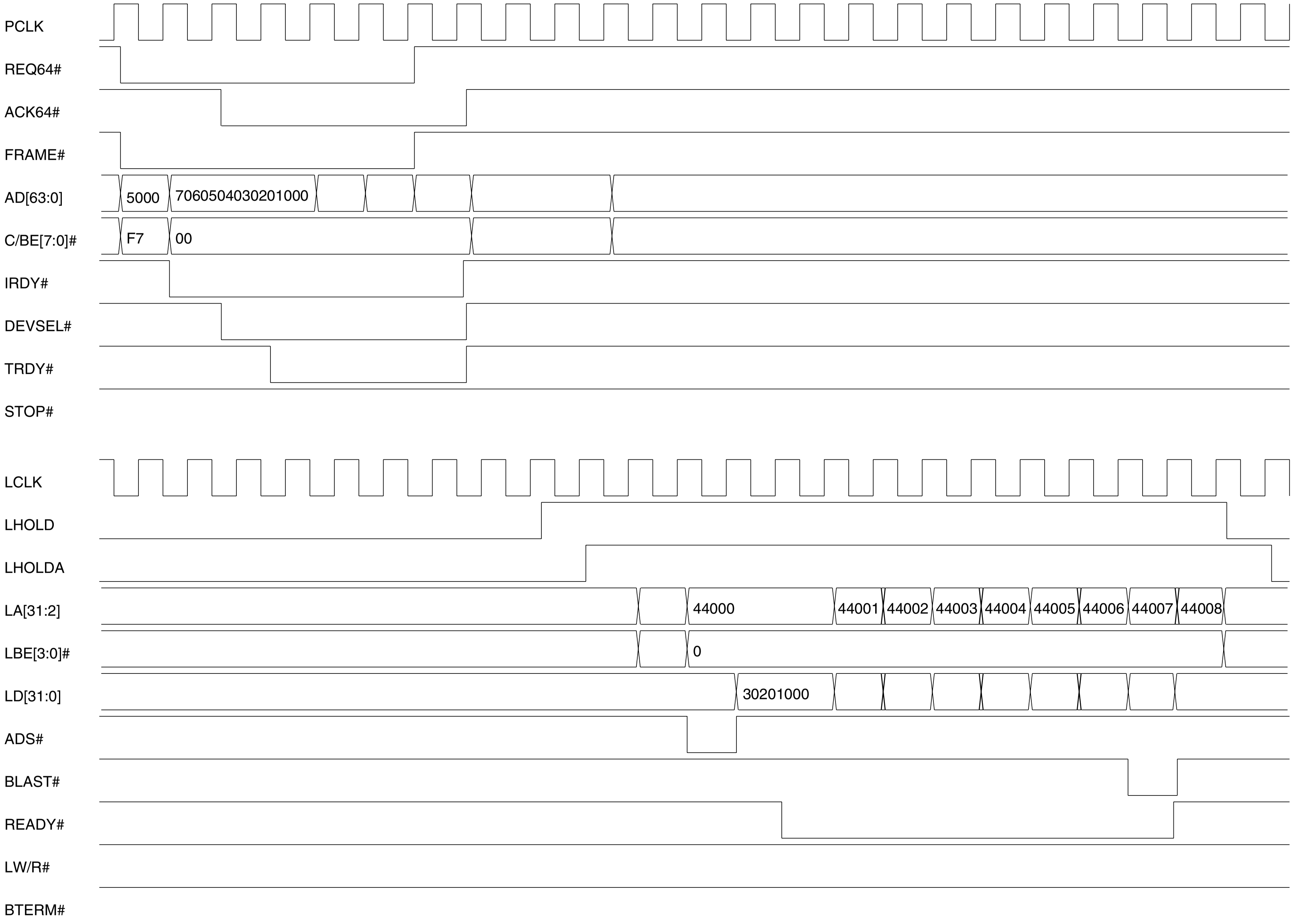
Note: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C0h.*

Timing Diagram 5-25. Direct Slave 64-Bit Burst Write of 4 Qwords (16-Bit Local Bus)



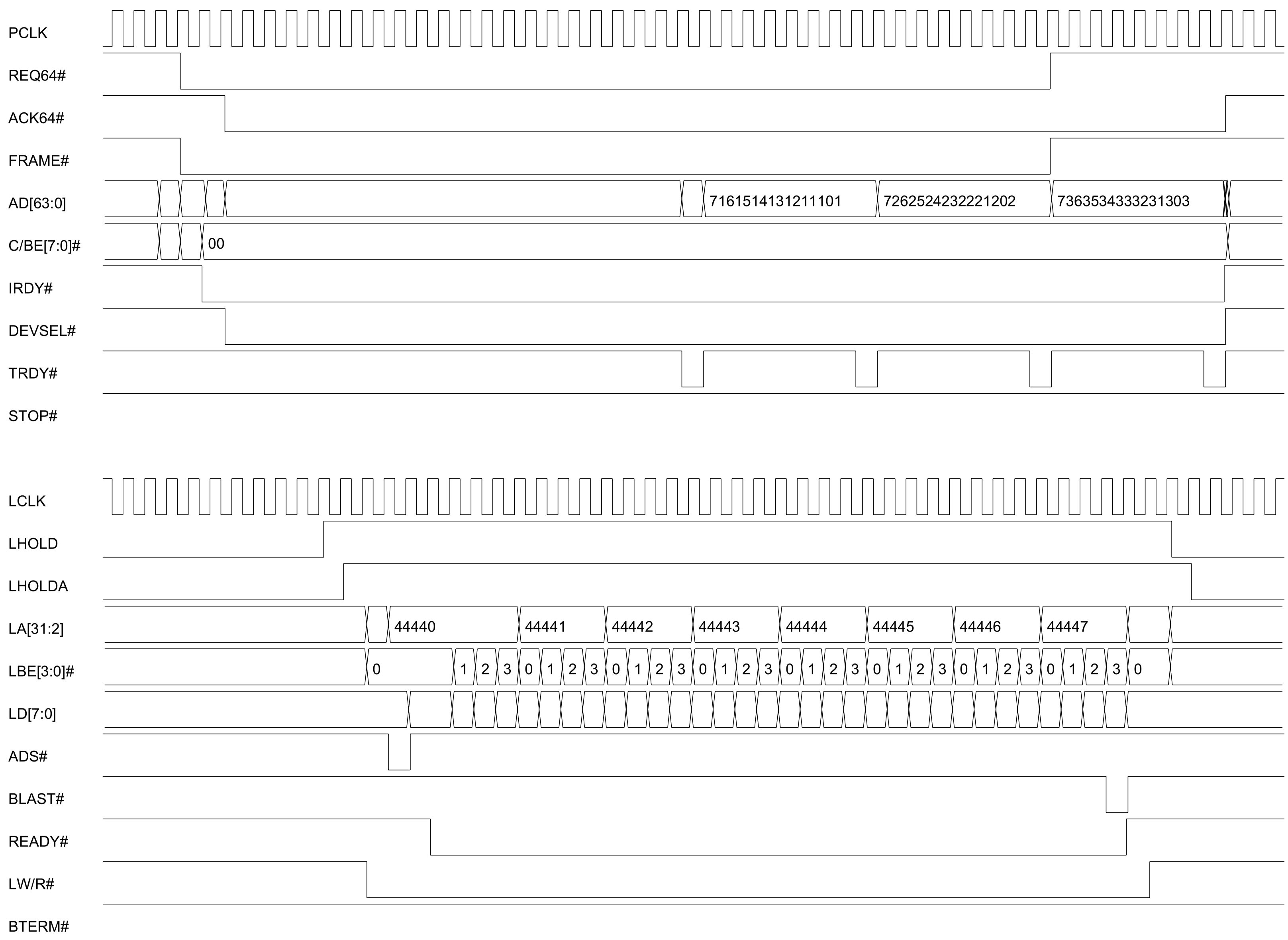
Note: *Key bit/register values are MARBR[24]=1 and LBRD1[15:0]=45C1h.*

Timing Diagram 5-26. Direct Slave 64-Bit Burst Write of 4 Qwords (32-Bit Local Bus)



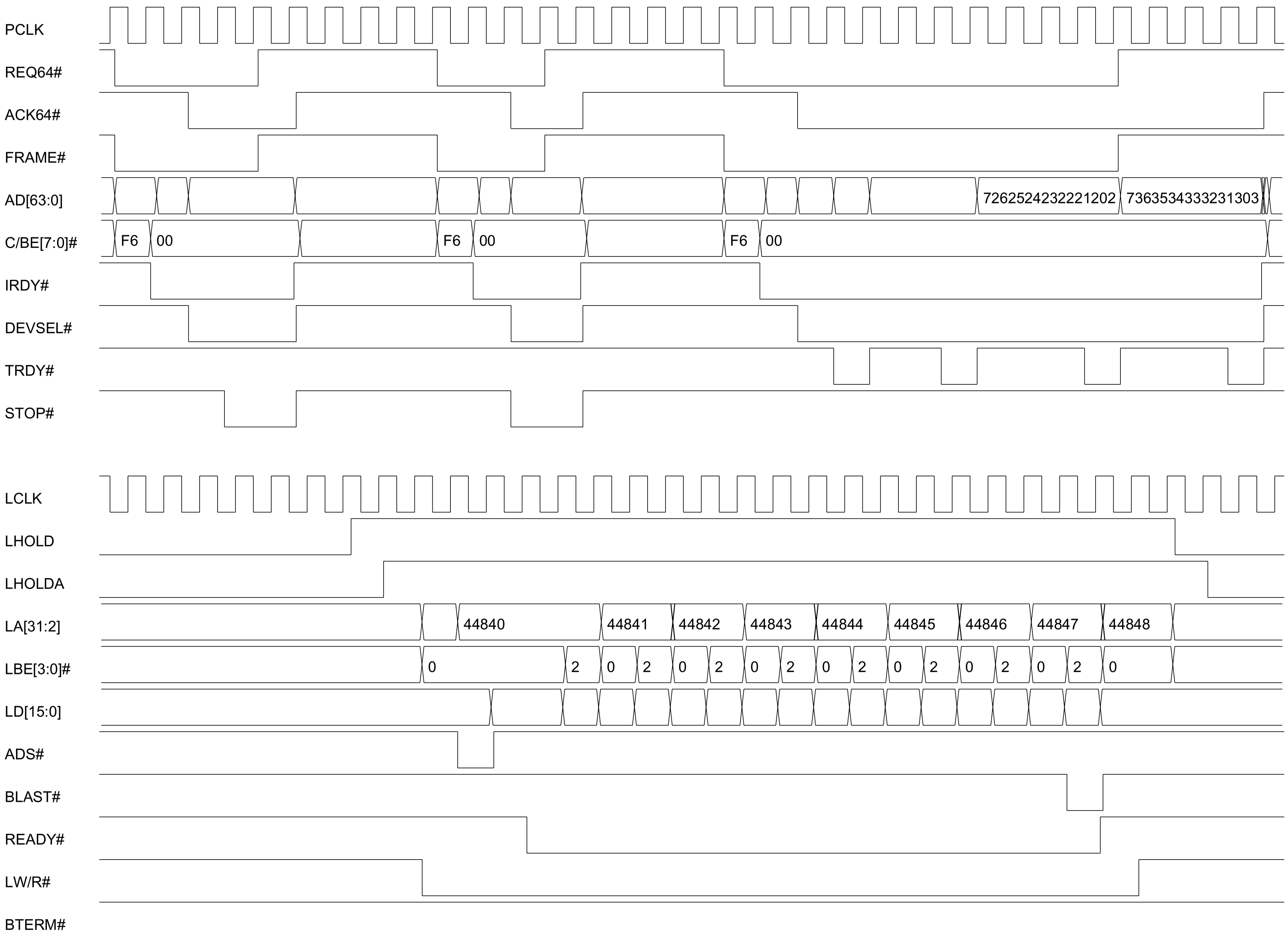
Note: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.*

Timing Diagram 5-27. Direct Slave 64-Bit Burst Read of 4 Qwords (8-Bit Local Bus)



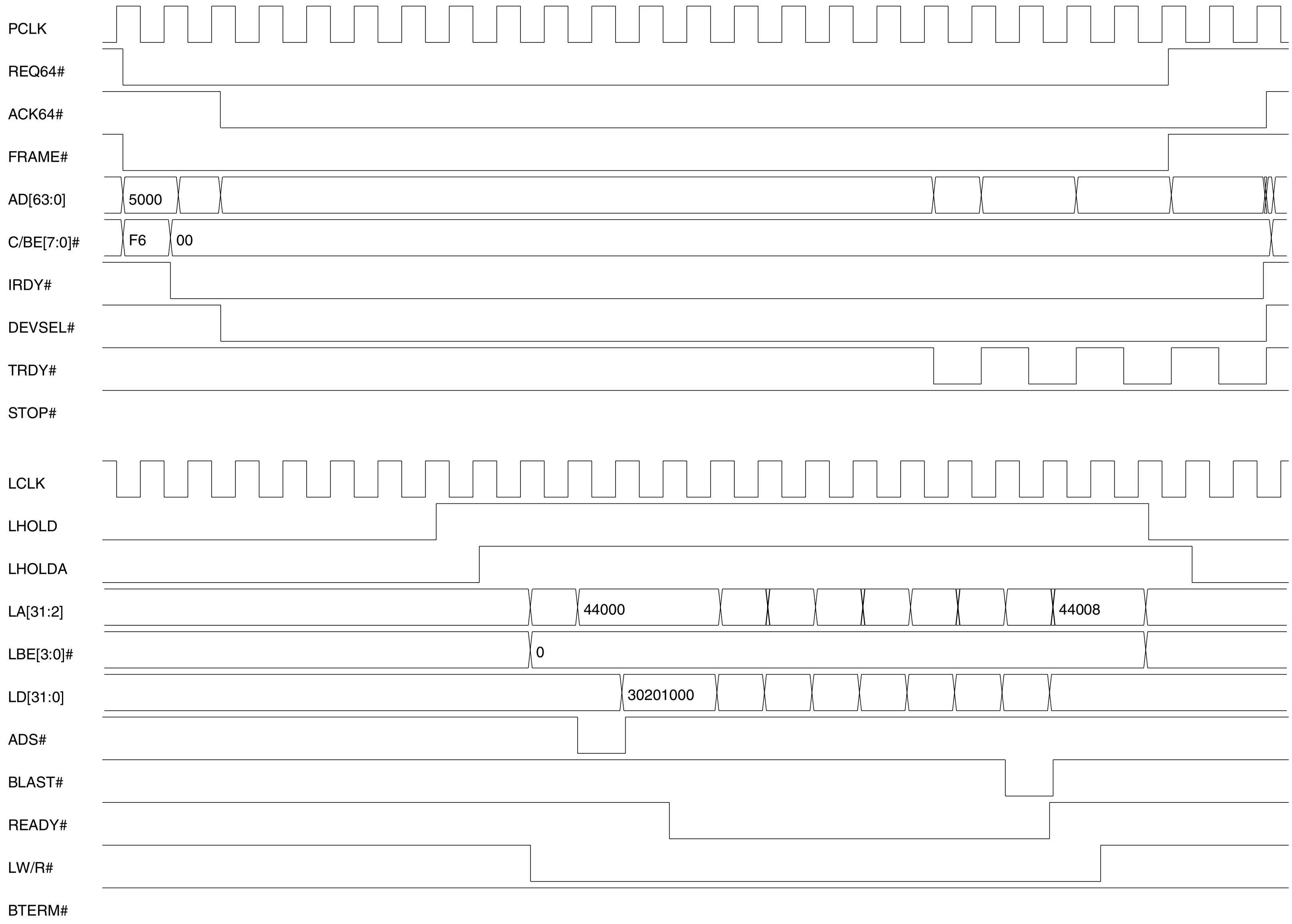
Note: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C0h.*

Timing Diagram 5-28. Direct Slave 64-Bit Burst Read of 4 Qwords (16-Bit Local Bus)



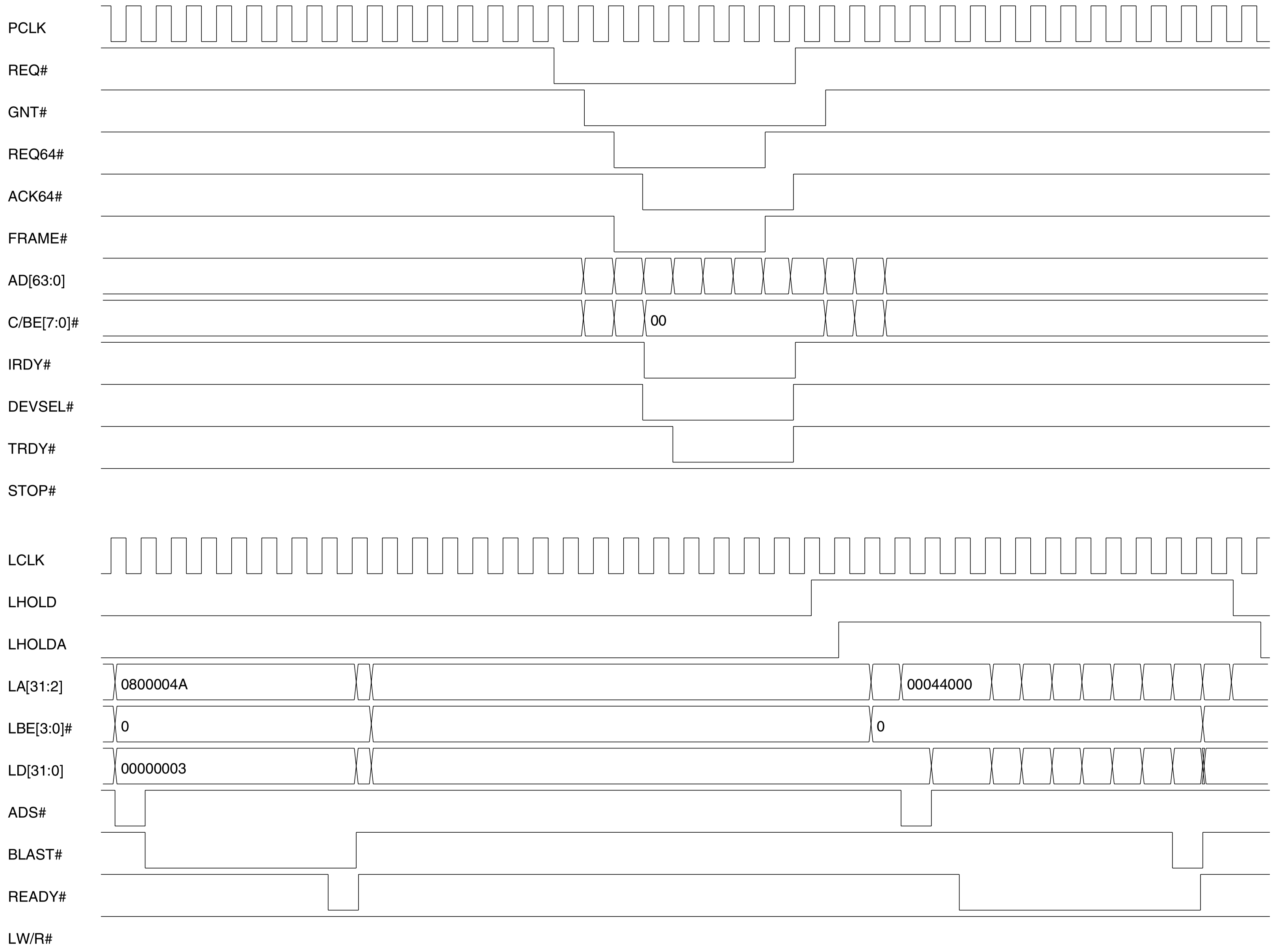
Note: *Key bit/register values are MARBR[24]=1 and LBRD1[15:0]=45C1h.*

Timing Diagram 5-29. Direct Slave 64-Bit Burst Read of 4 Qwords (32-Bit Local Bus)



Note: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.*

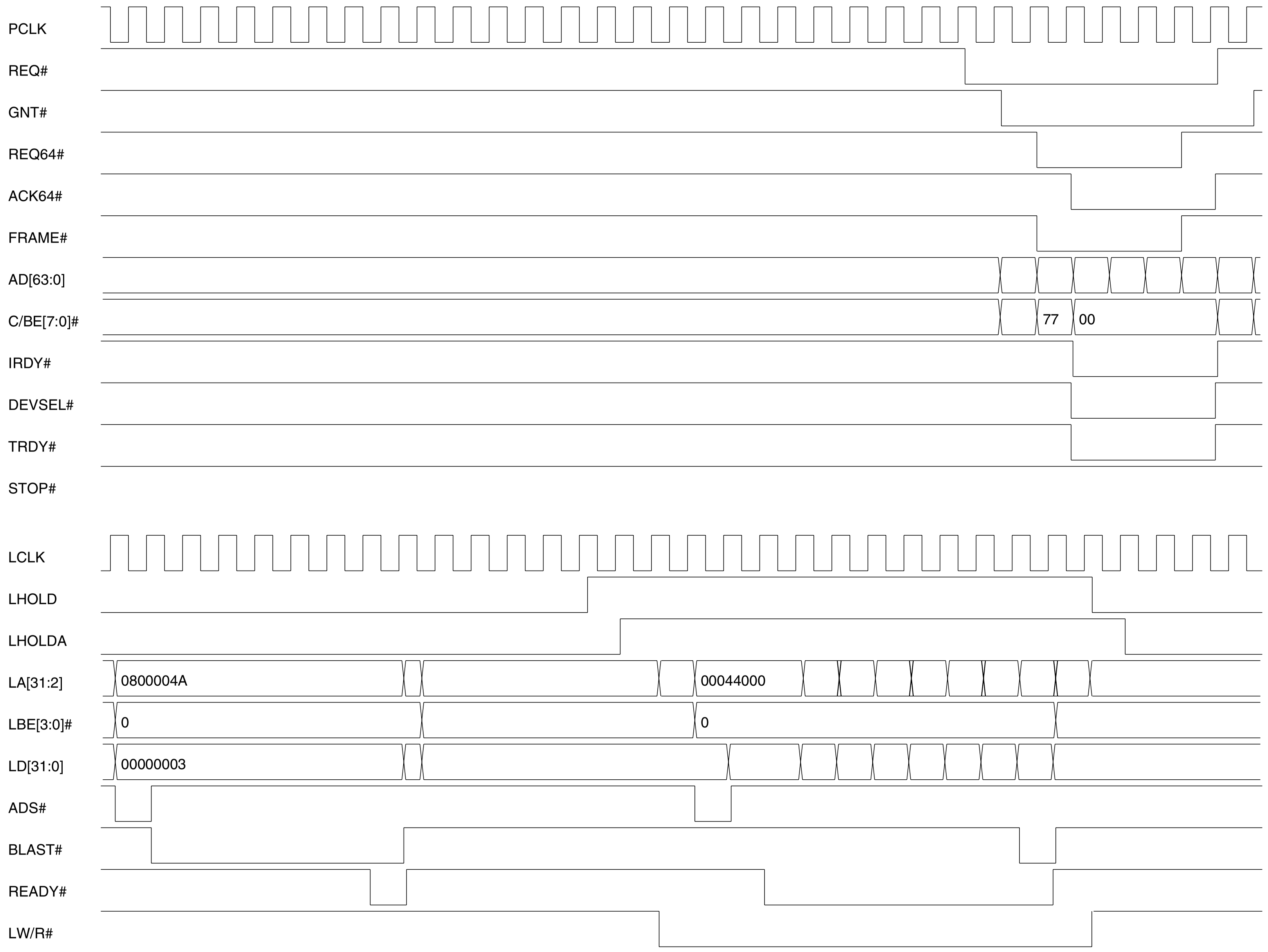
Timing Diagram 5-30. DMA PCI-to-Local (32-Bit Local Bus, 64-Bit PCI Bus)



Notes: *The writing of the registers to set up this 32-byte DMA Block mode transfer using DMA Channel 0 is not shown. Writing the DMACSR0 register (LOC:128h) with 03h enables and starts this DMA transfer.*

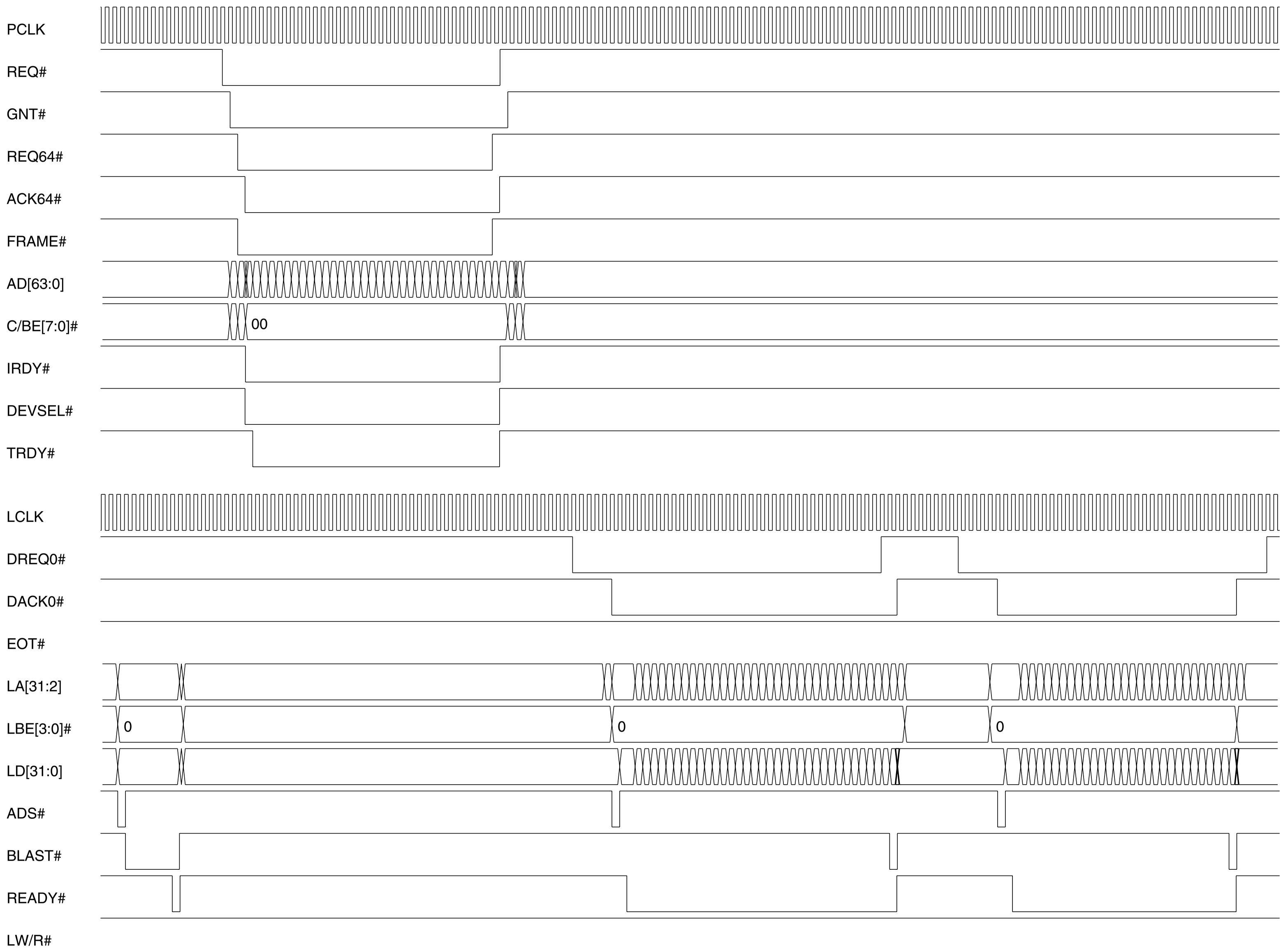
5.6.3 C Mode DMA Timing Diagrams

Timing Diagram 5-31. DMA Local-to-PCI (32-Bit Local Bus, 64-Bit PCI Bus)



Notes: *The writing of the registers to set up this 32-byte DMA Block mode transfer using DMA Channel 0 is not shown. Writing the DMACSR0 register (LOC:128h) with 03h enables and starts this DMA transfer.*

Timing Diagram 5-32. DMA PCI-to-Local Demand Mode (32-Bit Local Bus, 64-Bit PCI Bus)



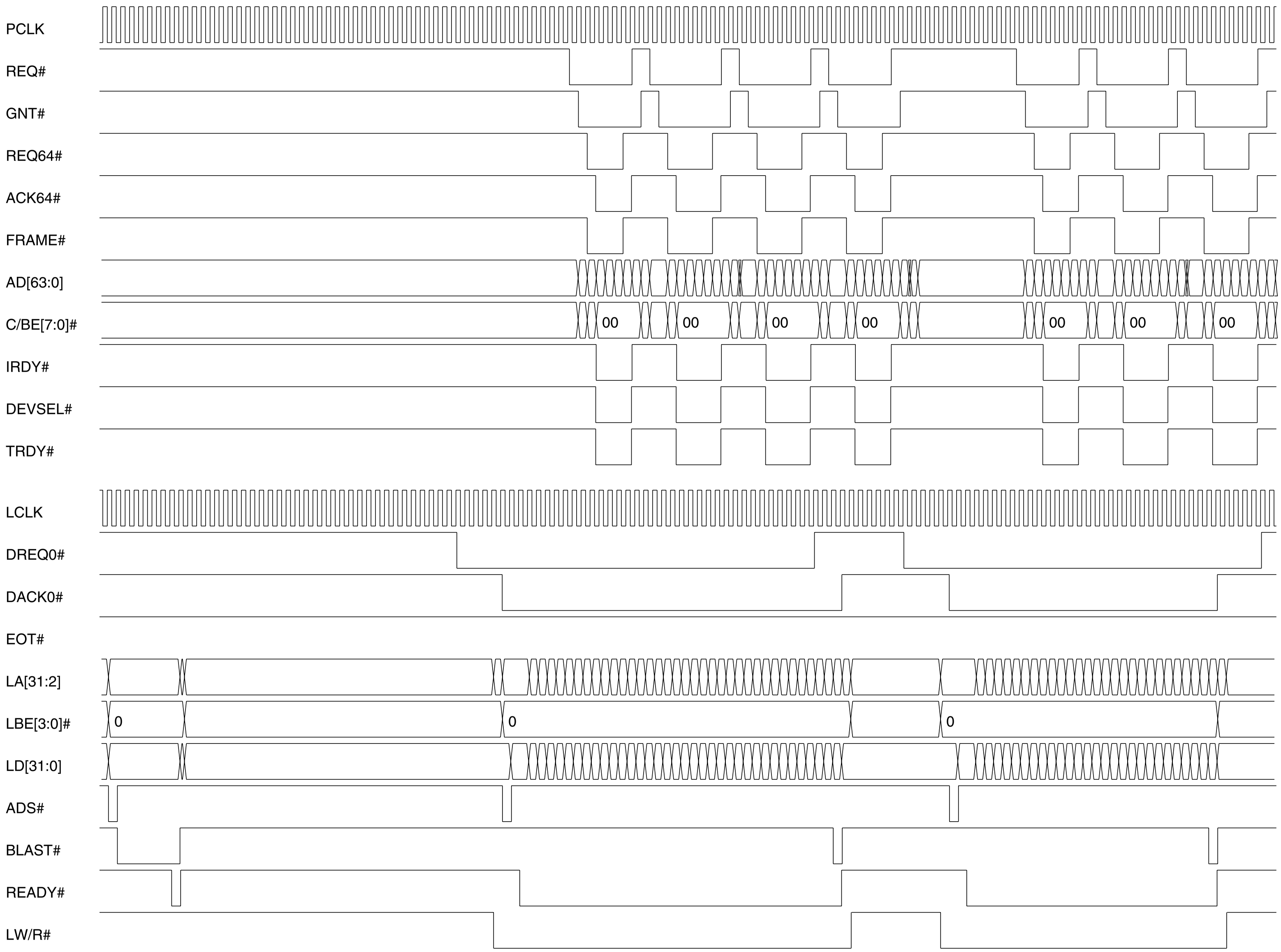
Notes: *The Channel 0 Demand mode DMA transfer starts when the DMACSR0[1:0] bits are written to 11b.*

The PCI 9656 does not attempt to arbitrate for and write data to the Local Bus until it detects the DREQ0# signal asserted.

The transfer is temporarily suspended when DREQ0# is de-asserted, then resumed upon its re-assertion.

The PCI STOP# signal (not shown) is not asserted during this transfer.

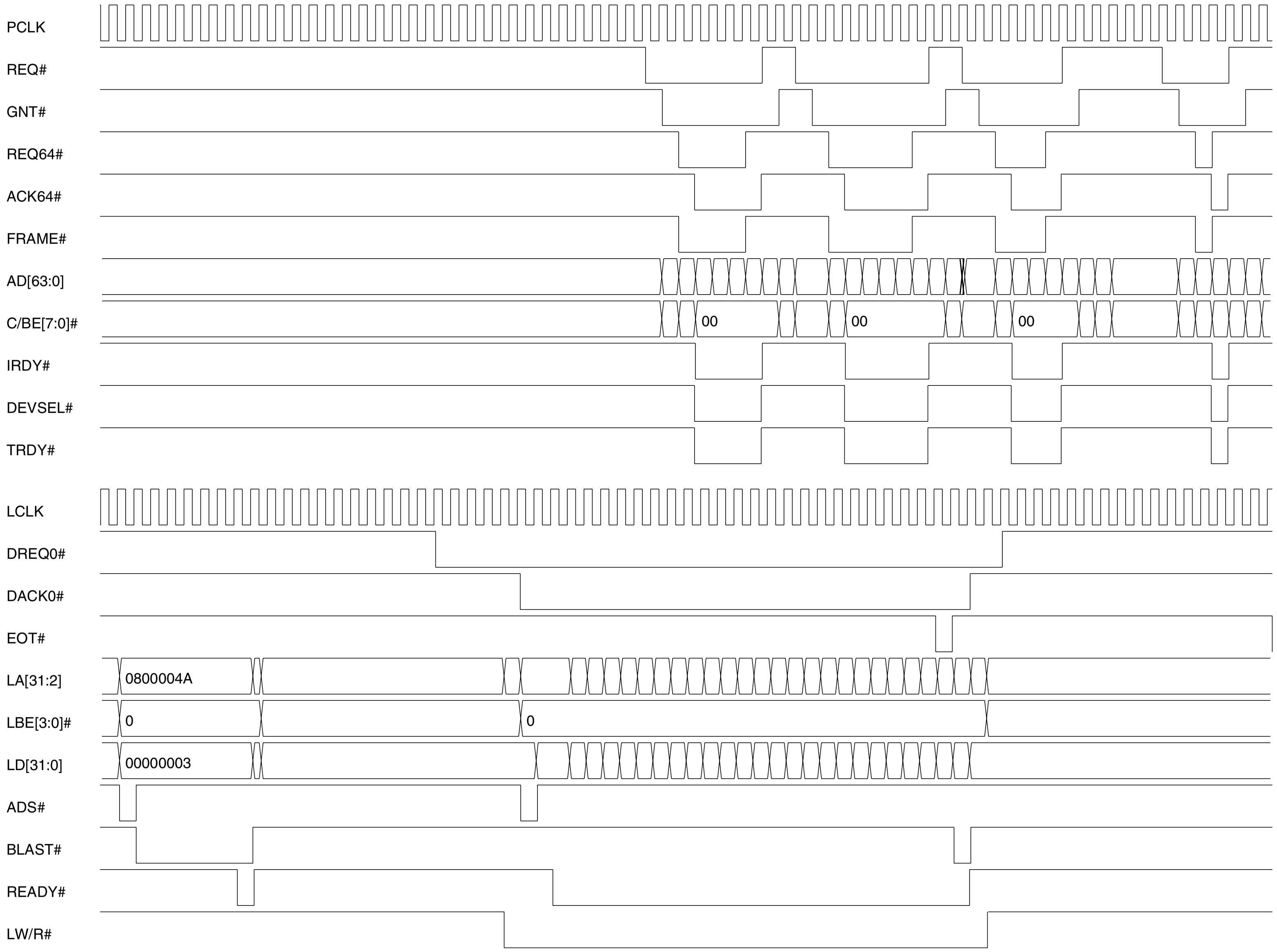
Timing Diagram 5-33. DMA Local-to-PCI Demand Mode (32-Bit Local Bus, 64-Bit PCI Bus)



Notes: *The Channel 0 Demand mode DMA transfer starts when the DMACSR0[1:0] bits are written to 11b and the PCI 9656 detects the DREQ0# signal asserted. Once started, the PCI 9656 arbitrates for and reads data from the Local Bus, then writes the data to the PCI Bus until the transfer is temporarily suspended when DREQ0# is de-asserted. The transfer resumes upon DREQ0# re-assertion.*

The PCI STOP# signal (not shown) is not asserted during this transfer.

Timing Diagram 5-34. DMA Local-to-PCI Demand Mode with EOT# Assertion (32-Bit Local Bus, 64-Bit PCI Bus)

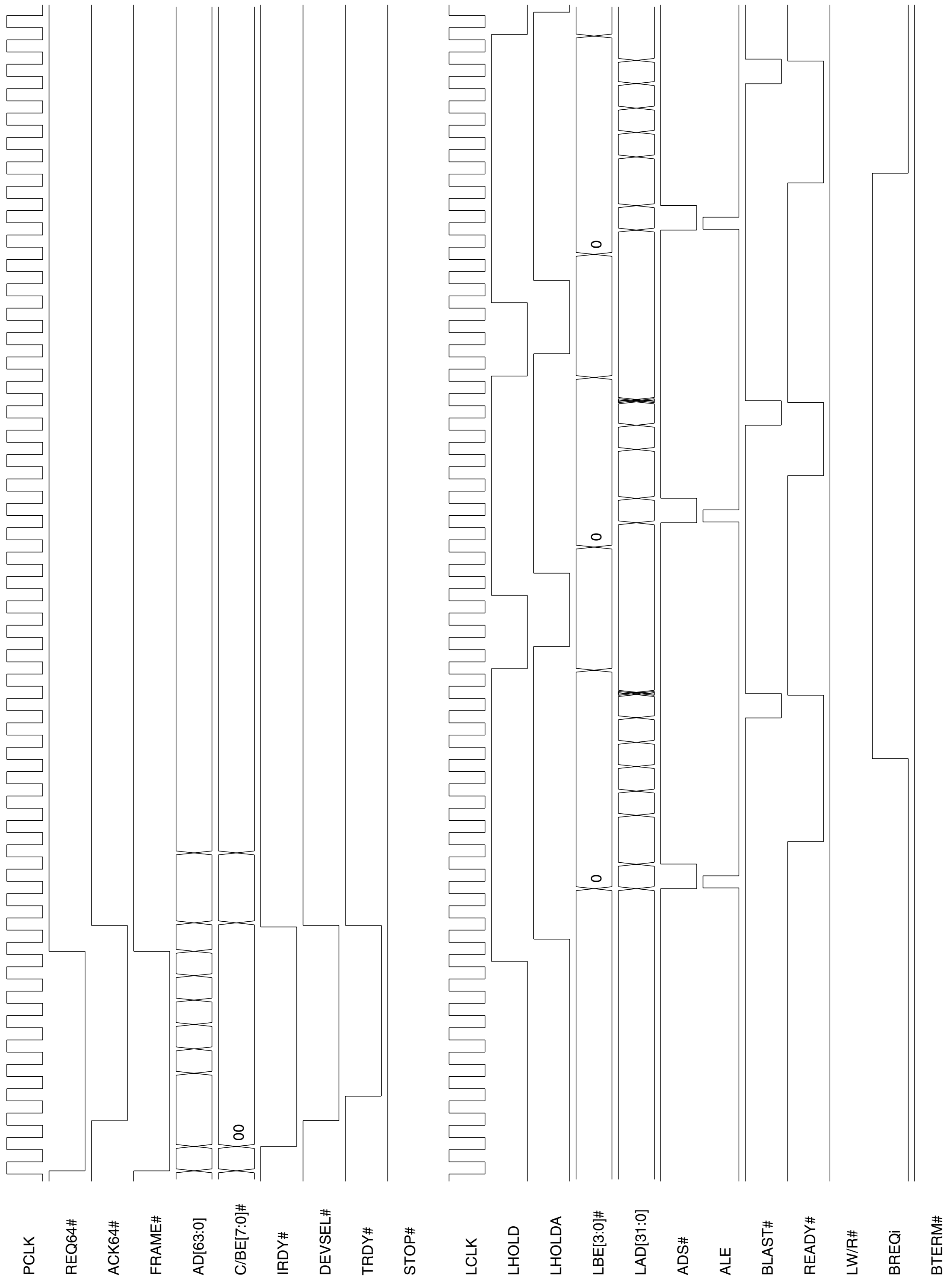


Notes: *The Channel 0 Demand mode DMA transfer starts when the DMACSR0[1:0] bits are written to 11b and the PCI 9656 detects the DREQ0# signal asserted. Once started, the PCI 9656 arbitrates for and reads data from the Local Bus, then writes the data to the PCI Bus until the transfer is terminated with an EOT# assertion. All data that is read into the DMA FIFO is written to the PCI Bus.*

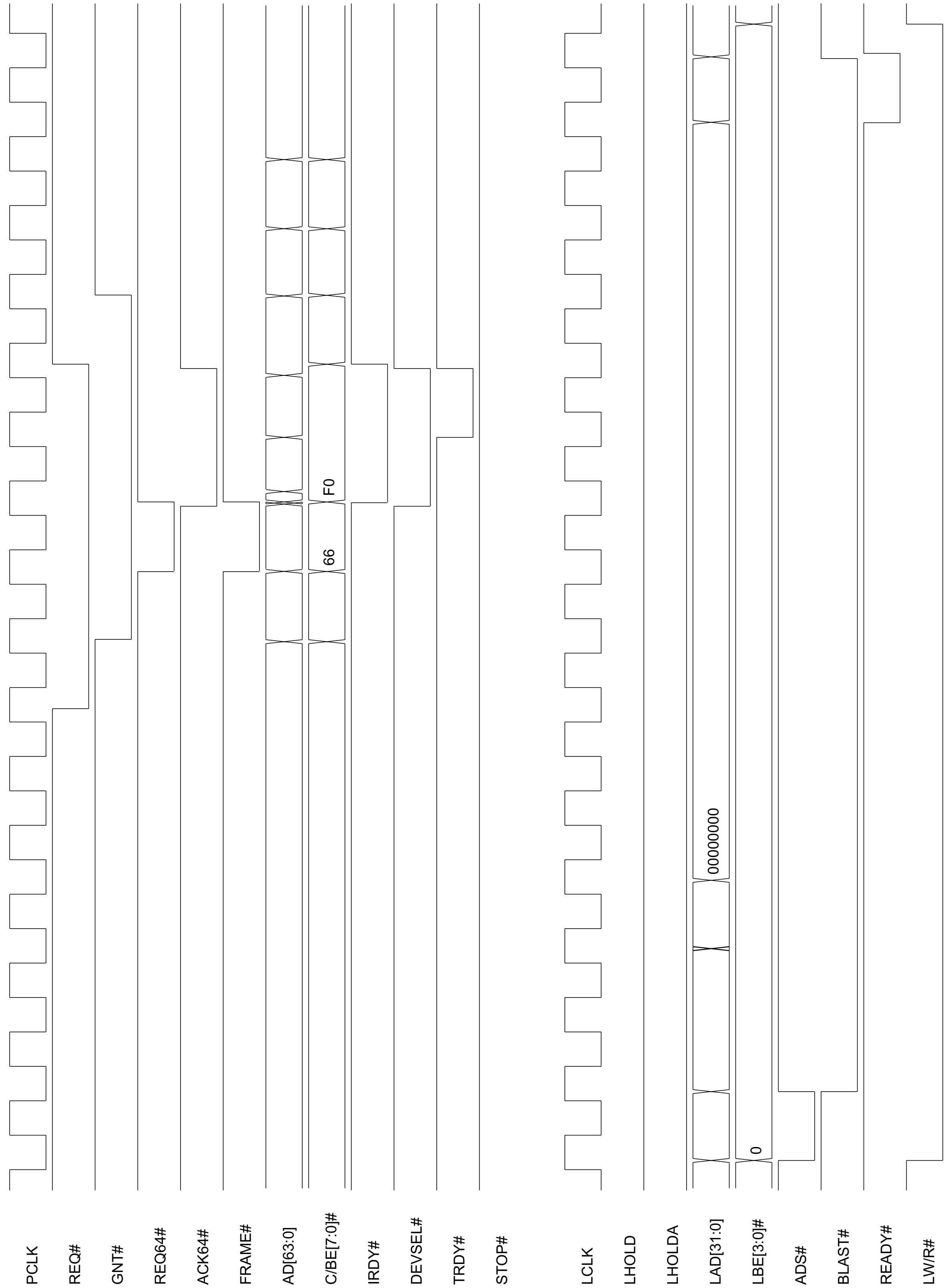
5.7 J MODE FUNCTIONAL TIMING DIAGRAMS

5.7.1 J Mode Direct Master Timing Diagrams

Timing Diagram 5-35. Direct Master Single Cycle Write (64-Bit PCI Bus)

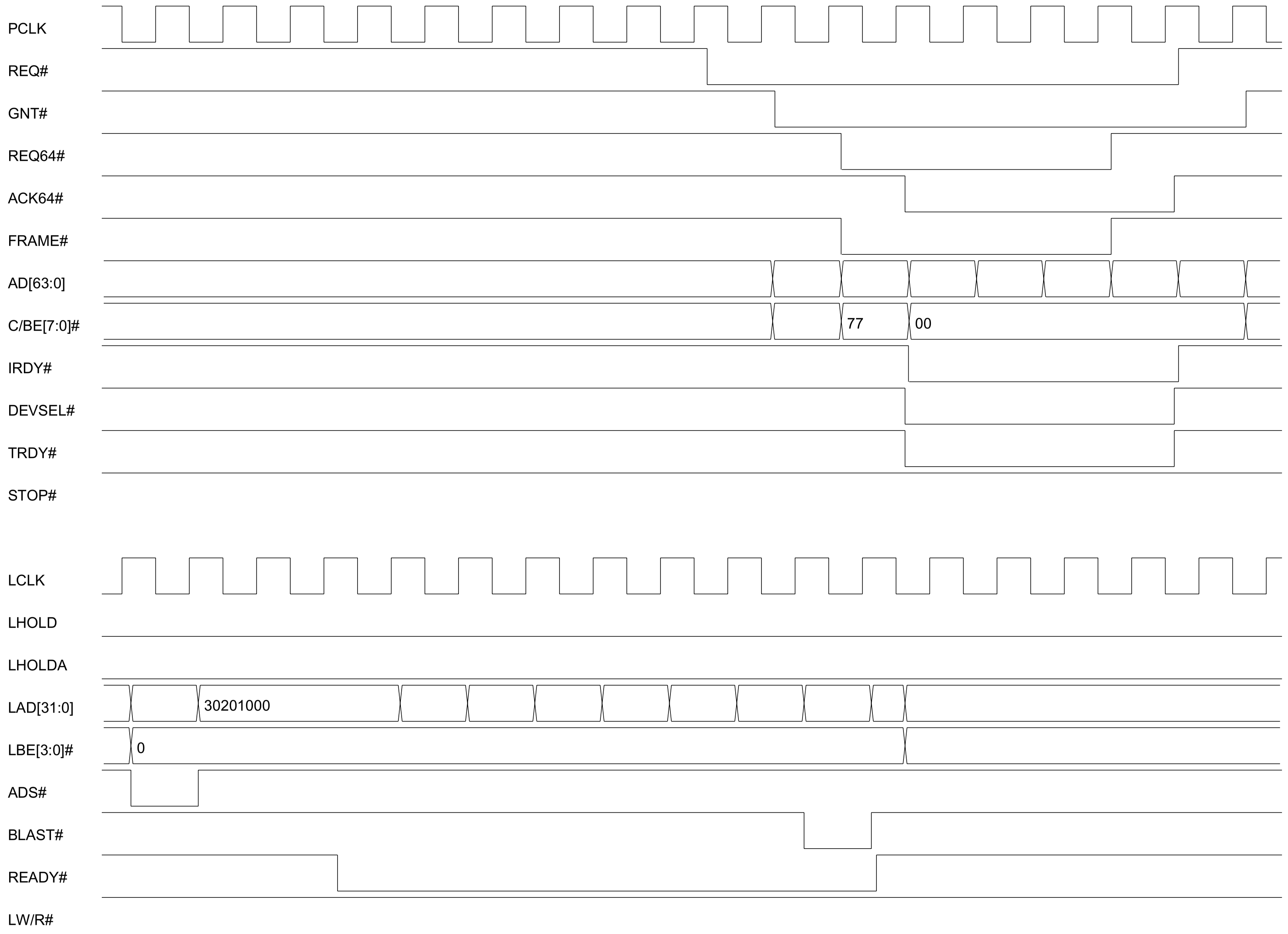


Timing Diagram 5-36. Direct Master Single Cycle Read (64-Bit PCI Bus)



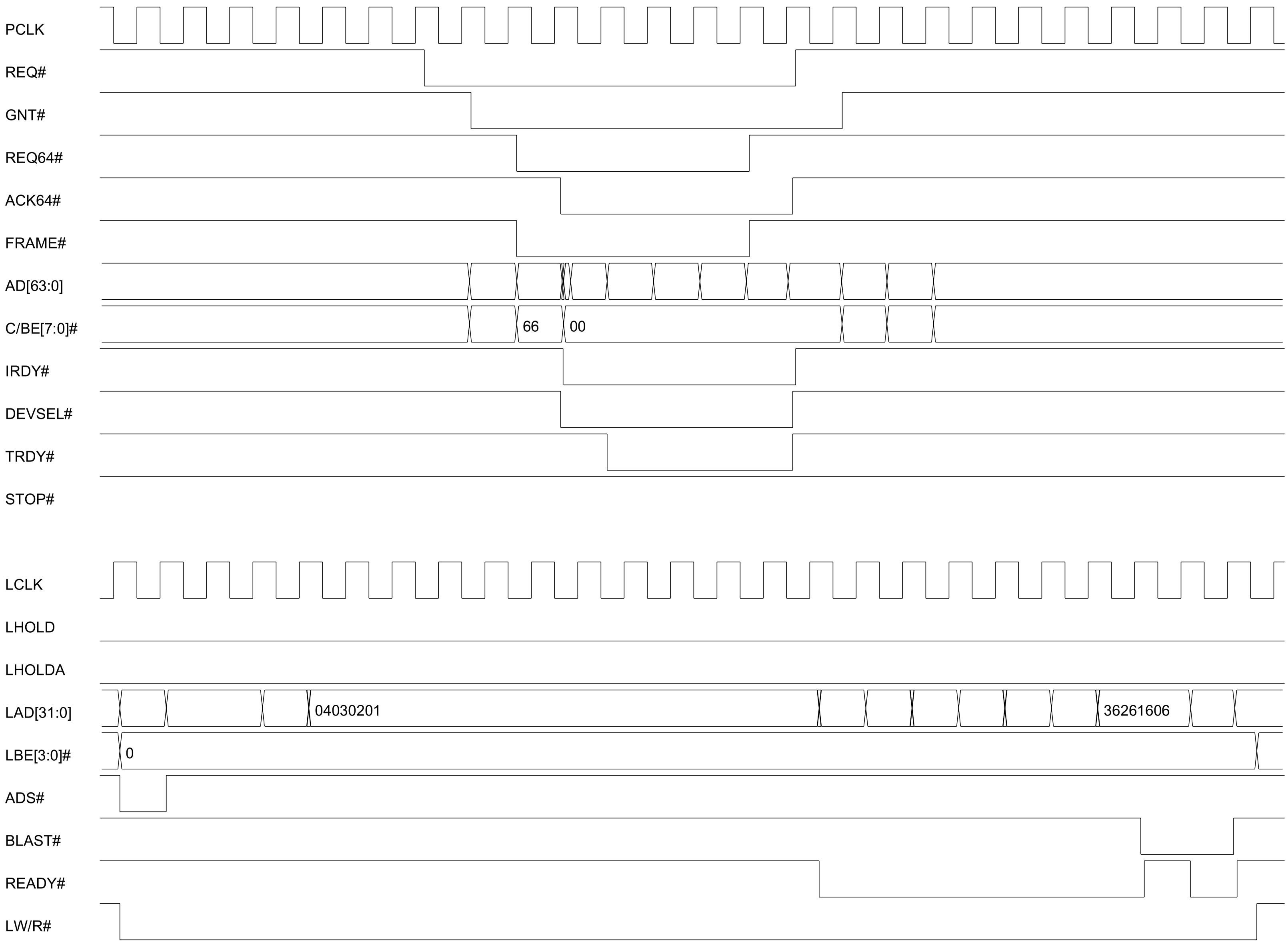
Note: *Key register value is DMPBAM[15:0]=00E3h.*

Timing Diagram 5-37. Direct Master Burst Write of 8 Lwords (64-Bit PCI Bus)



Note: *Key register value is DMPBAM[15:0]=000Bh.*

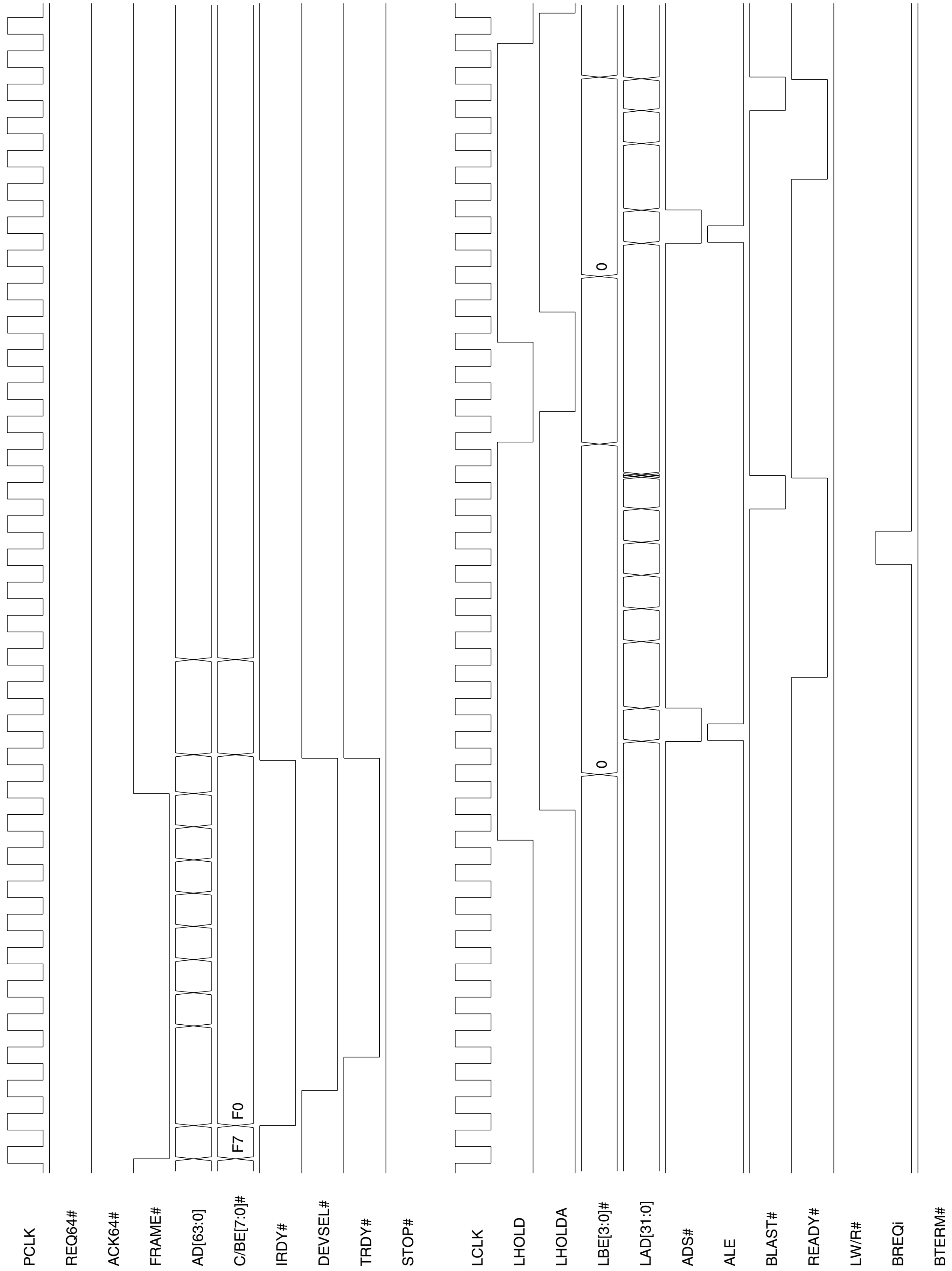
Timing Diagram 5-38. Direct Master Burst Read of 8 Lwords (64-Bit PCI Bus)



Note: *Key register value is DMPBAM[15:0]=000Bh.*

5.7.2 J Mode Direct Slave Timing Diagrams

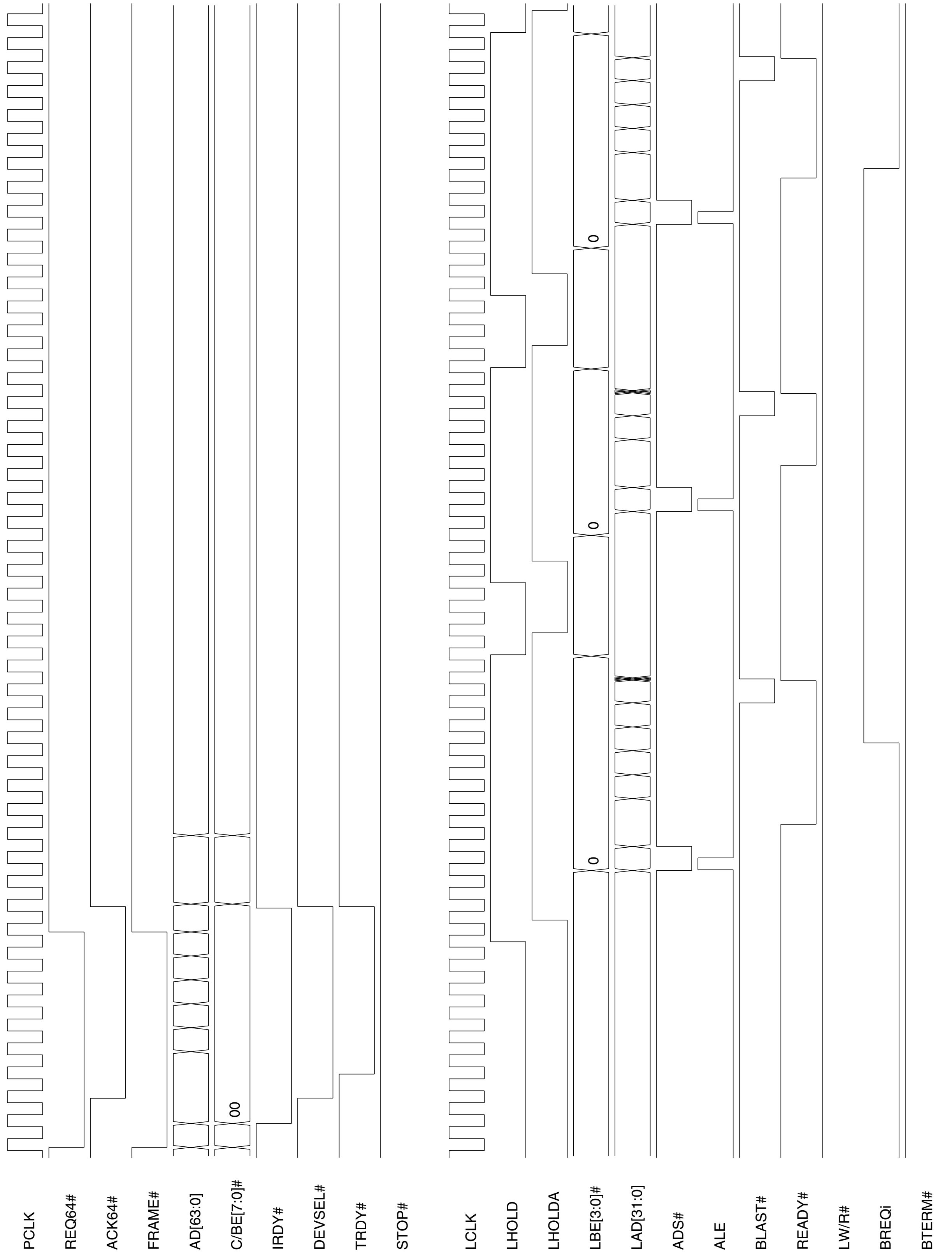
Timing Diagram 5-39. Direct Slave Burst Write Suspended by BREQi (32-Bit PCI Bus)



Notes: *This 9-Lword Direct Slave Burst Write transfer is suspended by a one-clock cycle BREQi assertion. The remaining three Lwords are transferred once the PCI 9656 is granted the Local Bus again.*

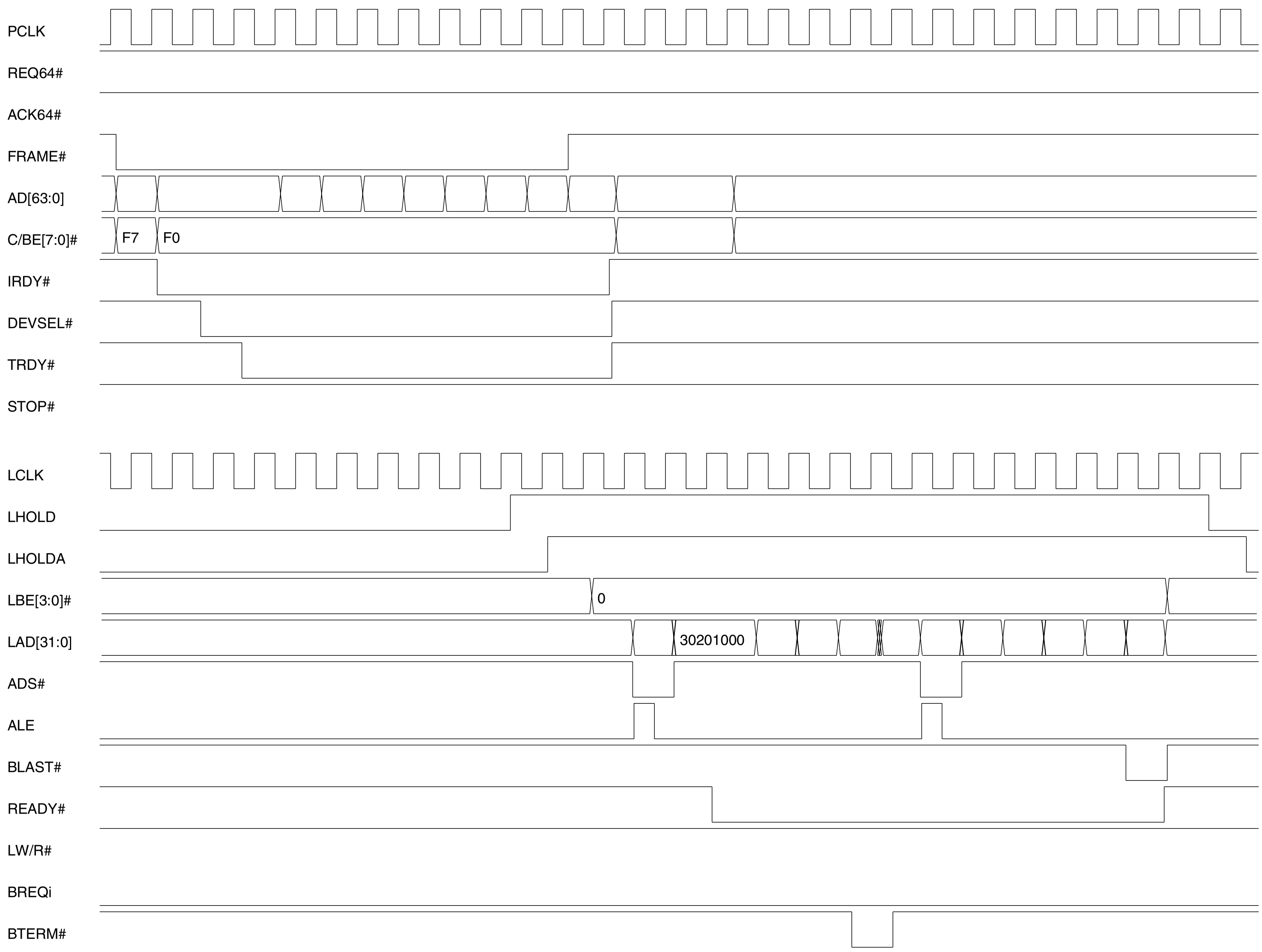
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.

Timing Diagram 5-40. Direct Slave Burst Write Suspended by BREQi (64-Bit PCI Bus)



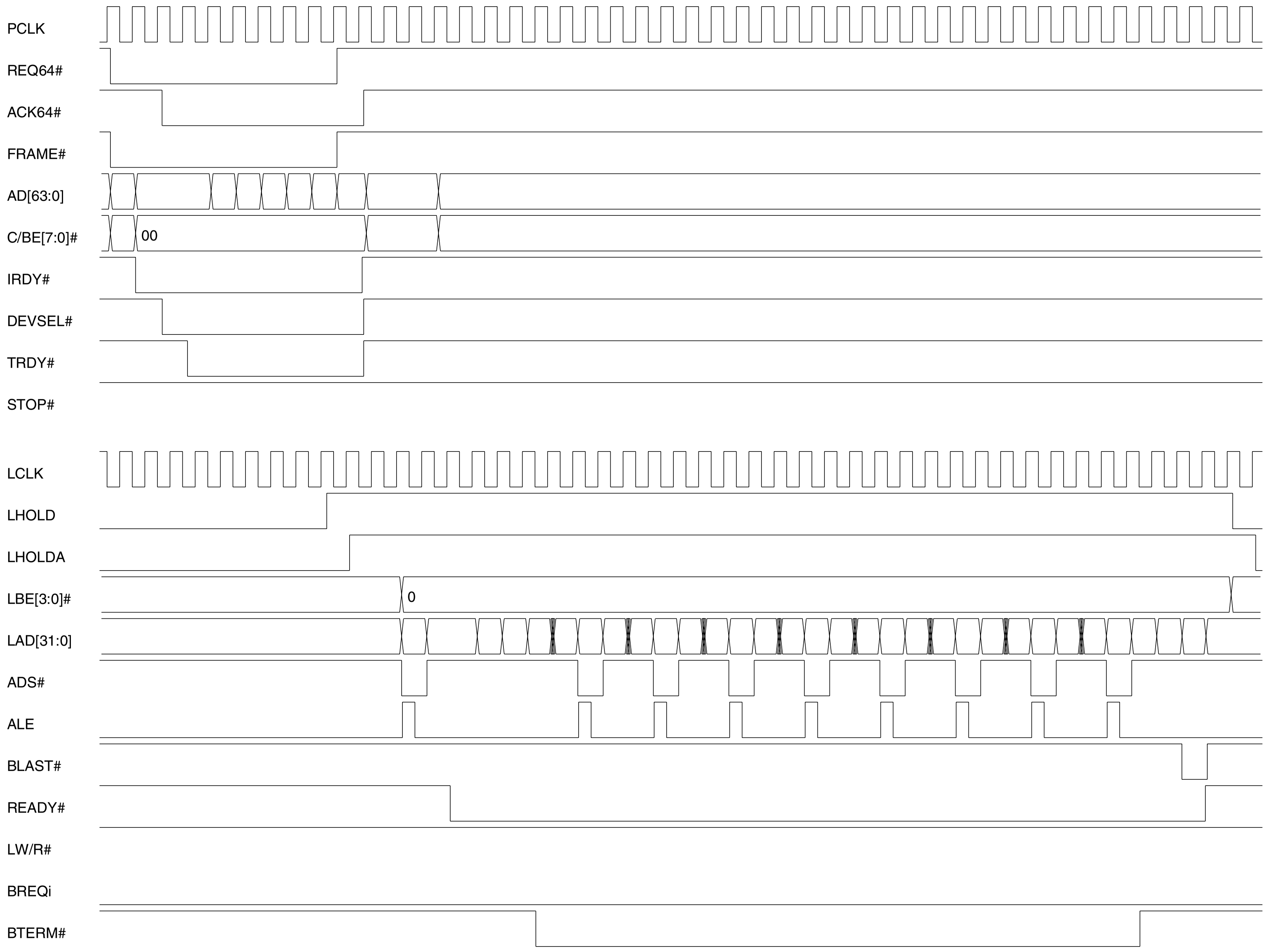
Notes: *This 14-Lword Direct Slave Burst Write transfer is suspended twice by a multi-clock cycle BREQi assertion.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.

Timing Diagram 5-41. Direct Slave Burst Write Interrupted by BTERM# (32-Bit PCI Bus)



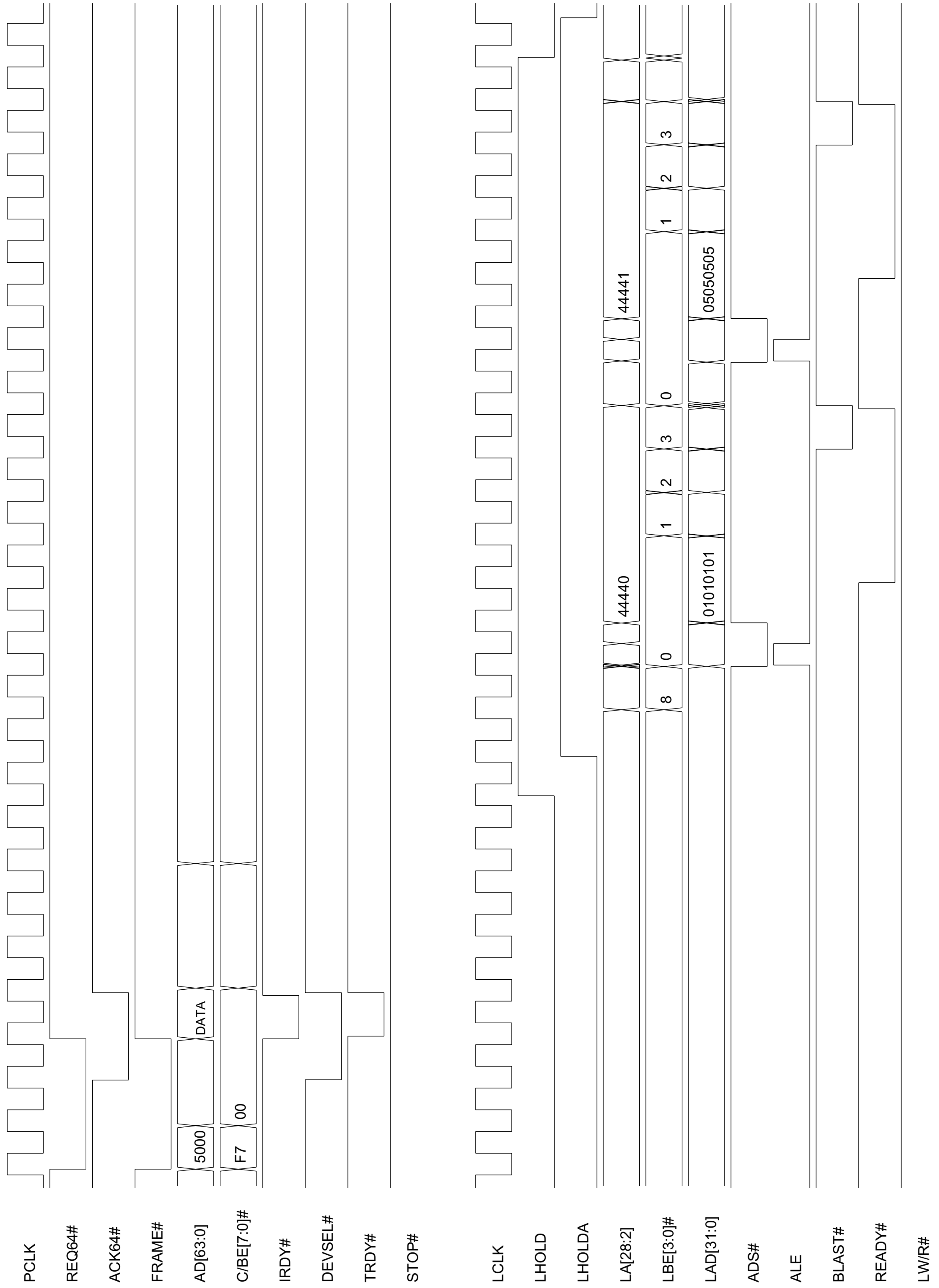
Notes: *This 9-Lword Direct Slave Burst Write transfer is interrupted by a one clock cycle BTERM# assertion. The remaining five Lwords are transferred after the PCI 9656 generates a new Address phase on the Local Bus. Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.*

Timing Diagram 5-42. Direct Slave Burst Write Interrupted by BTERM# (64-Bit PCI Bus)



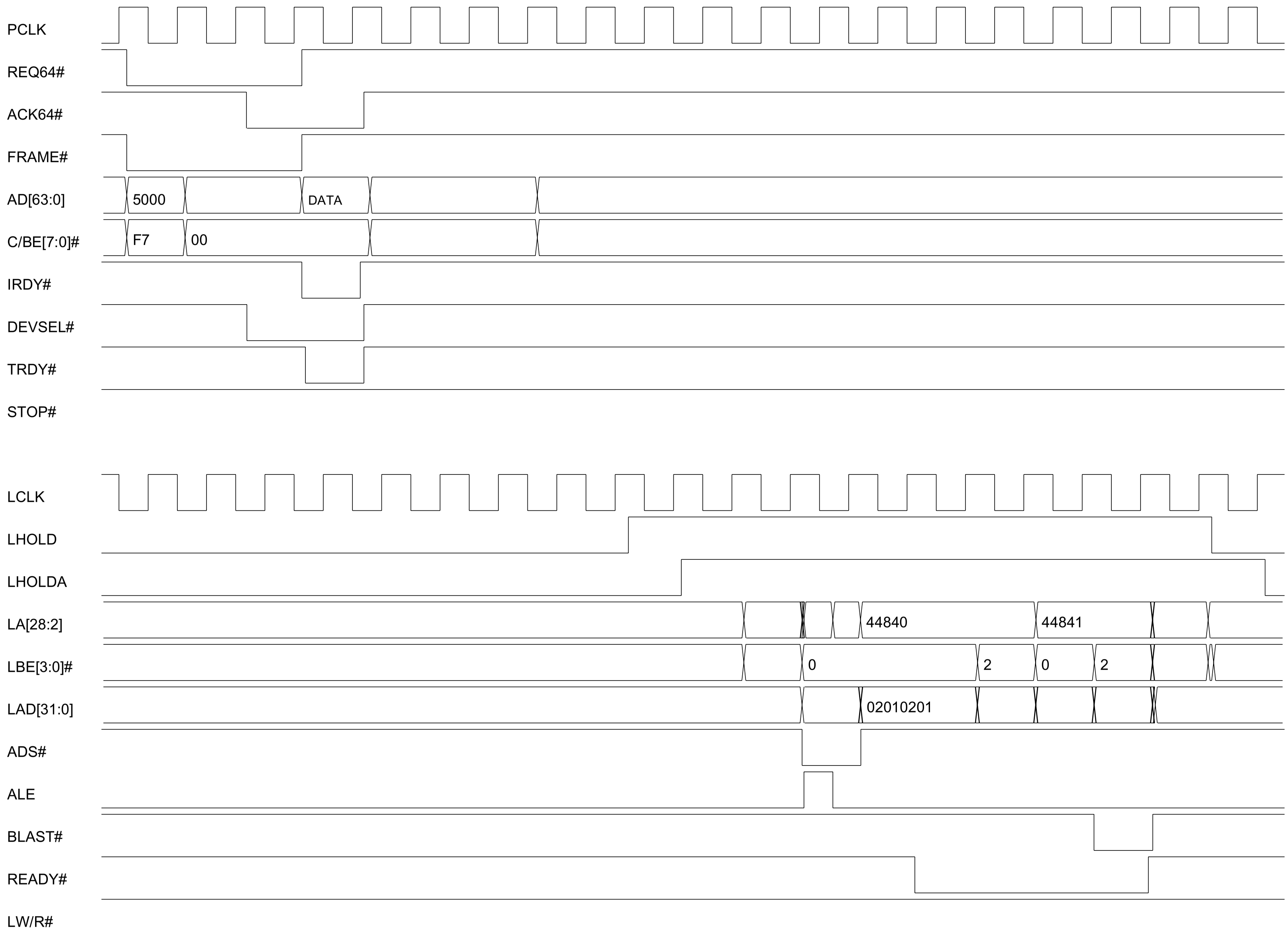
Notes: *This 14-Lword Direct Slave Burst Write transfer is interrupted multiple times by a multi-clock cycle BTERM# assertion.*
The PCI 9656 generates a new Address phase for each Lword of data transferred, as long as BTERM# remains asserted.
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.

Timing Diagram 5-43. Direct Slave 64-Bit Single Cycle Write (8-Bit Local Bus)



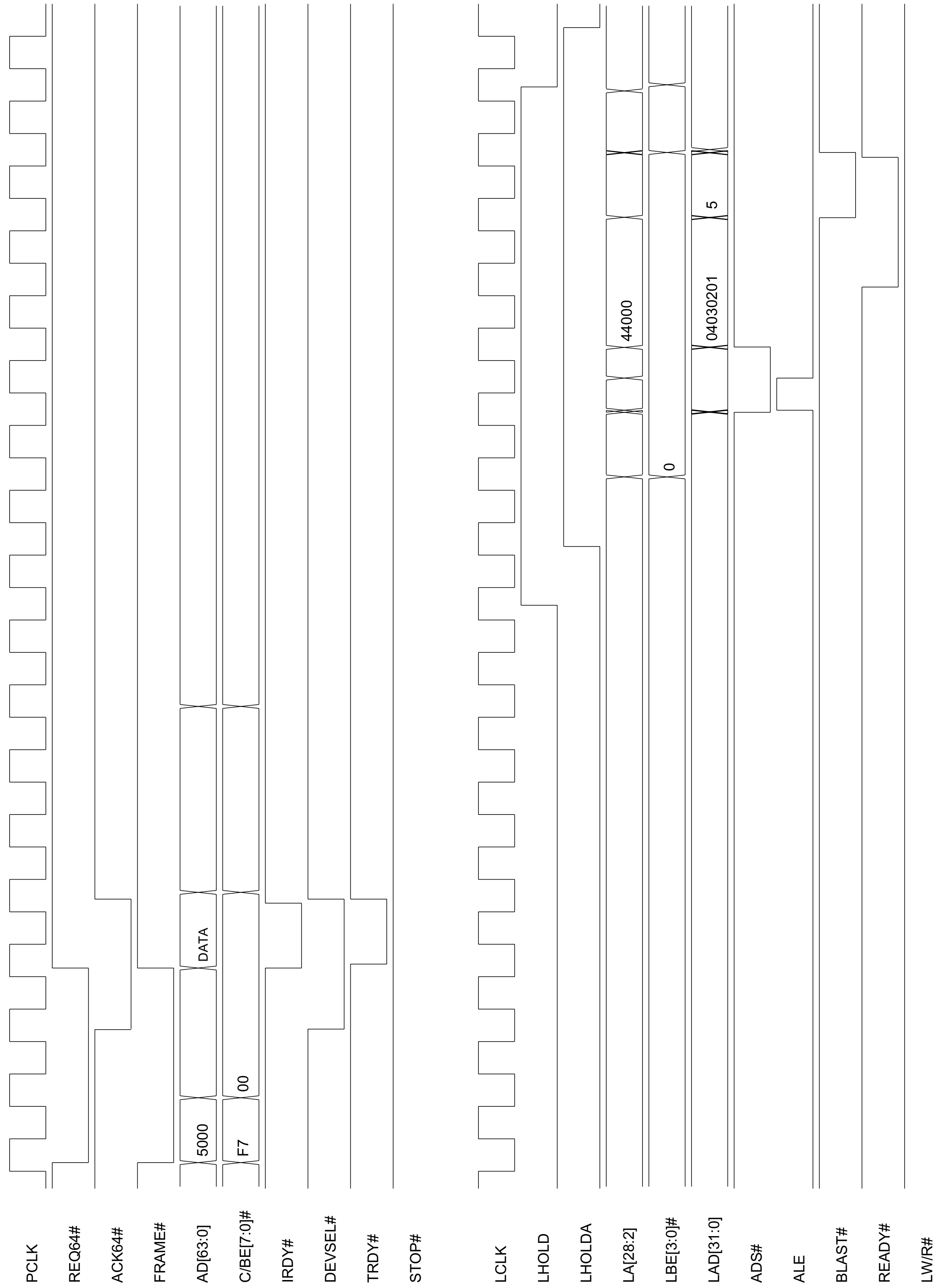
Notes: *Hex value for AD[63:0] DATA = 0807_0605_0403_0201h.*
Key bit/register values are MARBR[24]=1 and LBRD1[15:0]=1540h.

Timing Diagram 5-44. Direct Slave 64-Bit Single Cycle Write (16-Bit Local Bus)



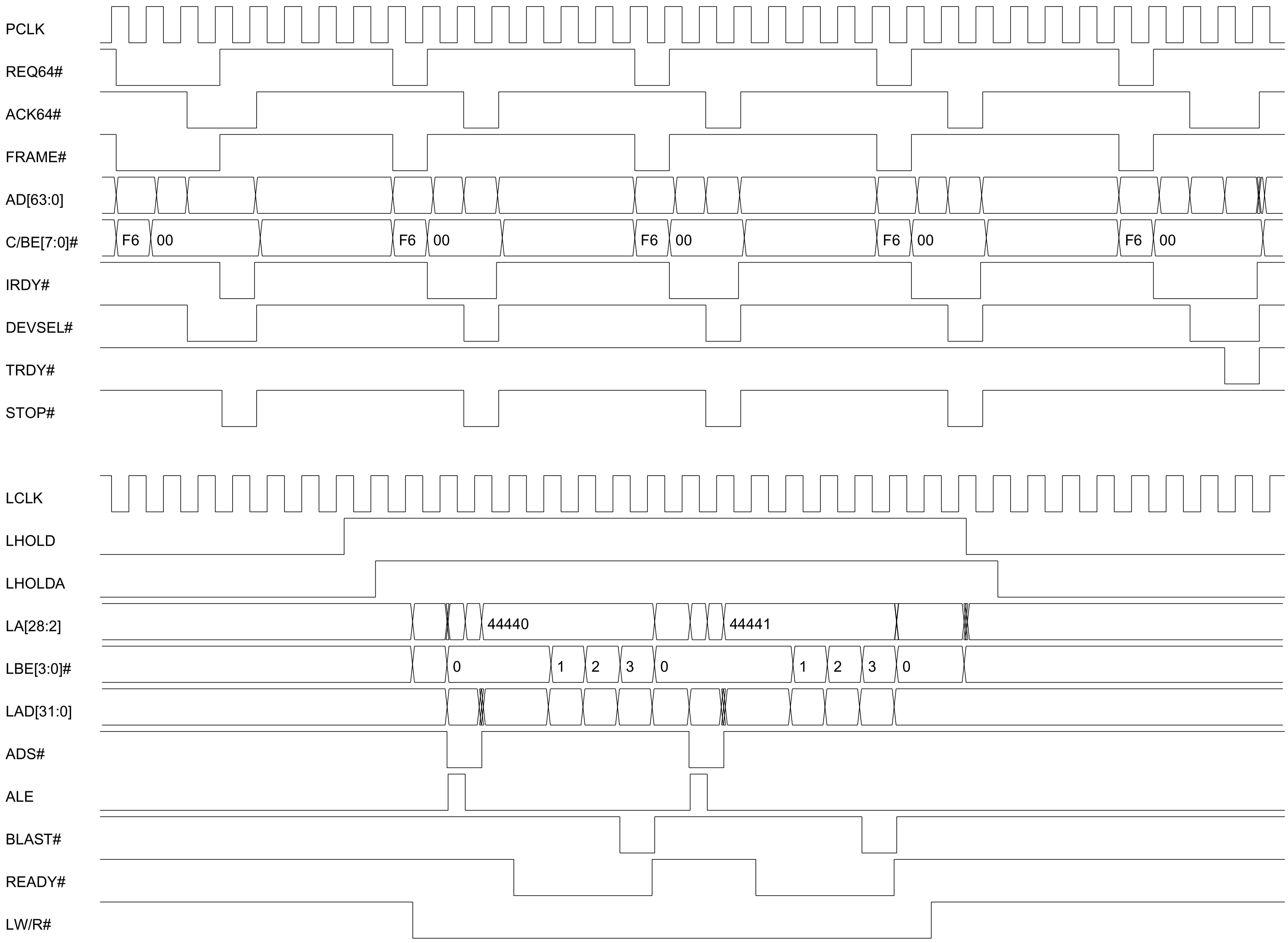
Notes: *Hex value for AD[63:0] DATA = 0807_0605_0403_0201h.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=2541h.

Timing Diagram 5-45. Direct Slave 64-Bit Single Cycle Write (32-Bit Local Bus)



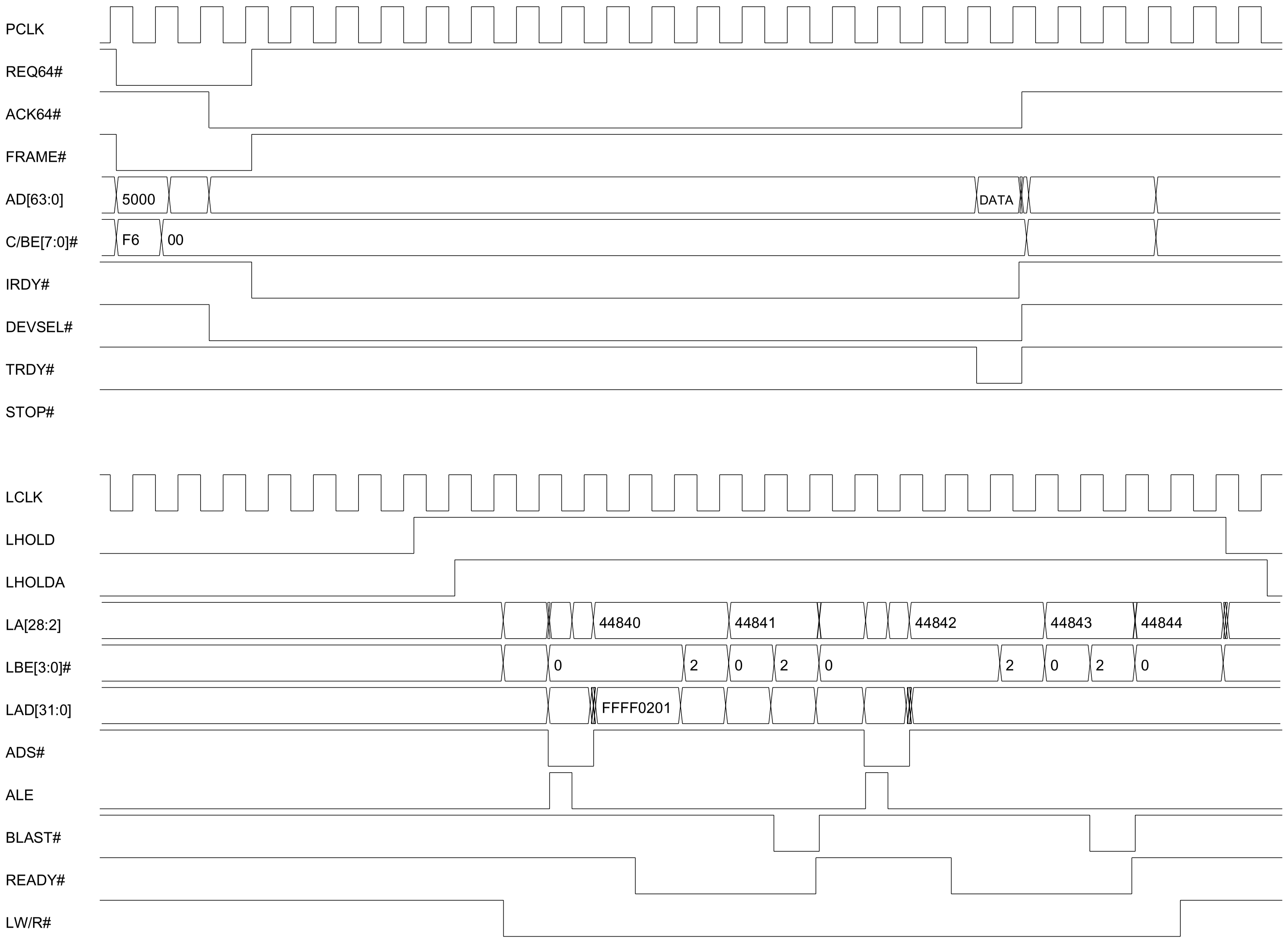
Notes: *Hex value for AD[63:0] DATA = 0000_0005_0403_0201h.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=1542h.

Timing Diagram 5-46. Direct Slave 64-Bit Single Cycle Read (8-Bit Local Bus)



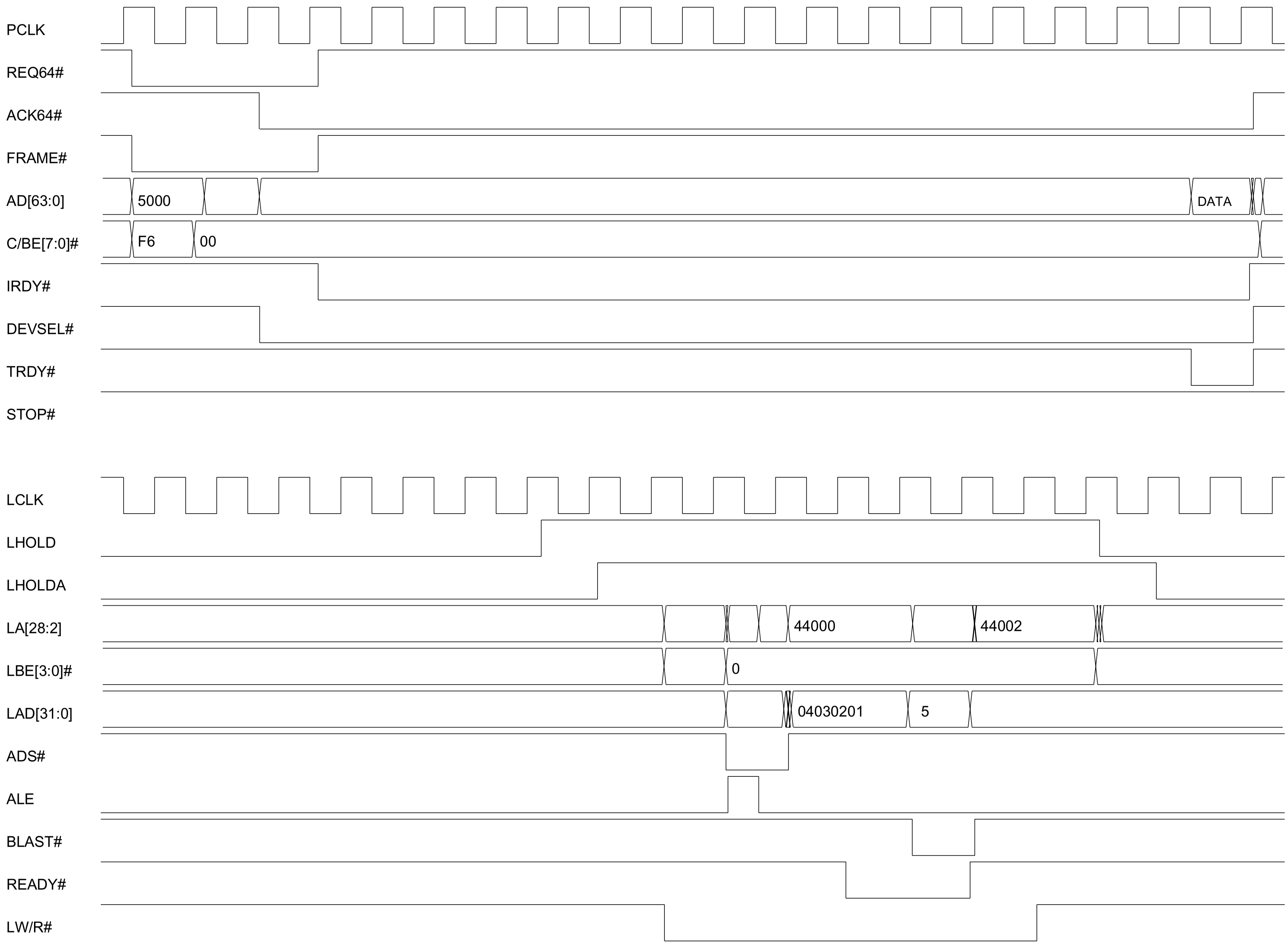
Note: *Key bit/register values are MARBR[24]=1 and LBRD1[15:0]=1540h.*

Timing Diagram 5-47. Direct Slave 64-Bit Single Cycle Read (16-Bit Local Bus)



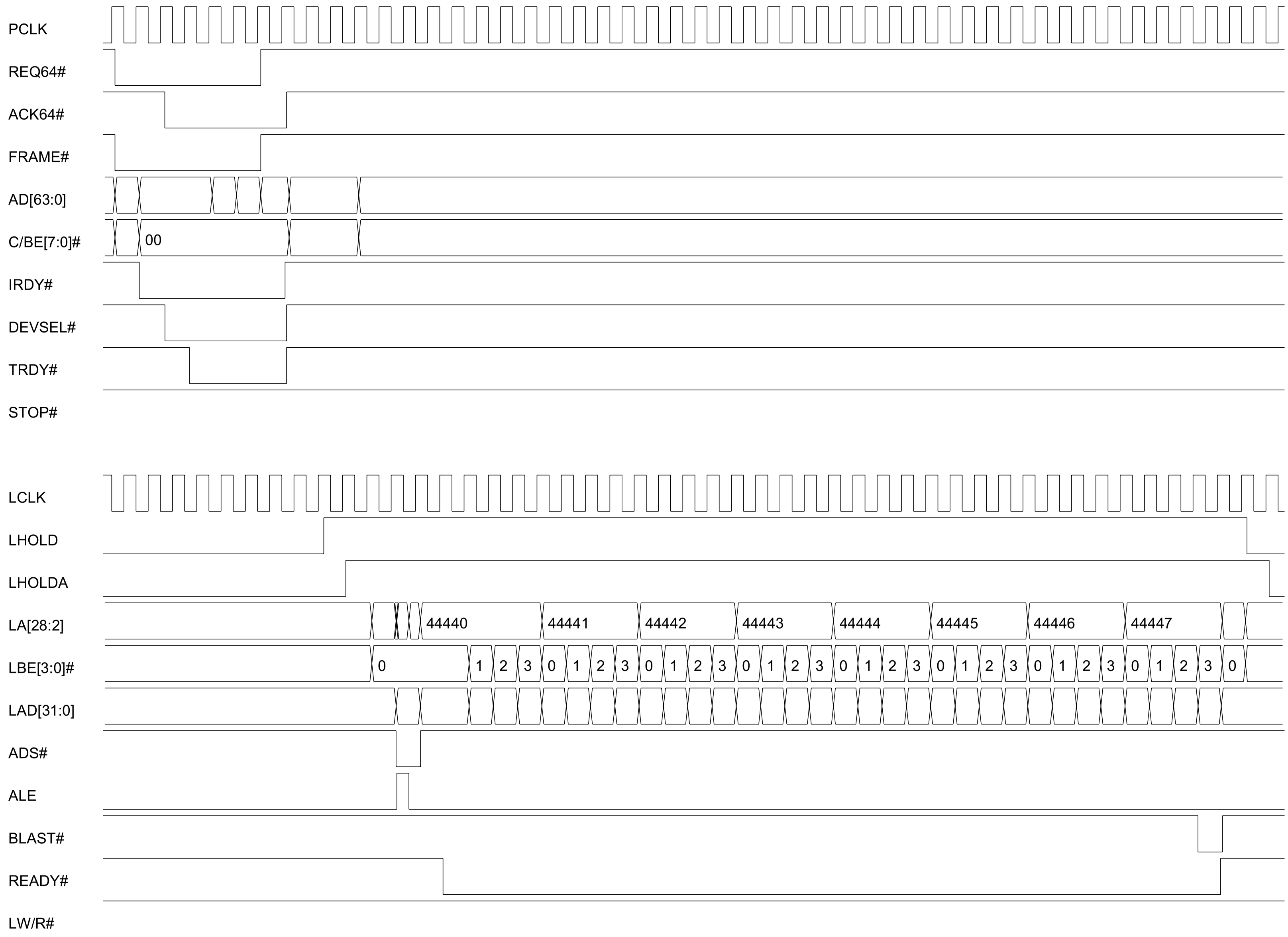
Notes: *Hex value for AD[63:0] DATA = 0807_0605_0403_0201h.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=2541h.
With prefetch enabled LBRD1[9]=0 and LBRD1[14:11]=0100b, two additional Lwords of data are prefetched.
The LAD[31:0] value for the additional two Lwords is not shown.

Timing Diagram 5-48. Direct Slave 64-Bit Single Cycle Read (32-Bit Local Bus)



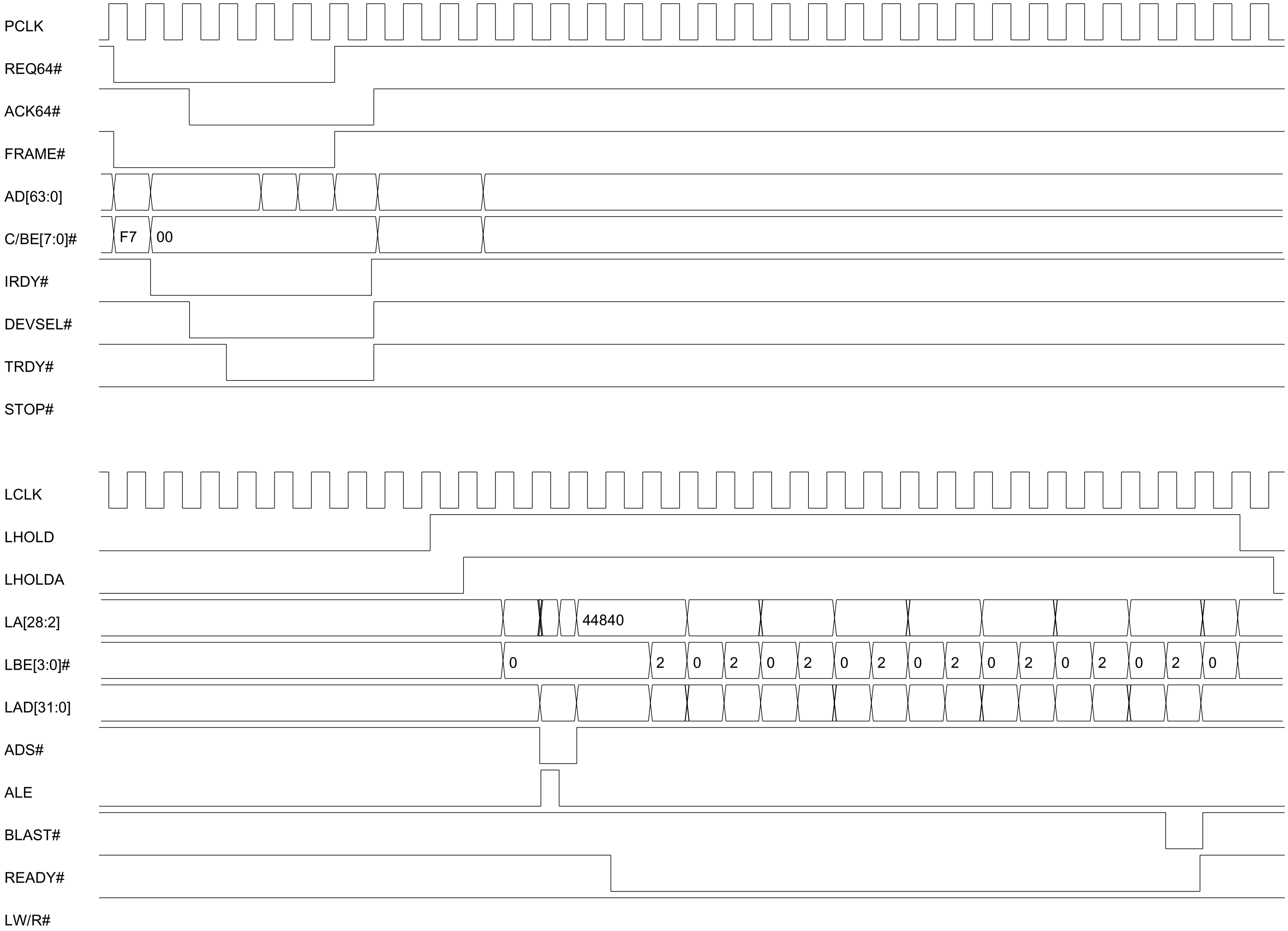
Notes: *Hex value for AD[63:0] DATA = 0000_0005_0403_0201h.*
Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=1542h.

Timing Diagram 5-49. Direct Slave 64-Bit Burst Write of 4 Qwords (8-Bit Local Bus)



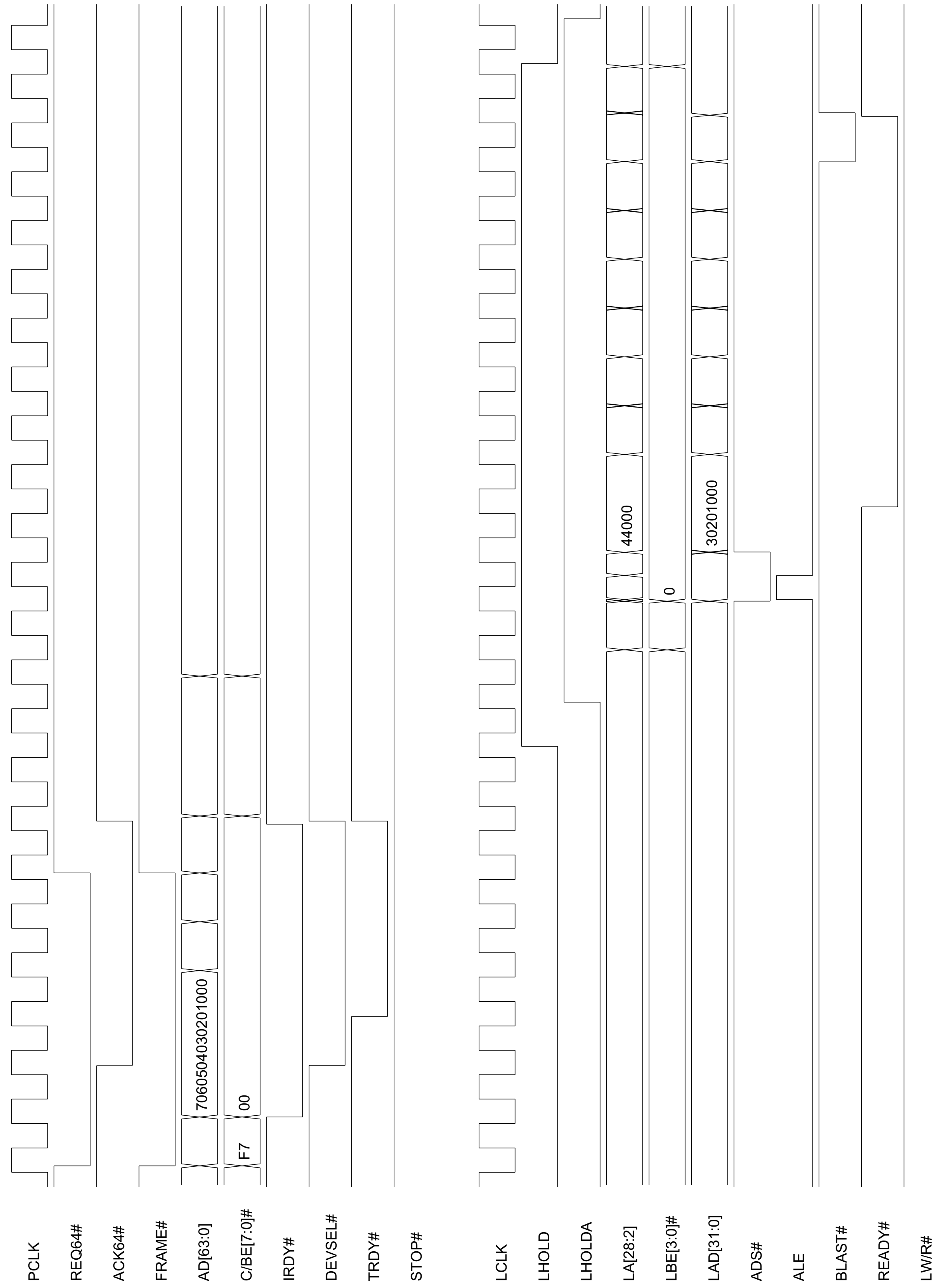
Note: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C0h.*

Timing Diagram 5-50. Direct Slave 64-Bit Burst Write of 4 Qwords (16-Bit Local Bus)



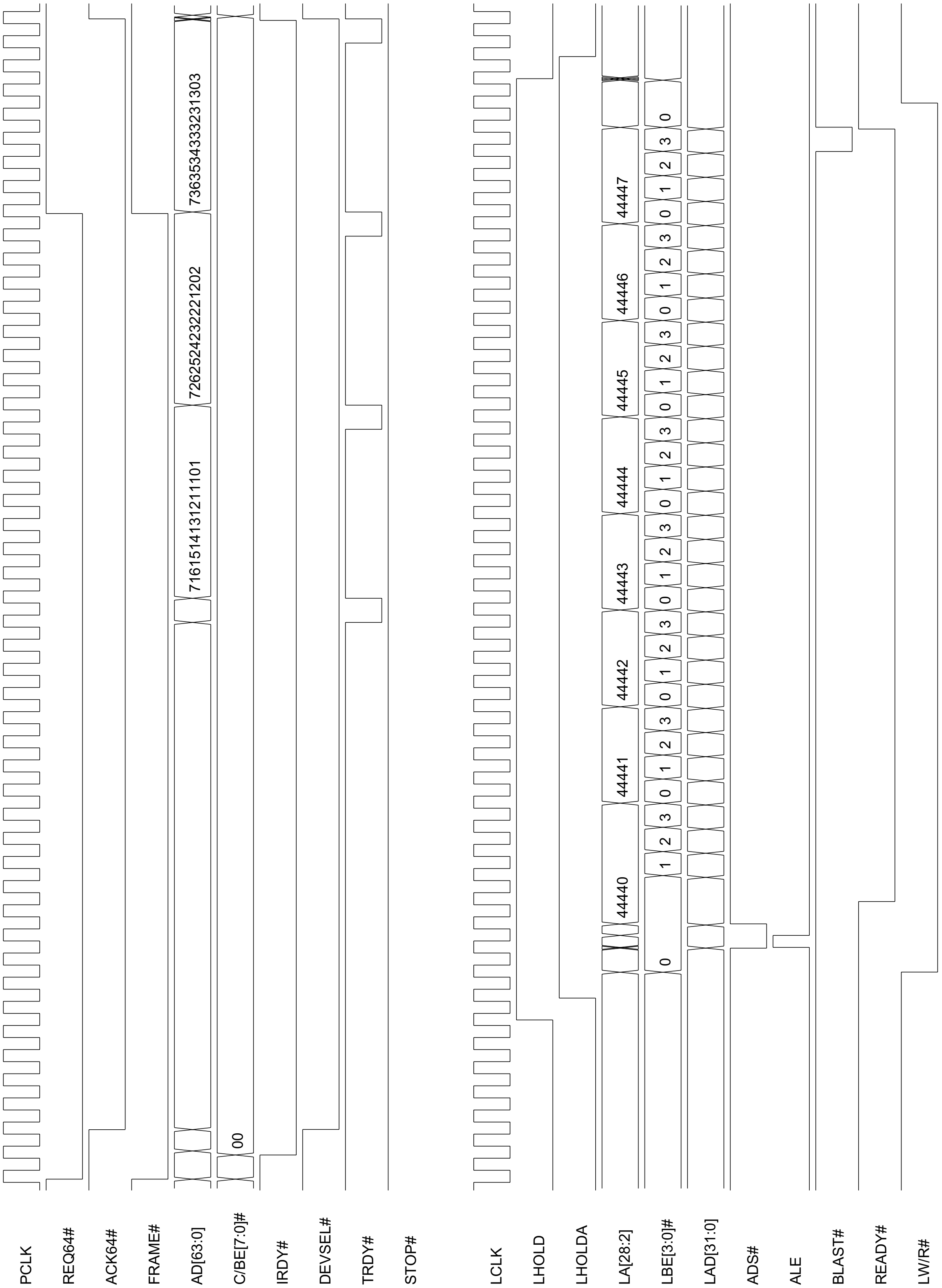
Note: *Key bit/register values are MARBR[24]=1 and LBRD1[15:0]=45C1h.*

Timing Diagram 5-51. Direct Slave 64-Bit Burst Write of 4 Qwords (32-Bit Local Bus)



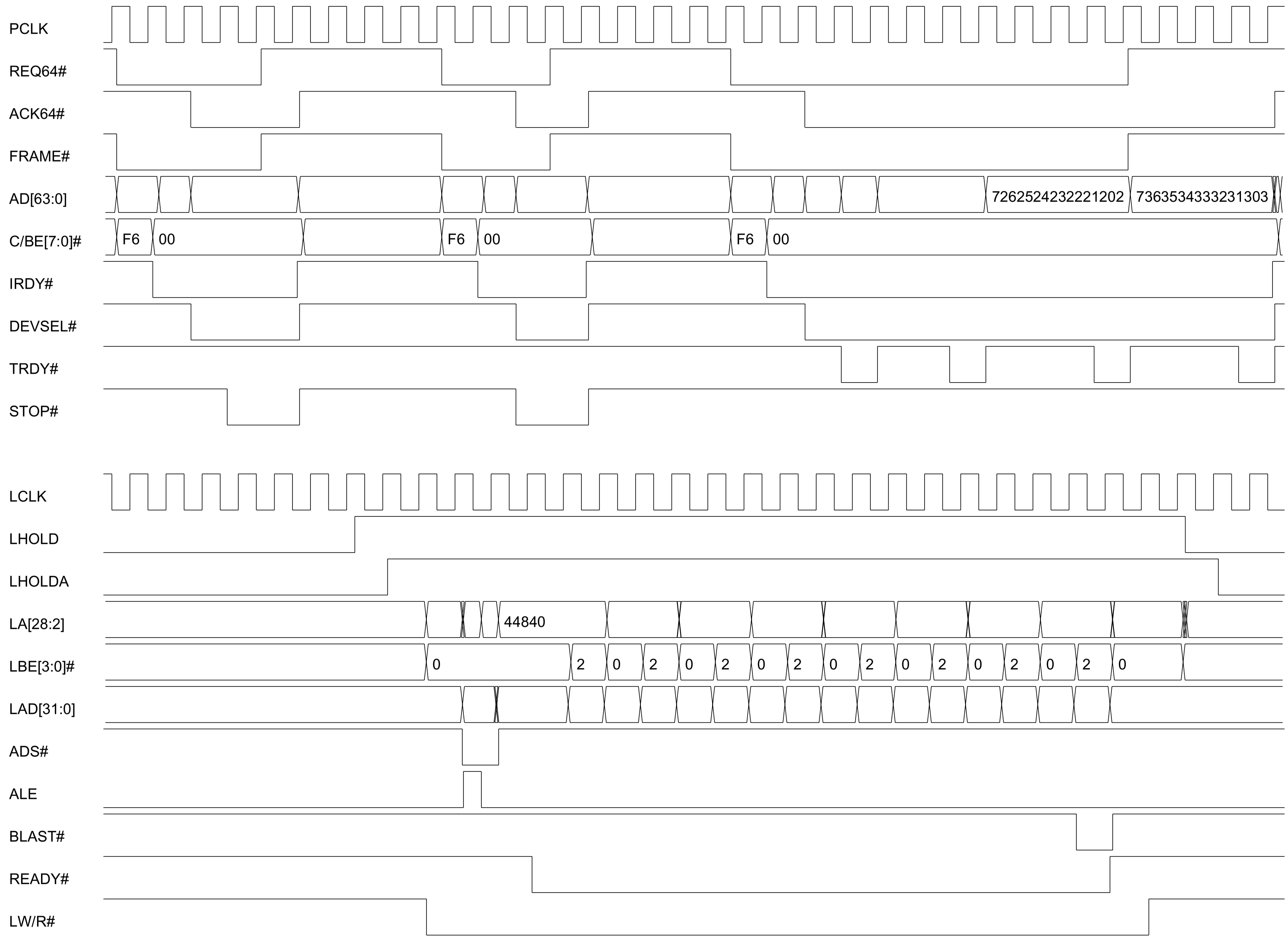
Note: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.*

Timing Diagram 5-52. Direct Slave 64-Bit Burst Read of 4 Qwords (8-Bit Local Bus)



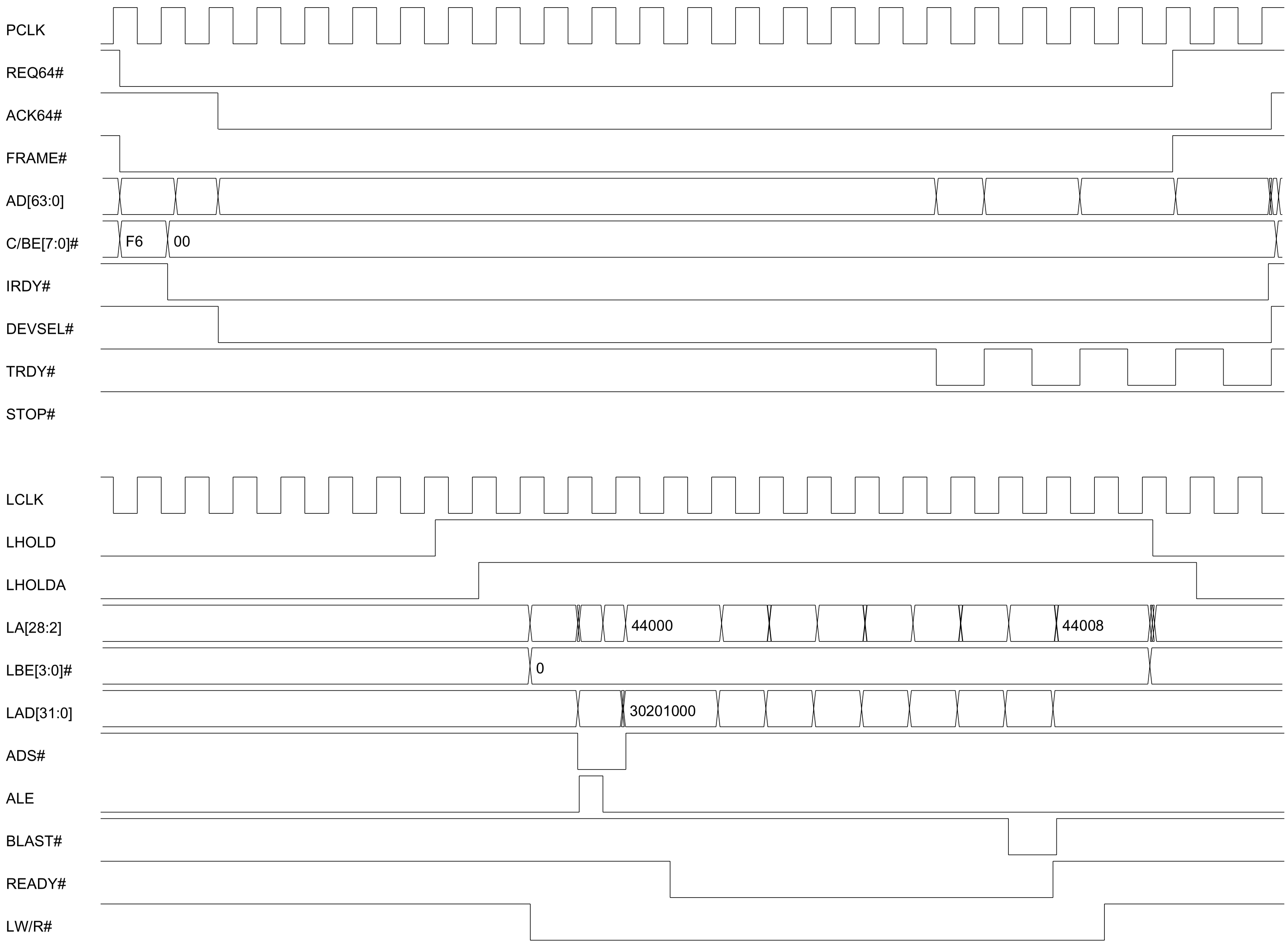
Note: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C0h.*

Timing Diagram 5-53. Direct Slave 64-Bit Burst Read of 4 Qwords (16-Bit Local Bus)



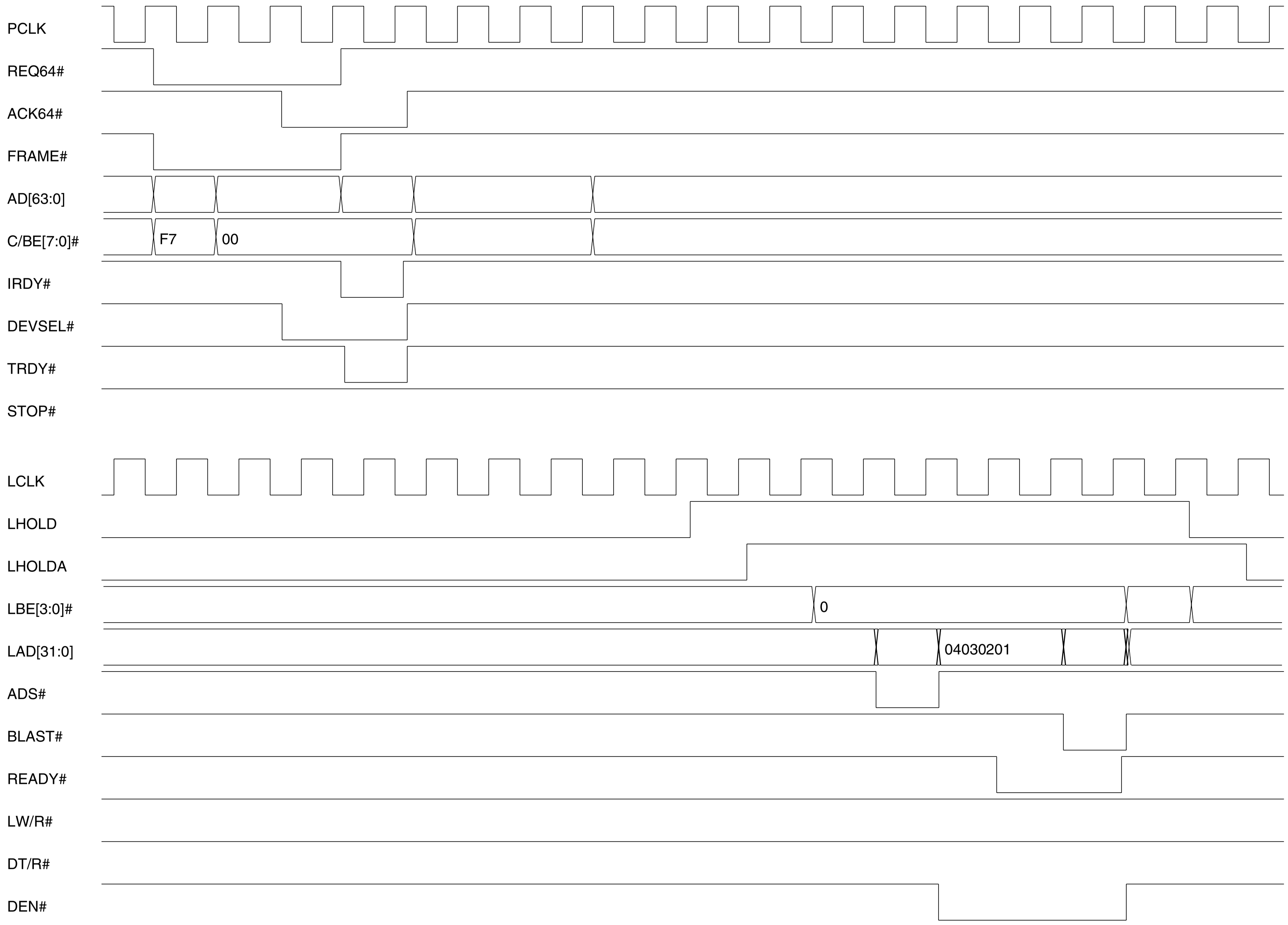
Note: *Key bit/register values are MARBR[24]=1 and LBRD1[15:0]=45C1h.*

Timing Diagram 5-54. Direct Slave 64-Bit Burst Read of 4 Qwords (32-Bit Local Bus)



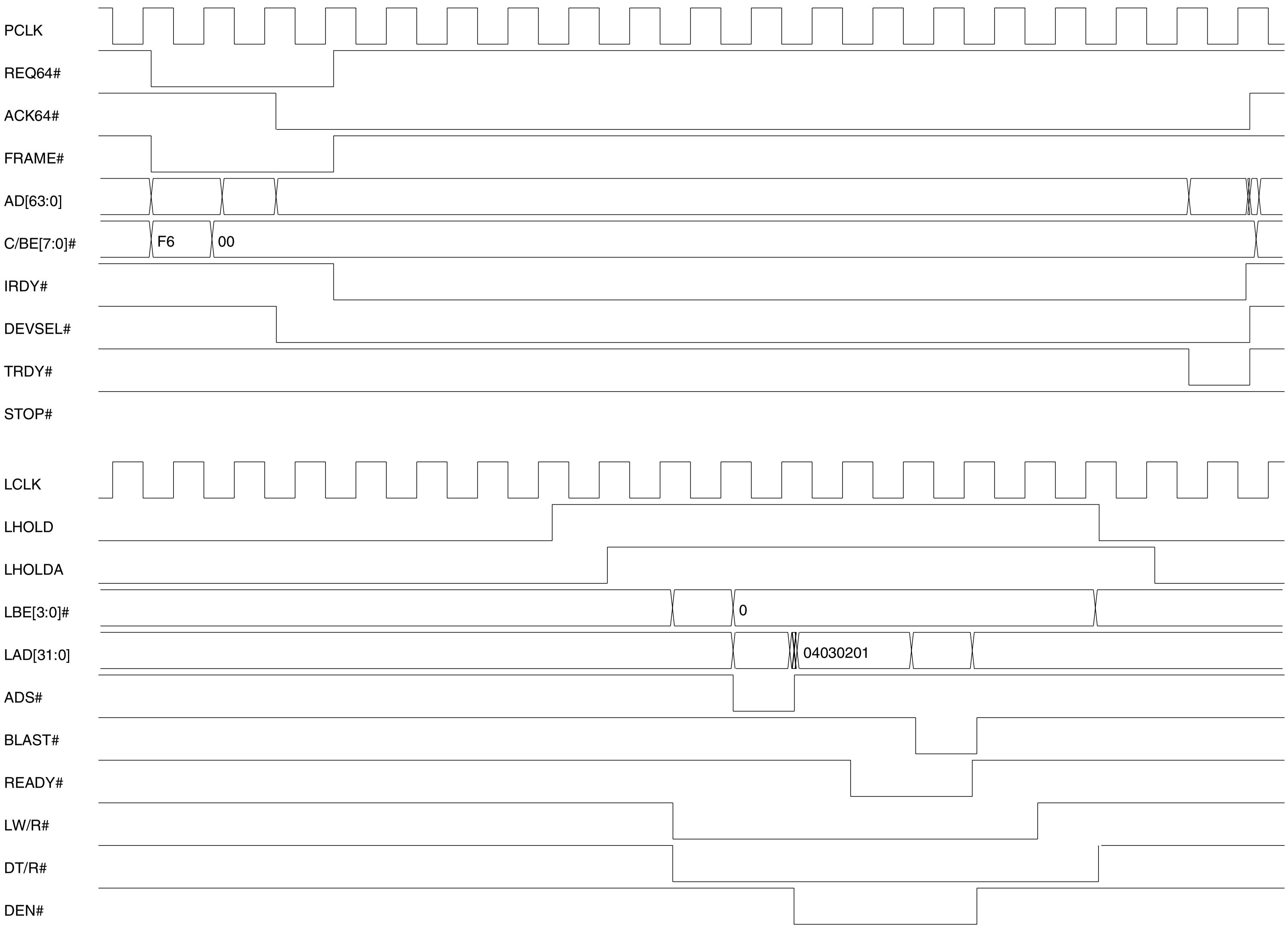
Note: *Key bit/register values are MARBR[24]=0 and LBRD1[15:0]=45C2h.*

Timing Diagram 5-55. Direct Slave Write with DT/R# and DEN#



Notes: *This 64-bit PCI single cycle Direct Slave write to a 32-bit Local Bus device/transceivers timing diagram includes the DT/R# and DEN# signals. The PCI 9656 always tri-states the DT/R# and DEN# signals, except during the time it owns the Local Bus (LHOLD and LHOLDA asserted) To process a Direct Slave or DMA transfer. Because the DT/R# and DEN# signal/pins are pulled up, they are seen as a 1 (high) in this diagram when the PCI 9656 is not driving them. DT/R# and DEN# maybe OR'ed to create the DIR signal for the transceiver(s).*

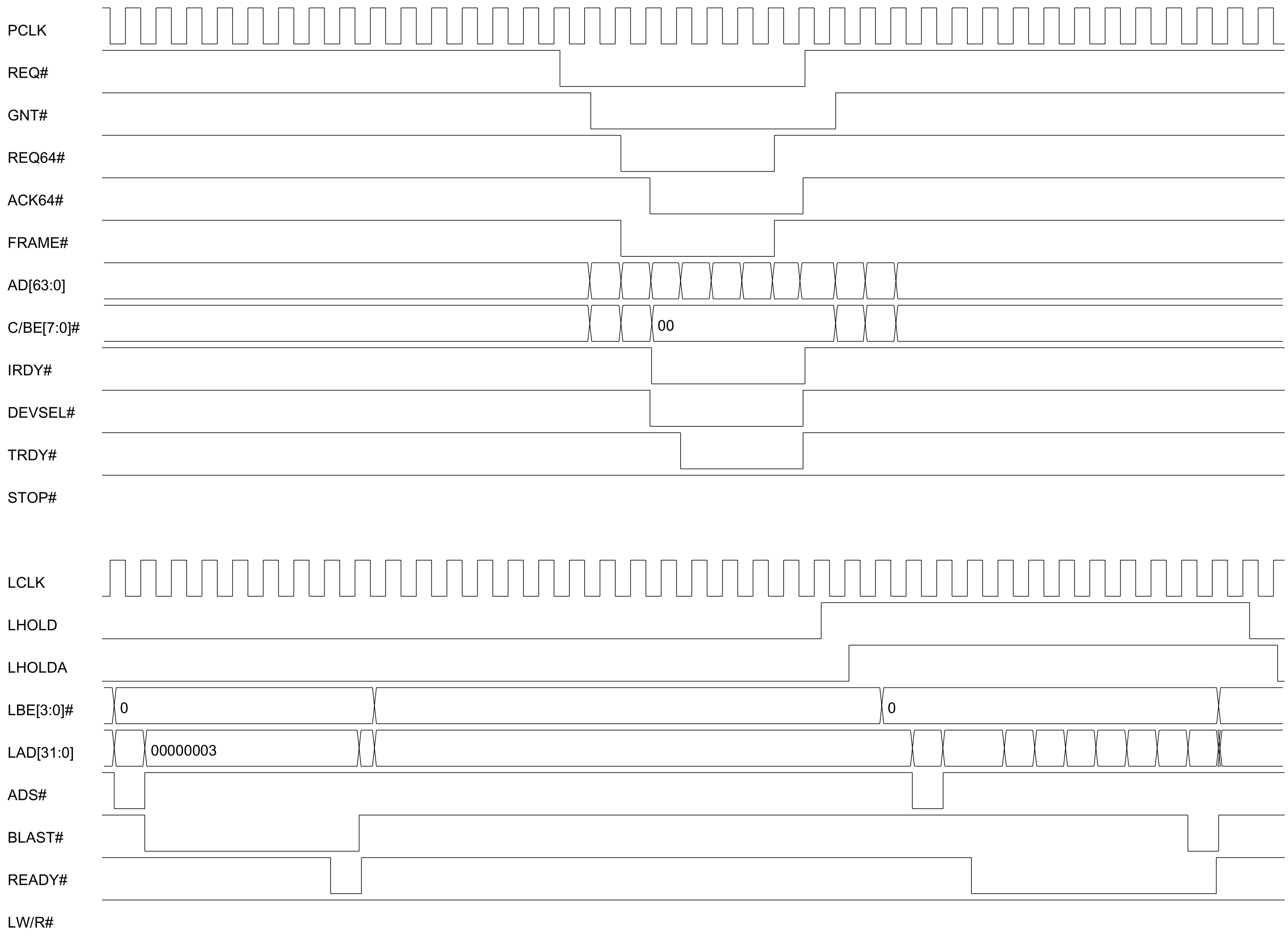
Timing Diagram 5-56. Direct Slave Read with DT/R# and DEN#



Notes: *This 64-bit PCI single cycle Direct Slave read from a 32-bit Local Bus device/transceivers timing diagram includes the DT/R# and DEN# signals. The PCI 9656 always tri-states the DT/R# and DEN# signals, except during the time it owns the Local Bus (LHOLD and LHOLDA asserted) to process a Direct Slave or DMA transfer. Because the DT/R# and DEN# signal/pins are pulled up, they are seen as a 1 (high) in this diagram when the PCI 9656 is not driving them. DT/R# and DEN# maybe OR'ed to create the DIR signal for the transceiver(s).*

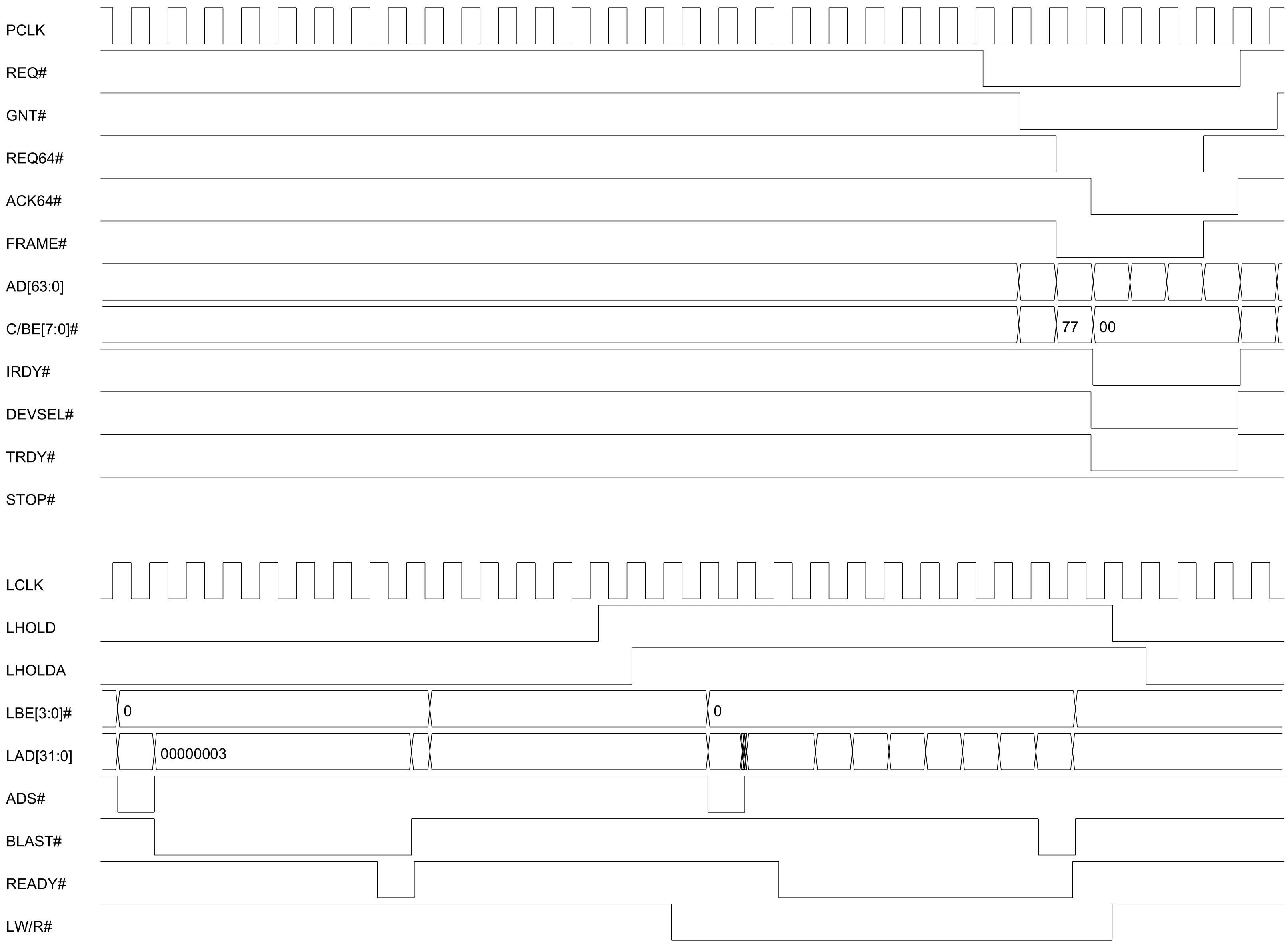
5.7.3 J Mode DMA Timing Diagrams

Timing Diagram 5-57. DMA PCI-to-Local (32-Bit Local Bus, 64-Bit PCI Bus)



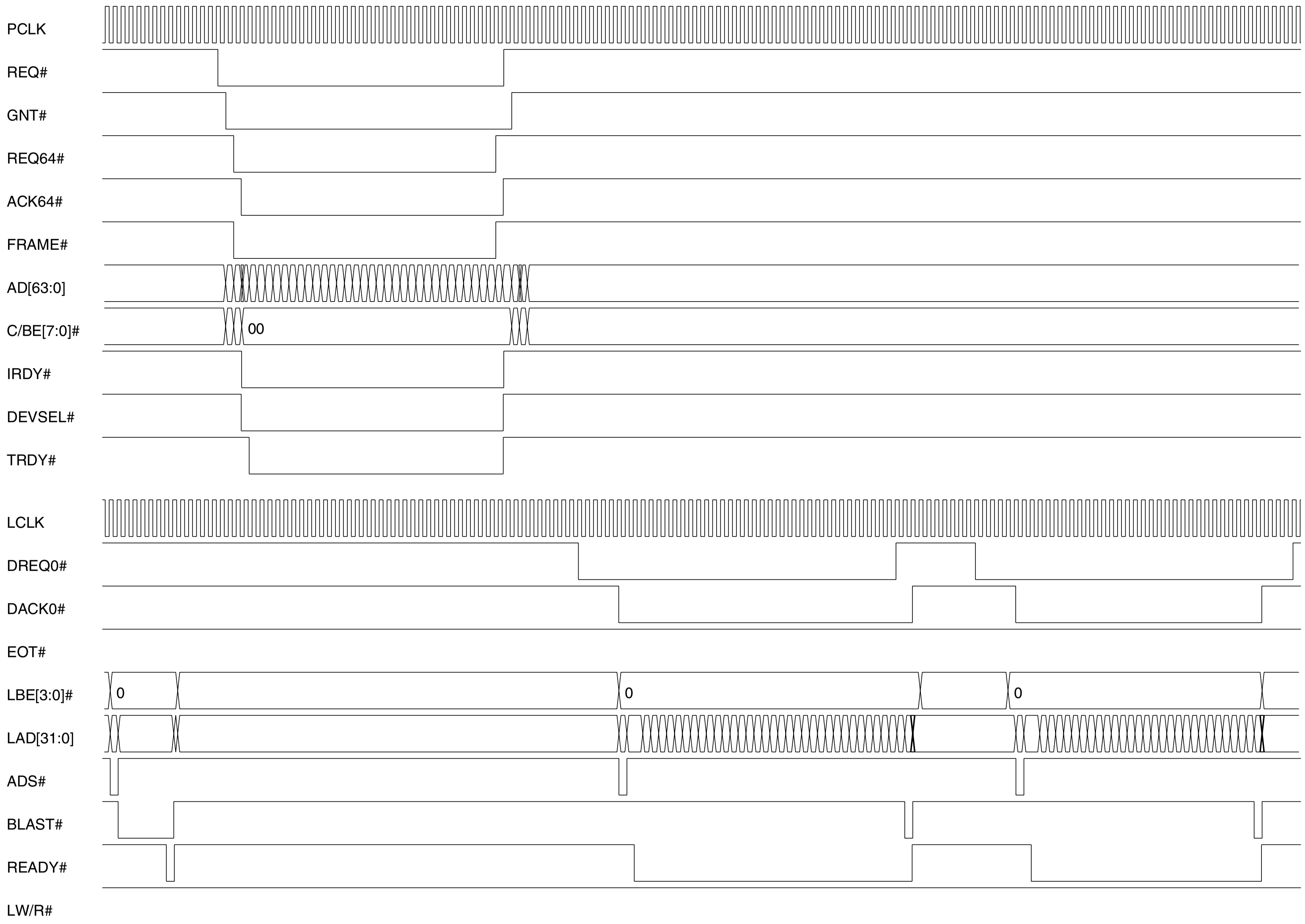
Note: *The writing of the registers to set up this 32-byte DMA Block mode transfer using DMA Channel 0 is not shown. Writing the DMACSR0 register (LOC:128h) with 03h enables and starts this DMA transfer.*

Timing Diagram 5-58. DMA Local-to-PCI (32-Bit Local Bus, 64-Bit PCI Bus)



Note: *The writing of the registers to set up this 32-byte DMA Block mode transfer using DMA Channel 0 is not shown. Writing the DMACSR0 register (LOC:128h) with 03h enables and starts this DMA transfer.*

Timing Diagram 5-59. DMA PCI-to-Local Demand Mode (32-Bit Local Bus, 64-Bit PCI Bus)



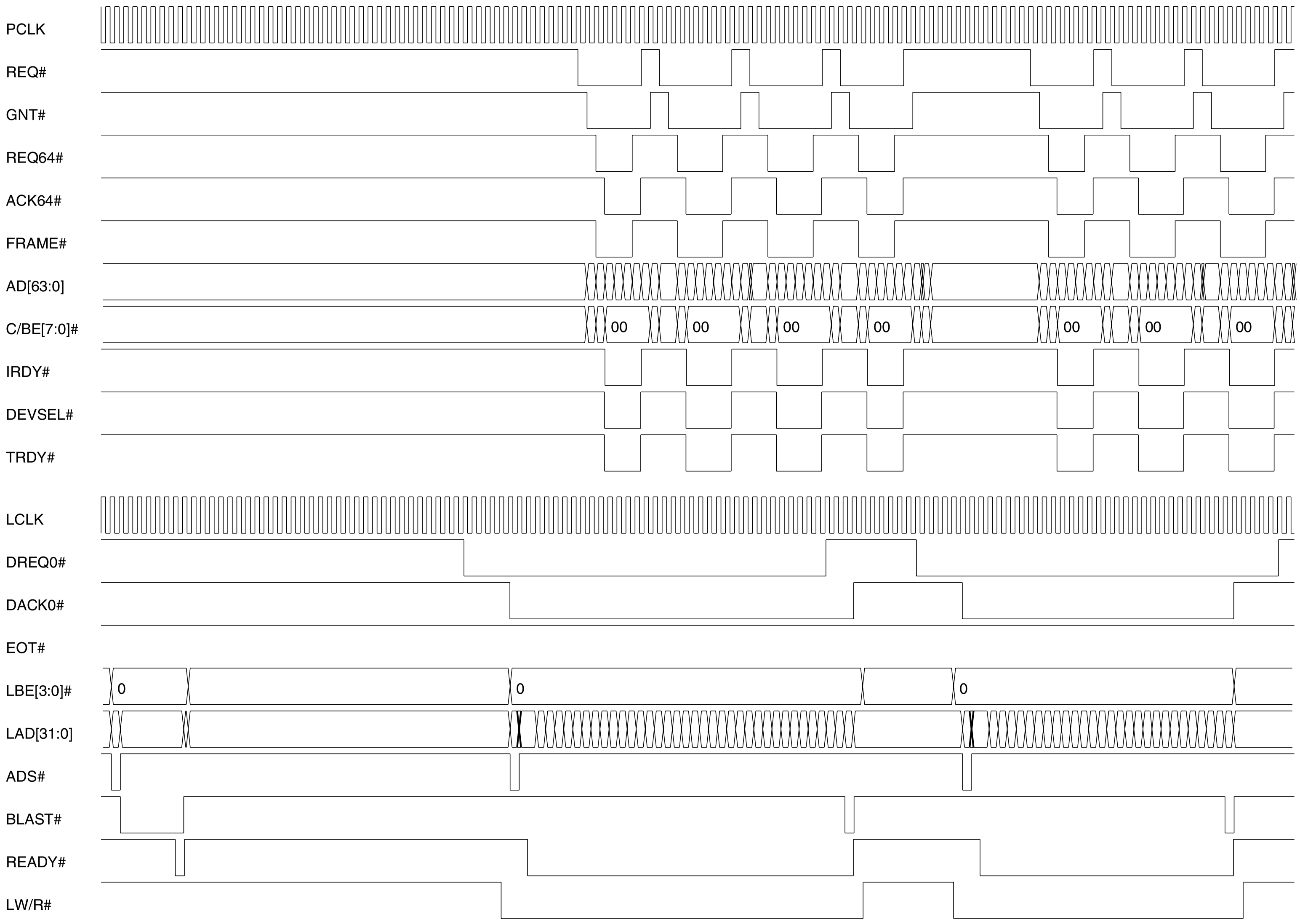
Notes: *The Channel 0 Demand mode DMA transfer starts when the DMACSR0[1:0] bits are written to 11b.*

The PCI 9656 does not attempt to arbitrate for and write data to the Local Bus until it detects the DREQ0# signal asserted.

The transfer is temporarily suspended when DREQ0# is de-asserted, then resumed upon its re-assertion.

The PCI STOP# signal (not shown) is not asserted during this transfer.

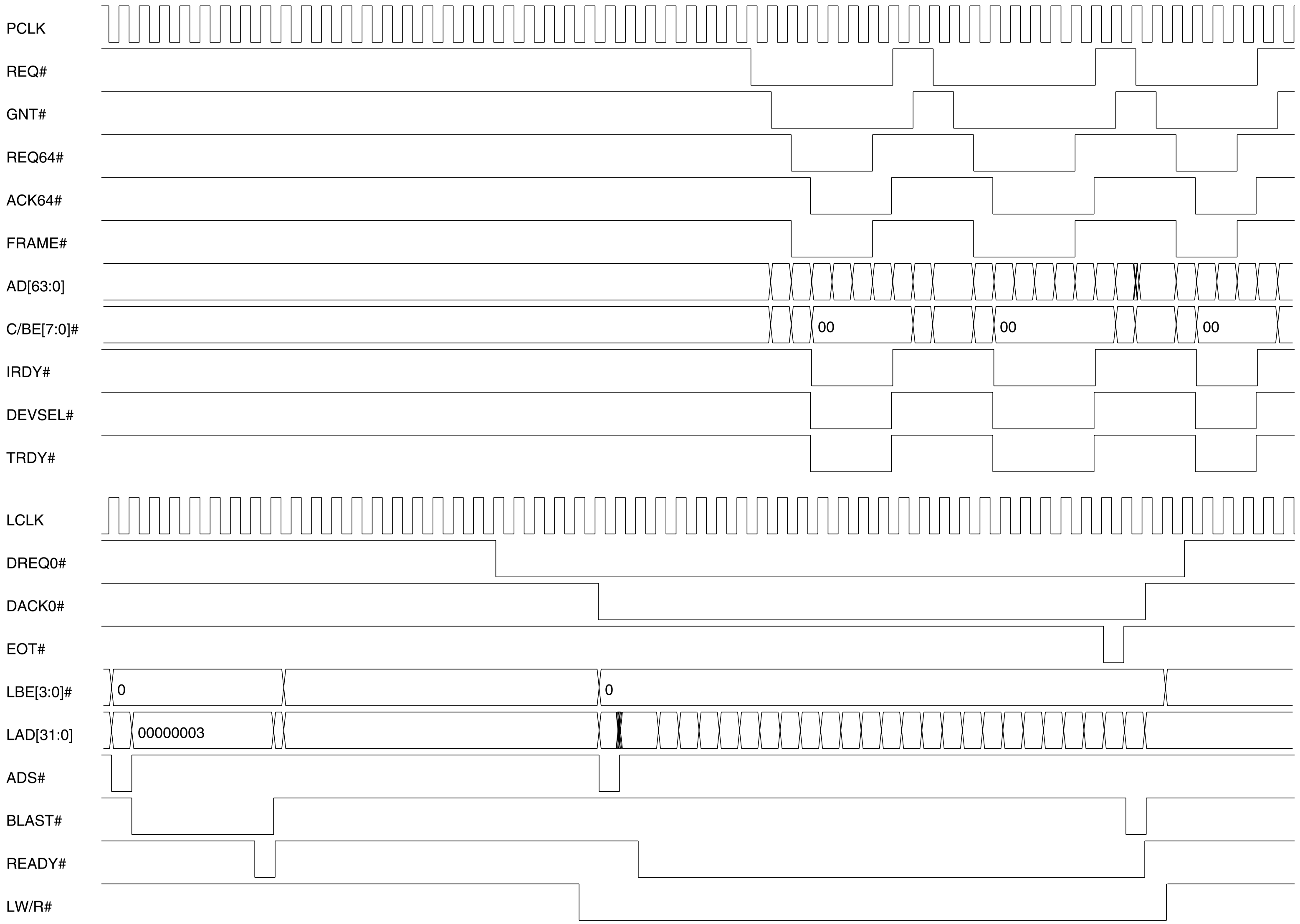
Timing Diagram 5-60. DMA Local-to-PCI Demand Mode (32-Bit Local Bus, 64-Bit PCI Bus)



Notes: *The Channel 0 Demand mode DMA transfer starts when the DMACSR0[1:0] bits are written to 11b and the PCI 9656 detects the DREQ0# signal asserted. Once started, the PCI 9656 arbitrates for and reads data from the Local Bus, then writes the data to the PCI Bus until the transfer is temporarily suspended when DREQ0# is de-asserted. The transfer resumes upon DREQ0# re-assertion.*

The PCI STOP# signal (not shown) is not asserted during this transfer.

Timing Diagram 5-61. DMA Local-to-PCI Demand Mode with EOT# assertion (32-Bit Local Bus, 64-Bit PCI Bus)

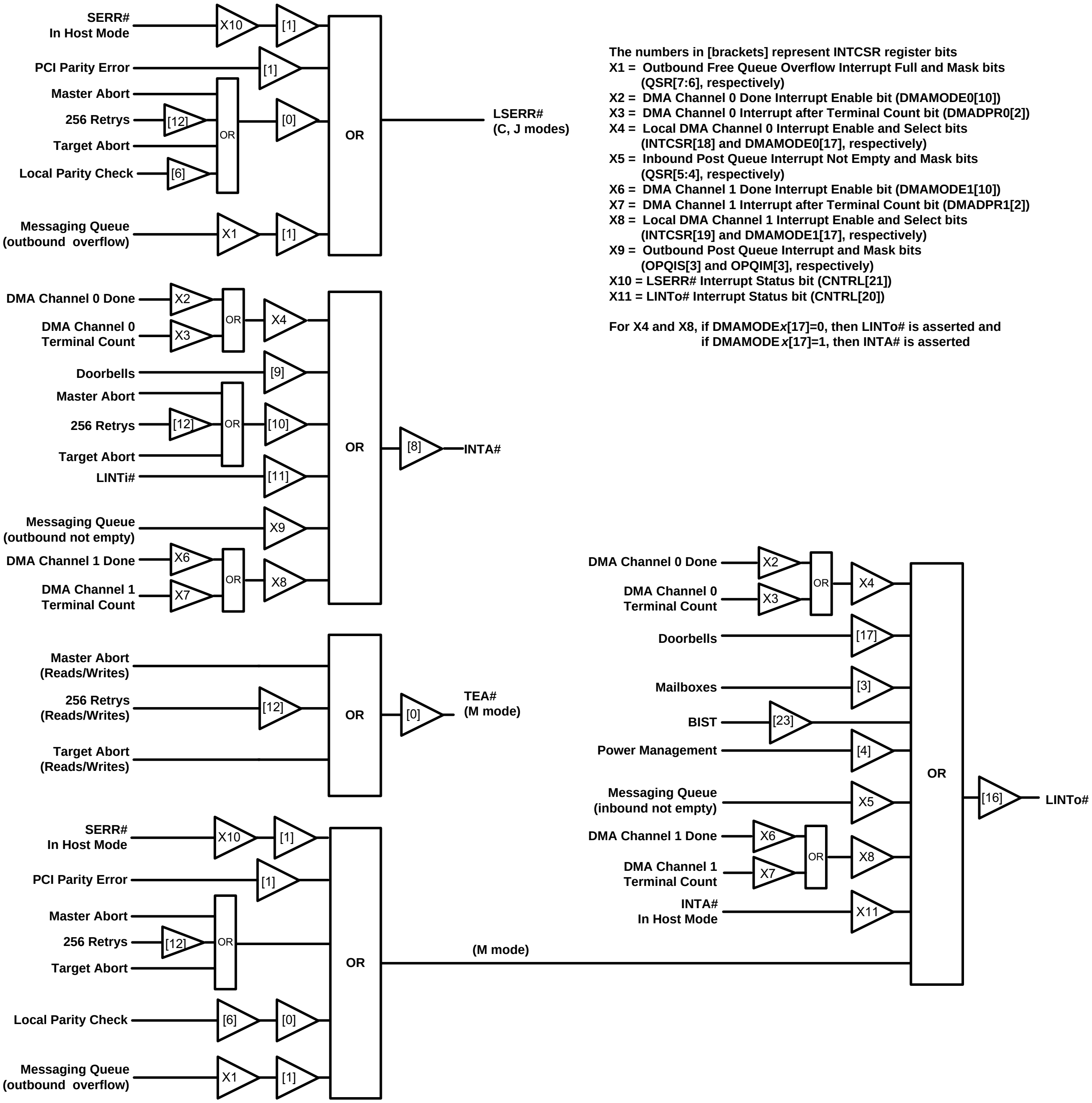


Notes: *The Channel 0 Demand mode DMA transfer starts when the DMACSR0[1:0] bits are written to 11b and the PCI 9656 detects the DREQ0# signal asserted. Once started, the PCI 9656 arbitrates for and reads data from the Local Bus, then writes the data to the PCI Bus until the transfer is terminated with an EOT# assertion. All data that is read into the DMA FIFO is written to the PCI Bus.*

6 INTERRUPTS, USER I/O, AND IDDQEN#

This section provides information about PCI 9656 interrupts, interrupt sources, user I/O pins, and IDDQEN# functionality.

6.1 INTERRUPTS



The numbers in [brackets] represent INTCSR register bits
X1 = Outbound Free Queue Overflow Interrupt Full and Mask bits (QSR[7:6], respectively)
X2 = DMA Channel 0 Done Interrupt Enable bit (DMAMODE0[10])
X3 = DMA Channel 0 Interrupt after Terminal Count bit (DMADPR0[2])
X4 = Local DMA Channel 0 Interrupt Enable and Select bits (INTCSR[18] and DMAMODE0[17], respectively)
X5 = Inbound Post Queue Interrupt Not Empty and Mask bits (QSR[5:4], respectively)
X6 = DMA Channel 1 Done Interrupt Enable bit (DMAMODE1[10])
X7 = DMA Channel 1 Interrupt after Terminal Count bit (DMADPR1[2])
X8 = Local DMA Channel 1 Interrupt Enable and Select bits (INTCSR[19] and DMAMODE1[17], respectively)
X9 = Outbound Post Queue Interrupt and Mask bits (OPQIS[3] and OPQIM[3], respectively)
X10 = LSERR# Interrupt Status bit (CNTRL[21])
X11 = LINTo# Interrupt Status bit (CNTRL[20])

For X4 and X8, if DMAMODE*x*[17]=0, then LINTo# is asserted and if DMAMODE *x*[17]=1, then INTA# is asserted

Figure 6-1. Interrupt and Error Sources

6.1.1 PCI Interrupts (INTA#)

If HOSTEN# is asserted (HOSTEN#=0), the PCI INTA# pin is an input. If HOSTEN# is de-asserted (HOSTEN#=1), PCI INTA# is an output, described as follows.

The source of a PCI 9656 PCI Interrupt (INTA#) assertion is one of the following:

- Local-to-PCI Doorbell register Write access
- Local Interrupt input (LINTi#) assertion
- Master/Target Abort Status condition
- DMA Channel *x* Done or Abort
- DMA Channel *x* Terminal Count is reached
- Messaging Outbound Post Queue not empty
- 256 consecutive PCI Retrys

INTA#, or individual sources of an interrupt, can be enabled, disabled, or cleared using various PCI 9656 bits described in this section. Additionally, the interrupt status bits of each interrupt source are provided.

The PCI 9656 PCI interrupt is a level output. Disabling an Interrupt Enable bit or clearing the cause of the interrupt clears (de-asserts) the INTA# interrupt pin.

6.1.2 Local Interrupt Input (LINTi#)

If the Local Interrupt Input is enabled (INTCSR[11]=1), asserting Local LINTi# can cause the PCI 9656 to assert a PCI interrupt (INTA#). The PCI Host processor can read the PCI 9656 Interrupt Control/ Status register (INTCSR) to determine whether an interrupt is pending as a result of the LINTi# assertion (INTCSR[15]=1).

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, regardless of the LINTi# pin state. The Local Interrupt Input Active bit is always (INTCSR[15]=1) active and reflects the LINTi# pin state, regardless of the INTCSR[11] bit setting.

Timing Diagram 6-1 illustrates INTA# behavior when LINTi# is asserted and de-asserted, if INTA# is enabled.

6.1.3 Local Interrupt Output (LINTo#)

The PCI 9656 Local Interrupt output (LINTo#) can be asserted by one of the following:

- PCI-to-Local Doorbell register Write access
- PCI-to-Local Mailbox register (MBOX0, MBOX1, MBOX2, and/or MBOX3) Write access
- PCI Built-In Self-Test interrupt
- DMA Channel *x* Done or Abort interrupt
- DMA Channel *x* Terminal Count is reached
- Messaging Inbound Post Queue is not empty
- PCI INTA# assertion, if the HOSTEN# pin is asserted low
- Power management state change

LINTo#, or individual sources of an interrupt, can be enabled, disabled, or cleared, using the PCI 9656 register bit(s) described in this section. Additionally, interrupt status bits for each interrupt source are provided.

The PCI 9656 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 the interrupt.

In M mode, a LINTo# interrupt is also asserted if any of the following conditions occur:

- PCI Bus Received Master or Target Abort bit is set (PCISR[13 or 12]=1, respectively)
- Detected Parity Error bit is set (PCISR[15]=1)
- Direct Master Write/Direct Slave Read Local Data Parity Check Error Status bit is set (INTCSR[7]=1)
- Messaging Outbound Free queue overflows
- PCI SERR# assertion if the HOSTEN# pin is asserted low

Each of these additional sources of LINTo# assertion in M mode may be masked by INTCSR[12, 6, 1:0], except for PCI Master or Target Aborts. (Refer to Figure 6-1.) The Local Interrupt Output Enable bit INTCSR[16] can be used to enable or disable the LINTo# interrupt pin. When in M mode, TEA# is always enabled and only 256 Retries can be masked from causing TEA# assertion. Local parity errors may be masked separately from asserting LINTo# using INTCSR[6]. The LINTo# signal is a level output that remains asserted as long as the Interrupt Status or Enable bits are set.

6.1.4 PCI Master/Target Abort Interrupt

The PCI 9656 sets the PCI Bus Received Master or Target Abort bit (PCISR[13 or 12]=1, respectively) when, as a PCI master, it detects a PCI Master or Target Abort. These status bits cause the PCI INTA# interrupt to be asserted if interrupts are enabled (INTCSR[10, 8]=11b).

The interrupt remains set as long as the Received Master or Target Abort bit remains set and the PCI Master/Target Abort interrupt is enabled. Use PCI Type 0 Configuration or Local accesses to clear the PCI Bus Received Master and Target Abort bits (PCISR[13:12]=00b, respectively).

The Interrupt Control/Status bits (INTCSR[26:24]) are latched at the time of a PCI Master or Target Abort interrupt. When an abort occurs, these bits provide information, such as which device was the Master at the time of the abort. Additionally, when a PCI Master or Target Abort occurs, the current PCI Address (PCI Abort Address) is stored in the PCI Abort Address bits (PABTADR[31:0]).

6.1.5 Mailbox Registers

The PCI 9656 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 (LINTo#) is asserted, if enabled (INTCSR[3, 16]=11b), when the PCI Host writes to one of the first four Mailbox registers (MBOX0, MBOX1, MBOX2, or MBOX3).

Regardless of whether LINTo# is enabled, a PCI write to one of these four Mailbox registers sets a corresponding status bit in INTCSR[31:28], provided the Mailbox interrupts are enabled (INTCSR[3]=1).

To clear the LINTo# assertion caused by the PCI Host write(s) of any of the four Mailbox registers, a Local Bus device must read each Mailbox register that was written to.

6.1.6 Doorbell Registers

The PCI 9656 has two 32-bit Doorbell registers. One is assigned to the PCI Bus interface; the other to the Local Bus interface.

A PCI host can assert a Local interrupt output (LINTo#) by writing any number other than all zeros (0) to the PCI-to-Local Doorbell bits (P2LDBELL [31:0]). The Local Interrupt remains asserted until all P2LDBELL bits are cleared to 0. A Local processor can assert a PCI interrupt (INTA#) by writing any number other than all zeros (0) to the Local-to-PCI Doorbell bits (L2PDBELL[31:0]). The PCI interrupt remains asserted until all L2PDBELL bits are cleared to 0.

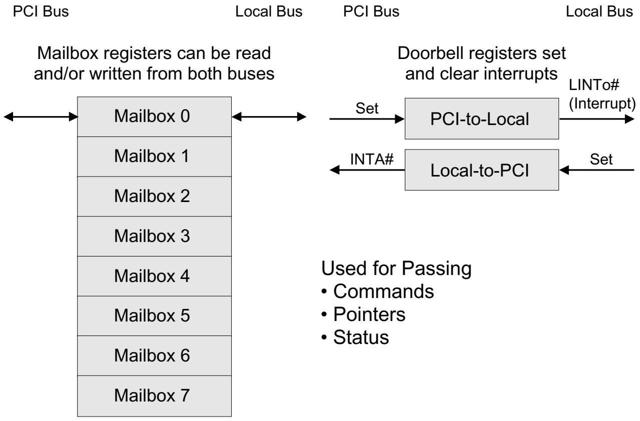


Figure 6-2. Mailbox and Doorbell Message Passing

6.1.6.1 Local-to-PCI Doorbell Interrupt

A Local Bus Master can assert a PCI interrupt (INTA#) by writing to the Local-to-PCI Doorbell bits (L2PDBELL[31:0]). The PCI Host processor can read the PCI Doorbell Interrupt Active bit (INTCSR[13]) to determine whether a PCI Doorbell interrupt is pending, and if so (INTCSR[13]=1), read the PCI 9656 Local-to-PCI Doorbell register.

Each Local-to-PCI Doorbell register bit is individually controlled. Local-to-PCI Doorbell register bits can be set only by the Local Bus. From the Local Bus, writing 1 to any bit position sets that bit and writing 0 has no effect. Local-to-PCI Doorbell register bits can be cleared only from the PCI Bus. From the PCI Bus, writing 1 to any bit position clears that bit and writing 0 has no effect.

The PCI Interrupt remains set as long as any of the Local-to-PCI Doorbell register bits are set and the PCI Doorbell Interrupt Enable bit is set and the PCI Interrupt is enabled (INTCSR[9:8]=11b, respectively).

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 (L2PDBELL) (or any Configuration register), the PCI 9656 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 (L2PDBELL) (or any Configuration register), the PCI 9656 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 interrupt output (LINTo#) by writing a one (1) to any of the PCI-to-Local Doorbell bits (P2LDBELL[31:0]). The Local processor can read the Local Doorbell Interrupt Active bit (INTCSR[20]) to determine whether a Local doorbell interrupt is pending, and, if so (INTCSR[20]=1), read the PCI 9656 PCI-to-Local Doorbell register.

Each PCI-to-Local Doorbell register bit is individually controlled. PCI-to-Local Doorbell register bits can be set only by the PCI Bus. From the PCI Bus, writing 1 to any bit position sets that bit and writing 0 has no effect. PCI-to-Local Doorbell register bits can be cleared only 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.*

The Local interrupt remains 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 9656 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 interrupt by performing a PCI Type 0 Configuration write that sets the PCI Built-In Self-Test 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.

The Local interrupt and INTCSR[23] remain set as long as PCIBISTR[6]=1. The Local Bus then resets INTCSR[23] by way of PCIBISTR[6] when the BIST interrupt completes.

***Note:** The PCI 9656 does **not** have an internal BIST.*

6.1.8 DMA Channel *x* Interrupts

A DMA channel can assert a PCI or Local interrupt when done (transfer is complete), or, in DMA Scatter/Gather mode, after a transfer is complete for the current descriptor. The DMA Channel Interrupt Select bit(s) (DMAMODE*x*[17]) selects whether to assert a PCI or Local interrupt (INTA# or LINTo#, respectively). The PCI or Local processor can read the DMA Channel Interrupt Active bit(s) (INTCSR[22 and/or 21]=1) to determine whether a DMA Channel interrupt is pending.

The DMA Channel Done bit(s) (DMACSR*x*[4]) can be used to determine whether an interrupt is one of the following:

- DMA Done interrupt
- Transfer complete for current descriptor interrupt
- DMA Transfer was aborted

The DMA Channel Done Interrupt Enable bit(s) (DMAMODE*x*[10]=1) enables a Done interrupt. In DMA Scatter/Gather mode, the DMA Channel *x* Descriptor Pointer register Channel Interrupt after Terminal Count bit(s) (DMADPR*x*[2]) specifies whether to assert an interrupt at the end of the transfer for the current descriptor.

A DMA Channel interrupt is cleared by writing a one (1) to the Channel Clear Interrupt bit(s) (DMACSR*x*[3]=1).

6.1.9 All Modes PCI SERR# (PCI NMI)

The PCI 9656 asserts an SERR# pulse if parity checking is enabled (PCICR[6]=1) and it detects a data parity error on any of the following:

- Direct Slave writes
- Direct Master reads
- Address parity error on Direct Slave reads or writes

The SERR# output can be enabled or disabled with the SERR# Enable bit (PCICR[8]).

6.1.10 M Mode PCI SERR#

The PCI 9656 also asserts SERR# and sets the Signaled System Error bit (PCISR[14]=1) if the Local Bus responds with TEA# to the PCI 9656 during Direct Slave and/or DMA transfers. The SERR# interrupt assertion can be masked by writing a value of 0 to the TEA# Input Interrupt Mask bit (LMISC1[5]=0).

6.1.11 All Modes PCI PERR# (PCI Parity Error)

If the Parity Error Response bit is set (PCICR[6]=1), the PCI 9656 sets the Master Data Parity Error bit (PCISR[8]=1) when the following conditions are met:

- The PCI 9656 asserted PERR#, or to acknowledge PERR# was asserted
- The PCI 9656 was the Bus Master for the operation in which the error occurred

Regardless of whether PERR# output is enabled (PCICR[6]=1), the PCI 9656 sets the Detected Parity Error bit (PCISR[15]=1) if it detects one of the following conditions:

- The PCI 9656 detected a parity error during a PCI Address phase
- The PCI 9656 detected a data parity error when it is the target of a write
- The PCI 9656 detected a data parity error when performing Master Read operation

6.1.12 M Mode Local Bus TEA# Signal

The Transfer Error Acknowledge (TEA#) signal is a wired-*OR* M mode bus signal that is asserted by a slave device on the Local Bus for one Local Bus Clock cycle.

The PCI 9656 supports TEA# assertion by the MPC850 or MPC860 processor (Local Bus Monitor Timeout logic) to the PCI 9656 when the PCI 9656 is a Local Bus master during a Direct Slave or DMA transfer Data phase. TEA# assertion to the PCI 9656 during a Direct Slave or DMA Data transfer causes the PCI 9656 to assert a new TS# for every TEA# until the Direct Slave Write or DMA FIFO is empty or the required read is complete through the Direct Slave Read or DMA FIFO. Additionally, SERR# is asserted (if enabled) to indicate a systems error to the PCI Bus, if the M Mode TEA# Input Interrupt Mask bit is disabled (LMISC1[5]=1). The PCI 9656 never asserts TEA# when it is a Local Bus master.

The PCI 9656 supports TEA# assertion to the MPC850 or MPC860 Processor memory controller when the PCI 9656 is a slave on the Local Bus during the Data phase of Direct Master IDMA/SDMA transfers.

In M mode, TEA# is enabled if INTCSR[0]=1. The PCI 9656 asserts TEA# when any of the following conditions occur:

- PCI 9656 Received Target Abort bit is set (PCISR[12]=1)
- 256 consecutive Master Retrys to a PCI Target converted to a Received Target Abort (INTCSR[27, 12]=11b and PCISR[12]=1)
- Received Master Abort bit is set (PCISR[13]=1)

If the PCI 9656 asserts TEA#, it stops driving the Local Bus on the next Local Bus Clock cycle.

If any of the three previously described Abort conditions occur, the following examples describe when the PCI 9656 asserts TEA# during the following types of transfers:

- For a pending Direct Master Posted Write transfer, TEA# is asserted during a Data phase on the ensuing Direct Master Read and Write accesses
- For an ongoing Direct Master Read/Write transfer, TEA# is asserted during a Data phase of the ongoing Direct Master Read/Write transfer

- For a pending (RETRY# asserted) Direct Master Delayed Read transfer, TEA# is asserted during a Data phase on the ensuing Direct Master Read and Write accesses
- For a DMA access on the PCI Bus while the MPC850 or MPC860 is arbitrating to perform a Direct Master Read or Write transfer, TEA# is asserted during a Data phase of the started Direct Master Read and Write transfer
- For a posted Direct Master write followed by a Direct Master Read transfer, TEA# is asserted during a Data phase of the ensuing Direct Master Read access

Note: *If enabled, LINTo# is immediately asserted upon detection of the Abort conditions that cause the PCI 9656 to assert TEA#.*

The PCI 9656 ignores TEA# assertion by the MPC850 or MPC860 Local Bus Monitor Timeout logic during Direct Master Read and Write IDMA/SDMA transfers.

6.1.13 C and J Modes Local LSERR# (Local NMI)

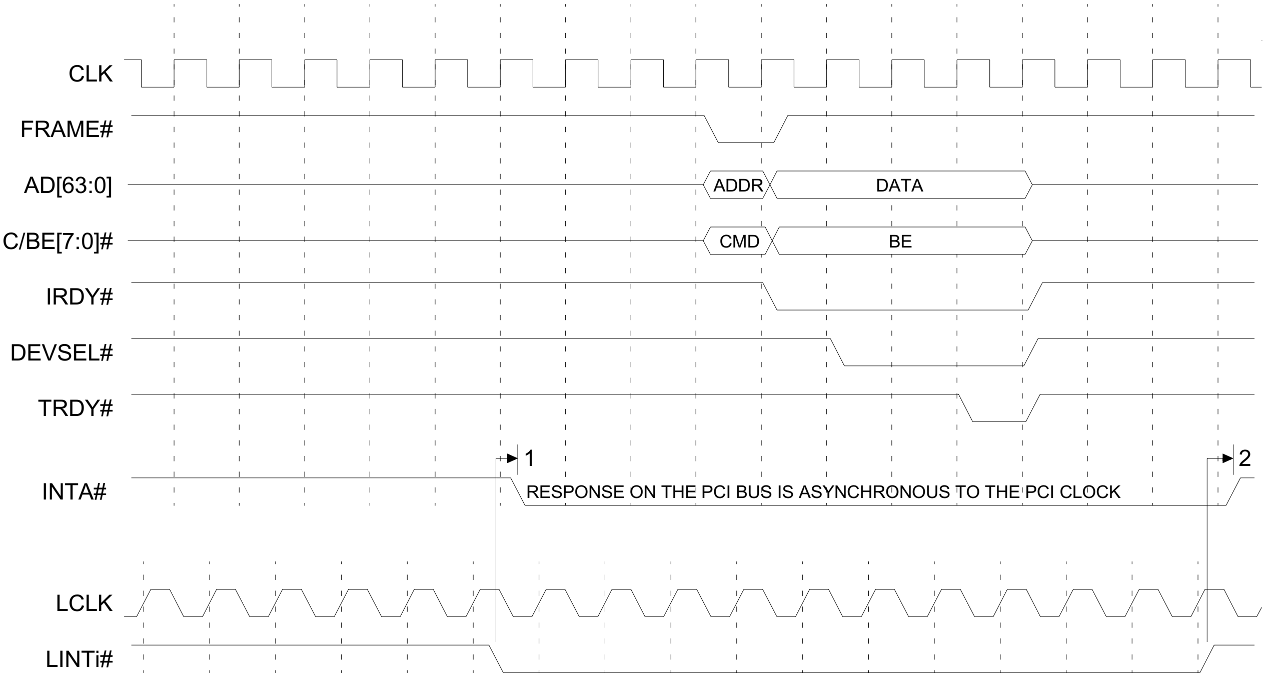
An LSERR# interrupt is asserted if any of the following conditions occur:

- PCI Bus Received Master or Target Abort bit is set (PCISR[13 or 12]=1, respectively)
- Detected Parity Error bit is set (PCISR[15]=1)
- Direct Master Write/Direct Slave Read Local Data Parity Check Error Status bit is set (INTCSR[7]=1)
- Messaging Outbound Free queue overflows
- PCI SERR# assertion if the HOSTEN# pin is asserted low

In C and J modes, LSERR# is always enabled. INTCSR[12, 6, 1:0] can be used to enable or disable LSERR# sources. (Refer to Figure 6-1.) LSERR# is a level output that remains asserted as long as the Interrupt Status or Enable bits are set.

6.1.14 Interrupt Timing Diagram

Timing Diagram 6-1. LINTi# Assertion Causes PCI Interrupt (INTA#) Assertion (All Modes)



Notes: *The PCI cycle has no impact on the LINTi# pin's ability to assert and hold INTA# asserted. Arrows 1 and 2 indicate that INTA# is directly asserted and de-asserted by the LINTi# signal assertion and de-assertion.*

Timing Diagram 6-1 was created using the Timing Designer tool. It is accurate and adheres to its specified protocol(s). This diagram shows transfers and signal transitions, but should not be relied upon to show exactly, on a clock-for-clock basis, where PCI 9656-driven signal transitions will occur.

6.2 USER I/O

The PCI 9656 supports user input and output pins, USERi and USERo (B14 and C14, respectively). Both are multiplexed with other signals. By default, the PCI 9656 configures these pins to be USERi and USERo. 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]). USERi is also used to configure the PCI 9656 for the desired PCI Bus behavior during chip initialization. [Refer to Section 2.4.2 (M mode) or Section 4.4.2 (C and J modes).]

6.3 IDDQEN# MULTI-FUNCTION SHARED PIN – POWER-ON FUNCTION

The IDDQEN# pin is a multiplexed pin with two functions:

1. Identifies the main power present status for the Power Management D_{3cold} function. (Refer to Section 8.1.3, “D_{3cold} Power State Support.”)
2. Used as an IDDQ enable to place the PCI 9656 output buffers into a quiescent state and disable the 1V voltage regulator.

To place the PCI 9656 into the IDDQ state, hold the IDDQEN# input signal (pin A10) in the asserted state. To configure the PCI 9656 for normal operation, hold IDDQEN# in its de-asserted state.

7 INTELLIGENT I/O (I_2O)

7.1 I_2O-COMPATIBLE MESSAGING 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 further details.)

Figures 7-1 and 7-2 illustrate I_2O architecture.

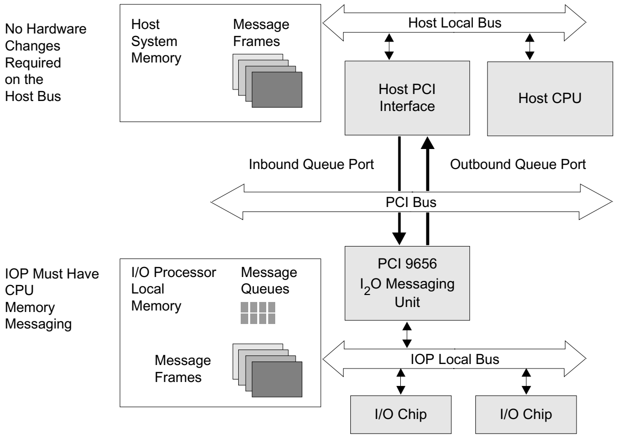


Figure 7-1. Typical I_2O Server/Adapter Card Design

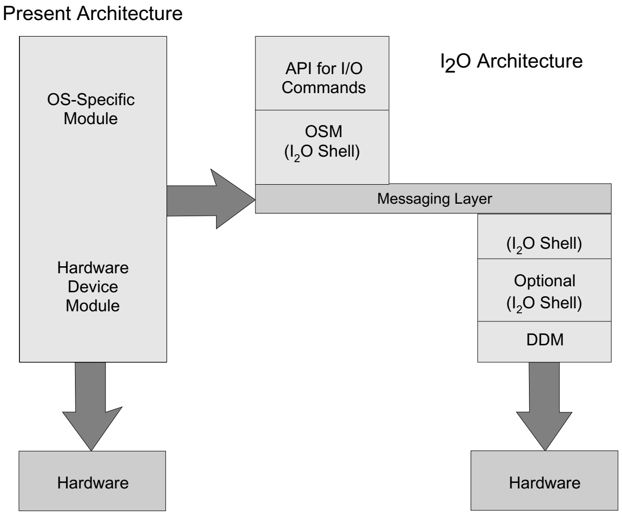


Figure 7-2. Driver Architecture Compared

7.1.1 Inbound Messages

Inbound messages reside in a pool of message frames (minimum 64-byte frames) allocated in the shared Local Bus I/O Processor (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 (MFAs) of available message frames in Local memory. The Inbound Post Queue FIFO holds the MFAs of all currently posted messages in Local Bus IOP memory.

External PCI agents, through the Inbound Queue Port location in PCI Address space, access the inbound circular FIFOs. (Refer to Table 7-2.) The Inbound Queue Port, when read by an external PCI agent, returns an Inbound Free List FIFO MFA. The external PCI agent places the MFA 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 Host 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 MFAs of available message frames in the Host system memory. The Outbound Post Queue FIFO holds the MFAs of all currently posted messages in the Local Bus (IOP) memory.

External PCI agents, through the Outbound Queue Port location in PCI Address space, access the outbound circular FIFOs. (Refer to Table 7-2.) 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 queue base address 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. 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
  (IPHPR and IFHPR, respectively)
- Inbound Post and Free Tail Pointer
  (IPTPR and IFTPR, respectively)
- Outbound Post and Free Head Pointer
  (OPHPR and OFHPR, respectively)
- Outbound Post and Free Tail Pointer
  (OPTPR and OFTPR, respectively)

The PCI 9656 automatically adds the Queue Base Address to the 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 PCI 9656 hardware.

Empty flags are set if the queues are disabled (MQCR[0]=0), or the head and tail pointers are equal. This occurs independently of how the head and tail pointers are set.

An empty flag is cleared, signifying not empty, only if the queues are enabled (MQCR[0]=1) and the 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.) Writes to a FIFO occur at the head of the FIFO and reads occur from the tail. Head and tail pointers are incremented by the Local processor or PCI 9656 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 PCI 9656 automatically wraps the pointers that it maintains. The IOP software must wrap the pointers that it maintains. When 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 to the start of the message frame. Each outbound MFA is specified as the offset from Host memory location 00000000h 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. The IOP allocates and initializes inbound message frames in shared IOP memory, using any suitable memory-allocation technique. The 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 wherever possible to ensure efficient use of the PCI Bus for message passing. (Refer to *I_2O r1.5* for further details about message passing implementation.)

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 (IFHPR). The Local processor must then increment the IFHPR 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). If the FIFO is empty (no free inbound message frames are currently available, head and tail pointers are equal), the PCI 9656 returns -1 (FFFFFFFFh). If the FIFO is not empty (head and tail pointers are not equal), the PCI 9656 reads the MFA pointed to by the Queue Base register + Inbound Free Tail Pointer register (IFTPR), returns its value, and increments the IFTPR 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 MFA to the Inbound Queue Port Address (40h). When the port is written, the PCI 9656 writes the MFA to the Inbound Post Queue FIFO location pointed to by the Queue Base register + FIFO Size + Inbound Post Head Pointer register (IPHPR). After the PCI 9656 writes the MFA to the Inbound Post Queue FIFO, it increments the IPHPR register.

The Inbound Post Tail Pointer register (IPTPR) points to the Inbound Post Queue FIFO location that 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 IPTPR register.

The PCI 9656 asserts a Local interrupt when the Inbound Post Queue FIFO is not empty. The Inbound Post Queue Interrupt Not Empty bit (QSR[5]) indicates the interrupt status. The interrupt is cleared when the Inbound Post Queue FIFO is empty. The Inbound Post Queue Interrupt Not Empty Mask bit can mask the interrupt (QSR[4]=1).

From the time a PCI Write transaction is received, Direct Slave accesses to the PCI 9656 are issued a Retry, until the data is written in Local memory and the Inbound Post Head Pointer register (IPHPR) is incremented.

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 MFA to the Outbound Post Queue FIFO location pointed to by the Queue Base register + Outbound Post Head Pointer register (OPHPR) + (2 * FIFO Size). The Local processor should then increment the OPHPR register.

A PCI master can obtain the MFA of the oldest posted message by reading the Outbound Queue Port Address (44h). If the FIFO is empty (no more outbound messages are posted, head and tail pointers are equal), the PCI 9656 returns -1 (FFFFFFFFh). If the Outbound Post Queue FIFO is not empty (head and tail pointers are not equal), the PCI 9656 reads the MFA pointed to by the Queue Base register + (2 * FIFO Size) + Outbound Post Tail Pointer register (OPTPR), returns its value, and increments the OPTPR register.

The PCI 9656 asserts a PCI interrupt when the Outbound Post Head Pointer register (OPHPR) is not equal to the Outbound Post Tail Pointer register (OPTPR). The Outbound Post Queue Interrupt bit (OPQIS[3]) indicates the interrupt status. When the pointers become equal, both the interrupt and OPQIS[3] 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 Interrupt Mask register can mask the interrupt (OPQIM[3]=1).

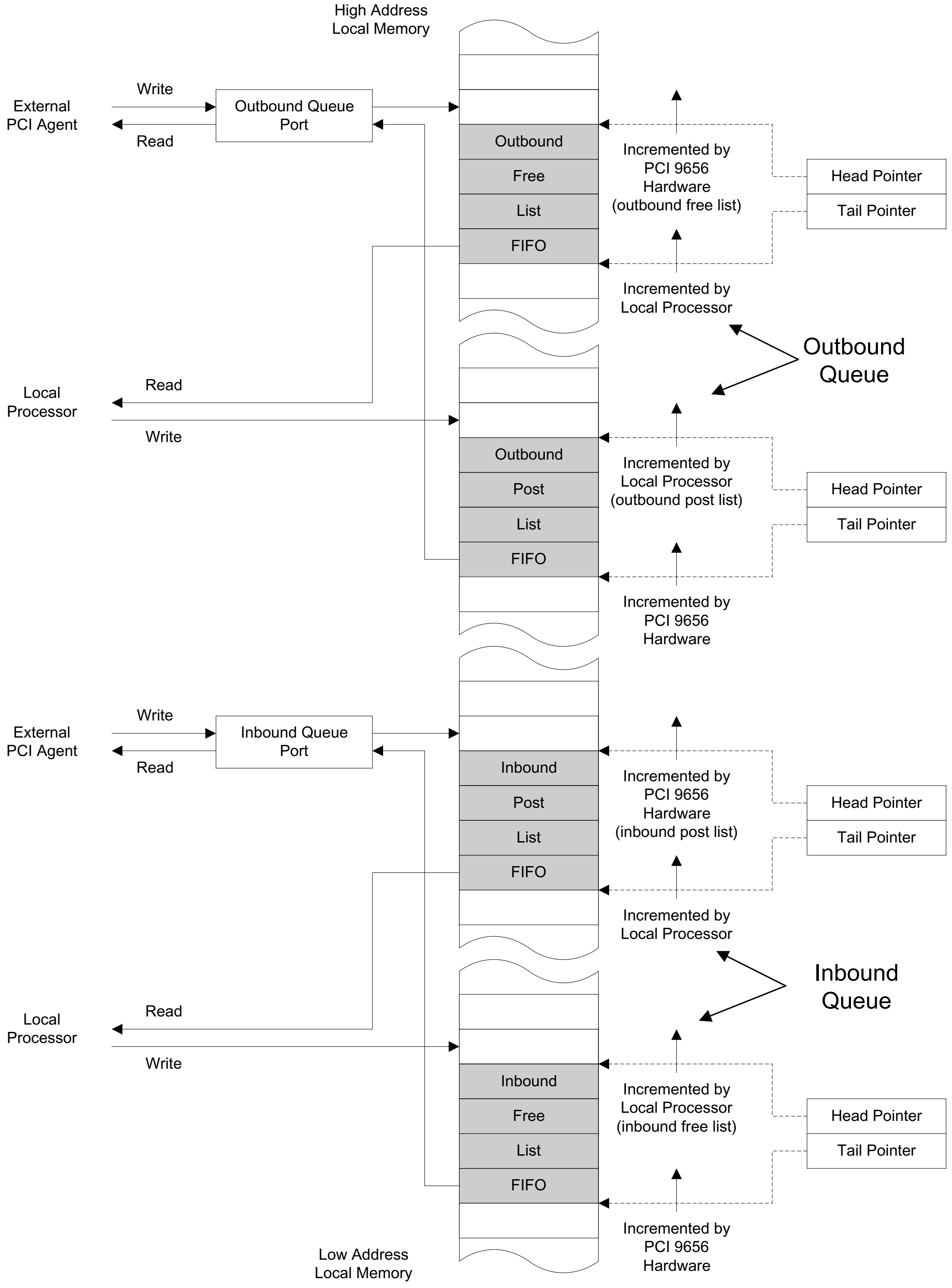


Figure 7-3. Circular FIFO Operation

7.1.7 Outbound Post Queue

To reduce read latency, prefetching from the tail of the queue occurs when 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 when 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 an MFA to the Outbound Queue Port Address (44h). When the port is written, the PCI 9656 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 (OFHPR). After the PCI 9656 writes the MFA to the Outbound Free List FIFO, it increments the OFHPR 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 at least one additional outbound free MFA 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 (OFTPR). After the IOP reads the MFA, it must increment the OFTPR 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 PCI 9656 also checks for Outbound Free List FIFO overflows. When the head pointer is incremented and becomes equal to the tail pointer, the Outbound Free List FIFO is full, and the PCI 9656 asserts a Local LINTo# (M mode) or LSERR# (C and J modes) interrupt. The interrupt is recorded in the Outbound Free Queue Overflow Interrupt Full bit (QSR[7]).

From the time the PCI Write transaction is received until the data is written into Local memory and the Outbound Free Head Pointer register (OFHPR) is incremented, any Direct Slave access to the PCI 9656 is issued a Retry.

7.1.10 I_2O Enable Sequence

To enable I_2O, the Local processor should perform the following:

- Initialize Space 1 address and range (minimum 1,024 bytes)
- 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]=1)
- Set the Local Init Status bit to “done” (LMISC1[2]=1)
- Disable all Direct Slave prefetch mechanisms (LBRD0[8] for Space 0, LBRD1[9] for Space 1, and/or LBRD0[9] for Expansion ROM)

Note: *The serial EEPROM must* ***not*** *set the Local Init Status bit so that the PCI 9656 issues Retrys to all PCI accesses until the Local Init Status bit is set to “done” by the Local processor.*

Enabling I_2O Decode causes the remapping of resources for use in I_2O mode (QSR[0]=1). When set, all Memory-Mapped Configuration registers and Space 1 share the PCIBAR0 register. PCI accesses to PCIBAR0 offset 00h to FFh result in accesses to the PCI 9656 internal Configuration registers.

Accesses above PCIBAR0 offset FFh result in Local Space accesses, beginning at offset 100h from the Remap PCI Address to Local Address Space 1 into the Remap PCIBAR3 Base Address to Local Address Space 1 Base Address bits (LAS1BA[31:4]). Therefore, space located at offset 00h to FFh from LAS1BA is not addressable from the PCI Bus using PCIBAR0.

Note: *Because PCI accesses to PCIBAR0 offset 00h to FFh result in internal Configuration accesses, the Inbound Free MFA 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	PCI 9656 hardware
Inbound Post List FIFO	Inbound Queue Port (Host write)	No	Yes, when Port is written	PCI 9656 hardware	Local processor
Outbound Post List FIFO	Outbound Queue Port (Host read)	Yes, when FIFO is not empty	No	Local processor	PCI 9656 hardware
Outbound Free List FIFO	Outbound Queue Port (Host write)	No	Yes, (LINTo#/LSERR#) when FIFO is full	PCI 9656 hardware	Local processor

8 PCI POWER MANAGEMENT

8.1 OVERVIEW

PCI Power Mgmt. r1.1 provides a standard mechanism for operating systems to control add-in cards for Power Management. The specification defines four PCI functional power states – D_0, D_1, D_2, and D_3. The D_0 and D_3 power states are required, while the D_1 and D_2 power states are optional. The D_0 power state represents the highest power consumption, and the D_3 power state represents the least.

- **D_0 (Uninitialized)** – Enters this state from power-on reset or from the D_{3hot} power state. Supports PCI Configuration cycles only and retains Power Management Event (PME) context.
- **D_0 (Active)** – All functions active.
- **D_1** – Uses less power than the D_0 power state, and more than the D_2 power state. 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 9656. This state is ***not*** supported with PCI clock frequencies above 33 MHz.

  In this state, the PCI 9656 supports PCI Configuration cycles if the PCI clock is running (memory, I/O, bus mastering, and interrupts are disabled). It also supports the Wake-Up Event function, but not standard PCI interrupts.
- **D_{3hot}** – Uses lower power than any other state. The PCI 9656 supports PCI Configuration cycles if the PCI clock is running. Supports the Wake-Up Event function, but not standard PCI interrupts. When programmed to the D_0 power state while in the D_{3hot} power state, an internal soft reset occurs. In this state, the PCI Bus drivers are disabled. PME context is retained during this soft reset.
- **D_{3cold}** – No power. All context is lost in this state, except for pins and logic that are powered by Card_V_{AUX} and 2.5V_{AUX}, respectively. (Refer to Section 8.1.3.)

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-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-driven bus activity. V_{CC} is applied to all devices on the bus.
- **B_3 (Off)** – Power to the bus is switched off. PME-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).

Device power 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 9656 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. A PCI BIOS is able to identify a New Capability function support when PCISR[4]=1. This bit is writable from the Local Bus, and readable from the PCI and Local 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 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 cleared to 0h. The default value for the PCI 9656 is 48h (Hot Swap).

For the PCI 9656 to change the power state and assert PME#, a PCI or Local master 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], respectively) are used by the Local Host to identify power state support
- PME_Support bits (PMC[15:11]) are used by the PCI 9656 to identify the PME# Support corresponding to a specific power state (PMCSR[1:0])

The Local Host then sets the PME_Status bit (PMCSR [15]=1) and the PCI 9656 asserts PME#. To clear the bit, the PCI Host must write 1 to the PME_Status bit (PMCSR[15]=1). To disable the PME# interrupt signal, either Host can write 0 to the PME_En bit (PMCSR[8]=0).

LINTo# is asserted each time the power state in PMCSR[1:0] changes. The transition from the D_{3hot} power state to the D_0 power state causes a soft reset. A soft reset should be initiated only from the PCI Bus because the Local Bus interface is reset during a soft reset. In Adapter mode (HOSTEN#=1), the PCI 9656 issues LRESET# and resets Local Configuration and Messaging Queue registers, Local Bus logic, PCI and Local DMA logic, and all FIFOs. [Refer to Section 3.1.1.3 (M mode) or Section 5.1.1.3 (C and J modes).] After power-on reset is complete, the PCI 9656 reloads its original values, overwriting the PCI Interrupt Line register (PCIILR) value (IRQ assignment) contents. The driver must save the PCIILR value before entering the Power Management state for restoration. To allow LINTo# to assert, set the LINTo# Enable and Power Management Interrupt Enable bits (INTCSR[16, 4]=11b, respectively) 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 combination. For Data_Select values from 8 to 15, the Data_Scale bits always return a zero (PMCSR[14:13]=00b).

PMDATA provides static operating data, *such as* power consumed or heat dissipation.

8.1.2 66 MHz PCI Clock D_2 Power State Support

The PCI 9656 provides full support for the D_2 power state at 33 MHz or less 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 9656 ***cannot*** support the D_2 power state at 66 MHz. To do that, the PCI 9656 requires an external control to avoid enabling the D_2 Power Management feature at a 66 MHz clock frequency. Default booting of the PCI 9656 sets D_2_Support to a disabled state (PMC[10]=0). 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 it, and enable D_2 Power Management support (PMC[10]=1) by way of the register access when the M66EN PCI connector pin is sampled false, de-asserted.

8.1.3 D_{3cold} Power State Support

The PCI 9656 provides full support for the D_{3cold} power state with PME# assertion and register contents storage. The PCI 9656 has all pins required by *PCI Power Mgmt. r1.1*. Special attention is necessary for the following pins:

- **2.5V_{AUX}** – Auxiliary power input pin routed to the D_{3cold} support core logic.
- **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 input pin whose signal is provided by add-in card Auxiliary Power Routing (refer to *PCI Power Mgmt. r1.1,* Figure 12) to enable the D_{3cold} Power Management Event assertion feature within the PCI 9656 silicon.
- **PME#** – Optional open drain (OD), asserted low signal intended to be driven low by the PCI 9656 to request a change in its current Power Management state and/or a Power Management event is requested by way of PMEREQ# assertion. The PCI 9656 requires external logic to avoid unexpected Wake-Up events that might occur when 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 pin used to request a Wake-Up event when the add-in card is in the D_{3cold} power state.
- **IDDQEN#** – Input signal providing main power status to the PCI 9656 D_{3cold} Power Management logic. This pin tracks the main power supply voltage to identify power-down events to activate the PCI 9656 D_{3cold} module and should be directly connected to the V_{CORE} voltage power supply (V_{CORE} = 2.5V). The IDDQEN# input buffer is custom designed with a unique voltage threshold that guarantees power sequence for the D_{3cold} module. The threshold during rising transition is Vth = 2.1V, and during falling transition is Vth = 1.9V. Because this pin is shared with other functions during the power-up event, refer to Section 6.3, “IDDQEN# Multi-Function Shared Pin – Power-On Function,” for further details about the function and circuit implementation.

Note: *Card_V_{AUX} powers all signal I/Os used for D_{3cold} Power Management support.*

8.1.4 System Changes Power Mode Example

1. The Host writes to the PCI 9656 Power Management Control/Status register (PMCSR) to change the power states.
2. The PCI 9656 sends a Local interrupt (LINTo#) to a Local CPU (LCPU).
3. The Local CPU has 200 µs to read the Power Management information from the PCI 9656 PMCSR register to implement the power-saving function.
4. After the Local CPU implements the power saving function, the PCI 9656 disables all Direct Slave accesses and PCI Interrupt output (INTA#). In addition, the Power Management driver disables the PCI 9656 Master Enable bit (PCICR[2]=0).

Notes: *In Power-Saving mode, all PCI and Local Configuration cycles are granted.*

The PCI 9656 automatically performs a soft reset to a Local Bus on D_3-to-D_0 power state transitions, then reloads the Configuration register values stored in the serial EEPROM.

8.1.5 Non-D_{3cold} Wake-Up Request Example

1. The add-in card (with a PCI 9656 device installed) is in a powered-down state.
2. The Local CPU performs a write to the PCI 9656 PMCSR register to request a Wake-Up event.
3. As soon as the request is detected, the PCI 9656 drives PME# out to the PCI Bus.
4. The PCI Host accesses the PCI 9656 PMCSR register to disable the PME# output signal and restores the PCI 9656 to the D_0 power state.
5. The PCI 9656 completes the Power Management task by issuing the Local interrupt (LINTo#) to the Local CPU, indicating that the power mode has changed.

THIS PAGE INTENTIONALLY LEFT BLANK.

9 COMPACTPCI HOT SWAP

The PCI 9656 is compliant with *PICMG 2.1 R2.0* requirements for Hot Swap Silicon, including Programming Interface 0 (PI = 0) and support for Precharge Voltage, Early Power, Initially Not Respond, and 64-Bit Initialization.

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 9656 uses five pins – BD_SEL#, CPCISW, ENUM#, LEDon#, and 64EN# – to implement the hardware aspects of Hot Swap functionality. The PCI 9656 uses the Hot Swap Capabilities register to implement the software aspects of Hot Swap.

The PCI 9656 supports the following features specified in *PICMG 2.1 R2.0*, for Hot Swap Silicon:

- *PICMG 2.1 R2.0*-compliant.
- 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** – Designed in accordance with *PICMG 2.1 R2.0*. The Capabilities Pointer is located within standard CSR space. New Capability Functions Support is enabled by PCISR[4].
- Incorporates remaining software connection control resources. Provides ENUM#, Hot Swap switch, and the blue Status LED.
- Early Power support.
- **Incorporates a 1V precharge voltage to the PCI I/O pins** – Incorporates a 1V regulator and precharge resistors. The silicon uses a built-in 1V voltage regulator to precharge the majority of the Hot Swap PCI I/O pins to 1V during the insertion/extraction process (each precharge resistor is individually switched in and out). Each Hot Swap PCI I/O pin has its own built-in 25KΩ precharge resistor, which is always connected to the 1V regulator. The silicon also switches in and out individual 10KΩ precharge resistors to precharge the ENUM#, INTA#, PME#, GNT0#/REQ#, REQ0#/GNT#, and RST# Hot Swap pins to V_{IO} during the insertion/extraction process.
- 64-bit Initialization support, using the 64EN# and/or REQ64# pins.

9.1.1 Silicon Behavior during Initialization on PCI Bus

The PCI 9656 supports an initialization-time PCI Initially Not Respond option that can be used by CompactPCI peripheral adapter cards designed for live insertion. [Refer to Section 2.4.2 (M mode) or Section 4.4.2 (C and J modes) for further details.] This option can be useful during PCI Bus initialization because, as *PICMG 2.1 R2.0*, Section 3.1.10, states, “it is far preferable for boards that are not ready for PCI accesses to Initially Not Respond.”

9.1.2 Configuration

In Adapter mode (HOSTEN#=1), the PCI 9656 supports *PICMG 2.1 R2.0* Programming Interface 0 (PI = 0). (Refer to *PICMG 2.1 R2.0*, Figures 29 and 30.) In Host mode (HOSTEN#=0), the PCI 9656 does ***not*** support *PICMG 2.1 R2.0* Programming Interface 0 (PI = 0). 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 9656. (Refer to *PICMG 2.1 R2.0* for further 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

For proper PCI 9656 operation during the Hot Swap insertion/extraction process, all PCI 9656 silicon and external components that affect booting of the PCI 9656 (*such as*, external pull-up and pull-down resistors) must be powered by Early Power. This includes the 2.5V_{AUX}, Card_V_{AUX}, V_{CORE}, V_{IO}, and V_{RING} pins. The PRESENT_DET pin is connected to ground to indicate no PCI Power Management D_{3cold} support because the CompactPCI backplane does ***not*** support PCI Power Management. Connect 2.5V_{AUX} to the V_{CORE} power supply, and Card_V_{AUX} to the V_{RING} power supply.

BD_SEL#, one of the shortest pins from the CompactPCI backplane, is used to enable/disable Back End Power. 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 9656 uses the BD_SEL# signal to either place in a high-impedance state or drive to logic 0 (ground) all Local output buffers during the insertion/extraction process. When BD_SEL# is de-asserted, all PCI and Local Bus signals are floating, with exception to LRESET# and LEDon#, which are driven to logic 0. In addition, the PCI 9656 dynamically connects the 1V and V_{IO} precharge resistors to all required PCI I/O buffers when the BD_SEL# pin is asserted. (Refer to Section 6.3, "IDDQEN# Multi-Function Shared Pin – Power-On Function," for further details.)

Note: *With BD_SEL# de-asserted, driving the Local signals to logic 0 (ground) is a legitimate practice during the insertion/extraction process because none of the attached components are powered on.*

To provide proper PCI 9656 operation, a pull-up resistor must be provided to the BD_SEL# pin or add-in card that connects to Early Power. The ENUM#, INTA#, PME#, GNT0#/REQ#, REQ0#/GNT#, and RST# PCI signals are internally precharged to V_{IO}. All other PCI Bus signals are internally precharged to 1V.

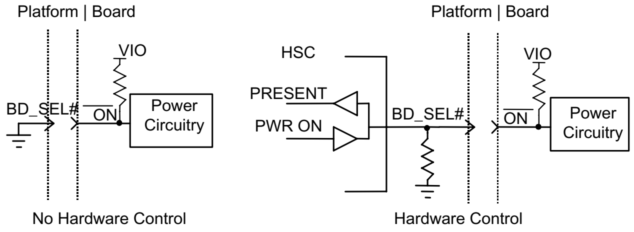


Figure 9-1. Redirection of BD_SEL#

9.2.1.2 Board Healthy

A second radial signal (HEALTHY#) 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. On platforms implementing this signaling, radially route these signals to a Hot Swap Controller.

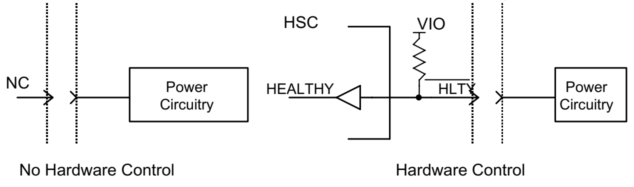


Figure 9-2. Board Healthy

9.2.1.3 Platform Reset

Reset (PCI_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 (Healthy) is reached, and they must honor the backplane reset. The Local board reset (Local_PCI_RST#) must be the logical *OR* of these two conditions. Local_PCI_RST# is connected to the PCI 9656 RST# input pin.

During a precharge voltage and platform reset, in insertion and extraction procedures, all PCI I/O buffers must be in a high-impedance state. The PCI 9656 supports this condition when the Host RST# is asserted. With full contact of the add-in card to the backplane, BD_SEL# is asserted, which ensures that the PCI 9656 asserts the LRESET# signal to complete a Local board reset task.

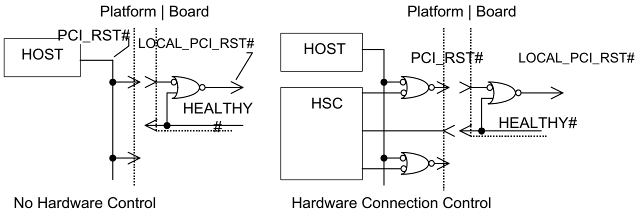


Figure 9-3. PCI Reset

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 Register accesses from the PCI or Local Bus.

These resources consist of four elements:

- ENUM# driven asserted 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 Status LED

A microswitch (switch), located in the CompactPCI Hot Swap board card-ejector mechanism, is used to signal impending board removal or position change. (Refer to *PICMG 2.1 R2.0*, Section 2.4, for detailed information regarding the insertion/extraction process.) When the switch is activated, it is necessary to wait for the LED to turn on, indicating that it is okay to remove the board. The PCI 9656 implements separate control logic for the microswitch and blue Status LED in two different pins (CPCISW and LEDon#, respectively).

When the ejector is opened or closed, the switch bounces for a time. The PCI 9656 uses internal de-bounce 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 panel of the CompactPCI Hot Swap board, is turned on 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 turned on by the hardware until the hardware connection process completes. The amount of time the LED remains on during the insertion process is the sum of the RST# pin assertion, plus the amount of time the Host takes to logically include the board and turn off the LED. The period of time the LED is on may be short. The LED remains *OFF* until the software uses it to indicate extraction is once again permitted.

The PCI 9656 uses an open-drain (OD) type output buffer to sink current for the external LED. The LED state is set by the LED Software On/Off Switch bit (HS_CSR[3]). The PCI 9656 asserts LEDon# during PCI resets (RST# asserted). The CPCISW input signal indicates ejector handle changes by turning on the blue Status LED. The appropriate status bit is set (HS_CSR[7 or 6]=1).

If the microswitch position frequently changes, ENUM# remains asserted, despite subsequent de-assertions.

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 drain (OD) 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 Swap system driver on the system Host manages the ENUM# sensing. Hot Swap boards assert ENUM# until serviced by the Hot Swap system driver.

The PCI 9656 evaluates the CPCISW input signal at power up, and clears the ENUM# Status – Insertion bit (HS_CSR[7]) to 0 if the CPCISW input is high (switch open), or to 1 if the CPCISW input is low (switch closed). When a board is inserted into the system and comes out of reset, the PCI 9656 acknowledges the ejector switch state. If this switch is open (ejector handle closed), the PCI 9656 asserts the ENUM# interrupt and sets the ENUM# Status – Insertion bit to 1 (HS_CSR[7]=1). 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 9656 acknowledges the ejector handle is open, asserts the ENUM# interrupt, and sets the ENUM# Status – Extraction bit (HS_CSR[6]=1). 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 64EN# (64-Bit Enable)

The 64EN# signal indicates whether the system is a 64- or 32-bit CompactPCI Bus. It is a static signal and should not change during normal operation. The PCI 9656 starts sensing this signal as soon as power is supplied. 64EN# has the following characteristics and capabilities:

- 64EN# true (asserted low) indicates a 64-bit CompactPCI Bus. The PCI 9656 immediately tri-states all CompactPCI signals until it communicates with the system.
- 64EN# false (de-asserted high) indicates a 32-bit CompactPCI Bus. The PCI 9656 re-configures itself to work in a 32-bit system and sets the unused PCI pins (ACK64#, AD[63:32], C/BE[7:4]#, PAR64, and REQ64#) to known values. This avoids the need for external pull-up resistors for the unused signals.
- The 64EN# pin eliminates the need for external 64-bit detection logic (on the REQ64# pin), while ensuring that the pins for the 64-bit PCI extension are correctly configured immediately after Early Power-On during Hot Swap insertion instead of RST# de-assertion.

9.2.2.4 Hot Swap Control/Status Register (HS_CSR)

The PCI 9656 directly supports Hot Swap, with a Control/Status register that 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 Swap 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#.

9.2.2.5 Hot Swap Capabilities Register

Hot Swap ID. Bits [7:0] (HS_CNTL[7:0]; PCI:48h, LOC:188h). These bits are set to a default value of 06h. To disable Hot Swap capabilities, clear this register to 0h.

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 cleared to 0h if this is the last capability in the structure. Otherwise, these bits must contain the default value of 4Ch.

Control/Status. Bits [23:16] (HS_CSR[7:0]; PCI:4Ah, LOC:18Ah). This 8-bit register is defined in Table 9-1.

31 24	23 16	15 8	7 0
Reserved	Control/ Status	Next_Cap Pointer	Hot Swap ID (06h)

Figure 9-4. Hot Swap Capabilities

Table 9-1. Hot Swap Control/Status

Bit	Description
23	ENUM# Status – Insertion. Writing 1 reports the ENUM# assertion for insertion process.
22	ENUM# Status – Extraction. Writing 1 reports the ENUM# assertion for removal process.
21:20	Programming Interface 0 (PI = 0).
19	LED Software On/Off Switch. Writing 1 asserts the LEDon# signal. Writing 0 de-asserts the LEDon# signal.
18	***Reserved.***
17	ENUM# Interrupt Mask (EIM). Writing 0 enables interrupt assertion. Writing 1 masks interrupt assertion.
16	***Reserved.***

THIS PAGE INTENTIONALLY LEFT BLANK.

10 PCI VITAL PRODUCT DATA (VPD)

10.1 OVERVIEW

VPD provides optional bits that can be used to identify and track a device. The bits can be set so that each device can be unique. In the original PCI specification, Device ID, Vendor ID, Revision ID, Class Code ID, and Subsystem Vendor ID were required in the Configuration Space Header and for basic device identification and configuration. Although this information allows a device to be configured, it is not sufficient to allow each device to be uniquely identified. The VPD optional information capability enables new support tools and reduces the cost of computer ownership.

In *PCI r2.2*, the VPD function defines a storage device and access method for VPD, as well as defining the Read-Only and Read/Write bits. The PCI 9656 stores the VPD in a serial EEPROM and access is through the New Capabilities function of the PCI Configuration Space.

10.2 VPD CAPABILITIES REGISTERS

The following sections describe the VPD Capabilities registers. As illustrated in Figure 10-1, they include the VPD Control and Data registers.

10.2.1 VPD Control Register

The VPD Control register is 32 bits wide and is documented as three smaller registers – VPD ID, Next_Cap Pointer, and VPD Address. (Refer to Figure 10-1 and Section 11.3, “PCI Configuration Registers.”)

VPD ID (PVPDID[7:0]; PCI:4Ch, LOC:18Ch). PCI-SIG assigned these bits a value of 03h. The VPD ID is hardwired.

Next_Cap Pointer (PVPD_NEXT[7:0]; PCI:4Dh, LOC:18Dh). These bits point to the next New Capability structure. The PCI 9656 defaults to 0h. Because VPD is the last feature of the New Capability structure, this field is cleared to 0h. Bits [1:0] are ***reserved*** by *PCI r2.2*, and should be cleared to 00b.

VPD Address (PVPDAD[14:0]; PCI:4Eh, LOC:18Eh). These bits specify the VPD byte address to be accessed. All accesses are 32 bits wide. For VPD writes, the byte address must be Lword-aligned (PVPDAD[1:0]=00b). For VPD reads, the byte address must be word-aligned (PVPDAD[0]=0). Bits [14:9] are ignored.

(PVPDAD[15]; PCI:4Eh, LOC:18Eh). The F bit of the VPD Address register controls the direction of the next VPD cycle and indicates when the VPD cycle is complete. For Write cycles, the 4 bytes of data are first written into the VPD Data bits, after which the VPD Address is written at the same time the F bit is set to 1. The F bit is cleared when the serial EEPROM Data transfer completes. For Read cycles, the VPD Address is written at the same time the F bit is cleared to 0. The F bit is set when 4 bytes of data are read from the serial EEPROM.

10.2.2 VPD Data Register

The VPD Data register is 32 bits wide and is documented as a single 32-bit register. (Refer to Figure 10-1.)

VPD Data (PVPDATA[31:0]; PCI:50h, LOC:190h). The PVPDATA register is used to read/write data to/from the VPD serial EEPROM. It is not, however, a pure read/write register. The data read from this register is the data read during the last VPD Read operation. The data written to this register is the data written to the serial EEPROM during a VPD Write operation. The register’s words are stored in the serial EEPROM, in Big Endian order, beginning at the serial EEPROM word address specified by the VPD Address bits (PVPDAD[8:1]). Four bytes are always transferred between the register and the serial EEPROM.

Register Bit Range	31	30 16	15 8	7 0
VPD Control Register	F (PVPDAD [15])	VPD Address (PVPDAD[14:0])	Next_Cap Pointer (0h) (PVPD_NEXT[7:0])	VPD ID (03h) (PVPDID[7:0])
VPD Data Register	VPD Data (PVPDATA[31:0])			

Figure 10-1. VPD Capabilities

10.3 VPD SERIAL EEPROM PARTITIONING

To support VPD, the serial EEPROM is partitioned into Read-Only and Read/Write portions. The boundary between Read-Only and Read/Write is set with the Serial EEPROM Location Starting at Lword Boundary for VPD Accesses bits (PROT_AREA[6:0]).

10.4 SEQUENTIAL READ-ONLY

The first 1536 bits, 192 bytes of the serial EEPROM contain Read-Only information. After power-on, the Read-Only portion of the serial EEPROM is loaded into the PCI 9656, using the serial EEPROM's Sequential Read protocol. Sequential words are read by holding EECS asserted, following the issuance of a serial EEPROM Read command.

10.5 RANDOM READ AND WRITE

The PCI 9656 can read and write the read/write portion of serial EEPROM, using the VPD function. It can also read the Read-Only portion of the serial EEPROM. The writable portion of the serial EEPROM starts at the Lword specified by PROT_AREA[6:0] and continues to the top of the serial EEPROM. The Serial EEPROM Location Starting at Lword Boundary for VPD Accesses bits (PROT_AREA[6:0]) designate this portion. This register is loaded at 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 Data, followed by Write Data (two 16-bit writes)
- 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 VPD write to the serial EEPROM, the following steps are necessary:

1. Disable EEDO Input (CNTRL[31]=0, default).
2. Change the write-protected serial EEPROM address in PROT_AREA[6:0] to the desired Lword location. Value of 0h makes the entire serial EEPROM writable.
3. Write desired data into the PVPDATA register.
4. Write the serial EEPROM destination address in the PVPDAD register, and the F bit to 1 (PVPDAD[15]=1). PVPDAD[1:0] must be 00b (address is Lword-aligned).
5. Poll the F bit until it changes to 0 (PVPDAD[15]=0), to ensure that the write completes.

To perform a VPD read from serial EEPROM, the following steps are necessary:

1. Disable EEDO Input (CNTRL[31]=0, default).
2. Write the serial EEPROM destination address in the PVPDAD register, and the F bit to 0 (PVPDAD[15]=0). PVPDAD[0] must be 0 (address is word-aligned).
3. Poll the F bit until it changes to 1 (PVPDAD[15]=1) to ensure that the Read data is available.
4. Read back the PVPDATA register to obtain the requested data.

11 REGISTERS

11.1 SUMMARY OF REGISTER DIFFERENCES

This section summarizes how the PCI 9656 registers differ from those in the PCI 9054. Refer to the subsequent sections for a full explanation of each PCI 9656 register.

Table 11-1. Summary of PCI 9656 and PCI 9054 Register Differences

PCI Offset	Local Offset	Register	Bit	Description
06h	06h	PCI Status	5	66 MHz-Capable.
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 Delayed Write Mode.
			5	Direct Slave Write FIFO Full Condition.
			7:6	***Reserved***.
100h	1A0h	PCI Arbiter Control	0	PCI Arbiter Enable.
			1	PCI 9656 High Priority.
			2	Early Grant Release.
			3	PCI Arbiter Parking on PCI 9656.
			31:4	***Reserved.***
104h	1A4h	PCI Abort Address	31:0	PCI Abort Address.
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	Software Reset when HOSTEN#=1 or 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.***
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.
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.
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.

Table 11-1. Summary of PCI 9656 and PCI 9054 Register Differences (Continued)

PCI Offset	Local Offset	Register	Bit	Description
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.
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.
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).
42h	182h	Power Management Capabilities	2:0	Value is loadable by way of serial EEPROM.
			15:9	Value is loadable by way of serial EEPROM.
44h	184h	Power Management Control/Status	14:8	Value is loadable by way of serial EEPROM.
47h	187h	Power Management Data	7:0	Value is loadable by way of serial EEPROM.

11.2 REGISTER ADDRESS MAPPING

Table 11-2. PCI Configuration Register Address Mapping

PCI Configuration Register Address	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9656 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	PCI Device ID		PCI Vendor ID		Local	Yes
04h	04h	PCI Status		PCI Command		Yes	No
08h	08h	PCI Class Code			PCI Revision ID	Local	Yes
0Ch	0Ch	PCI Built-In Self-Test	PCI Header Type	PCI Bus Latency Timer	PCI Cache Line Size	Yes	No
10h	10h	PCI Base Address for Memory Accesses to Local, Runtime, DMA, and Messaging Queue Registers (PCIBAR0)				Yes	No
14h	14h	PCI Base Address for I/O Accesses to Local, Runtime, DMA, and Messaging Queue Registers (PCIBAR1)				Yes	No
18h	18h	PCI Base Address for Accesses to Local Address Space 0 (PCIBAR2)				Yes	No
1Ch	1Ch	PCI Base Address for Accesses to Local Address Space 1 (PCIBAR3)				Yes	No
20h	20h	PCI Base Address 4 ***(Reserved)***				No	No
24h	24h	PCI Base Address 5 ***(Reserved)***				No	No
28h	28h	PCI Cardbus Information Structure (CIS) Pointer ***(Not supported)***				No	No
2Ch	2Ch	PCI Subsystem ID		PCI Subsystem Vendor ID		Local	Yes
30h	30h	PCI Base Address for Local Expansion ROM				Yes	No
34h	34h	***Reserved***			New Capability Pointer	Local	No
38h	38h	***Reserved***				No	No
3Ch	3Ch	PCI Maximum Latency	PCI Minimum Grant	PCI Interrupt Pin	PCI Interrupt Line	Yes	Yes
40h	180h	Power Management Capabilities		Power Management Next Capability Pointer	Power Management Capability ID	Local [31:21, 19:8]	Yes [31:25, 18:16]
44h	184h	Power Management Data	PMCSR Bridge Support Extensions ***(Reserved)***	Power Management Control/Status		PCI [15, 12:8, 1:0], Local [31:24, 15:8, 1:0]	Yes [31:24, 14:8]
48h	188h	***Reserved***	Hot Swap Control/Status	Hot Swap Next Capability Pointer	Hot Swap Control (Capability ID)	PCI [23:22, 19, 17], Local [23:22, 17, 15:0]	Yes [15:0]
4Ch	18Ch	F	PCI VPD Address	PCI VPD Next Capability Pointer	PCI VPD Capability ID	PCI [31:16], Local [31:8]	No
50h	190h	PCI VPD Data				Yes	No

Notes: *Refer to PCI r2.2 for definitions of the registers listed in this table.*
Where Writable bit numbers are not listed, refer to the individual register descriptions to determine which bits are writable.

Table 11-3. Local Configuration Register Address Mapping

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9656 family and to ensure compatibility with future enhancements, write 0 to all unused bits.				PCI/Local Writable	Serial EEPROM Writable
		31 24	23 16	15 8	7 0		
00h	80h	Direct Slave Local Address Space 0 Range				Yes	Yes
04h	84h	Direct Slave Local Address Space 0 Local Base Address (Remap)				Yes	Yes
08h	88h	Mode/DMA Arbitration				Yes	Yes
0Ch	8Ch	Local Miscellaneous Control 2	Serial EEPROM Write-Protected Address Boundary	Local Miscellaneous Control 1	Big/Little Endian Descriptor	Yes	Yes
10h	90h	Direct Slave Expansion ROM Range				Yes	Yes
14h	94h	Direct Slave Expansion ROM Local Base Address (Remap) and BREQo Control				Yes	Yes
18h	98h	Local Address Space 0/Expansion ROM Bus Region Descriptor				Yes	Yes
1Ch	9Ch	Local Range for Direct Master-to-PCI				Yes	Yes
20h	A0h	Local Base Address for Direct Master-to-PCI Memory				Yes	Yes
24h	A4h	Local Base Address for Direct Master-to-PCI I/O Configuration				Yes	Yes
28h	A8h	PCI Base Address (Remap) for Direct Master-to-PCI Memory				Yes	Yes
2Ch	ACh	PCI Configuration Address for Direct Master-to-PCI I/O Configuration				Yes	Yes
F0h	170h	Direct Slave Local Address Space 1 Range				Yes	Yes
F4h	174h	Direct Slave Local Address Space 1 Local Base Address (Remap)				Yes	Yes
F8h	178h	Local Address Space 1 Bus Region Descriptor				Yes	Yes
FCh	17Ch	Direct Master PCI Dual Address Cycles Upper Address				Yes	No
100h	1A0h	PCI Arbiter Control				Yes	Yes
104h	1A4h	PCI Abort Address				No	No

Notes: *PCI offset registers 100h and 104h are accessible only by way of the PCIBAR0 register.*

Refer to the individual register descriptions to determine which bits are writable.

Table 11-4. Runtime Register Address Mapping

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9656 family and to ensure compatibility with future enhancements, write 0 to all unused bits.			PCI/Local Writable	Serial EEPROM Writable
		31 16	15 8	7 0		
40h	C0h	Mailbox 0 (refer to Notes)			Yes	Yes
44h	C4h	Mailbox 1 (refer to Notes)			Yes	Yes
48h	C8h	Mailbox 2			Yes	No
4Ch	CCh	Mailbox 3			Yes	No
50h	D0h	Mailbox 4			Yes	No
54h	D4h	Mailbox 5			Yes	No
58h	D8h	Mailbox 6			Yes	No
5Ch	DCh	Mailbox 7			Yes	No
60h	E0h	PCI-to-Local Doorbell			Yes	No
64h	E4h	Local-to-PCI Doorbell			Yes	No
68h	E8h	Interrupt Control/Status			Yes	No
6Ch	ECh	Serial EEPROM Control, PCI Command Codes, User I/O Control, and Init Control			Yes	No
70h	F0h	Device ID	Vendor ID		No	No
74h	F4h	***Unused***		Revision ID	No	No
78h	C0h	Mailbox 0 (refer to Notes)			Yes	Yes
7Ch	C4h	Mailbox 1 (refer to Notes)			Yes	Yes

Notes: *The Mailbox 0 register (MBOX0) is always accessible at PCI address 78h, Local address C0h. The Mailbox 1 register (MBOX1) is always accessible at PCI address 7Ch, Local address C4h.*

When I$_2$O Decode is disabled (QSR[0]=0), MBOX0 and MBOX1 are also accessible at PCI addresses 40h and 44h for PCI 9054 compatibility. When I$_2$O Decode is enabled (QSR[0]=1), the Inbound and Outbound Queue pointers are accessed at PCI addresses 40h and 44h, replacing MBOX0 and MBOX1 in PCI Address space.

Refer to the individual register descriptions to determine which bits are writable.

Table 11-5. DMA Register Address Mapping

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9656 family and to ensure compatibility with future enhancements, write 0 to all unused bits.			PCI/Local Writable	Serial EEPROM Writable
		31 16	15 8	7 0		
80h	100h	DMA Channel 0 Mode			Yes	No
84h/88h‡	104h/108h‡	DMA Channel 0 PCI Address			Yes	No
88h/8Ch‡	108h/10Ch‡	DMA Channel 0 Local Address			Yes	No
8Ch/84h‡	10Ch/104h‡	DMA Channel 0 Transfer Size (Bytes)			Yes	No
90h	110h	DMA Channel 0 Descriptor Pointer			Yes	No
94h	114h	DMA Channel 1 Mode			Yes	No
98h/9Ch‡	118h/11Ch‡	DMA Channel 1 PCI Address			Yes	No
9Ch/A0h‡	11Ch/120h‡	DMA Channel 1 Local Address			Yes	No
A0h/98h‡	120h/118h‡	DMA Channel 1 Transfer Size (Bytes)			Yes	No
A4h	124h	DMA Channel 1 Descriptor Pointer			Yes	No
A8h	128h	***Reserved***	DMA Channel 1 Command/ Status	DMA Channel 0 Command/ Status	Yes	No
ACh	12Ch	DMA Arbitration			Yes	Yes
B0h	130h	DMA Threshold			Yes	No
B4h	134h	DMA Channel 0 PCI Dual Address Cycles Upper Address			Yes	No
B8h	138h	DMA Channel 1 PCI Dual Address Cycles Upper Address			Yes	No

Notes: *‡ PCI and Local Configuration offset depends upon the DMAMODEx[20] setting(s).*

Refer to the individual register descriptions to determine which bits are writable.

Table 11-6. Messaging Queue (I_2O) Register Address Mapping

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9656 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	No	No
34h	B4h	Outbound Post Queue Interrupt Mask	Yes	No
40h	–	Inbound Queue Port	PCI	No
44h	–	Outbound Queue Port	PCI	No
C0h	140h	Messaging Queue Configuration	Yes	No
C4h	144h	Queue Base Address	Yes	No
C8h	148h	Inbound Free Head Pointer	Yes	No
CCh	14Ch	Inbound Free Tail Pointer	Yes	No
D0h	150h	Inbound Post Head Pointer	Yes	No
D4h	154h	Inbound Post Tail Pointer	Yes	No
D8h	158h	Outbound Free Head Pointer	Yes	No
DCh	15Ch	Outbound Free Tail Pointer	Yes	No
E0h	160h	Outbound Post Head Pointer	Yes	No
E4h	164h	Outbound Post Tail Pointer	Yes	No
E8h	168h	Queue Status/Control	Yes	No

Notes: *When I_2O Decode is enabled (QSR[0]=1), the PCI Master (Host or other 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 Memory Base Address (programmed in PCIBAR0) to the start of the message frame. This means that all inbound message frames should reside in PCIBAR0 Memory space.

Each Outbound MFA is specified by I_2O as an offset from system address 00000000h. 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.

Refer to the individual register descriptions to determine which bits are writable.

11.3 PCI CONFIGURATION REGISTERS

All registers may be written to or read from in Byte, Word, or Lword accesses.

Note: *"Yes" in the register Read and Write columns indicates the register is writable by the PCI and Local Buses.*

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 the device. Defaults to the PLX PCI-SIG-issued Vendor ID, 10B5h, if blank or if no serial EEPROM is present.	Yes	Local/ Serial EEPROM	10B5h
31:16	**Device ID.** Identifies the particular device. Defaults to the PLX part number for the PCI interface chip (9656h) if blank or no serial EEPROM is present.	Yes	Local/ Serial EEPROM	9656h

Register 11-2. (PCICR; PCI:04h, LOC:04h) PCI Command

Bit	Description	Read	Write	Value after Reset
0	**I/O Space.** Writing 1 allows the device to respond to I/O Space accesses. Writing 0 disables the device from responding to I/O Space accesses.	Yes	Yes	0
1	**Memory Space.** Writing 1 allows the device to respond to Memory Space accesses. Writing 0 disables the device from responding to Memory Space accesses.	Yes	Yes	0
2	**Master Enable.** Writing 1 allows device to behave as a Bus Master. Writing 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 1 enables the Memory Write and Invalidate mode for Direct Master and DMA. ***Note:*** *Refer to DMPBAM[9] for Direct Master and DMAMODEx[13] for DMA.*	Yes	Yes	0
5	**VGA Palette Snoop. *Not Supported.***	Yes	No	0
6	**Parity Error Response.** Writing 1 enables PERR# to be asserted when a parity error is detected. Writing 0 disables PERR# from being asserted when a parity error is detected. Parity error status is reported in PCISR[15], regardless of this bit's value.	Yes	Yes	0
7	**Stepping Control.** Controls whether a device does address/data stepping. Writing 0 indicates the device never does stepping. Writing 1 indicates the device always does stepping. ***Note:*** *Hardwired to 0.*	Yes	No	0
8	**SERR# Enable.** Writing 1 enables the SERR# driver. Writing 0 disables the 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 1 indicates fast back-to-back transfers can occur to any agent on the bus. Writing 0 indicates fast back-to-back transfers can occur only 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 1 supports New Capabilities Functions. If enabled, the first New Capability Function ID is located at the PCI Configuration Space offset determined by the New Capabilities linked list pointer value at offset 34h. 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	***Reserved*.**	Yes	No	0
7	**Fast Back-to-Back Capable.** Writing 1 indicates an adapter can accept fast back-to-back transactions. ***Note:*** *Hardwired to 1.*	Yes	No	1
8	**Master Data Parity Error.** Set to 1 when the following three conditions are met: 1) PERR# is asserted; 2) PCI 9656 was Bus Master for operation in which error occurred; and 3) Parity Error Response bit is set (PCICR[6]=1). Writing 1 clears this bit to 0.	Yes	Yes/Clr	0
10:9	**DEVSEL# Timing.** Indicates timing for DEVSEL# assertion. Writing 01b sets these bits to medium. ***Note:*** *Hardwired to 01b.*	Yes	No	01b
11	**Signaled Target Abort.** When set to 1, indicates the PCI 9656 has signaled a Target Abort. Writing 1 clears this bit to 0.	Yes	Yes/Clr	0
12	**Received Target Abort.** When set to 1, indicates the PCI 9656 has received a Target Abort signal. Writing 1 clears this bit to 0.	Yes	Yes/Clr	0
13	**Received Master Abort.** When set to 1, indicates the PCI 9656 has received a Master Abort signal. Writing 1 clears this bit to 0.	Yes	Yes/Clr	0
14	**Signaled System Error (SERR# Status).** When set to 1, indicates the PCI 9656 has reported a system error on SERR#. Writing 1 clears this bit to 0.	Yes	Yes/Clr	0
15	**Detected Parity Error.** When set to 1, indicates the PCI 9656 has detected a PCI Bus parity error, even if parity error handling is disabled [the Parity Error Response bit in the Command register is cleared (PCICR[6]=0)]. One of three conditions can cause this bit to be set when the PCI 9656 detects a parity error: 1) During a PCI Address phase; 2) During a PCI Data phase when the PCI 9656 is the Target of a write; 3) During a Direct Master or DMA Read operation. Writing 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 9656.	Yes	Local/ Serial EEPROM	Current Rev # (BAh)

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 9656 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 9656, 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 registers 10h through 3Fh in configuration space. Header Type 0 is defined for all PCI devices other than PCI-to-PCI bridges (Header Type 1) and Cardbus bridges (Header Type 2).	Yes	Local	0h
7	**Multi-Function Device.** Value of 1 indicates multiple (up to eight) functions (logical devices) each containing its own, individually addressable configuration space, 64 Lwords in size.	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	**Built-In Self-Test 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	00b
6	**PCI Built-In Self-Test Interrupt Enable.** The PCI Bus writes 1 to enable BIST interrupts. Generates a BIST interrupt to the Local Bus. Reset by the Local Bus when BIST is complete. The software should fail the device if BIST is not complete after 2s. ***Note:*** *Refer to the INTCSR[23] register bit for BIST interrupt status.*	Yes	Yes	0
7	**Built-In Self-Test Support.** Returns 1 if the device supports BIST. Returns 0 if the device is not BIST-compatible.	Yes	Local	0

Register 11-10. (PCIBAR0; PCI:10h, LOC:10h) PCI Base Address for Memory Accesses to Local, Runtime, DMA, and Messaging Queue Registers

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Value of 0 indicates this register maps into Memory space. ***Note:*** *Hardwired to 0.*	Yes	No	0
2:1	**Register Location.** Value of 00b locates anywhere in 32-bit Memory Address space. ***Note:*** *Hardwired to 00b.*	Yes	No	00b
3	**Prefetchable.** Writing 1 indicates there are no side effects on reads. Does not affect PCI 9656 operation. ***Note:*** *Hardwired to 0.*	Yes	No	0
8:4	**Memory Base Address.** Memory Base address for access to Local, Runtime, DMA, and Messaging Queue registers (requires 512 bytes). ***Note:*** *Hardwired to 0h.*	Yes	No	0h
31:9	**Memory Base Address.** Memory Base address for access to Local, Runtime, DMA, and Messaging Queue 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 the start of the message frame.*

Register 11-11. (PCIBAR1; PCI:14h, LOC:14h) PCI Base Address for I/O Accesses to Local, Runtime, DMA, and Messaging Queue Registers

Bit	Description	Read	Write	Value after Reset
0	**I/O Space Indicator.** Value of 1 indicates this 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, DMA, and Messaging Queue 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, DMA, and Messaging Queue registers.	Yes	Yes	0h

Note: *PCIBAR1 can be enabled or disabled by setting or clearing the I/O Base Address Register Enable bit (LMISC1[0]), respectively.*

Register 11-12. (PCIBAR2; PCI:18h, LOC:18h) PCI Base Address for Accesses to Local Address Space 0

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing 0 indicates the register maps into Memory space. Writing 1 indicates the register maps into I/O space. (Bit is writable by way of the LAS0RR register.)	Yes	No	0
2:1	**Register Location (If Memory Space).** Values: 00b = Locate anywhere in 32-bit Memory Address space 01b = *PCI r2.1*, Locate below 1-MB Memory Address space *PCI r2.2*, ***Reserved*** 10b or 11b = ***Reserved*** (Specified in the LAS0RR register.) When mapped into I/O space (PCIBAR2[0]=1), bit 1 is always 0 and bit 2 is included in the Base Address (PCIBAR2[31:4]).	Yes	PCIBAR2[0]=0: No PCIBAR2[0]=1: Bit 1 No, Bit 2 Yes	00b
3	**Prefetchable (If Memory Space).** Writing 1 indicates there are no side effects on reads. Reflects value of LAS0RR[3] and provides only status to the system. Does not affect PCI 9656 operation. The associated Bus Region Descriptor register (LBRD0) controls prefetching functions of this address space. When mapped into I/O space (PCIBAR2[0]=1), bit 3 is included in the Base Address (PCIBAR2[31:4]).	Yes	Memory: No I/O: Yes	0
31:4	**Base Address.** Base Address for access to Local Address Space 0.	Yes	Yes	0h

Notes: *Configure LAS0RR before configuring PCIBAR2.*

If allocated, Local Address Space 0 can be enabled or disabled by setting or clearing LAS0BA[0], respectively.

Register 11-13. (PCIBAR3; PCI:1Ch, LOC:1Ch) PCI Base Address for Accesses to Local Address Space 1

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing 0 indicates the register maps into Memory space. Writing 1 indicates the register maps into I/O space. (Bit is writable by way of the LAS1RR register.)	Yes	No	0
2:1	**Register Location.** Values: 00b = Locate anywhere in 32-bit Memory Address space 01b = *PCI r2.1*, Locate below 1-MB Memory Address space *PCI r2.2*, ***Reserved*** 10b or 11b = ***Reserved*** (Specified in the LAS1RR register.) When mapped into I/O space (PCIBAR3[0]=1), bit 1 is always 0 and bit 2 is included in the Base Address (PCIBAR3[31:4]).	Yes	PCIBAR3[0]=0: No PCIBAR3[0]=1: Bit 1 No, Bit 2 Yes	00b
3	**Prefetchable (If Memory Space).** Writing 1 indicates there are no side effects on reads. Reflects value of LAS1RR[3] and provides only status to the system. Does not affect PCI 9656 operation. The associated Bus Region Descriptor register (LBRD1) controls prefetching functions of this address space. When mapped into I/O space (PCIBAR3[0]=1), bit 3 is included in the Base Address (PCIBAR3[31:4]).	Yes	Memory: No I/O: Yes	0
31:4	**Base Address.** Base address for access to Local Address Space 1. When I_2O Decode is enabled (QSR[0]=1), PCIBAR3[31:4] return 0h.	Yes	Yes	0h

Notes: *Configure LAS1RR before configuring PCIBAR3.*

If allocated, Local Address Space 1 can be enabled or disabled by setting or clearing LAS1BA[0], respectively.

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 Information Structure Pointer

Bit	Description	Read	Write	Value after Reset
31:0	**Cardbus Information Structure (CIS) Pointer for PC Cards.** ***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).** The PLX PCI-SIG-issued Vendor ID is 10B5h.	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	9656h

Register 11-19. (PCIERBAR; PCI:30h, LOC:30h) PCI Base Address for Local Expansion ROM

Bit	Description	Read	Write	Value after Reset
0	**Address Decode Enable.** Writing 1 indicates a device accepts Memory accesses to the Expansion ROM address. Writing 0 indicates a device does not accept accesses to Expansion ROM address. Clear this bit 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
7:0	**New Capability Pointer.** Provides an offset into PCI Configuration Space for location of the Power Management capability in the New Capabilities Linked List. ***Note:*** *For the New Capability Pointer features, these bits must always contain the default value of 40h.*	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 the PCI 9656 INTA# output.	Yes	Yes/Serial EEPROM	0h

Register 11-22. (PCIIPR; PCI:3Dh, LOC:3Dh) PCI Interrupt Pin

Bit	Description	Read	Write	Value after Reset
7:0	**Interrupt Pin.** Indicates which interrupt pin the device uses. The following values are valid: 0 = No Interrupt pin 1 = INTA# The PCI 9656 supports only INTA#.	Yes	Local/ Serial EEPROM	1h

Register 11-23. (PCIMGR; PCI:3Eh, LOC:3Eh) PCI Minimum Grant

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 Maximum Latency

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.** The PCI-SIG-issued Capability ID for Power Management is 01h.	Yes	No	01h

Register 11-26. (PMNEXT; PCI:41h, LOC:181h) Power Management Next Capability Pointer

Bit	Description	Read	Write	Value after Reset
7:0	**Next_Cap Pointer.** Provides an offset into PCI Configuration space for location of the Hot Swap capability in the New Capabilities Linked List. If Power Management is the last capability in the list, clear to 0h. Otherwise, these bits must contain the default value of 48h.	Yes	Local	48h

Register 11-27. (PMC; PCI:42h, LOC:182h) Power Management Capabilities

Bit	Description	Read	Write	Value after Reset
2:0	**Version.** Reading a 010b value indicates this function complies with *PCI Power Mgmt. r1.1*.	Yes	Local/ Serial EEPROM	010b
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. Because the PCI 9656 does not require the PCI clock for PME#, clear this bit to 0 in the serial EEPROM.	Yes	Local	0
4	***Reserved.***	Yes	No	0
5	**Device-Specific Implementation (DSI).** When set to 1, the PCI 9656 requires special initialization following a transition to a D_0 uninitialized state before a generic class device driver is able to use the PCI 9656.	Yes	Local	0
8:6	**AUX_Current.** Refer to *PCI Power Mgmt. r1.1*.	Yes	Local	000b
9	**D_1_Support.** When set to 1, the PCI 9656 supports the D_1 power state.	Yes	Local/ Serial EEPROM	0
10	**D_2_Support.** When set to 1, the PCI 9656 supports the D_2 power state.	Yes	Local/ Serial EEPROM	0
15:11	**PME_Support.** Indicates power states in which the PCI 9656 may assert PME#. Values: XXXX1b = PME# can be asserted from D_0 XXX1Xb = PME# can be asserted from D_1 XX1XXb = PME# can be asserted from D_2 X1XXXb = PME# can be asserted from D_{3hot} 1XXXXb = PME# can be asserted from D_{3cold} ***Note:*** *"X" is "Don't Care."*	Yes	Local/ Serial EEPROM	00000b

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. Values: 00b = D_0 10b = D_2 01b = D_1 11b = D_{3hot} In a D_{3hot} power state, PCI Memory and I/O accesses are disabled, as well as PCI interrupts, and only configuration or PME# assertion is allowed. The same is true for the D_2 power state, if the corresponding D_2_Support bit is set (PMC[10]=1). Transition from a D_{3hot} to a D_0 power state causes a soft reset.	Yes	Yes	00b
7:2	***Reserved.***	Yes	No	0h
8	**PME_En.** Writing 1 enables PME# to be asserted. ***Notes:*** *Value after reset is indeterminate (either 1 or 0) at time of initial operating system boot when the PRESENT_DET pin is connected to power (D_{3cold} power state support).* *Value after reset is 0 when the PRESENT_DET pin is connected to ground (no D_{3cold} power state support).*	Yes	Yes/Serial EEPROM	Sticky bit, see Note
12:9	**Data_Select.** Selects which data to report through the PMDATA register and Data_Scale (PMCSR[14:13] bits.	Yes	Yes/Serial EEPROM	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 bits (PMCSR[12:9]). When the Local CPU initializes the Data_Scale values, the Data_Select bits must be used to determine which Data_Scale value the Local CPU 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/ Serial EEPROM	00b
15	**PME_Status.** Indicates PME# is being driven if the PME_En bit is set (PMCSR[8]=1). Writing 1 from the Local Bus sets this bit; writing 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 (*for example*, PMC[15:11]=1h). ***Note:*** *Value after reset is indeterminate (either 1 or 0) at time of initial operating system boot when the PRESENT_DET pin is connected to power (D_{3cold} power state support). Value after reset is 0 when the PRESENT_DET pin is connected to ground (no D_{3cold} power state support).*	Yes	Local/Set, PCI/Clr	Sticky bit, see Note

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]). Values: 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/ Serial EEPROM	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.** The Hot Swap Capability ID is 06h. To disable Hot Swap capabilities, clear this register to 0h.	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
7:0	**Next_Cap Pointer.** Provides an offset into PCI Configuration space for location of the VPD capability in the New Capabilities Linked List. If Hot Swap is the last capability in the list, clear to 0h. Otherwise, these bits must contain the default value of 4Ch.	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 Mask (EIM).** Writing 0 enables interrupt assertion. Writing 1 masks interrupt assertion.	Yes	Yes/Clr	0
2	***Reserved.***	Yes	No	0
3	**LED Software On/Off Switch.** Writing 1 asserts the LEDon# signal. Writing 0 de-asserts the LEDon# signal.	Yes	PCI	0
5:4	**Programming Interface 0 (PI = 0).**	Yes	No	00b
6	**ENUM# Status – Extraction.** Writing 1 reports the ENUM# assertion for removal process.	Yes	Yes/Clr	0
7	**ENUM# Status – Insertion.** Writing 1 reports the ENUM# assertion for insertion process.	Yes	Yes/Clr	0
15:8	***Reserved.***	Yes	No	0h

Register 11-34. (PVPDID; PCI:4Ch, LOC:18Ch) PCI Vital Product Identification

Bit	Description	Read	Write	Value after Reset
7:0	**VPD ID.** The PCI-SIG-issued Capability ID for VPD is 03h.	Yes	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
7:0	**Next_Cap Pointer.** Because VPD is the last capability in the list, clear to 0h.	Yes	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.** VPD byte address to be accessed. All accesses are 32 bits wide. For VPD writes, the byte address must be Lword-aligned (bits [1:0]=00b). For VPD reads, the byte address must be word-aligned (bit [0]=0). PVPDAD[14:9] are ignored.	Yes	Yes	0h
15	**F.** Controls the direction of the next VPD cycle and indicates when the VPD cycle is complete. Writing 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 1 along with the VPD address causes a write of VPD information from PVPDATA into a storage component. The hardware clears this bit to 0 after the Write operation is complete.	Yes	Yes	0

Register 11-37. (PVPDATA; PCI:50h, LOC:190h) PCI VPD Data

Bit	Description	Read	Write	Value after Reset
31:0	**VPD Data.** (Refer to Section 10.2.2.)	Yes	Yes	0h

11.4 LOCAL CONFIGURATION REGISTERS

Note: *"Yes" in the register Read and Write columns indicates the register is writable by the PCI and Local Buses.*

Register 11-38. (LAS0RR; PCI:00h, LOC:80h) Direct Slave Local Address Space 0 Range

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing 0 indicates Local Address Space 0 maps into PCI Memory space. Writing 1 indicates Local Address Space 0 maps into PCI I/O space.	Yes	Yes/Serial EEPROM	0
2:1	When mapped into Memory space (LAS0RR[0]=0), the only valid value is 00b. Locate anywhere in 32-bit PCI Address space. When mapped into I/O space (LAS0RR[0]=1), bit 1 must be cleared to 0. Bit 2 is included with LAS0RR[31:3] to indicate the decoding range.	Yes	Yes/Serial EEPROM	00b
3	When mapped into Memory space (LAS0RR[0]=0), writing 1 indicates reads are prefetchable (does not affect PCI 9656 operation, but is used for system status). When mapped into I/O space (LAS0RR[0]=1), this bit is included with LAS0RR[31:4, 2] to indicate the decoding range.	Yes	Yes/Serial EEPROM	0
31:4	Specifies which PCI Address bits to use for decoding a PCI access to Local Address 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:*** *LAS0RR range (**not** the Range register) must be power of 2. "Range register value" is two's complement of the range.* *Per PCI r2.2, user should limit I/O-mapped spaces to 256 bytes per space.*	Yes	Yes/Serial EEPROM	FFF0000h

Register 11-39. (LAS0BA; PCI:04h, LOC:84h) Direct Slave Local Address Space 0 Local Base Address (Remap)

Bit	Description	Read	Write	Value after Reset
0	**Local Address Space 0 Enable.** Writing 1 enables decoding of PCI addresses for Direct Slave access to Local Address Space 0. Writing 0 disables decoding.	Yes	Yes/Serial EEPROM	0
1	***Reserved.***	Yes	No	0
3:2	If Local Address Space 0 is mapped into Memory space (LAS0RR[0]=0), LAS0BA[3:2] must be 00b. When mapped into I/O space (LAS0RR[0]=1), included with LAS0BA[31:4] for remapping.	Yes	Yes/Serial EEPROM	00b
31:4	**Remap PCIBAR2 Base Address to Local Address Space 0 Base Address.** The PCIBAR2 base address translates to the Local Address Space 0 Base Address programmed in this register. A Direct Slave access to an offset from PCIBAR2 maps to the same offset from this Local Base Address. ***Note:*** *Remap Address value must be a multiple of the LAS0RR range (**not** the Range register).*	Yes	Yes/Serial EEPROM	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.** **M Mode:** Number of Local Bus Clock cycles to occur before de-asserting BB# and releasing the Local Bus. The Local Bus Latency Timer starts counting upon BB# assertion. When MARBR[16]=1, specifies the minimum delay (in LCLK cycles) before interrupting a continuing Direct Slave or DMA transfer and releasing the Local Bus by de-asserting BB#. The Local Bus Latency Timer starts counting upon TS# assertion. Refer to Sections 3.4.4.13 and 3.4.4.15 for further details. **C and J Modes:** Number of Local Bus Clock cycles to occur before de-asserting LHOLD and releasing the Local Bus. The Local Bus Latency Timer starts counting upon LHOLDA assertion. When MARBR[16]=1, specifies the minimum delay (in LCLK cycles) before interrupting a continuing Direct Slave or DMA transfer and releasing the Local Bus by de-asserting LHOLD. The Local Bus Latency Timer starts counting upon ADS# assertion. Refer to Sections 5.4.4.12 and 5.4.4.14 for further details.	Yes	Yes/Serial EEPROM	0h
15:8	**Local Bus Pause Timer.** Valid only during DMA transfers. **M Mode:** Number of Local Bus Clock cycles to occur before re-asserting BR# after releasing the Local Bus. When MARBR[17]=1, specifies the minimum delay (in LCLK cycles) before re-asserting BR# to resume a previously interrupted DMA transfer. **C and J Modes:** Number of Local Bus Clock cycles to occur before re-asserting LHOLD after releasing the Local Bus. When MARBR[17]=1, specifies the minimum delay (in LCLK cycles) before re-asserting LHOLD to resume a previously interrupted DMA transfer.	Yes	Yes/Serial EEPROM	0h

Register 11-40. (MARBR; PCI:08h or ACh, LOC:88h or 12Ch) Mode/DMA Arbitration (Continued)

Bit	Description	Read	Write	Value after Reset
16	**Local Bus Latency Timer Enable.** Writing 1 enables the Latency Timer. Writing 0 disables the Latency Timer.	Yes	Yes/Serial EEPROM	0
17	**Local Bus Pause Timer Enable.** Writing 1 enables the Pause Timer. Writing 0 disables the Pause Timer.	Yes	Yes/Serial EEPROM	0
18	**Local Bus BREQi Enable (C and J Modes Only)**. Writing 1 enables the Local Bus BREQi. When BREQi is asserted, the PCI 9656 de-asserts LHOLD and releases the Local Bus.	Yes	Yes/Serial EEPROM	0
20:19	**DMA Channel Priority.** Writing 00b indicates a rotational priority scheme. Writing 01b indicates Channel 0 has priority. Writing 10b indicates Channel 1 has priority. Value of 11b is ***reserved***.	Yes	Yes/Serial EEPROM	00b
21	**Local Bus Direct Slave Release Bus Mode.** **C and J Modes Only:** When set to 1, the PCI 9656 de-asserts LHOLD and releases the Local Bus when either of the following occurs: • Direct Slave Write FIFO becomes empty during a Direct Slave write • Direct Slave Read FIFO becomes full during a Direct Slave read ***Note:*** *For M mode, this bit must be cleared to 0.*	Yes	Yes/Serial EEPROM	1
22	**Direct Slave PCI LOCK# Input Enable.** Writing 1 enables Direct Slave locked sequences. Writing 0 disables Direct Slave locked sequences.	Yes	Yes/Serial EEPROM	0
23	**PCI Request Mode.** Writing 1 causes the PCI 9656 to de-assert REQ# when the PCI 9656 asserts FRAME# during a Master cycle. Writing 0 causes the PCI 9656 to leave REQ# asserted for the entire Bus Master cycle.	Yes	Yes/Serial EEPROM	0
24	**PCI Compliance Enable.** When set to 1, the PCI 9656 performs all PCI Read and Write transactions in compliance with *PCI r2.2*. Setting this bit enables Direct Slave Delayed Reads, 2^{15} PCI clock timeout on Retrys, 16- and 8-clock PCI latency rules, and enables the option to select PCI Read No Write mode (Retries for writes) (MARBR[25]). Value of 0 causes TRDY# to remain de-asserted on reads until Read data is available. ***Note:*** *Refer to Section 3.4.2.4 (M mode) or Section 5.4.2.4 (C and J modes) for further details.*	Yes	Yes/Serial EEPROM	0
25	**PCI Read No Write Mode (PCI Retries for Writes).** When PCI Compliance is enabled (MARBR[24]=1), value of 1 forces a PCI Retry on writes if a Delayed Read is pending. Value of 0 (or MARBR[24]=0) allows writes to occur while a Delayed Read is pending.	Yes	Yes/Serial EEPROM	0
26	**PCI Read with Write Flush Mode.** Value of 1 flushes a pending Delayed Read cycle if a Write cycle is detected. Value of 0 does not affect a pending Delayed Read when a Write cycle occurs.	Yes	Yes/Serial EEPROM	0
27	**Gate Local Bus Latency Timer with BREQi (C and J Modes Only).** The Local Bus Latency Timer counts only while BREQi is asserted (BREQi=1) and this bit is set to 1. The Local Bus Latency Timer does not timeout until the number of Local clocks programmed into MARBR[7:0] is reached. When cleared to 0, or if the Local Bus Latency Timer is disabled (MARBR[16]=0), BREQi assertion causes the PCI 9656 to release the Local Bus. (Refer to Section 4.2.1.)	Yes	Yes/Serial EEPROM	0

Register 11-40. (MARBR; PCI:08h or ACh, LOC:88h or 12Ch) Mode/DMA Arbitration (Continued)

Bit	Description	Read	Write	Value after Reset
28	**PCI Read No Flush Mode.** Writing 1 submits a request to not flush the Direct Slave Read FIFO when a PCI Read cycle completes (Direct Slave Read Ahead mode). Writing 0 submits a request to flush the Direct Slave Read FIFO if a PCI Read cycle completes.	Yes	Yes/Serial EEPROM	0
29	When cleared to 0, reads from the PCI Configuration register address 00h return the Device and Vendor IDs. When set to 1, reads from the PCI Configuration register address 00h return the Subsystem and Subsystem Vendor IDs.	Yes	Yes/Serial EEPROM	0
30	**Direct Master Write FIFO Full Status Flag.** When set to 1, the Direct Master Write FIFO is almost full. Reflects the DMPAF pin value, as determined by the Programmable Almost Full Flag value in DMPBAM[10, 8:5].	Yes	No	0
31	**BIGEND#/WAIT# I/O Select (M Mode Only).** Writing 1 selects the Wait I/O functionality of the signal. Writing 0 selects Big Endian input functionality.	Yes	Yes/Serial EEPROM	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 1 specifies use of Big Endian data ordering for Local accesses to the Configuration registers. Writing 0 specifies Little Endian ordering.	Yes	Yes/Serial EEPROM	0
1	**Direct Master Big Endian Mode (Address Invariance).** Writing 1 specifies use of Big Endian data ordering for Direct Master accesses. Writing 0 specifies Little Endian ordering.	Yes	Yes/Serial EEPROM	0
2	**Direct Slave Local Address Space 0 Big Endian Mode (Address Invariance).** Writing 1 specifies use of Big Endian data ordering for Direct Slave accesses to Local Address Space 0. Writing 0 specifies Little Endian ordering.	Yes	Yes/Serial EEPROM	0
3	**Direct Slave Expansion ROM Space Big Endian Mode (Address Invariance).** Writing 1 specifies use of Big Endian data ordering for Direct Slave accesses to Expansion ROM. Writing 0 specifies Little Endian ordering.	Yes	Yes/Serial EEPROM	0
4	**Big Endian Byte Lane Mode.** Writing 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 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/Serial EEPROM	0
5	**Direct Slave Local Address Space 1 Big Endian Mode (Address Invariance).** Writing 1 specifies use of Big Endian data ordering for Direct Slave accesses to Local Address Space 1. Writing 0 specifies Little Endian ordering.	Yes	Yes/Serial EEPROM	0
6	**DMA Channel 1 Big Endian Mode (Address Invariance).** Writing 1 specifies use of Big Endian data ordering for DMA Channel 1 accesses to the Local Bus. Writing 0 specifies Little Endian ordering.	Yes	Yes/Serial EEPROM	0
7	**DMA Channel 0 Big Endian Mode (Address Invariance).** Writing 1 specifies use of Big Endian data ordering for DMA Channel 0 accesses to the Local Bus. Writing 0 specifies Little Endian ordering.	Yes	Yes/Serial EEPROM	0

Register 11-42. (LMISC1; PCI:0Dh, LOC:8Dh) Local Miscellaneous Control

Bit	Description	Read	Write	Value after Reset
0	**I/O Base Address Register Enable.** If set to 1, the PCI Base Address for I/O accesses to Local, Runtime, DMA, and Messaging Queue Registers register (PCIBAR1) is enabled. If cleared to 0, PCIBAR1 is disabled. This option is intended for embedded systems only. Set this bit to 1 for PC platforms.	Yes	Yes/ Serial EEPROM	1
1	**I/O Base Address Register Shift.** If the I/O Base Address register is disabled and this bit is cleared to 0 (LMISC1[1:0]=00b), then PCIBAR2 and PCIBAR3 remain at PCI Configuration addresses 18h and 1Ch. If the I/O Base Address register is disabled and this bit is set to 1 (LMISC1[1:0]=10b), 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 I/O Base Address register space cannot be accepted by the system BIOS.	Yes	Yes/ Serial EEPROM	0
2	**Local Init Status.** Reading 1 indicates Local Init is done. Responses to PCI accesses prior to this bit being set are determined by the USERi state when the PCI 9656 receives an external reset at PCI RST# de-assertion in Adapter mode (HOSTEN#=1), or LRESET# input de-assertion in Host mode (HOSTEN#=0). ***Note:*** *Refer to Section 2.4.2 (M mode) or Section 4.4.2 (C and J modes) for further details.*	Yes	Local/ Serial EEPROM	0
3	**Direct Master (PCI Initiator) Write FIFO Flush during PCI Master Abort.** When cleared to 0, the PCI 9656 flushes the Direct Master Write FIFO each time a PCI Master/Target Abort or Retry timeout occurs. When set to 1, the PCI 9656 retains data in the Direct Master Write FIFO.	Yes	Yes/ Serial EEPROM	0
4	**M Mode Direct Master Delayed Read Enable.** Writing 1 enables the PCI 9656 to operate in Delayed Transaction mode for Direct Master reads. The PCI 9656 issues a RETRY# to the M mode Master and prefetches Read data from the PCI Bus.	Yes	Yes/ Serial EEPROM	0
5	**M Mode TEA# Input Interrupt Mask.** When set to 1, TEA# input causes SERR# output to assert on the PCI Bus, if enabled (PCICR[8]=1), and the Signaled System Error (SERR# Status) bit is set (PCISR[14]=1). Writing 0 masks the TEA# input to assert SERR#. PCISR[14] is set to 1 in both cases.	Yes	Yes/ Serial EEPROM	0
6	**M Mode Direct Master Write FIFO Almost Full RETRY# Output Enable.** When set to 1, the PCI 9656 issues a RETRY# when the Direct Master Write FIFO is almost full.	Yes	Yes/ Serial EEPROM	0
7	**Disconnect with Flush Read FIFO.** When PCI Compliance is enabled (MARBR[24]=1), value of 1 causes acceptance of a new Read request with flushing of the Read FIFO when a Direct Slave Read request does not match an existing, pending Delayed Read in the Read FIFO. Value of 0, or clearing of the PCI Compliance Enable bit (MARBR[24]=0), causes a new Direct Slave Read request to be Retried when a Delayed Read is pending in the Read FIFO.	Yes	Yes/ Serial EEPROM	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 Location Starting at Lword Boundary (48 Lwords = 192 Bytes) for VPD Accesses.** Value is the number of Lwords that are VPD write-protected (starting from serial EEPROM byte address 00h). The corresponding serial EEPROM byte (VPD) address is this value multiplied by 4. Default value of 30h sets the boundary at serial EEPROM byte (VPD) address C0h. Value of 40h write protects a 2K bit serial EEPROM. A maximum value of 7Fh write-protects all but 2 words in a 4K bit serial EEPROM. Any serial EEPROM address below this boundary is Read-Only. ***Note:*** *PCI 9656 configuration data is stored below Lword address 19h.*	Yes	Yes/Serial EEPROM	30h
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	**TA#/READY# Timeout Enable.** Value of 1 enables TA# (M mode) or READY# (C and J modes) timeout.	Yes	Yes/Serial EEPROM	0
1	**TA#/READY# Timeout Select.** Values: 1 = 1,024 clocks 0 = 32 clocks	Yes	Yes/Serial EEPROM	0
4:2	**Direct Slave Delayed Write Mode.** Delay in LCLK of BR# (M mode) or LHOLD (C and J modes) assertion for PCI Burst writes. 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/Serial EEPROM	000b
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 9656. Value of 0 Retries all Direct Slave Read accesses when the Direct Slave Write FIFO is full with Direct Slave Write data.	Yes	Yes/Serial EEPROM	0
7:6	***Reserved*.**	Yes	No	00b

Register 11-45. (EROMRR; PCI:10h, LOC:90h) Direct Slave Expansion ROM Range

Bit	Description	Read	Write	Value after Reset
0	**Address Decode Enable.** Bit 0 can be enabled only from the serial EEPROM. To disable, clear 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). The minimum range, if enabled, is 2 KB, and maximum range allowed by *PCI r2.2* is 16 MB. ***Notes:*** *EROMRR range (**not** the Range register) must be power of 2. "Range register value" is two's complement of the range.* *EROMRR should be programmed by way of the serial EEPROM to a value of 0h, unless Expansion ROM is present on the Local Bus. If the value is not 0h, system BIOS may attempt to allocate Expansion ROM address space and then access it at the Local Address space specified in EROMBA[31:11] (default value is 0h) to determine whether the Expansion ROM image is valid. If the image is not valid, as defined in Section 6.3.1.1 (PCI Expansion ROM Header Format) of PCI r2.2, the system BIOS unmaps the Expansion ROM address space that it initially allocated, by writing 0h to PCIERBAR.*	Yes	Yes/Serial EEPROM	0h

Register 11-46. (EROMBA; PCI:14h, LOC:94h) Direct Slave Expansion ROM Local Base Address (Remap) and BREQo Control

Bit	Description	Read	Write	Value after Reset
3:0	**RETRY# Signal Assertion Delay Clocks (M Mode).** 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 the PCI 9656 samples BB# de-assertion by the external Local Bus Arbiter (LSB is 8 or 64 clocks). **Backoff Request Delay Clocks (C and J Modes).** Number of Local Bus clocks in which a Direct Slave LHOLD request is pending and a Local Direct Master access is in progress and not being granted the bus (LHOLDA) before asserting BREQo. BREQo remains asserted until the PCI 9656 receives LHOLDA (LSB is 8 or 64 clocks).	Yes	Yes/Serial EEPROM	0h
4	**Local Bus Backoff Enable.** Writing 1 enables the PCI 9656 to assert RETRY# (M mode) or BREQo (C and J modes).	Yes	Yes/Serial EEPROM	0
5	**Backoff Timer Resolution.** Writing 1 changes the LSB of the Backoff Timer from 8 to 64 clocks.	Yes	Yes/Serial EEPROM	0
10:6	***Reserved.***	Yes	No	0h
31:11	**Remap PCI Expansion ROM 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 EROMRR range (**not** the Range register).*	Yes	Yes/Serial EEPROM	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	**Local Address Space 0 Local Bus Data Width.** Writing of the following values indicates the associated bus data width: 00b = 8 bit 01b = 16 bit 10b or 11b = 32 bit	Yes	Yes/Serial EEPROM	11b
5:2	**Local Address Space 0 Internal Wait State Counter (Address-to-Data; Data-to-Data; 0 to 15 Wait States).**	Yes	Yes/Serial EEPROM	0h
6	**Local Address Space 0 TA#/READY# Input Enable.** Writing 1 enables TA# (M mode) or READY# (C and J modes) input. Writing 0 disables TA#/READY# input.	Yes	Yes/Serial EEPROM	1
7	**Local Address Space 0 Continuous Burst Enable.** When bursting is enabled (LBRD0[24]=1), writing 1 enables Continuous Burst mode and writing 0 enables Burst-4 mode. In C and J modes, writing 1 additionally enables BTERM# input, which when asserted overrides the READY# input state (if READY# is enabled, LBRD0[6]=1). ***Notes:*** *In M mode, this bit is referred to as the "BI mode" bit. In C and J modes, this bit is referred to as the "BTERM# Input Enable" bit.* *Refer to Section 2.2.5 (M mode) or Section 4.2.5 (C and J modes) for further details.*	Yes	Yes/Serial EEPROM	0
8	**Local Address Space 0 Prefetch Disable.** When mapped into Memory space (LAS0RR[0]=0), writing 0 enables Read prefetching. Writing 1 disables prefetching. If prefetching is disabled, the PCI 9656 disconnects after each Memory read.	Yes	Yes/Serial EEPROM	0
9	**Expansion ROM Space Prefetch Disable.** Writing 0 enables Read prefetching. Writing 1 disables prefetching. If prefetching is disabled, the PCI 9656 disconnects after each Memory read.	Yes	Yes/Serial EEPROM	0
10	**Local Address Space 0/Expansion ROM Space Prefetch Counter Enable.** When set to 1 and Memory prefetching is enabled, the PCI 9656 prefetches up to the number of Lwords specified in the Prefetch Counter (LBRD0[14:11]). When cleared to 0, the PCI 9656 ignores the count and continues prefetching as follows: • **If PCI Read No Flush mode is enabled (MARBR[28]=1)** – Continues prefetching until the Direct Slave Read FIFO is full, or • **If PCI Read No Flush mode is disabled (MARBR[28]=0)** – Continues prefetching until PCI read completion, at which time the Direct Slave Read FIFO is flushed.	Yes	Yes/Serial EEPROM	0
14:11	**Local Address Space 0/Expansion ROM Space Prefetch Counter.** Number of Lwords to prefetch during Memory Read cycles (0 to 15). A count of zero selects a prefetch of 16 Lwords.	Yes	Yes/Serial EEPROM	0h
15	***Reserved.***	Yes	No	0

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
17:16	**Expansion ROM Space Local Bus Data Width.** Writing of the following values indicates the associated bus data width: 00b = 8 bit 01b = 16 bit 10b or 11b = 32 bit	Yes	Yes/Serial EEPROM	11b
21:18	**Expansion ROM Space Internal Wait State Counter (Address-to-Data; Data-to-Data; 0 to 15 Wait States).**	Yes	Yes/Serial EEPROM	0h
22	**Expansion ROM Space TA#/READY# Input Enable.** Writing 1 enables TA# (M mode) or READY# (C and J modes) input. Writing 0 disables TA#/READY# input.	Yes	Yes/Serial EEPROM	1
23	**Expansion ROM Space Continuous Burst Enable.** When bursting is enabled (LBRD0[26]=1), writing 1 enables Continuous Burst mode and writing 0 enables Burst-4 mode. In C and J modes, writing 1 additionally enables BTERM# input, which when asserted overrides the READY# input state (if READY# is enabled, LBRD0[22]=1). ***Notes:*** *In M mode, this bit is referred to as the "BI mode" bit. In C and J modes, this bit is referred to as the "BTERM# Input Enable" bit.* *Refer to Section 2.2.5 (M mode) or Section 4.2.5 (C and J modes) for further details.*	Yes	Yes/Serial EEPROM	0
24	**Local Address Space 0 Burst Enable.** Writing 1 enables bursting. Writing 0 disables bursting.	Yes	Yes/Serial EEPROM	0
25	**Extra Long Load from Serial EEPROM.** Writing 1 loads the Subsystem ID, Local Address Space 1 registers, and PCI Arbiter register values. Writing 0 indicates not to load them.	Yes	Serial EEPROM Only	0
26	**Expansion ROM Space Burst Enable.** Writing 1 enables bursting. Writing 0 disables bursting.	Yes	Yes/Serial EEPROM	0
27	**Direct Slave PCI Write Mode.** Writing 0 indicates the PCI 9656 should disconnect from the PCI Bus when the Direct Slave Write FIFO is full. Writing 1 indicates the PCI 9656 should de-assert TRDY# when the Direct Slave Write FIFO is full. ***Note:*** *This bit is used for all three Local Address spaces – Space 0, Space 1, and Expansion ROM.*	Yes	Yes/Serial EEPROM	0
31:28	**Direct Slave Retry Delay Clocks.** Number of PCI clocks (multiplied by 8) from the beginning of a Direct Slave access until a PCI Retry is issued, if the transfer has not completed. Valid for Read cycles only if MARBR[24]=0. Valid for Write cycles only if LBRD0[27]=1. ***Note:*** *These bits are used for all three Local Address spaces – Space 0, Space 1, and Expansion ROM.*	Yes	Yes/Serial EEPROM	4h (32 clocks)

Register 11-48. (DMRR; PCI:1Ch, LOC:9Ch) Local Range 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:*** *DMRR range (**not** the Range register) must be power of 2. “Range register value” is two’s complement of the range.*	Yes	Yes/Serial EEPROM	0h

Register 11-49. (DMLBAM; PCI:20h, LOC:A0h) Local Base Address 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 DMRR range (**not** the Range register).*	Yes	Yes/Serial EEPROM	0h

Register 11-50. (DMLBAI; PCI:24h, LOC:A4h) Local Base Address 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 PCI Configuration Space accesses. ***Notes:*** *Local Base Address value must be a multiple of the DMRR range (**not** the Range register). I/O address space is 64 KB.* *Refer to DMPBAM[13] for the I/O Remap Address option.*	Yes	Yes/Serial EEPROM	0h

Register 11-51. (DMPBAM; PCI:28h, LOC:A8h) PCI Base Address (Remap) for Direct Master-to-PCI Memory

Bit	Description	Read	Write	Value after Reset
0	**Direct Master Memory Access Enable.** Writing 1 enables decode of Direct Master Memory accesses within the address space defined by the DMLBAM and DMRR registers. Writing 0 disables decode of Direct Master Memory accesses.	Yes	Yes/Serial EEPROM	0
1	**Direct Master I/O Access Enable.** Writing 1 enables decode of Direct Master I/O accesses within the 64 KB address space beginning at the DMLBAI base address. Writing 0 disables decode of Direct Master I/O accesses.	Yes	Yes/Serial EEPROM	0
2	**Direct Master Read Ahead Mode.** Writing 1 submits a request to not flush the Read FIFO if the Local Read cycle completes (Direct Master Read Ahead mode). This is valid only when DMPBAM[12, 3]=00b. Writing 0 submits a request to flush the Read FIFO if a Local Read cycle completes. ***Caution:*** *To prevent constant locking of the PCI Bus, do **not** simultaneously enable this bit and Direct Master PCI Read mode (DMPBAM[2, 4]≠11b).*	Yes	Yes/Serial EEPROM	0
12, 3	**Direct Master Read Prefetch Size Control.** Values: 00b = The PCI 9656 continues to prefetch Read data from the PCI Bus until the Direct Master Read access is finished. This may result in an additional four unneeded Lwords being prefetched from the 32-bit PCI Bus, or four additional unneeded Qwords being prefetched from the 64-bit PCI Bus. 01b = Prefetch up to 4 Lwords from the 32-bit PCI Bus. Prefetch up to 4 Qwords from the 64-bit PCI Bus. 10b = Prefetch up to 8 Lwords from the 32-bit PCI Bus. Prefetch up to 8 Qwords from the 64-bit PCI Bus. 11b = Prefetch up to 16 Lwords from the 32-bit PCI Bus. Prefetch up to 16 Qwords from the 64-bit PCI Bus. ***Caution:*** *Direct Master Burst reads must **not** exceed the programmed limit.*	Yes	Yes/Serial EEPROM	00b
4	**Direct Master PCI Read Mode.** Writing 0 indicates the PCI 9656 should release the PCI Bus when the Read FIFO becomes full. Writing 1 indicates the PCI 9656 should retain the PCI Bus and de-assert IRDY# when the Read FIFO becomes full. ***Caution:*** *To prevent constant locking of the PCI Bus, do **not** simultaneously enable this bit and Direct Master Read Ahead mode (DMPBAM[2, 4]≠11b).*	Yes	Yes/Serial EEPROM	0
10, 8:5	**Programmable Almost Full Flag.** When the number of entries in the 32-Qword Direct Master Write FIFO exceeds a (programmed value +1), MDREQ#/DMPAF (M mode) or DMPAF (C and J modes) is asserted high.	Yes	Yes/Serial EEPROM	00000b
9	**Memory Write and Invalidate Mode.** When set to 1, the PCI 9656 waits for 8 or 16 Lwords to be written from the Local Bus before starting a PCI access. All Memory Write and Invalidate cycles to the PCI Bus must be 8 or 16 Lword bursts.	Yes	Yes/Serial EEPROM	0
11	**Direct Master Prefetch Limit.** Writing 1 causes the PCI 9656 to terminate a prefetch at 4-KB boundaries and restart when the boundary is crossed. Writing 0 results in continuous prefetch over the boundary space.	Yes	Yes/Serial EEPROM	0
13	**I/O Remap Select.** Writing 1 forces PCI Address bits [31:16] to all zeros (0) for PCI I/O Direct Master transactions. Writing 0 uses DMPBAM[31:16] as PCI Address bits [31:16] for PCI I/O Direct Master transactions.	Yes	Yes/Serial EEPROM	0
15:14	**Direct Master Delayed Write Mode.** Delays PCI Bus request after Direct Master Burst Write cycle has started. Values: 00b = No delay; start cycle immediately 01b = Delay 4 PCI clocks 10b = Delay 8 PCI clocks 11b = Delay 16 PCI clocks	Yes	Yes/Serial EEPROM	00b

Register 11-51. (DMPBAM; PCI:28h, LOC:A8h) PCI Base Address (Remap) for Direct Master-to-PCI Memory (Continued)

Bit	Description	Read	Write	Value after Reset
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 DMRR range (**not** the Range register).*	Yes	Yes/Serial EEPROM	0h

Register 11-52. (DMCFGA; PCI:2Ch, LOC:ACh) PCI Configuration Address for Direct Master-to-PCI I/O Configuration

Bit	Description	Read	Write	Value after Reset
1:0	**Configuration Cycle Type.** Values: 00b = Type 0 01b = Type 1	Yes	Yes/Serial EEPROM	00b
7:2	**Register Number.**	Yes	Yes/Serial EEPROM	0h
10:8	**Function Number.**	Yes	Yes/Serial EEPROM	000b
15:11	**Device Number.**	Yes	Yes/Serial EEPROM	0h
23:16	**Bus Number.**	Yes	Yes/Serial EEPROM	0h
30:24	***Reserved.***	Yes	No	0h
31	**Configuration Enable.** Writing 1 allows Local-to-PCI I/O accesses to be converted to a PCI Configuration cycle. Parameters in this register are used to assert the PCI Configuration address. ***Note:*** *Refer to the Direct Master Configuration Cycle example in Section 3.4.1.9.1 (M mode) or Section 5.4.1.7.1 (C and J modes) for further details.*	Yes	Yes/Serial EEPROM	0

Register 11-53. (LAS1RR; PCI:F0h, LOC:170h) Direct Slave Local Address Space 1 Range

Bit	Description	Read	Write	Value after Reset
0	**Memory Space Indicator.** Writing 0 indicates Local Address Space 1 maps into PCI Memory space. Writing 1 indicates Local Address Space 1 maps into PCI I/O space.	Yes	Yes/Serial EEPROM	0
2:1	When mapped into Memory space (LAS1RR[0]=0), the only valid value is 00b. Locate anywhere in 32-bit PCI Address space. When mapped into I/O space (LAS1RR[0]=1), bit 1 must be cleared to 0. Bit 2 is included with LAS1RR[31:3] to indicate the decoding range.	Yes	Yes/Serial EEPROM	00b
3	When mapped into Memory space (LAS1RR[0]=0), writing 1 indicates reads are prefetchable (does not affect PCI 9656 operation, but is used for system status). When mapped into I/O space (LAS1RR[0]=1), included with LAS1RR[31:4, 2] to indicate the decoding range.	Yes	Yes/Serial EEPROM	0
31:4	Specifies which PCI Address bits to use for decoding a PCI access to Local Address 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. When I_2O Decode is enabled (QSR[0]=1), defines PCIBAR0 (minimum range is 1,024 bytes). ***Notes:*** *LAS1RR range (**not** the Range register) must be power of 2. “Range register value” is two’s complement of the range.* *Per PCI r2.2, user should limit I/O-mapped spaces to 256 bytes per space.*	Yes	Yes/Serial EEPROM	FFF0000h

Register 11-54. (LAS1BA; PCI:F4h, LOC:174h) Direct Slave Local Address Space 1 Local Base Address (Remap)

Bit	Description	Read	Write	Value after Reset
0	**Local Address Space 1 Enable.** Writing 1 enables decoding of PCI addresses for Direct Slave access to Local Address Space 1. Writing 0 disables decoding.	Yes	Yes/Serial EEPROM	0
1	***Reserved.***	Yes	No	0
3:2	If Local Address Space 1 is mapped into Memory space (LAS1RR[0]=0), LAS1BA[3:2] must be 00b. When mapped into I/O space (LAS1RR[0]=1), included with LAS1BA[31:4] for remapping.	Yes	Yes/Serial EEPROM	00b
31:4	**Remap PCIBAR3 Base Address to Local Address Space 1 Base Address.** The PCIBAR3 base address translates to the Local Address Space 1 Base Address programmed in this register. A Direct Slave access to an offset from PCIBAR3 maps to the same offset from this Local Base Address. ***Note:*** *Remap Address value must be a multiple of the LAS1RR range (**not** the Range register).*	Yes	Yes/Serial EEPROM	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	**Local Address Space 1 Local Bus Data Width.** Writing of the following values indicates the associated bus data width: 00b = 8 bit 01b = 16 bit 10b or 11b = 32 bit	Yes	Yes/Serial EEPROM	11b
5:2	**Local Address Space 1 Internal Wait State Counter (Address-to-Data; Data-to-Data; 0 to 15 Wait States).**	Yes	Yes/Serial EEPROM	0h
6	**Local Address Space 1 TA#/READY# Input Enable.** Writing 1 enables TA# (M mode) or READY# (C and J modes) input. Writing 0 disables TA#/READY# input.	Yes	Yes/Serial EEPROM	1
7	**Local Address Space 1 Continuous Burst Enable.** When bursting is enabled (LBRD1[8]=1), writing 1 enables Continuous Burst mode and writing 0 enables Burst-4 mode. In C and J modes, writing 1 additionally enables BTERM# input, which when asserted overrides the READY# input state (if READY# is enabled, LBRD1[6]=1). ***Notes:*** *In M mode, this bit is referred to as the "BI mode" bit. In C and J modes, this bit is referred to as the "BTERM# Input Enable" bit.* *Refer to Section 2.2.5 (M mode) or Section 4.2.5 (C and J modes) for further details.*	Yes	Yes/Serial EEPROM	0
8	**Local Address Space 1 Burst Enable.** Writing 1 enables bursting. Writing 0 disables bursting.	Yes	Yes/Serial EEPROM	0
9	**Local Address Space 1 Prefetch Disable.** When mapped into Memory space (LAS1RR[0]=0), writing 0 enables Read prefetching. Writing 1 disables prefetching. If prefetching is disabled, the PCI 9656 disconnects after each Memory read.	Yes	Yes/Serial EEPROM	0
10	**Local Address Space 1 Prefetch Counter Enable.** When set to 1 and Memory prefetching is enabled, the PCI 9656 prefetches up to the number of Lwords specified in the Prefetch Counter (LBRD1[14:11]). When cleared to 0, the PCI 9656 ignores the count and continues prefetching as follows: • **If PCI Read No Flush mode is enabled (MARBR[28]=1)** – Continues prefetching until the Direct Slave Read FIFO is full, or • **If PCI Read No Flush mode is disabled (MARBR[28]=0)** – Continues prefetching until PCI read completion, at which time the Direct Slave Read FIFO is flushed.	Yes	Yes/Serial EEPROM	0
14:11	**Local Address Space 1 Prefetch Counter.** Number of Lwords to prefetch during Memory Read cycles (0 to 15). A count of zero selects a prefetch of 16 Lwords.	Yes	Yes/Serial EEPROM	0h
31:15	***Reserved.***	Yes	No	0h

Register 11-56. (DMDAC; PCI:FCh, LOC:17Ch) Direct Master PCI Dual Address Cycles 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 cleared to 0, the PCI 9656 performs 32-bit address Direct Master access.	Yes	Yes	0h

Register 11-57. (PCIARB; PCI:100h, LOC:1A0h) PCI Arbiter Control

Bit	Description	Read	Write	Value after Reset
0	**PCI Arbiter Enable.** Value of 0 indicates the PCI Arbiter is disabled and REQ0# and GNT0# are used by the PCI 9656 to acquire PCI Bus use. Value of 1 indicates the PCI Arbiter is enabled.	Yes	Local/ Serial EEPROM	0
1	**PCI 9656 High Priority.** Value of 0 indicates the PCI 9656 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 9656 (when using the internal PCI Arbiter).	Yes	Yes/ Serial EEPROM	0
2	**Early Grant Release.** Value of 0 indicates the PCI 9656 holds the current GNT# asserted until another Master requests use of the PCI Bus. Value of 1 indicates the PCI 9656 always de-asserts the current GNT# when FRAME# is asserted (when using the internal PCI Arbiter).	Yes	Yes/ Serial EEPROM	0
3	**PCI Arbiter Parking on PCI 9656.** Value of 1 indicates the PCI Arbiter parks the grant on the PCI 9656. Value of 0 indicates the PCI Arbiter parks the grant on the current PCI Master (when using the internal PCI Arbiter).	Yes	Yes/ Serial EEPROM	0
31:4	***Reserved.***	Yes	No	0h

Note: *PCIARB* ***cannot*** *be accessed from the PCI Bus by an I/O access, because of the PCI r2.2 PCI I/O space 256-byte limitation.*

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 PCI address where the abort occurred is written to this register.	Yes	No	0h

Note: *PABTADR* ***cannot*** *be accessed from the PCI Bus by an I/O access, because of the PCI r2.2 PCI I/O space 256-byte limitation.*

11.5 RUNTIME REGISTERS

Note: *"Yes" in the register Read and Write columns indicates the register is writable by the PCI and Local Buses.*

Register 11-59. (MBOX0; PCI:40h or 78h, LOC:C0h) Mailbox 0

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.** ***Note:*** *The Inbound Queue Port register (IQP) replaces this register at PCI:40h when I_2O Decode is enabled (QSR[0]=1). MBOX0 is always accessible at PCI address 78h and Local address C0h. Refer to Section 7.1.10 for further details.*	Yes	Yes/Serial EEPROM	0h

Register 11-60. (MBOX1; PCI:44h or 7Ch, LOC:C4h) Mailbox 1

Bit	Description	Read	Write	Value after Reset
31:0	**32-Bit Mailbox Register.** ***Note:*** *The Outbound Queue Port register (OQP) replaces this register at PCI:44h when I_2O Decode is enabled (QSR[0]=1). MBOX1 is always accessible at PCI address 7Ch and Local address C4h. Refer to Section 7.1.10 for further details.*	Yes	Yes/Serial EEPROM	0h

Register 11-61. (MBOX2; PCI:48h, LOC:C8h) Mailbox 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 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 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 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 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 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 output (LINTo#) 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 1 to a particular bit. The Local processor can clear a Doorbell bit by writing 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 (INTA#). The PCI Bus Master can then read this register to determine which Doorbell bit was set. The Local processor sets the doorbell by writing 1 to a particular bit. The PCI Bus Master can clear a Doorbell bit by writing 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 Interrupt Sources (Bit 0).** **M mode:** Writing 1 enables the TEA# pin and enables LINTo# to be asserted upon detection of a Local parity error. **C and J modes:** Writing 1 enables LSERR# to be asserted upon detection of a Local parity error or PCI Abort. ***Note:*** *Refer to Figure 6-1, "Interrupt and Error Sources," for further details.*	Yes	Yes	0
1	**Enable Interrupt Sources (Bit 1).** **M mode:** Writing 1 enables LINTo# to be asserted upon detection of an SERR# assertion in Host mode, or detection of a PCI parity error or a messaging queue outbound overflow. **C and J modes:** Writing 1 enables LSERR# to be asserted upon detection of an SERR# assertion in Host mode, or detection of a PCI parity error or a messaging queue outbound overflow. ***Note:*** *Refer to Figure 6-1, "Interrupt and Error Sources," for further details.*	Yes	Yes	0
2	**Generate PCI Bus SERR# Interrupt.** When cleared to 0, writing 1 asserts the PCI Bus SERR# interrupt.	Yes	Yes	0
3	**Mailbox Interrupt Enable.** Writing 1 enables a Local interrupt output (LINTo#) to be asserted when the PCI Bus writes to MBOX0 through MBOX3. To clear a LINTo# 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 1 enables a Local interrupt output (LINTo#) 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 Management Control/Status register Power State bits (PMCSR[1:0]). Writing 1 clears the interrupt. Writable from the PCI Bus only in the D_0 power state.	Yes	Yes/Clr	0
6	**Direct Master Write/Direct Slave Read Local Data Parity Check Error Enable.** **M Mode:** Writing 1 enables a Local Bus Data Parity Error signal to be asserted through the LINTo# pin. INTCSR[0] must be enabled for this to have an effect. **C and J Modes:** Writing 1 enables a Local Bus Data Parity Error signal to be asserted through the LSERR# 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 9656 has detected a Local data parity check error, even if Parity Check Error is disabled (INTCSR[6]=0). Writing 1 clears this bit to 0.	Yes	Yes/Clr	0

Register 11-69. (INTCSR; PCI:68h, LOC:E8h) Interrupt Control/Status (Continued)

Bit	Description	Read	Write	Value after Reset
8	**PCI Interrupt Enable.** Writing 1 enables PCI interrupts (INTA#).	Yes	Yes	1
9	**PCI Doorbell Interrupt Enable.** Writing 1 enables Local-to-PCI Doorbell interrupts. Used in conjunction with the PCI Interrupt Enable bit (INTCSR[8]). Clearing the L2PDBELL register bits that caused the interrupt also clears the interrupt.	Yes	Yes	0
10	**PCI Abort Interrupt Enable.** Value of 1 enables a Master Abort or Master detection of a Target Abort to assert a PCI interrupt (INTA#). Used in conjunction with the PCI Interrupt Enable bit (INTCSR[8]). Clearing the Received Master and Target Abort bits (PCISR[13:12]) also clears the PCI interrupt.	Yes	Yes	0
11	**Local Interrupt Input Enable.** Writing 1 enables a Local interrupt input (LINTi#) assertion to assert a PCI interrupt (INTA#). Used in conjunction with the PCI Interrupt Enable bit (INTCSR[8]). De-asserting LINTi# also clears the interrupt.	Yes	Yes	0
12	**Retry Abort Enable.** Writing 1 enables the PCI 9656 to treat 256 consecutive Master Retrys to a Target as a Target Abort. Writing 0 enables the PCI 9656 to attempt Master Retrys 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 Master or Target Abort interrupt is active.	Yes	No	0
15	**Local Interrupt Input Active.** When set to 1, indicates the Local interrupt input (LINTi#) is active.	Yes	No	0
16	**Local Interrupt Output Enable.** Writing 1 enables Local interrupt output (LINTo#).	Yes	Yes	1
17	**Local Doorbell Interrupt Enable.** Writing 1 enables PCI-to-Local Doorbell interrupts. Used in conjunction with the Local Interrupt Output Enable bit (INTCSR[16]). Clearing the P2LDBELL register bits that caused the interrupt also clears the interrupt.	Yes	Yes	0
18	**DMA Channel 0 Interrupt Enable.** Writing 1 enables DMA Channel 0 interrupts. Used in conjunction with the DMA Channel 0 Interrupt Select bit (DMAMODE0[17]). Setting the DMA Channel 0 Clear Interrupt bit (DMACSR0[3]=1) also clears the interrupt.	Yes	Yes	0
19	**DMA Channel 1 Interrupt Enable.** Writing 1 enables DMA Channel 1 interrupts. Used in conjunction with the DMA Channel 1 Interrupt Select bit (DMAMODE1[17]). Setting the DMA Channel 1 Clear Interrupt bit (DMACSR1[3]=1) also clears the interrupt.	Yes	Yes	0
20	**Local Doorbell Interrupt Active.** Reading 1 indicates the Local Doorbell interrupt is active.	Yes	No	0
21	**DMA Channel 0 Interrupt Active.** Reading 1 indicates the DMA Channel 0 interrupt is active.	Yes	No	0
22	**DMA Channel 1 Interrupt Active.** Reading 1 indicates the DMA Channel 1 interrupt is active.	Yes	No	0
23	**Built-In Self-Test (BIST) Interrupt Active.** Reading 1 indicates the BIST interrupt is active. The BIST interrupt is enabled by writing 1 to the PCI Built-In Self-Test Interrupt Enable bit (PCIBISTR[6]=1). Clearing the Enable bit (PCIBISTR[6]=0) also clears the interrupt. ***Note:*** *Refer to the PCIBISTR register for a description of the self-test.*	Yes	No	0

Register 11-69. (INTCSR; PCI:68h, LOC:E8h) Interrupt Control/Status (Continued)

Bit	Description	Read	Write	Value after Reset
24	Reading 0 indicates the Direct Master was the Bus Master during a Master or Target Abort.	Yes	No	1
25	Reading 0 indicates that DMA Channel 0 was the Bus Master during a Master or Target Abort.	Yes	No	1
26	Reading 0 indicates that DMA Channel 1 was the Bus Master during a Master or Target Abort.	Yes	No	1
27	Reading 0 indicates that the PCI 9656 asserted a Target Abort after 256 consecutive Master Retrys to a Target.	Yes	No	1
28	Reading 1 indicates that 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 1 indicates that 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 1 indicates that 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 1 indicates that 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	1110b
7:4	**PCI Write Command Code for DMA.**	Yes	Yes	0111b
11:8	**PCI Memory Read Command Code for Direct Master.**	Yes	Yes	0110b
15:12	**PCI Memory Write Command Code for Direct Master.**	Yes	Yes	0111b
16	**General-Purpose Output.** Writing 1 causes USERo output to go high. Writing 0 causes USERo output to go low.	Yes	Yes	1
17	**General-Purpose Input.** Reading 1 indicates the USERi input pin is high. Reading 0 indicates the USERi pin is low.	Yes	No	–
18	**USERi or LLOCKi# Pin Select.** Writing 1 selects USERi to be an input to the PCI 9656. Writing 0 selects LLOCKi# as an input to the PCI 9656.	Yes	Yes	1
19	**USERo or LLOCKo# Pin Select.** Writing 1 selects USERo to be an output from the PCI 9656. Writing 0 selects LLOCKo# as an output from the PCI 9656.	Yes	Yes	1
20	**LINTo# Interrupt Status.** When HOSTEN# is enabled, reading 1 indicates the LINTo# interrupt is active by way of the PCI INTA# interrupt. Writing 1 clears this bit to 0.	Yes	Yes/Clr	0
21	**TEA#/LSERR# Interrupt Status.** When HOSTEN# is enabled, reading 1 indicates the TEA# (M mode) or LSERR# (C and J modes) interrupt is active by way of the SERR# PCI System Error. Writing 1 clears this bit to 0.	Yes	Yes/Clr	0
23:22	***Reserved.***	Yes	No	00b
24	**Serial EEPROM Clock for PCI or Local Bus Reads or Writes to Serial EEPROM (EESK).** Toggling this bit toggles the serial EEPROM clock output.	Yes	Yes	0
25	**Serial EEPROM Chip Select (EECS).** For PCI or Local Bus reads or writes to the serial EEPROM, setting this bit to 1 provides the serial EEPROM chip select output.	Yes	Yes	0
26	**Write Bit to Serial EEPROM (EEDI).** For writes, the EEDI output signal is input to the serial EEPROM. Clocked into the serial EEPROM by the serial EEPROM clock. ***Note:*** *Refer to Section 2.4.3 (M mode) or Section 4.4.3 (C and J modes) for further details.*	Yes	Yes	0
27	**Read Bit from Serial EEPROM (EEDO).** [Refer to Section 2.4.3 (M mode) or Section 4.4.3 (C and J modes).]	Yes	No	–
28	**Serial EEPROM Present.** When set to 1, indicates that a blank or programmed serial EEPROM is present (Serial EEPROM Start bit detected).	Yes	No	0
29	**Reload Configuration Registers.** When cleared to 0, writing 1 causes the PCI 9656 to reload the Local Configuration registers from the serial EEPROM. ***Note:*** *This bit is not self-clearing.*	Yes	Yes	0
30	**Software Reset when HOSTEN#=1.** Writing 1 holds the PCI 9656 Local Bus logic in a reset state, and asserts LRESET# output. Contents of the PCI Configuration registers and the Runtime registers are not reset. A software reset can be cleared only from the PCI Bus. **Software Reset when HOSTEN#=0.** Writing 1 holds the PCI 9656 PCI Bus logic and DMA Configuration registers in a reset state, and asserts RST# output. Contents of the Local Configuration, Runtime, and Messaging Queue registers are not reset. A software reset can be cleared only from the Local Bus.	Yes	Yes	0
31	**EEDO Input Enable.** When set to 1, the EEDI/EEDO I/O buffer is placed in a bus high-impedance state, enabling the serial EEPROM data to be read. The serial EEPROM data resides in CNTRL[27]. Clear to 0 to allow VPD cycles to occur.	Yes	Yes	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 the device. Hardwired to the PLX PCI-SIG-issued Vendor ID, 10B5h.	Yes	No	10B5h
31:16	**Device ID.** Identifies the particular device. Hardwired to the PLX part number for PCI interface chip 9656h.	Yes	No	9656h

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 9656.	Yes	No	Current Rev # (BAh)

11.6 DMA REGISTERS

Note: *"Yes" in the register Read and Write columns indicates the register is writable by the PCI and Local Buses.*

Register 11-73. (DMAMODE0; PCI:80h, LOC:100h) DMA Channel 0 Mode

Bit	Description	Read	Write	Value after Reset
1:0	**DMA Channel 0 Local Bus Data Width.** Writing of the following values indicates the associated bus data width: 00b = 8 bit 01b = 16 bit 10b or 11b = 32 bit	Yes	Yes	11b
5:2	**DMA Channel 0 Internal Wait State Counter (Address-to-Data; Data-to-Data; 0 to 15 Wait States).**	Yes	Yes	0h
6	**DMA Channel 0 TA#/READY# Input Enable.** Writing 1 enables TA# (M mode) or READY# (C and J modes) input. Writing 0 disables TA#/READY# input.	Yes	Yes	1
7	**DMA Channel 0 Continuous Burst Enable.** When bursting is enabled (DMAMODE0[8]=1), writing 1 enables Continuous Burst mode and writing 0 enables Burst-4 mode. In C and J modes, writing 1 additionally enables BTERM# input, which when asserted overrides the READY# input state (if READY# is enabled, DMAMODE0[6]=1). ***Notes:*** *In M mode, this bit is referred to as the "BI mode" bit.* *In C and J modes, this bit is referred to as the "BTERM# Input Enable" bit.* *Refer to Section 2.2.5 (M mode) or Section 4.2.5 (C and J modes) for further details.*	Yes	Yes	0
8	**DMA Channel 0 Local Burst Enable.** Writing 1 enables Local bursting. Writing 0 disables Local bursting.	Yes	Yes	0
9	**DMA Channel 0 Scatter/Gather Mode.** Writing 1 indicates DMA Scatter/Gather mode is enabled. For Scatter/Gather mode, the DMA source and destination addresses and byte count are loaded from memory in PCI or Local Address spaces. Writing 0 indicates DMA Block mode is enabled.	Yes	Yes	0
10	**DMA Channel 0 Done Interrupt Enable.** Writing 1 enables an interrupt when done. Writing 0 disables an interrupt when done. If DMA Clear Count mode is enabled (DMAMODE0[16]=1), the interrupt does not occur until the byte count is cleared.	Yes	Yes	0
11	**DMA Channel 0 Local Addressing Mode.** Writing 1 holds the Local Address Bus constant. Writing 0 indicates the Local Address is incremented.	Yes	Yes	0

Register 11-73. (DMAMODE0; PCI:80h, LOC:100h) DMA Channel 0 Mode (Continued)

Bit	Description	Read	Write	Value after Reset
12	**DMA Channel 0 Demand Mode.** Writing 1 causes the DMA Channel 0 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. The 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	**DMA Channel 0 Memory Write and Invalidate Mode for DMA Transfers.** When set to 1, the PCI 9656 performs Memory Write and Invalidate cycles to the PCI Bus. The PCI 9656 supports Memory Write and Invalidate sizes of 8 or 16 Lwords. The 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 9656 performs Write transfers, rather than Memory Write and Invalidate transfers. Transfers must start and end at cache line boundaries. PCICR[4] must be set to 1.	Yes	Yes	0
14	**DMA Channel 0 EOT# Enable.** Writing 1 enables the EOT# input pin. Writing 0 disables the EOT# input pin. If DMAMODE0[14] and DMAMODE1[14]=00b, the EOT# pin becomes the DMPAF pin.	Yes	Yes	0
15	**DMA Channel 0 Fast/Slow Terminate Mode Select.** **M Mode:** Writing 0 sets the PCI 9656 into Slow Terminate mode. As a result, BDIP# is de-asserted at the nearest 16-byte boundary and the DMA transfer stops. Writing 1 sets the PCI 9656 into Fast Terminate mode, and indicates BDIP# output is disabled. As a result, the PCI 9656 DMA transfer terminates immediately when EOT# (if enabled) is asserted, or during DMA Demand mode when DREQ0# is de-asserted. **C and J Modes:** Writing 0 sets the PCI 9656 into Slow Terminate mode. As a result, BLAST# is asserted on the last Data transfer to terminate the DMA transfer. Writing 1 sets the PCI 9656 into Fast Terminate mode, and indicates the PCI 9656 DMA transfer terminates immediately when EOT# (if enabled) is asserted, or during DMA Demand mode when DREQ0# is de-asserted.	Yes	Yes	0
16	**DMA Channel 0 Clear Count Mode.** Writing 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 1 routes the DMA Channel 0 interrupt to the PCI interrupt (INTA#). Writing 0 routes the DMA Channel 0 interrupt to the Local interrupt output (LINTo#).	Yes	Yes	0
18	**DMA Channel 0 DAC Chain Load.** When set to 1, enables the descriptor to load the PCI Dual Address Cycles value. Otherwise, the descriptor loads the DMADAC0 register contents.	Yes	Yes	0
19	**DMA Channel 0 EOT# End Link.** Used only for DMA Scatter/Gather transfers. Value of 1 indicates that when EOT# is asserted, the DMA transfer ends the current Scatter/Gather link and continues with the remaining Scatter/Gather transfers. Value of 0 indicates that when EOT# is asserted, the DMA transfer ends the current Scatter/Gather transfer and does not continue with the remaining Scatter/Gather transfers.	Yes	Yes	0

Register 11-73. (DMAMODE0; PCI:80h, LOC:100h) DMA Channel 0 Mode (Continued)

Bit	Description	Read	Write	Value after Reset
20	**DMA Channel 0 Ring Management Valid Mode Enable.** Value of 0 indicates the Ring Management Valid bit (DMASIZ0[31]) is ignored. Value of 1 indicates the DMA descriptors are processed only when the Ring Management Valid bit is set (DMASIZ0[31]=1). If the Valid bit is set, the transfer count is 0, and the descriptor is not the last descriptor in the chain. The DMA Channel 0 DMA Controller then moves to the next descriptor in the chain. ***Note:*** *Descriptor Memory fields are re-ordered when this bit is set.*	Yes	Yes	0
21	**DMA Channel 0 Ring Management Valid Stop Control.** Value of 0 indicates the DMA Scatter/Gather controller continuously polls a descriptor with the Valid bit cleared to 0 (invalid descriptor) if Ring Management Valid Mode is enabled (DMAMODE0[20]=1). Value of 1 indicates the Scatter/Gather controller stops polling when the Ring Management Valid bit with a value of 0 is detected (DMASIZ0[31]=0). In this case, the CPU must restart the DMA Channel 0 DMA Controller by setting the Start bit (DMACSR0[1]=1). A pause clearing the Start bit (DMACSR0[1]=0) sets the DMA Done bit (DMACSR0[4]=1).	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	**DMA Channel 0 PCI Address.** Indicates from where in PCI Memory space DMA transfers (reads or writes) start. Value is a physical address.	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	**DMA Channel 0 Local Address.** 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 Channel 0 Transfer Size (Bytes).** Indicates the number of bytes to transfer during a DMA operation.	Yes	Yes	0h
30:23	***Reserved.***	Yes	No	0h
31	**DMA Channel 0 Ring Management Valid.** When Ring Management Valid Mode is enabled (DMAMODE0[20]=1), indicates the validity of this DMA descriptor.	Yes	Yes	0

Register 11-77. (DMADPR0; PCI:90h, LOC:110h) DMA Channel 0 Descriptor Pointer

Bit	Description	Read	Write	Value after Reset
0	**DMA Channel 0 Descriptor Location.** Writing 1 indicates PCI Address space. Writing 0 indicates Local Address space.	Yes	Yes	0
1	**DMA Channel 0 End of Chain.** Writing 1 indicates end of chain. Writing 0 indicates not end of chain descriptor. (Same as DMA Block mode.)	Yes	Yes	0
2	**DMA Channel 0 Interrupt after Terminal Count.** Writing 1 causes an interrupt to be asserted after the terminal count for this descriptor is reached. Writing 0 disables interrupts from being asserted.	Yes	Yes	0
3	**DMA Channel 0 Direction of Transfer.** Writing 1 indicates transfers from the Local Bus to the PCI Bus. Writing 0 indicates transfers from the PCI Bus to the Local Bus.	Yes	Yes	0
31:4	**DMA Channel 0 Next Descriptor Address.** Qword-aligned (DMADPR0[3:0]=0h).	Yes	Yes	0h

Register 11-78. (DMAMODE1; PCI:94h, LOC:114h) DMA Channel 1 Mode

Bit	Description	Read	Write	Value after Reset
1:0	**DMA Channel 1 Local Bus Data Width.** Writing of the following values indicates the associated bus data width: 00b = 8 bit 01b = 16 bit 10b or 11b = 32 bit	Yes	Yes	11b
5:2	**DMA Channel 1 Internal Wait State Counter (Address-to-Data; Data-to-Data; 0 to 15 Wait States).**	Yes	Yes	0h
6	**DMA Channel 1 TA#/READY# Input Enable.** Writing 1 enables TA# (M mode) or READY# (C and J modes) input. Writing 0 disables TA#/READY# input.	Yes	Yes	1
7	**DMA Channel 0 Continuous Burst Enable.** When bursting is enabled (DMAMODE1[8]=1), writing 1 enables Continuous Burst mode and writing 0 enables Burst-4 mode. In C and J modes, writing 1 additionally enables BTERM# input, which when asserted overrides the READY# input state (if READY# is enabled, DMAMODE1[6]=1). ***Notes:*** *In M mode, this bit is referred to as the "BI mode" bit.* *In C and J modes, this bit is referred to as the "BTERM# Input Enable" bit.* *Refer to Section 2.2.5 (M mode) or Section 4.2.5 (C and J modes) for further details.*	Yes	Yes	0
8	**DMA Channel 1 Local Burst Enable.** Writing 1 enables Local bursting. Writing 0 disables Local bursting.	Yes	Yes	0
9	**DMA Channel 1 Scatter/Gather Mode.** Writing 1 indicates DMA Scatter/Gather mode is enabled. For Scatter/Gather mode, the DMA source and destination addresses and byte count are loaded from memory in PCI or Local Address spaces. Writing 0 indicates DMA Block mode is enabled.	Yes	Yes	0
10	**DMA Channel 1 Done Interrupt Enable.** Writing 1 enables an interrupt when done. Writing 0 disables an interrupt when done. If DMA Clear Count mode is enabled (DMAMODE1[16]=1), the interrupt does not occur until the byte count is cleared.	Yes	Yes	0
11	**DMA Channel 1 Local Addressing Mode.** Writing 1 holds the Local Address Bus constant. Writing 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	**DMA Channel 1 Demand Mode.** Writing 1 causes the DMA Channel 1 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. The 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	**DMA Channel 1 Memory Write and Invalidate Mode for DMA Transfers.** When set to 1, the PCI 9656 performs Memory Write and Invalidate cycles to the PCI Bus. The PCI 9656 supports Memory Write and Invalidate sizes of 8 or 16 Lwords. The 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 9656 performs Write transfers, rather than Memory Write and Invalidate transfers. Transfers must start and end at cache line boundaries. PCICR[4] must be set to 1.	Yes	Yes	0
14	**DMA Channel 1 EOT# Enable.** Writing 1 enables the EOT# input pin. Writing 0 disables the EOT# input pin. If DMAMODE0[14] and DMAMODE1[14]=00b, the EOT# pin becomes the DMPAF pin.	Yes	Yes	0
15	**DMA Channel 1 Fast/Slow Terminate Mode Select.** **M Mode:** Writing 0 sets the PCI 9656 into Slow Terminate mode. As a result, BDIP# is de-asserted at the nearest 16-byte boundary and the DMA transfer stops. Writing 1 sets the PCI 9656 into Fast Terminate mode, and indicates BDIP# output is disabled. As a result, the PCI 9656 DMA transfer terminates immediately when EOT# (if enabled) is asserted, or during DMA Demand mode when DREQ1# is de-asserted. **C and J Modes:** Writing 0 sets the PCI 9656 into Slow Terminate mode. As a result, BLAST# is asserted on the last Data transfer to terminate the DMA transfer. Writing 1 sets the PCI 9656 into Fast Terminate mode, and indicates the PCI 9656 DMA transfer terminates immediately when EOT# (if enabled) is asserted, or during DMA Demand mode when DREQ1# is de-asserted.	Yes	Yes	0
16	**DMA Channel 1 Clear Count Mode.** Writing 1 clears the byte count in each Scatter/Gather descriptor when the corresponding DMA transfer is complete.	Yes	Yes	0
17	**DMA Channel 1 Interrupt Select.** Writing 1 routes the DMA Channel 1 interrupt to the PCI interrupt (INTA#). Writing 0 routes the DMA Channel 1 interrupt to the Local interrupt output (LINTo#).	Yes	Yes	0
18	**DMA Channel 1 DAC Chain Load.** When set to 1, enables the descriptor to load the PCI Dual Address Cycles value. Otherwise, the descriptor loads the DMADAC1 register contents.	Yes	Yes	0
19	**DMA Channel 1 EOT# End Link.** Used only for DMA Scatter/Gather transfers. Value of 1 indicates that when EOT# is asserted, the DMA transfer ends the current Scatter/Gather link and continues with the remaining Scatter/Gather transfers. Value of 0 indicates that when EOT# is asserted, the DMA transfer ends the current Scatter/Gather transfer and does not continue with the remaining Scatter/Gather transfers.	Yes	Yes	0

Register 11-78. (DMAMODE1; PCI:94h, LOC:114h) DMA Channel 1 Mode (Continued)

Bit	Description	Read	Write	Value after Reset
20	**DMA Channel 1 Ring Management Valid Mode Enable.** Value of 0 indicates the Ring Management Valid bit (DMASIZ1[31]) is ignored. Value of 1 indicates the DMA descriptors are processed only when the Ring Management Valid bit is set (DMASIZ1[31]=1). If the Valid bit is set, the transfer count is 0, and the descriptor is not the last descriptor in the chain. The DMA Channel 1 DMA Controller then moves to the next descriptor in the chain. ***Note:*** *Descriptor Memory fields are re-ordered when this bit is set.*	Yes	Yes	0
21	**DMA Channel 1 Ring Management Valid Stop Control.** Value of 0 indicates the DMA Scatter/Gather controller continuously polls a descriptor with the Valid bit cleared to 0 (invalid descriptor) if Ring Management Valid Mode is enabled (DMAMODE1[20]=1). Value of 1 indicates the Scatter/Gather controller stops polling when the Ring Management Valid bit with a value of 0 is detected (DMASIZ1[31]=0). In this case, the CPU must restart the DMA Channel 1 DMA Controller by setting the Start bit (DMACSR1[1]=1). A pause clearing the Start bit (DMACSR1[1]=0) sets the DMA Done bit (DMACSR1[4]=1).	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	**DMA Channel 1 PCI Address.** Indicates from where in PCI Memory space DMA transfers (reads or writes) start. Value is a physical address.	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	**DMA Channel 1 Local Address.** 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 Channel 1 Transfer Size (Bytes).** Indicates the number of bytes to transfer during a DMA operation.	Yes	Yes	0h
30:23	***Reserved.***	Yes	No	0h
31	**DMA Channel 1 Ring Management Valid.** When Ring Management Valid Mode is enabled (DMAMODE1[20]=1), indicates the validity of this DMA descriptor.	Yes	Yes	0

Register 11-82. (DMADPR1; PCI:A4h, LOC:124h) DMA Channel 1 Descriptor Pointer

Bit	Description	Read	Write	Value after Reset
0	**DMA Channel 1 Descriptor Location.** Writing 1 indicates PCI Address space. Writing 0 indicates Local Address space.	Yes	Yes	0
1	**DMA Channel 1 End of Chain.** Writing 1 indicates end of chain. Writing 0 indicates not end of chain descriptor. (Same as DMA Block mode.)	Yes	Yes	0
2	**DMA Channel 1 Interrupt after Terminal Count.** Writing 1 causes an interrupt to be asserted after the terminal count for this descriptor is reached. Writing 0 disables interrupts from being asserted.	Yes	Yes	0
3	**DMA Channel 1 Direction of Transfer.** Writing 1 indicates transfers from the Local Bus to the PCI Bus. Writing 0 indicates transfers from the PCI Bus to the Local Bus.	Yes	Yes	0
31:4	**DMA Channel 1 Next Descriptor Address.** Qword-aligned (DMADPR1[3:0]=0h).	Yes	Yes	0h

Register 11-83. (DMACSR0; PCI:A8h, LOC:128h) DMA Channel 0 Command/Status

Bit	Description	Read	Write	Value after Reset
0	**DMA Channel 0 Enable.** Writing 1 enables the channel to transfer data. Writing 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	**DMA Channel 0 Start.** Writing 1 causes the channel to start transferring data if the channel is enabled.	No	Yes/Set	0
2	**DMA Channel 0 Abort.** Writing 1 causes the channel to abort the current transfer. The DMA Channel 0 Enable bit must be cleared (DMACSR0[0]=0). Sets the DMA Channel 0 Done bit (DMACSR0[4]=1) when the abort is complete.	No	Yes/Set	0
3	**DMA Channel 0 Clear Interrupt.** Writing 1 clears DMA Channel 0 interrupts.	No	Yes/Clr	0
4	**DMA Channel 0 Done.** Reading 1 indicates the transfer is complete. The transfer may be complete either because the DMA transfer finished successfully, or that the DMA transfer was aborted when software set the Abort bit (DMACSR0[2]=1). Reading 0 indicates the Channel transfer is not complete.	Yes	No	1
7:5	***Reserved.***	Yes	No	000b

Register 11-84. (DMACSR1; PCI:A9h, LOC:129h) DMA Channel 1 Command/Status

Bit	Description	Read	Write	Value after Reset
0	**DMA Channel 1 Enable.** Writing 1 enables the channel to transfer data. Writing 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	**DMA Channel 1 Start.** Writing 1 causes channel to start transferring data if the channel is enabled.	No	Yes/Set	0
2	**DMA Channel 1 Abort.** Writing 1 causes the channel to abort the current transfer. The DMA Channel 1 Enable bit must be cleared (DMACSR1[0]=0). Sets the DMA Channel 1 Done bit (DMACSR1[4]=1) when the abort is complete.	No	Yes/Set	0
3	**DMA Channel 1 Clear Interrupt.** Writing 1 clears DMA Channel 1 interrupts.	No	Yes/Clr	0
4	**DMA Channel 1 Done.** Reading 1 indicates the transfer is complete. The transfer may be complete either because the DMA transfer finished successfully, or that the DMA transfer was aborted when software set the Abort bit (DMACSR1[2]=1). Reading 0 indicates the Channel transfer is not complete.	Yes	No	1
7:5	***Reserved.***	Yes	No	000b

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.”

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 Qword entries (plus 1, times 2) in the FIFO before requesting the Local Bus for writes. Nybble values 0h through Eh may be used. (Refer to Table 11-7.) (15 - C0PLAF) > C0LPAE.	Yes	Yes	0h
7:4	**DMA Channel 0 Local-to-PCI Almost Empty (C0LPAE).** Number of empty Qword entries (plus 1, times 2) in the FIFO before requesting the Local Bus for reads. Nybble values 0h through Eh may be used. (Refer to Table 11-7.) (15 - C0PLAF) > C0LPAE.	Yes	Yes	0h
11:8	**DMA Channel 0 Local-to-PCI Almost Full (C0LPAF).** Number of full Qword entries (plus 1, times 2) in the FIFO before requesting the PCI Bus for writes. Nybble values 0h through Eh may be used. (Refer to Table 11-7.)	Yes	Yes	0h
15:12	**DMA Channel 0 PCI-to-Local Almost Empty (C0PLAE).** Number of empty Qword entries (plus 1, times 2) in the FIFO before requesting the PCI Bus for reads. Nybble values 0h through Eh may be used. (Refer to Table 11-7.)	Yes	Yes	0h
19:16	**DMA Channel 1 PCI-to-Local Almost Full (C1PLAF).** Number of full Qword entries (plus 1, times 2) in the FIFO before requesting the Local Bus for writes. Nybble values 0h through Eh may be used. (Refer to Table 11-7.) (15 - C1PLAF) > C1LPAE.	Yes	Yes	0h
23:20	**DMA Channel 1 Local-to-PCI Almost Empty (C1LPAE).** Number of empty Qword entries (plus 1, times 2) in the FIFO before requesting the Local Bus for reads. Nybble values 0h through Eh may be used. (Refer to Table 11-7.) (15 - C1PLAF) > C1LPAE.	Yes	Yes	0h
27:24	**DMA Channel 1 Local-to-PCI Almost Full (C1LPAF).** Number of full Qword entries (plus 1, times 2) in the FIFO before requesting the PCI Bus for writes. Nybble values 0h through Eh may be used. (Refer to Table 11-7.)	Yes	Yes	0h
31:28	**DMA Channel 1 PCI-to-Local Almost Empty (C1PLAE).** Number of empty Qword entries (plus 1, times 2) sin the FIFO before requesting the PCI Bus for reads. Nybble values 0h through Eh may be used. (Refer to Table 11-7.)	Yes	Yes	0h

Note: *For all DMATHR 4-bit fields, value of Fh is* ***reserved****.*

Nybble values 0h through Eh may be used to set the number of full or empty Qword entries for each 4-bit field of the DMATHR register, as listed in Table 11-7.

Table 11-7. DMA Threshold Nybble Values

Nybble Value	Setting	Nybble Value	Setting	Nybble Value	Setting
0h	2 Qwords	5h	12 Qwords	Ah	22 Qwords
1h	4 Qwords	6h	14 Qwords	Bh	24 Qwords
2h	6 Qwords	7h	16 Qwords	Ch	26 Qwords
3h	8 Qwords	8h	18 Qwords	Dh	28 Qwords
4h	10 Qwords	9h	20 Qwords	Eh	30 Qwords

Register 11-87. (DMADAC0; PCI:B4h, LOC:134h) DMA Channel 0 PCI Dual Address Cycles 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 cleared to 0h, the PCI 9656 performs a 32-bit address DMA Channel 0 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 cleared to 0h, the PCI 9656 performs a 32-bit address DMA Channel 1 access.	Yes	Yes	0h

11.7 MESSAGING QUEUE (I_2O) REGISTERS

Note: *"Yes" in the register Read and Write columns indicates the register is writable by the PCI and Local Buses.*

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	000b
3	**Outbound Post Queue Interrupt.** Set when the Outbound Post Queue is not empty. Not affected by the Interrupt Mask bit (OPQIM[3]) state.	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	000b
3	**Outbound Post Queue Interrupt Mask.** Writing 1 masks the Outbound Post Queue 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 output (LINTo#) 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 interrupt (LINTo# for M mode, or LSERR# for C and J modes) 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 (INTA#) 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 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: MQCR[5:1] Number of entries Total size 00001b 4-KB entries 64 KB 00010b 8-KB entries 128 KB 00100b 16-KB entries 256 KB 01000b 32-KB entries 512 KB 10000b 64-KB entries 1 MB	Yes	Yes	00001b
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	00b
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	00b
19:2	**Inbound Free Tail Pointer.** Local Memory offset for the Inbound Free Queue. Maintained by the hardware and incremented by the modulo of 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	00b
19:2	**Inbound Post Head Pointer.** Local Memory offset for the Inbound Post Queue. Maintained by the hardware and incremented by the modulo of 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	00b
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	00b
19:2	**Outbound Free Head Pointer.** Local Memory offset for the Outbound Free Queue. Maintained by the hardware and incremented by the modulo of 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	00b
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	00b
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	00b
19:2	**Outbound Post Tail Pointer.** Local Memory offset for the Outbound Post Queue. Maintained by the hardware and incremented by the modulo of 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 to 1, replaces the MBOX0 and MBOX1 registers with the Inbound and Outbound Queue Port registers and redefines Space 1 as the PCI Memory Base Address 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 the PCI Memory Base Address.	Yes	Yes	0
1	**Queue Local Space Select.** When cleared 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 1 causes prefetching to occur from the Outbound Post Queue if the queue is not empty.	Yes	Yes	0
3	**Inbound Free Queue Prefetch Enable.** Writing 1 causes prefetching to occur from the Inbound Free Queue if the queue is not empty.	Yes	Yes	0
4	**Inbound Post Queue Interrupt Not Empty Mask.** Writing 1 masks the Inbound Post Queue Not Empty Interrupt from generating a Local interrupt. Value of 0 clears the mask.	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 (QSR[4]).	Yes	No	0
6	**Outbound Free Queue Overflow Interrupt Full Mask.** When set to 1, masks the Local LINTo# (M mode) or LSERR# (C and J modes) (NMI) 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 LINTo# (M mode) or LSERR# (C and J modes) interrupt is asserted. Writing 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 9656 pin. Tables 12-4 through 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 Tables 12-10 through 12-12 correspond to the PCI 9656 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 PCI 9656 pinout, refer to Section 14, "Physical Specifications."

Table 12-1 lists abbreviations used in this section to represent various pin types.

Table 12-1. Pin Type Abbreviations

Abbreviation	Pin Type
I/O	Input and output
I	Input only
O	Output only
TS	Tri-state
OD	Open drain
PCI	PCI-compliant
TP	Totem pole
STS	Sustained tri-state, driven high for one CLK before float
DTS	Driven tri-state, driven inactive for one-half CLK before float

Note: *Some pins change type when in Reset/BDSEL mode. This is indicated in the Pin Type column for the associated pins. Pins with a pin type that makes no mention of Reset/BDSEL mode have only the one pin type, which is indicated in the Pin Type column entry for that pin, regardless of the silicon mode. For pins that include a Reset/BDSEL mode type, the other type specified is the type in all other silicon modes.*

Reset/BDSEL mode is defined as:

- *HOSTEN# de-asserted and RST# asserted (PCI Reset input in Peripheral mode), or*
- *HOSTEN# asserted and LRESET# asserted (Local Bus reset input in Host mode), or*
- *BD_SEL# de-asserted (CompactPCI Hot Swap Precharge mode).*

12.2 PULL-UP AND PULL-DOWN RESISTORS

IDDQEN# has an internal 100KΩ pull-down resistor. Each pin listed in Table 12-2 has an internal 100KΩ pull-up resistor to V_{RING}.

Table 12-2. Pins with Internal Pull-Up Resistors

Mode	Pin
All	CCS#, DACK[1:0]#, DREQ[1:0]#, EEDI/EEDO, HOSTEN#, LINTi#, LINTo#, LRESET#, PMEREQ#
M only	BDIP#, BI#, BIGEND#/WAIT#, BURST#, DP[0:3], LA[0:1, 3:31], LD[0:31], MDREQ#/DMPAF/EOT#, RD/WR#, TA#, TEA#, TS#, TSIZ[0:1]
C and J only	ADS#, BIGEND#, BLAST#, BTERM#, DMPAF/EOT#, DP[3:0], LA[28:2], LBE[3:0]#, LSERR#, LW/R#, READY#, WAIT#
C only	LA[31:30], LD[31:0]
J only	DEN#, DT/R#, LAD[31:0]

Due to the weak value of the internal pull-up resistors for all pins listed in Table 12-2, it is strongly advised that external pull-up to V_{RING} or pull-down resistors (3KΩ to 10KΩ are recommended) be used on those signal pins when implementing them in a design. However, depending upon the application, some signal pins listed in Table 12-2 may remain unconnected, driven or tied, high or low. External pull-up or pull-down resistors on the Address and/or Data Buses are recommended, but optional.

Although the EEDI/EEDO pin has an internal pull-up resistor, it may be necessary to add an external pull-up or pull-down resistor. [Refer to Table 2-12, "Serial EEPROM Guidelines" (M mode), or Table 4-14, "Serial EEPROM Guidelines" (C and J modes), for further details.]

The PCI pins listed in Table 12-4 have no internal resistors. Refer to *PCI r2.2* for external pull-up requirements. If the PME# pin is not used, it requires an external pull-up resistor; otherwise, refer to Section 8, “PCI Power Management.”

Table 12-3 lists pins that do not have internal resistors.

Table 12-3. Pins with No Internal Resistors

Mode	Pin
All	BD_SEL#, CPCISW, EECS, EESK, ENUM#, LCLK, LEDon#, MODE[1:0], 64EN#, TCK, TDI, TDO, TRST#, TMS, USERi/LLOCKi#, USERo/LLOCKo#
M only	BB#, BG#, BR#, LA2, RETRY#
C and J only	BREQi, BREQo, LHOLD, LHOLDA
C only	LA29
J only	ALE

Because the Open Drain (OD) pins (*for example*, LEDon# and ENUM#) are created using I/O buffers, they must be pulled-up to ensure their input buffer is at a known state. The same is true for some of the Output (O) pins (*for example*, RETRY#/BREQo), which are tri-stated when not driven (STS).

Recommendations for the multiplexed pins listed in Table 12-3 are as follows:

- **Pin A17, RETRY# (M mode) or BREQo (C or J mode)** – RETRY# requires an external pull-up resistor. BREQo requires an external pull-down resistor.
- **Pin B14, USERi/LLOCKi#** – Refer to Section 2.4.2 (M mode) or Section 4.4.2 (C and J modes).
- **Pin B17, BG# (M mode) or LHOLDA (C or J mode)** – BG# is an input pin that should be pulled high, or driven high or low. LHOLDA is an input pin that should be pulled low or driven high or low.
- **Pin B18, BR# (M mode) or LHOLD (C or J mode)** – BR# is an output pin that is always driven by the PCI 9656. LHOLD is an output pin that floats during reset or BD_SEL# assertion Therefore, an external pull-up or pull-down resistor is ***not*** required.
- **Pin C14, USERo/LLOCKo#** – An external pull-up or pull-down resistor is ***not*** required.
- **Pin C16, BB# (M mode) or BREQi (C or J mode)** – BB# requires an external pull-up resistor (a value of 4.7KΩ or less is recommended). BREQi should be pulled, tied, or driven low.
- **Pin V14, LA[2] (M mode), LA[29] (C mode), or ALE (J mode)** – LA[2] or LA[29] requires an external pull-up resistor. ALE requires an external pull-down resistor.

The following outlines recommendations for the non-multiplexed pins listed in Table 12-3.

The MODE[1:0] pins should be tied high or low. The LCLK pin does ***not*** require an external pull-up nor pull-down resistor.

If the internal PCI Arbiter is not used (disabled), external pull-up resistors are required for the REQ[6:1]# input and GNT[6:1]# output pins. If the internal PCI Arbiter is used (enabled), the REQ[6:1]# pins must be driven or pulled to a known state. When the PCI Arbiter is enabled, the GNT[6:1]# output pins do ***not*** require external pull-up resistors, and may be connected or remain unconnected (floating).

For non-CompactPCI systems, tie BD_SEL# to ground, connect ENUM# and 64EN# to an external pull-up resistor, and connect CPCISW to a pull-up resistor tied to V_{RING} to simulate an unlocked switch or to a pull-down resistor to simulate a locked switch. Because LEDon# is driven while the PCI RST# signal is asserted, LEDon# requires its own external pull-up resistor. For CompactPCI Hot Swap implementations, refer to *PICMG 2.1 R2.0* for external logic and resistor requirements for these pins.

Serial EEPROM interface pins EECS and EESK are outputs that are always driven by the PCI 9656 and may be connected or remain unconnected (floating).

For designs that do not implement JTAG, ground the TRST# pin, drive, pull, or tie the TCK, TDI, and TMS inputs to a known value, and pull the TDO output up with an external pull-up resistor. *IEEE Standard 1149.1-1990* requires pull-up resistors on the TDI, TMS, and TRST# pins. To remain *PCI r2.2*-compliant, no internal pull-up resistors are provided on JTAG pins in the PCI 9656; therefore, the pull-up resistors must be externally added to the PCI 9656 when implementing JTAG.

12.3 PINOUT COMMON TO ALL BUS MODES

Table 12-4. PCI System Bus Interface Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
ACK64#	64-Bit Transfer Acknowledge	1	I/O STS PCI	N1	The PCI Slave asserts ACK64# in response to REQ64# to acknowledge a 64-bit Data transfer.
AD[63:0]	Address and Data	64	I/O TS PCI	T4, U3, W1, V3, Y2, W4, V4, U5, Y3, Y4, V5, W5, Y5, V6, U7, W6, Y6, V7, W7, Y7, V8, W8, Y8, V9, W9, Y9, W10, V10, Y10, Y11, W11, V11, A5, D7, C6, B5, A4, C5, B4, A3, B3, B2, A2, C3, B1, C2, D2, D3, H3, H2, H1, J4, J3, J2, J1, K2, K1, L2, L3, L4, M1-M4	PCI multiplexed Address/Data Bus.
C/BE[7:0]#	Bus Command and Byte Enables	8	I/O TS PCI	T2, U1, T3, U2, D5, E4, G1, K3	PCI multiplexed Command and Byte Enable pins. During the Address phase of a transaction, defines the bus command. During the Data phase, used as Byte Enables.
DEVSEL#	Device Select	1	I/O STS PCI	E1	When asserted (output), indicates that the driving device has decoded its address as the Target of the current access. As an input, indicates that a slave device on the bus is selected.
FRAME#	Cycle Frame	1	I/O STS PCI	C1	Driven by the current Master to indicate beginning and duration of an access. Asserted to indicate a bus transaction is beginning, and while asserted, Data transfers may continue. When de-asserted, the transaction is in the final Data phase.
GNT0# REQ#	Internal Arbiter Grant 0 External Arbiter Request	1	O TP PCI ***Note:*** *If [(RST# asserted) or (BD_SEL# de-asserted)], pin goes Hi-Z.*	C7	Multiplexed output pin. GNT0#: When the internal PCI Arbiter is enabled (PCIARB[0]=1), the PCI GNT0# signal is an output to an arbitrating master. The PCI 9656 arbiter asserts GNT0# to grant the PCI Bus to the Master. REQ#: When the internal PCI Arbiter is disabled (PCIARB[0]=0), GNT0# becomes the REQ# output from the PCI 9656 to an external arbiter. The PCI 9656 asserts REQ# to request the PCI Bus.

Table 12-4. PCI System Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
GNT[6:1]#	Internal Arbiter Grant 6 to 1	6	O TP PCI ***Note:*** *If [(RST# asserted)* *or* *(BD_SEL# de-asserted)* *or* *(PCIARB[0]=0)],* *pins go Hi-Z.*	W12, U11, P4, R2, R1, N3	When the internal PCI Arbiter is enabled (PCIARB[0]=1), the PCI GNT[6:1]# signals are outputs, one each to an arbitrating master. The PCI 9656 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. If the PCI Arbiter is disabled, pull-up resistors are required (recommended value 1KΩ to 4.7KΩ).*
IDSEL	Initialization Device Select	1	I PCI	C4	Used as a chip select during Configuration Read and Write transactions.
INTA#	Interrupt A	1	I/O OD PCI	B7	As an input, in Host mode, causes LINTo# assertion when asserted. As an output, in Adapter mode, driven by the PCI 9656 to generate a PCI Interrupt request.
IRDY#	Initiator Ready	1	I/O STS PCI	D1	Indicates the ability of the initiating agent (Bus Master) to complete the current Data phase of the transaction.
LOCK#	Lock	1	I/O STS PCI	G4	Indicates an atomic operation that may require multiple transactions to complete.
PAR	Parity	1	I/O TS PCI	G2	Even parity across AD[31:0] and C/BE[3:0]#. 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.
PAR64	Upper 32 Bits Parity	1	I/O TS PCI	V1	Even parity across AD[63:32] and C/BE[7:4]#. PAR64 is stable and valid one clock after the Address phase. For Data phases, PAR64 is stable and valid one clock after either IRDY# is asserted on a Write transaction or TRDY# is asserted on a Read transaction. Once PAR64 is valid, it remains valid until one clock after the current Data phase completes.
PCLK	PCI Clock	1	I PCI	L1	Provides timing for all transactions on PCI and is an input to every PCI device. The PCI 9656 PCI Bus operates up to 66 MHz. ***Note:*** *On Expansion boards, trace length for the PCI PCLK signal must be 2.5 inches ±0.1 inches, and must be routed to only one load, per PCI r2.2.*
PERR#	Parity Error	1	I/O STS PCI	F2	Reports data parity errors during all PCI transactions, except during a special cycle.
PME#	Power Management Event	1	O OD PCI	B9	Asserted to alert system to a Power Management Event (PME). ***Note:*** *If PME# is implemented, a field-effect transistor (FET) should be used to isolate the signal when power is removed from the card. (Refer to PCI Power Mgmt. r1.1.)*

Table 12-4. PCI System Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
REQ0# GNT#	Internal Arbiter Request 0 External Arbiter Grant	1	I PCI	B6	Multiplexed input pin. REQ0#: When the internal PCI Arbiter is enabled (PCIARB[0]=1), the PCI REQ0# signal is an input from an arbitrating master. REQ0# is asserted to the PCI 9656 arbiter by the Master to request the PCI Bus. GNT#: When the internal PCI Arbiter is disabled (PCIARB[0]=0), REQ0# becomes the GNT# input to the PCI 9656 from an external arbiter. The arbiter asserts GNT# to grant the PCI Bus to the PCI 9656. REQ[6:1]# are not used.
REQ[6:1]#	Internal Arbiter Request 6 to 1	6	I PCI	V12, Y12, R3, T1, P2, P1	When the internal PCI Arbiter is enabled (PCIARB[0]=1), the PCI REQ[6:1]# signals are inputs, one each from an arbitrating master. REQ[6:1]# is asserted to the PCI 9656 arbiter by the corresponding master to request the PCI Bus. ***Notes:*** *If the PCI Arbiter is disabled, pull-up resistors are required (recommended value 1KΩ to 4.7KΩ).* *The PCI Arbiter pins are type "I", independent of the PCI Arbiter Enable bit.*
REQ64#	64-Bit Transfer Request	1	I/O STS PCI	N2	When in Adapter mode, the PCI 9656 samples REQ64# at RST# de-assertion, to configure a 32- or 64-bit PCI backplane. When in Host mode, the PCI 9656 asserts REQ64# during RST# to configure a 64-bit PCI backplane. In either mode, asserted with FRAME# by the PCI Master to request a 64-bit Data transfer. ***Note:*** *If the PCI backplane is configured as a 32-bit PCI width (REQ64#=1 during RST# de-assertion), REQ64# assertion in subsequent transfers is illegal.*
RST#	Reset	1	*If (HOSTEN# asserted)* O TP PCI *else* I PCI	A6	As an input, in Adapter mode, RST# assertion resets the entire PCI 9656. LRESET# is asserted as long as RST# is asserted. As an output, in Host mode, LRESET# assertion causes an RST# assertion. RST# assertion causes the entire PCI Bus to be reset.
SERR#	Systems Error	1	I/O OD PCI	G3	As an input, in Host mode, SERR# assertion causes TEA#/LSERR# to be generated whenever a PCI error occurs. As an output, SERR# reports address parity errors or any other system error that indicates catastrophic results.

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	F3	Indicates that the current Target is requesting the Master to stop the current transaction.
TRDY#	Target Ready	1	I/O STS PCI	E3	Indicates ability of the Target agent (selected device) to complete the current Data phase of the transaction.
Total		103			

Table 12-5. JTAG Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
TCK	Test Clock Input	1	I PCI	A8	Clock source for the PCI 9656 Test Access Port (TAP). TCK may be any frequency from 0 to 10 MHz.
TDI	Test Data Input	1	I PCI	A7	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	C8	Used to transmit serial data from the PCI 9656 TAP. Data from the selected shift register is shifted out of TDO.
TMS	Test Mode Select	1	I PCI	B8	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 PCI	D9	Resets JTAG. Should be toggled or held at 0 for the PCI 9656 to properly function.
Total		5			

Table 12-6. CompactPCI Hot Swap Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
64EN#	CompactPCI 64-Bit Enable	1	I PCI	U12	Input that enables 64-bit CompactPCI operation. For non-CompactPCI systems, this pin should be pulled high.
BD_SEL#	CompactPCI Board Select	1	I PCI	C9	CompactPCI board select for Hot Swap system. For non-CompactPCI systems, this pin should be grounded.
CPCISW	CompactPCI Switch Sense	1	I PCI	Y14	Input that monitors CompactPCI board latch status. For non-CompactPCI systems, tie this pin to a pull-up resistor tied to V_{RING} to simulate an unlocked switch or to a pull-down resistor to simulate a locked switch.
ENUM#	Enumeration	1	O OD PCI	Y13	Interrupt output asserted when an adapter using the PCI 9656 has been inserted or is ready to be removed from a PCI slot.
LEDon#	CompactPCI LED On	1	O OD 24 mA	V13	Hot Swap board indicator LED. Controlled by the LED Software On/Off Switch bit (HS_CSR[3]). Asserted during PCI reset.
Total		5			

Table 12-7. System Pins

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
IDDQEN#	Double Multiplexed Function Signal	1	I	A10	Multiplexed input pin, with two functions: • Identifies main power present status for the Power Management D_{3cold} function. (Refer to Section 8.1.3, “D_{3cold} Power State Support.”) • As an IDDQ enable, puts the PCI 9656 output buffers into a quiescent state. Asserting IDDQEN# to logic low (0) along with BD_SEL# to logic high (1) forces the PCI 9656 into a quiescent state. All analog power is disabled and I/Os are tri-stated.
MODE[1:0]	Bus Mode	2	I	A19, A20	Selects the PCI 9656 bus operation mode: **Mode 0** **Mode 1** **Bus Mode** 1 1 M 0 0 C 1 0 J 0 1 ***Reserved*** ***Note:*** *The MODE input level must be stable at power-on.*
HOSTEN#	Host Enable	1	I	C15	HOSTEN# assertion (Host mode) configures the PCI 9656 as a host bridge, setting reset and interrupt signal directions for system board applications. *For example*, PCI RST# is an output. HOSTEN# de-assertion (Adapter mode) configures the PCI 9656 as a peripheral bridge, setting reset and interrupt signal directions for peripheral board applications. *For example*, PCI RST# is an input.
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 ***Note:*** *If (BD_SEL# not asserted), pin goes Hi-Z.*	B12	Serial EEPROM chip select.
EEDI/EEDO	Serial EEPROM Data In/ Serial EEPROM Data Out	1	I/O TS 12 mA ***Note:*** *If (CNTRL[31]=1), pin goes Hi-Z.*	B11	Multiplexed Write and Read data to the serial EEPROM pin.
EESK	Serial EEPROM Clock	1	O TP 12 mA ***Note:*** *If (BD_SEL# not asserted), pin goes Hi-Z.*	A12	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	D10	2.5V to PME logic during the D_{3cold} power state. For Power Management systems, connect directly to 2.5V regulated auxiliary power from Card_V_{AUX}. Otherwise, connect directly to 2.5V.
Card_V_{AUX}	PME 3.3V D_{3cold} Power	1	I	C10	3.3V to PME logic during the D_{3cold} power state. (Refer to *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	A11	When sampled as 1, $3.3V_{AUX}$ power is present and PME# assertion in the D_{3cold} power state is supported. When sampled as 0, $3.3V_{AUX}$ power is not present and PME# assertion in the D_{3cold} power state is not supported by PMC[15]. (Refer to *PCI Power Mgmt. r1.1*, Figure 12.) For non-Power Management systems, connect directly to ground.
V_{CORE}	Core Power	6	I	C11, C19, E2, P3, U9, U19	2.5V to core.
V_{IO}	PCI System Voltage	4	I	A9, F1, V2, W13	System signaling environment voltage select, 3.3 or 5V, from the PCI Bus.
V_{RING}	I/O Ring Power	26	I	A1, D4, D6, D8, D11, D13, D15, D17, F4, F17, H4, H17, K4, L17, N4, N17, R4, R17, U4, U6, U8, U10, U13, U15, U17, W2	3.3V to I/O ring.
V_{SS}	Ground	18	I	J9-J12, K9-K12, L9-L12, M9-M12, W3, Y1	Ground.
Total		57			

12.4 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 DTS 24 mA	C16	As an input, the PCI 9656 monitors BB# to determine whether an external Master has ended a Bus cycle. As an output, the PCI 9656 asserts BB# after an external Local Bus Arbiter has granted ownership of the Local Bus and BB# is detected de-asserted. It is recommended that an external pull-up resistor value of 4.7KΩ be applied to guarantee a fast transition to the inactive state when the PCI 9656 relinquishes Local Bus ownership.
BDIP#	Burst Data in Progress	1	I/O TS 24 mA	B16	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 9656 during the Data phase of a Burst transaction. The PCI 9656 de-asserts before the last Burst Data phase on the bus.
BG#	Bus Grant	1	I	B17	The external Local Bus Arbiter asserts BG# in response to BR#. Indicates that the requesting Master is next.
BI#	Burst Inhibit	1	I	C20	When asserted, indicates that the Target device does not support Burst transactions.
BIGEND# WAIT#	Big Endian Select WAIT I/O	1	*If (MARBR[31]=0)* I *else* I/O TS 24 mA	C12	Multiplexed I/O pin. Default functionality is BIGEND#, Big Endian input (MARBR[31])=0). BIGEND# (Input): 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# (I/O): As an input, the Local Bus Master may assert WAIT# to pause the PCI 9656 (insert wait states) during a Direct Master access Data phase. As an output, the PCI 9656 may be programmed to insert wait states (to pause the Local Slave) by asserting WAIT# for a predefined number of Local Bus clock cycles during Direct Slave transfers.

Table 12-10. M Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
BR#	Bus Request	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pin goes Hi-Z.*	B18	The PCI 9656 asserts BR# to request use of the Local Bus. The external Local Bus Arbiter asserts BG# when the Master is next in line for bus ownership.
BURST#	Burst	1	I/O TS 24 mA	A18	As an input, driven by the Master along with address and data, indicating that a Burst transfer is in progress. As an output, driven by the PCI 9656 along with address and data, indicating that a Burst transfer is in progress.
CCS#	Configuration Register Select	1	I	D12	Internal PCI 9656 registers are selected when CCS# is asserted low during Local Bus accesses to the PCI 9656.
DACK0#	DMA Channel 0 Demand Mode Acknowledge	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pins go Hi-Z.*	A13	When DMA Channel 0 is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is in process.
DACK1#	DMA Channel 1 Demand Mode Acknowledge	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pins go Hi-Z.*	C13	When DMA Channel 1 is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is in process.
DP[0:3]	Data Parity	4	I/O TS 24 mA	D18, B20, C18, B19	Parity is even for each of up to four byte lanes on the Local Bus. Parity is checked for writes to or on reads by the PCI 9656. Parity is asserted for reads from or writes by the PCI 9656.

Table 12-10. M Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
DREQ0#	DMA Channel 0 Demand Mode Request	1	I	B13	When DMA Channel 0 is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request.
DREQ1#	DMA Channel 1 Demand Mode Request	1	I	A14	When DMA Channel 1 is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request.
LA[0:31]	Local Address Bus	32	I/O TS 24 mA	W14, Y15, V14, W15, Y16, U14, V15, W16, Y17, V16, W17, Y18, U16, V17, W18, Y19, V18, W19, Y20, W20, V19, U18, T17, V20, U20, T18, T19, T20, R18, P17, P19, P20	Carries the 32 bits of the physical Address Bus.
LCLK	Local Processor Clock	1	I	D20	Local clock input.
LD[0:31]	Local Data Bus	32	I/O TS 24 mA	N18, N19, N20, M17, M18, M19, M20, L19, L18, L20, K20, K19, K18, K17, J20, J19, J18, J17, H20, H19, H18, G20, G19, F20, G18, F19, E20, G17, F18, E19, E18, D19	When the PCI 9656 is the Local Bus Master, it carries 8-, 16-, or 32-bit data quantities, depending upon bus data-width configuration. All Master accesses to the PCI 9656 are 32 bits only.
LINTi#	Local Interrupt Input	1	I	B15	When asserted, causes a PCI interrupt (INTA#).
LINTo#	Local Interrupt Output	1	O OD 24 mA	A15	Synchronous output that remains asserted as long as the interrupt is enabled and the interrupt condition exists.
LRESET#	Local Bus Reset	1	*If (HOSTEN# asserted)* I *else* O TP 24 mA	A16	As an input, in Host mode, RST# is generated as long as LRESET# is asserted. Resets the PCI 9656. As an output, in Adapter mode, LRESET# is asserted when the PCI 9656 is in reset (RST#=0). May be used to reset the back-end logic on the board.

Table 12-10. M Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
MDREQ# DMPAF EOT#	IDMA Data Transfer Request Direct Master Programmable Almost Full End of Transfer for Current DMA Channel	1	*If (DMAMODE0[14]=1 and/or DMAMODE1[14]=1)* I *else* O TP 24 mA ***Note for 2nd case:*** *If [(HOSTEN# not asserted and RST# asserted) or (HOSTEN# and LRESET# asserted) or BD_SEL# not asserted)], pin goes Hi-Z.*	D14	Multiplexed I/O pin. Default functionality is MDREQ#/DMPAF output (DMAMODE0[14] and DMAMODE1[14]=00b). MDREQ# (Output): IDMA M mode Data transfer request start. When asserted, indicates that Data transfer should start. De-asserted only when the number of Direct Master FIFO entries exceed the value programmed in DMPBAM[10, 8:5]. DMPAF (Output): Direct Master Write FIFO Almost Full status output. Programmable through DMPBAM[10, 8:5]. EOT# (Input): 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	B10	Asserted during a D_{3cold} power state to request a Power Management Event.
RD/WR#	Read/Write	1	I/O TS 24 mA	R19	Asserted high for reads and low for writes.
RETRY#	Retry	1	O DTS 24 mA	A17	Driven by the PCI 9656 when it is a slave to indicate the Local Master must release the bus.
TA#	Transfer Acknowledge	1	I/O DTS 24 mA	E17	As an input, when the PCI 9656 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 9656, indicates a Write Data transfer is complete or that Read data on the bus is valid.
TEA#	Transfer Error Acknowledge	1	I/O OD 24 mA	D16	Driven by the Target device, indicating that an error condition occurred during a Bus cycle.
TS#	Address Strobe	1	I/O TS 24 mA	C17	A Local master asserts TS# to start a Bus access.
TSIZ[0:1]	Transfer Size	2	I/O TS 24 mA	R20, P18	Driven by current Master along with the address, indicating the data-transfer size. (Refer to Section 3.4.2.9.4 for further details.)

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	B14	Multiplexed input pin. Default functionality is USERi (CNTRL[18]=1). USERi: General-purpose input that can be read in the PCI 9656 Configuration registers. Also used to select the PCI 9656 behavior in response to PCI accesses during initialization. (Refer to Section 2.4.2.) LLOCKi#: Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9656 for direct Local access to the PCI Bus.
USERo LLOCKo#	User Output Local Lock Output	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pin goes Hi-Z.*	C14	Multiplexed output pin. Default functionality is USERo (CNTRL[19]=1). USERo: General-purpose output controlled from the PCI 9656 Configuration registers. LLOCKo#: Indicates an atomic operation for a Direct Slave PCI-to-Local Bus access may require multiple transactions to complete.
Total		95			

12.5 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	C17	Indicates the valid address and start of a new Bus access. ADS# is asserted for the first clock of the Bus access.
BIGEND#	Big Endian Select	1	I	C12	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.
BLAST#	Burst Last	1	I/O TS 24 mA	A18	As an input, the Local Bus Master asserts BLAST# to indicate the last Data transfer of a Bus access. As an output, the PCI 9656 asserts BLAST# to indicate the last Data transfer of the Bus access.
BREQi	Bus Request In	1	I	C16	Asserted to indicate a Local Bus Master requires the bus. If enabled through the PCI 9656 Configuration registers, the PCI 9656 releases the bus during a Direct Slave or DMA transfer if BREQi is asserted.
BREQo	Bus Request Out	1	O DTS 24 mA	A17	If the Backoff Timer expires, the PCI 9656 asserts BREQo until it is granted the Local Bus.
BTERM#	Burst Terminate	1	I/O DTS 24 mA	C20	As an input, BTERM# assertion causes the PCI 9656 (as the Local Master) to break the transfer (typically a burst). If the transfer is not completed, the PCI 9656 generates a new Address cycle and continues the transfer. As an output, the PCI 9656 (as a Local target) only asserts BTERM# to request the Master to break the transfer if a PCI Abort condition is detected.
CCS#	Configuration Register Select	1	I	D12	Internal PCI 9656 registers are selected when CCS# is asserted low during Local Bus accesses to the PCI 9656.

Table 12-11. C 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 ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pins go Hi-Z.*	A13	When DMA Channel 0 is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is in process.
DACK1#	DMA Channel 1 Demand Mode Acknowledge	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pins go Hi-Z.*	C13	When DMA Channel 1 is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is in process.
DMPAF EOT#	Direct Master Programmable Almost Full End of Transfer for Current DMA Channel	1	*If (DMAMODE0[14]=1 and/or DMAMODE1[14]=1)* I *else* O TP 24 mA ***Note for 2nd case:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pin goes Hi-Z.*	D14	Multiplexed I/O pin. Default functionality is DMPAF output (DMAMODE0[14] and DMAMODE1[14]=00b). DMPAF (Output): Direct Master Write FIFO Almost Full status output. Programmable through DMPBAM[10, 8:5]. EOT# (Input): 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	D18, B20, C18, B19	Parity is even for each of up to four byte lanes on the Local Bus. Parity is checked for writes to or on reads by the PCI 9656. Parity is asserted for reads from or writes by the PCI 9656.
DREQ0#	DMA Channel 0 Demand Mode Request	1	I	B13	When DMA Channel 0 is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request.

Table 12-11. C Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
DREQ1#	DMA Channel 1 Demand Mode Request	1	I	A14	When DMA Channel 1 is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request.
LA[31:2]	Local Address Bus	30	I/O TS 24 mA	W14, Y15, V14, W15, Y16, U14, V15, W16, Y17, V16, W17, Y18, U16, V17, W18, Y19, V18, W19, Y20, W20, V19, U18, T17, V20, U20, T18, T19, T20, R18, P17	Carries the upper 30 bits of the physical Address Bus. Incremented during bursts to indicate successive Data cycles.
LBE[3:0]#	Local Byte Enables	4	I/O TS 24 mA	R20, P18, P19, P20	Encoded, based on the bus data-width configuration, as follows: **32-Bit Bus** The four Byte Enables indicate which of the four bytes are valid 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)
LCLK	Local Processor Clock	1	I	D20	Local clock input.
LD[31:0]	Local Data Bus	32	I/O TS 24 mA	N18, N19, N20, M17, M18, M19, M20, L19, L18, L20, K20, K19, K18, K17, J20, J19, J18, J17, H20, H19, H18, G20, G19, F20, G18, F19, E20, G17, F18, E19, E18, D19	When the PCI 9656 is the Local Bus Master, it carries 8-, 16-, or 32-bit data quantities, depending upon bus data-width configuration. All Master accesses to the PCI 9656 are 32 bits only.

Table 12-11. C Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LHOLD	Local Hold Request	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)], pin goes Hi-Z.*	B18	Asserted to request use of the Local Bus.
LHOLDA	Local Hold Acknowledge	1	I	B17	The external Local Bus Arbiter asserts LHOLDA when bus ownership is granted in response to LHOLD. The Local Bus should not be granted to the PCI 9656, unless requested by LHOLD.
LINTi#	Local Interrupt Input	1	I	B15	When asserted, causes a PCI interrupt (INTA#).
LINTo#	Local Interrupt Output	1	O OD 24 mA	A15	Synchronous output that remains asserted as long as the interrupt is enabled and the interrupt condition exists.
LRESET#	Local Bus Reset	1	*If (HOSTEN# asserted)* I *else* O TP 24 mA	A16	As an input, in Host mode, RST# is generated as long as LRESET# is asserted. Resets the PCI 9656. As an output, in Adapter mode, LRESET# is asserted when the PCI 9656 is in reset (RST#=0). May be used to reset the back-end logic on the board.
LSERR#	Local System Error Interrupt Output	1	O OD 24 mA	D16	Synchronous output that remains asserted as long as the interrupt is enabled and the interrupt condition exists.
LW/R#	Local Write/Read	1	I/O TS 24 mA	R19	Asserted low for reads and high for writes.
PMEREQ#	PME Request	1	I	B10	Asserted during a D_{3cold} power state to request a Power Management Event.
READY#	Ready I/O	1	I/O DTS 24 mA	E17	A Local slave asserts READY# to indicate 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(s) expires (WAIT# output de-asserted).

Table 12-11. C Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
USERi LLOCKi#	User Input Local Lock Input	1	I	B14	Multiplexed input pin. Default functionality is USERi (CNTRL[18]=1). USERi: General-purpose input that can be read in the PCI 9656 Configuration registers. Also used to select the PCI 9656 behavior in response to PCI accesses during initialization. (Refer to Section 4.4.2.) LLOCKi#: Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9656 for direct Local access to the PCI Bus.
USERo LLOCKo#	User Output Local Lock Output	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pin goes Hi-Z.*	C14	Multiplexed output pin. Default functionality is USERo (CNTRL[19]=1). USERo: General-purpose output controlled from the PCI 9656 Configuration registers. LLOCKo#: Indicates an atomic operation for a Direct Slave PCI-to-Local Bus access may require multiple transactions to complete.
WAIT#	Wait I/O	1	I/O TS 24 mA	B16	As an input, the Local Bus Master may assert WAIT# to pause the PCI 9656 (insert wait states) during a Direct Master access Data phase. As an output, the PCI 9656 may be programmed to insert wait states (to pause the Local Slave) by asserting WAIT# for a predefined number of Local Bus clock cycles during Direct Slave transfers.
Total		95			

12.6 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	C17	Indicates a valid address and start of a new Bus access. ADS# asserts for the first clock of the Bus access.
ALE	Address Latch Enable	1	I/O TS 24 mA	V14	Indicates the valid address and start of a new Bus access. ALE asserts for the first clock of the Bus access. As an input, the PCI 9656 latches the incoming address on the positive edge of LCLK, following ALE assertion. The ALE pulse should be similar to the example provided in Figure 13-3, “PCI 9656 ALE Output Delay to the Local Clock.” As an output, refer to Figure 13-3.
BIGEND#	Big Endian Select	1	I	C12	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.
BLAST#	Burst Last	1	I/O TS 24 mA	A18	As an input, the Local Bus Master asserts BLAST# to indicate the last Data transfer of a Bus access. As an output, the PCI 9656 asserts BLAST# to indicate the last Data transfer of the Bus access.
BREQi	Bus Request In	1	I	C16	Asserted to indicate a Local Bus Master requires the bus. If enabled through the PCI 9656 Configuration registers, the PCI 9656 releases the bus during a Direct Slave or DMA transfer if BREQi is asserted.
BREQo	Bus Request Out	1	O DTS 24 mA	A17	If the Backoff Timer expires, the PCI 9656 asserts BREQo until it is granted the Local Bus.
BTERM#	Burst Terminate	1	I/O DTS 24 mA	C20	As an input, assertion causes the PCI 9656 (as the Local Master) to break the transfer (typically a burst). If the transfer is not completed, the PCI 9656 generates a new Address cycle and continues the transfer. As an output, the PCI 9656 (as a Local target) only asserts BTERM# to request the Master to break the transfer if a PCI Abort condition is detected.
CCS#	Configuration Register Select	1	I	D12	Internal PCI 9656 registers are selected when CCS# is asserted low during Local Bus accesses to the PCI 9656.

Table 12-12. J 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 ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pins go Hi-Z.*	A13	When DMA Channel 0 is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is in process.
DACK1#	DMA Channel 1 Demand Mode Acknowledge	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pins go Hi-Z.*	C13	When DMA Channel 1 is programmed through the Configuration registers to operate in Demand mode, this output indicates a DMA transfer is in process.
DEN#	Data Enable	1	O TS 24 mA	Y15	When asserted low, the Local Bus device can drive the Local Data Bus. 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	*If (DMAMODE0[14]=1 and/or DMAMODE1[14]=1)* I *else* O TP 24 mA ***Note for 2nd case:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pin goes Hi-Z.*	D14	Multiplexed I/O pin. Default functionality is DMPAF output (DMAMODE0[14] and DMAMODE1[14]=00b). DMPAF (Output): Direct Master Write FIFO Almost Full status output. Programmable through DMPBAM[10, 8:5]. EOT# (Input): 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	D18, B20, C18, B19	Parity is even for each of up to four byte lanes on the Local Bus. Parity is checked for writes to or on reads by the PCI 9656. Parity is asserted for reads from or writes by the PCI 9656.

Table 12-12. J Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
DREQ0#	DMA Channel 0 Demand Mode Request	1	I	B13	When DMA Channel 0 is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request.
DREQ1#	DMA Channel 1 Demand Mode Request	1	I	A14	When DMA Channel 1 is programmed through the Configuration registers to operate in Demand mode, this input serves as a DMA request.
DT/R#	Data Transmit/ Receive	1	O TS 24 mA	W14	When asserted High, indicates the PCI 9656 is driving data onto the Local Data Bus. Used in conjunction with DEN#, to provide control for data transceivers attached to the Local Bus. When asserted Low, indicates that the PCI 9656 receives data. Refer also to Timing Diagram 5-55 and Timing Diagram 5-56.
LA[28:2]	Local Address Bus	27	I/O TS 24 mA	W15, Y16, U14, V15, W16, Y17, V16, W17, Y18, U16, V17, W18, Y19, V18, W19, Y20, W20, V19, U18, T17, V20, U20, T18, T19, T20, R18, P17	Provides the current Lword address (except the upper three bits [31:29]) during any phase of an access. Incremented on successive Data cycles during Burst cycles. ***Note:*** *LA[1:0] may also be obtained by using the LBE[1:0]# pins.*
LAD[31:0]	Local Address/ Data Bus	32	I/O TS 24 mA	N18, N19, N20, M17, M18, M19, M20, L19, L18, L20, K20, K19, K18, K17, J20, J19, J18, J17, H20, H19, H18, G20, G19, F20, G18, F19, E20, G17, F18, E19, E18, D19	During an Address phase: As a Local Bus master, the PCI 9656 provides a 32-bit address for 8-bit data quantities, and LAD[31:0] provide byte addressing. For 16-bit data quantities, LAD[31:1] provide word addressing and LAD[0] is driven to 0. For 32-bit data quantities, LAD[31:2] provide Lword addressing and LAD[1:0] are driven to 00b. As a Local Bus slave, all Master accesses to the PCI 9656 can only be 32-bit quantities (LAD[1:0] are ignored). The input address is latched into the PCI 9656, on the positive edge of LCLK during ADS# assertion or following ALE assertion. During a Data phase: As a Local Bus master, the PCI 9656 provides an 8-bit data quantity on LAD byte lanes, 16-bit data quantities on LAD word lanes, and 32-bit data quantities on LAD[31:0], depending on the bus data-width access. As a Local Bus slave, 32-bit Data Bus for reading or writing to the PCI 9656.

Table 12-12. J Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LBE[3:0]#	Local Byte Enables	4	I/O TS 24 mA	R20, P18, P19, P20	Encoded, based on the bus data-width configuration, as follows: **32-Bit Bus** The four Byte Enables indicate which of the four bytes are valid 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#, LA1, and BLE#, respectively: LBE3# Byte High Enable (BHE#) – LAD[15:8] LBE2# ***not used*** LBE1# Address bit 1 (LA1) LBE0# Byte Low Enable (BLE#) – LAD[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)
LCLK	Local Processor Clock	1	I	D20	Local clock input.
LHOLD	Local Hold Request	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted)* *or* *(HOSTEN# and LRESET# asserted)* *or* *BD_SEL# not asserted)],* *pin goes Hi-Z.*	B18	Asserted to request use of the Local Bus.
LHOLDA	Local Hold Acknowledge	1	I	B17	The external Local Bus Arbiter asserts LHOLDA when bus ownership is granted in response to LHOLD. The Local Bus should not be granted to the PCI 9656, unless requested by LHOLD.
LINTi#	Local Interrupt Input	1	I	B15	When asserted, causes a PCI interrupt (INTA#).
LINTo#	Local Interrupt Output	1	O OD 24 mA	A15	Synchronous output that remains asserted as long as the interrupt is enabled and the interrupt condition exists.

Table 12-12. J Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LRESET#	Local Bus Reset	1	*If (HOSTEN# asserted)* I *else* O TP 24 mA	A16	As an input, in Host mode, RST# is generated as long as LRESET# is asserted. Resets the PCI 9656. As an output, in Adapter mode, LRESET# is asserted when the PCI 9656 is in reset (RST#=0). May be used to reset the back-end logic on the board.
LSERR#	Local System Error Interrupt Output	1	O OD 24 mA	D16	Synchronous output that remains asserted as long as the interrupt is enabled and the interrupt condition exists.
LW/R#	Local Write/Read	1	I/O TS 24 mA	R19	Asserted low for reads and high for writes.
PMEREQ#	PME Request	1	I	B10	Asserted during a D_{3cold} power state to request a Power Management Event.
READY#	Ready I/O	1	I/O DTS 24 mA	E17	A Local slave asserts READY# to indicate 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(s) expires (WAIT# output de-asserted).
USERi LLOCKi#	User Input Local Lock Input	1	I	B14	Multiplexed input pin. Default functionality is USERi (CNTRL[18]=1). USERi: General-purpose input that can be read in the PCI 9656 Configuration registers. Also used to select the PCI 9656 behavior in response to PCI accesses during initialization. (Refer to Section 4.4.2.) LLOCKi#: Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9656 for direct Local access to the PCI Bus.
USERo LLOCKo#	User Output Local Lock Output	1	O TP 24 mA ***Note:*** *If [(HOSTEN# not asserted and RST# asserted) or (HOSTEN# and LRESET# asserted) or BD_SEL# not asserted)], pin goes Hi-Z.*	C14	Multiplexed output pin. Default functionality is USERo (CNTRL[19]=1). USERo: General-purpose output controlled from the PCI 9656 Configuration registers. LLOCKo#: Indicates an atomic operation for a Direct Slave PCI-to-Local Bus access may require multiple transactions to complete.

Table 12-12. J Mode Local Bus Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
WAIT#	Wait I/O	1	I/O TS 24 mA	B16	As an input, the Local Bus Master may assert WAIT# to pause the PCI 9656 (insert wait states) during a Direct Master access Data phase. As an output, the PCI 9656 may be programmed to insert wait states (to pause the Local Slave) by asserting WAIT# for a predefined number of Local Bus clock cycles during Direct Slave transfers.
Total		95			

12.7 JTAG INTERFACE

The PCI 9656 provides a JTAG Boundary Scan interface which can be used to debug a pin's board connectivity.

12.7.1 *IEEE 1149.1* Test Access 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* specifications 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 – TCK, TDI, TDO, TMS – and the optional TRST# signal.
- **JTAG Clock Requirements** – The TCK signal frequency can range from DC to 10 MHz.
- **JTAG Reset Requirements** – JTAG debug port logic is reset at the same time as a system reset. There are two methods for putting its JTAG TAP controller into the Test-Logic-Reset state:
  - Upon receiving TRST#, the JTAG TAP controller returns to the Test-Logic Reset state.
  - Hold the PCI 9656 TMS pin (B8) high while transitioning the PCI 9656 TCK pin (A8) five times.

12.7.2 JTAG Instructions

The JTAG debug port provides the standard **EXTEST**, **SAMPLE/PRELOAD**, **BYPASS**, and **IDCODE** instructions. Invalid instructions behave as the **BYPASS** instruction.

The PCI 9656 returns the **IDCODE** values listed in Table 12-13.

Table 12-14 lists the JTAG instructions, along with their input codes.

Table 12-13. PCI 9656BA JTAG IDCODE Value

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Version				Part Number (9656 when converted to decimal)																PLX Manufacturer Identity											
0	0	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	0	1	1	1	0	0	1	1	0	1	1

Table 12-14. JTAG Instructions

Instruction	Input Code	Comments
EXTEST	00	*IEEE Standard 1149.1-1990*
IDCODE	01	
SAMPLE/PRELOAD	10	
BYPASS	11	

12.7.3 JTAG Boundary Scan

Boundary Scan Description Language (BSDL), *IEEE 1149.1b-1994*, is a supplement to *IEEE Standard 1149.1-1990* and *IEEE 1149.1a-1993*, *IEEE Standard Test Access Port and Boundary-Scan Architecture*. BSDL, a subset of the *IEEE 1076-1993 Standard 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 chip pins. Each pin has a logical type of in, out, in out, buffer, or linkage that defines the logical direction of signal flow.

The physical pin map correlates the chip logical ports 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

Due to JTAG Boundary Scan design with the PCI clock being masked, normal operation of the PCI 9656 is ***not*** allowed during JTAG Boundary Scan test. If the above condition is encountered during JTAG, the PCI 9656 should go through a hardware reset before normal operation can occur.

The BD_SEL# pin is a JTAG-compliance enable signal and should always be asserted low during JTAG Boundary Scan operations.

12.7.4 JTAG Reset Input TRST#

The TRST# input pin is the asynchronous JTAG logic reset. When TRST# is asserted, it causes the PCI 9656 TAP controller to initialize. In addition, when the TAP controller is initialized, it selects the PCI 9656 normal logic path (core-to-I/O). It is recommended that the designer take the following into consideration when implementing the asynchronous JTAG logic reset on a board:

- If JTAG functionality is required, one of the following should be considered:
  - The TRST# input signal should use a low-to-high transition once during the PCI 9656 boot-up, along with the RST# signal.
  - Hold the PCI 9656 TMS pin (B8) high while transitioning the PCI 9656 TCK pin (A8) five times.
- If JTAG functionality is not required, the TRST# signal must be directly connected to ground.

Note: *IEEE Standard 1149.1-1990 requires pull-up resistors on the TDI, TMS, and TRST# pins. To remain PCI r2.2-compliant, no internal pull-up resistors are provided on JTAG pins in the PCI 9656; therefore, the pull-up resistors must be externally added to the PCI 9656 when implementing JTAG.*

13 ELECTRICAL SPECIFICATIONS

13.1 3.3 AND 5V MIXED-VOLTAGE DEVICES AND POWER SEQUENCE

The PCI 9656 I/O buffers are 3.3V-based, but 5V tolerant. They are capable of receiving 5V signals and their output levels meet 5V TTL specifications.

The power supply for the PCI 9656 I/O ring (V_{RING}) is 3.3V, and for core logic (V_{CORE}) it is 2.5V. To fully comply with *PCI r2.2*, a third power supply (V_{IO}) is also supported. V_{IO} connects to a cathode of the high clamp diode in PCI buffers. In a 5V system, set V_{IO} to 5V, and in a 3.3V system, to 3.3V.

As in all multi-supply devices, care should be taken to properly power-on the PCI 9656.

There are five different chip power sources – V_{RING}, Card_V_{AUX}, V_{CORE}, 2.5V_{AUX}, and V_{IO}. To properly sequence power to these five sources, the only requirement is that V_{CORE}, 2.5V_{AUX}, and V_{IO} must receive power no later than 10 ms after V_{RING} and Card_V_{AUX} receive power. It is recommended that V_{CORE}, 2.5V_{AUX}, and V_{IO} receive power first or as close to V_{RING} and Card_V_{AUX} as reasonable (up to the 10 ms maximum). Relative to each other, V_{CORE}, 2.5V_{AUX}, and V_{IO} may receive power in any order. Also, relative to each other, V_{CORE} and 2.5V_{AUX} may receive power in either order.

Caution: *Violating the above power sequencing requirement* ***will*** *damage the device.*

Note: *For non-Power Management designs, connect 2.5V_{AUX} to the V_{CORE} power supply, and Card_V_{AUX} to the V_{RING} power supply. Connect the PRESENT_DET signal to ground.*

13.2 GENERAL ELECTRICAL SPECIFICATIONS

Table 13-1. Absolute Ratings

Specification	Rating
Storage Temperature	-65 to +125°C
Ambient Temperature with Power Applied (Industrial)	-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 Power Consumption	830 mW

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 (Industrial)	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

Note: *Values guaranteed by design.*

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
33.6	31.5	30.3

Table 13-5. Electrical Characteristics over Operating Range

Parameter	Description	Test Conditions		Min	Max	Units	Buffer Type
V_{OH}[1]	Output High Voltage	V_{DD} = Min $V_{IN} = V_{IH}$ or V_{IL}	I_{OH} = -12.0 mA	2.4	–	V	Local
V_{OL}[1]	Output Low Voltage		I_{OL} = 12 mA	–	0.4	V	
V_{OH}[2]	Output High Voltage		I_{OH} = -24.0 mA	2.4	–	V	
V_{OL}[2]	Output Low Voltage		I_{OL} = 24 mA	–	0.4	V	
V_{IH}	Input High Level	–	–	2.0	5.5	V	
V_{IL}	Input Low Level	–	–	-0.5	0.8	V	
V_{OH3}	PCI 3.3V Output High Voltage	V_{DD} = Min $V_{IN} = V_{IH}$ or V_{IL}	I_{OH} = -500 µA	0.9 V_{DD}	–	V	PCI
V_{OL3}	PCI 3.3V Output Low Voltage		I_{OL} = 1500 µA	–	0.1 V_{DD}	V	
V_{IH3}	PCI 3.3V Input High Level	–	–	0.5 V_{DD}	V_{DD} +0.5	V	
V_{IL3}	PCI 3.3V Input Low Level	–	–	-0.5	0.3 V_{DD}	V	
I_{IL}	Input Leakage Current	$V_{SS} \le V_{IN} \le V_{DD}$, V_{DD} = Max		-10	+10	µA	
I_{LPC}[3]	DC Current Per Pin during Precharge	V_P = 0.8 to 1.2V		–	1.0	mA	

Table 13-5. Electrical Characteristics over Operating Range (Continued)

Parameter	Description	Test Conditions	Min	Max	Units	Buffer Type
I_{OZ}	Tri-state Output Leakage Current	V_{DD} = Max	-10	+10	µA	PCI
I_{DD}[4] (I/O Ring)	Power Supply Current for I/O Ring	I/O Ring V_{DD} = 3.6V PCLK = 66 MHz, LCLK = 66 MHz C = 50 pF (PCI and Local)	–	95	mA	
I_{DD} (Core)	Power Supply Current for Core	Core V_{DD} = 2.63V PCLK = 66 MHz, LCLK = 66 MHz	–	185	mA	
I_{CCL} I_{CCH} I_{CCZ}	Quiescent Power Supply Current	V_{CC} = Max V_{IN} = GND or V_{CC}	–	50	µA	

[1] *For 12 mA I/O or output cells (Local Bus).*

[2] *For 24 mA I/O or output cells (Local Bus). The PCI 9656 Local Bus is proprietary to PLX; therefore, the values reported are best-case estimates. TSMC marks the I/O buffer drive strength at 24 mA. The V_{OL}/V_{OH} lists a 90% worst-case scenario; however, this is still an approximation. Therefore, 10% margin still remains for the worst case. In addition, the 24 mA drive strength is estimated to have 10% uncertainty.*

[3] *Value guaranteed by design.*
I_{LPC} is the DC current flowing from V_{DD} to Ground during precharge, as both PMOS and NMOS devices remain on during precharge. It is ***not*** *the leakage current flowing into or out of the pin under precharge.*

[4] *76 PCI Bus and 40 Local Bus I/Os, simultaneously switching.*

13.3 LOCAL INPUTS

Figure 13-1 illustrates the PCI 9656 Local input setup and hold timing. Tables 13-6 through 13-8 list the Local Bus Worst Case T_{SETUP} and T_{HOLD} values for each Local Bus mode. T_{SETUP} is the setup time, the time that an input signal is stable before the rising edge of LCLK. T_{HOLD} is the time that an input signal is stable after the rising edge of LCLK.

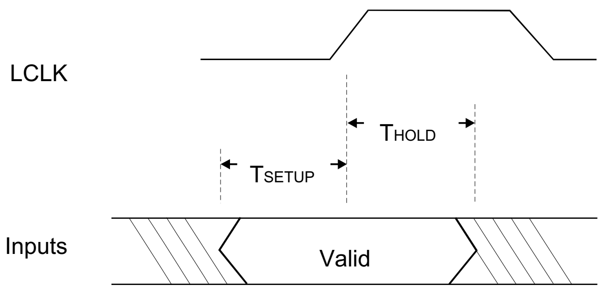


Figure 13-1. PCI 9656 Local Input Setup and Hold Timing

Table 13-6. M Mode Local Bus Worst Case Input AC Timing Specifications

Signals (Synchronous Inputs)	**T_{SETUP}**	**T_{HOLD}**
	V_{CC} = 3.0V, T_a = 85°C	
BB#	2.7 ns	1 ns
BDIP#	3.8 ns	1 ns
BG#	2.9 ns	1 ns
BI#	4.0 ns	1 ns
BIGEND#/WAIT#	3.8 ns	1 ns
BURST#	4.1 ns	1 ns
CCS#	2.9 ns	1 ns
DP[0:3]	2.9 ns	1 ns
DREQ[1:0]#	3.3 ns	1 ns
LA[0:31]	3.6 ns	1 ns
LD[0:31]	3.1 ns	1 ns
MDREQ#/DMPAF/**EOT#**	4.2 ns	1 ns
RD/WR#	3.5 ns	1 ns
TA#	4.1 ns	1 ns
TEA#	4.4 ns	1 ns
TS#	2.1 ns	1 ns
TSIZ[0:1]	3.6 ns	1 ns
USERi/LLOCKi#	3.2 ns	1 ns
Clock Input Frequency	**Min**	**Max**
Local	0 MHz	66 MHz
PCI	0 MHz	66 MHz

Table 13-7. C Mode Local Bus Worst Case Input AC Timing Specifications

Signals (Synchronous Inputs)	T_{SETUP}	T_{HOLD}
	V_{CC} = 3.0V, T_a = 85°C	
ADS#	2.1 ns	1 ns
BIGEND#	4.0 ns	1 ns
BLAST#	3.4 ns	1 ns
BREQi	0.3 ns	1 ns
BTERM#	4.0 ns	1 ns
CCS#	2.9 ns	1 ns
DMPAF/**EOT#**	4.2 ns	1 ns
DP[3:0]	2.9 ns	1 ns
DREQ[1:0]#	3.3 ns	1 ns
LA[31:2]	3.4 ns	1 ns
LBE[3:0]#	3.6 ns	1 ns
LD[31:0]	3.1 ns	1 ns
LHOLDA	2.5 ns	1 ns
LW/R#	3.5 ns	1 ns
READY#	4.0 ns	1 ns
USERi/LLOCKi#	2.9 ns	1 ns
WAIT#	4.0 ns	1 ns
Clock Input Frequency	**Min**	**Max**
Local	0 MHz	66 MHz
PCI	0 MHz	66 MHz

Table 13-8. J Mode Local Bus Worst Case Input AC Timing Specifications

Signals (Synchronous Inputs)	T_{SETUP}	T_{HOLD}
	V_{CC} = 3.0V, T_a = 85°C	
ADS#	2.1 ns	1 ns
ALE	1.7 ns	1 ns
BIGEND#	4.0 ns	1 ns
BLAST#	3.4 ns	1 ns
BREQi	0.3 ns	1 ns
BTERM#	4.2 ns	1 ns
CCS#	2.9 ns	1 ns
DMPAF/**EOT#**	4.2 ns	1 ns
DP[3:0]	2.9 ns	1 ns
DREQ[1:0]#	3.3 ns	1 ns
LA[28:2]	3.4 ns	1 ns
LAD[31:0]	3.1 ns	1 ns
LBE[3:0]#	3.6 ns	1 ns
LHOLDA	2.5 ns	1 ns
LW/R#	3.5 ns	1 ns
READY#	4.2 ns	1 ns
USERi/LLOCKi#	2.9 ns	1 ns
WAIT#	4.0 ns	1 ns
Clock Input Frequency	**Min**	**Max**
Local	0 MHz	66 MHz
PCI	0 MHz	66 MHz

13.4 LOCAL OUTPUTS

Figure 13-2 illustrates the PCI 9656 Local output delay timing. Tables 13-9 through 13-11 list the T_{VALID} (MAX) values for each Local Bus mode. T_{VALID} is output valid (clock-to-out), the time after the rising edge of LCLK until the output is stable. T_{VALID} (MIN) is no less than 2.2 ns for each signal for all Local Bus modes.

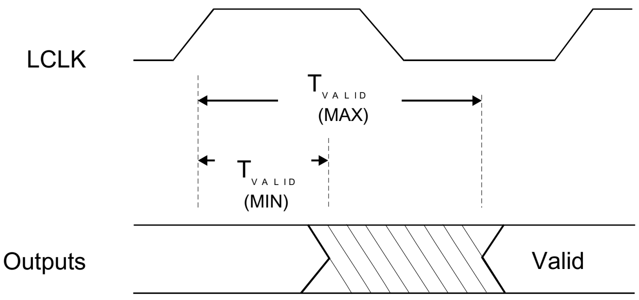


Figure 13-2. PCI 9656 Local Output Delay

Table 13-9. M Mode Local Bus T_{VALID} (MAX) Output AC Timing Specifications

Signals (Synchronous Outputs)	**Output T_{VALID}** **C_L = 50 pF, V_{CC} = 3.0V, T_a = 85°C**
BB#	6.8 ns
BDIP#	6.4 ns
BIGEND#/**WAIT#**	6.3 ns
BR#	6.8 ns
BURST#	6.3 ns
DACK[1:0]#	6.3 ns
DP[0:3]	6.8 ns
LA[0:31]	6.8 ns
LD[0:31]	6.3 ns
MDREQ#/DMPAF/ EOT#	6.6 ns
RD/WR#	6.3 ns
RETRY#	6.8 ns
TA#	7.2 ns
TEA#	7.5 ns
TS#	6.3 ns
TSIZ[0:1]	6.3 ns
USERo/LLOCKo#	6.3 ns

Notes: *On high-to-low transitions, output T_{VALID} values increase/decrease by 16 ps for each increase/decrease of 1 pF.*

On low-to-high transitions, output T_{VALID} values increase/decrease by 20 ps for each increase/decrease of 1 pF.

On high-to-low transitions, the slew rate at 50 pF loading is 1.93V/ns typical; 0.94V/ns worst case.

On low-to-high transitions, the slew rate at 50 pF loading is 1.15V/ns typical; 0.70V/ns worst case.

Table 13-10. C Mode Local Bus T_{VALID} (MAX) Output AC Timing Specifications

Signals (Synchronous Outputs)	Output T_{VALID} C_L = 50 pF, V_{CC} = 3.0V, T_a = 85°C
ADS#	6.3 ns
BLAST#	6.3 ns
BREQo	6.8 ns
BTERM#	6.8 ns
DACK[1:0]#	6.3 ns
DMPAF/EOT#	6.6 ns
DP[3:0]	6.8 ns
LA[31:2]	6.8 ns
LBE[3:0]#	6.3 ns
LD[31:0]	6.4 ns
LHOLD	6.8 ns
LSERR#	7.5 ns
LW/R#	6.3 ns
READY#	7.2 ns
USERo/LLOCKo#	6.3 ns
WAIT#	6.4 ns

Notes: *On high-to-low transitions, output T_{VALID} values increase/decrease by 16 ps for each increase/decrease of 1 pF.*

On low-to-high transitions, output T_{VALID} values increase/decrease by 20 ps for each increase/decrease of 1 pF.

On high-to-low transitions, the slew rate at 50 pF loading is 1.93V/ns typical; 0.94V/ns worst case.

On low-to-high transitions, the slew rate at 50 pF loading is 1.15V/ns typical; 0.70V/ns worst case.

Table 13-11. J Mode Local Bus T_{VALID} (MAX) Output AC Timing Specifications

Signals (Synchronous Outputs)	Output T_{VALID} C_L = 50 pF, V_{CC} = 3.0V, T_a = 85°C
ADS#	6.3 ns
ALE	Refer to Figure 13-3
BLAST#	6.3 ns
BREQo	6.8 ns
BTERM#	6.8 ns
DACK[1:0]#	6.3 ns
DEN#	6.4 ns
DMPAF/EOT#	6.6 ns
DP[3:0]	6.8 ns
DT/R#	6.3 ns
LA[28:2]	6.4 ns
LAD[31:0]	6.4 ns
LBE[3:0]#	6.3 ns
LHOLD	6.8 ns
LSERR#	7.5 ns
LW/R#	6.3 ns
READY#	7.2 ns
USERo/LLOCKo#	6.3 ns
WAIT#	6.4 ns

Notes: *On high-to-low transitions, output T_{VALID} values increase/decrease by 16 ps for each increase/decrease of 1 pF.*

On low-to-high transitions, output T_{VALID} values increase/decrease by 20 ps for each increase/decrease of 1 pF.

On high-to-low transitions, the slew rate at 50 pF loading is 1.93V/ns typical; 0.94V/ns worst case.

On low-to-high transitions, the slew rate at 50 pF loading is 1.15V/ns typical; 0.70V/ns worst case.

13.5 ALE OUTPUT DELAY TIMING FOR ALL LOCAL BUS CLOCK RATES

Figure 13-3 illustrates the PCI 9656BA ALE output delay timing for any processor/Local Bus clock rate.

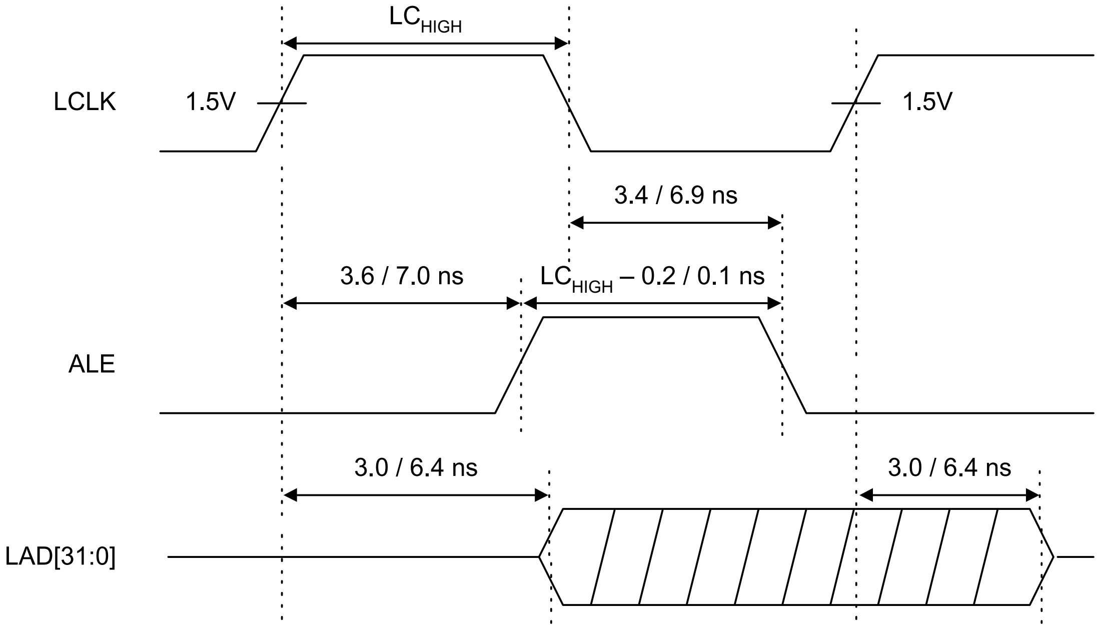


Figure 13-3. PCI 9656 ALE Output Delay to the Local Clock

Notes: *LC_{HIGH} is the time, in ns, that the processor/Local Bus clock is high.*

Times listed in Figure 13-3 are “minimum/maximum” values.

THIS PAGE INTENTIONALLY LEFT BLANK.

14 PHYSICAL SPECIFICATIONS

14.1 MECHANICAL DIMENSIONS

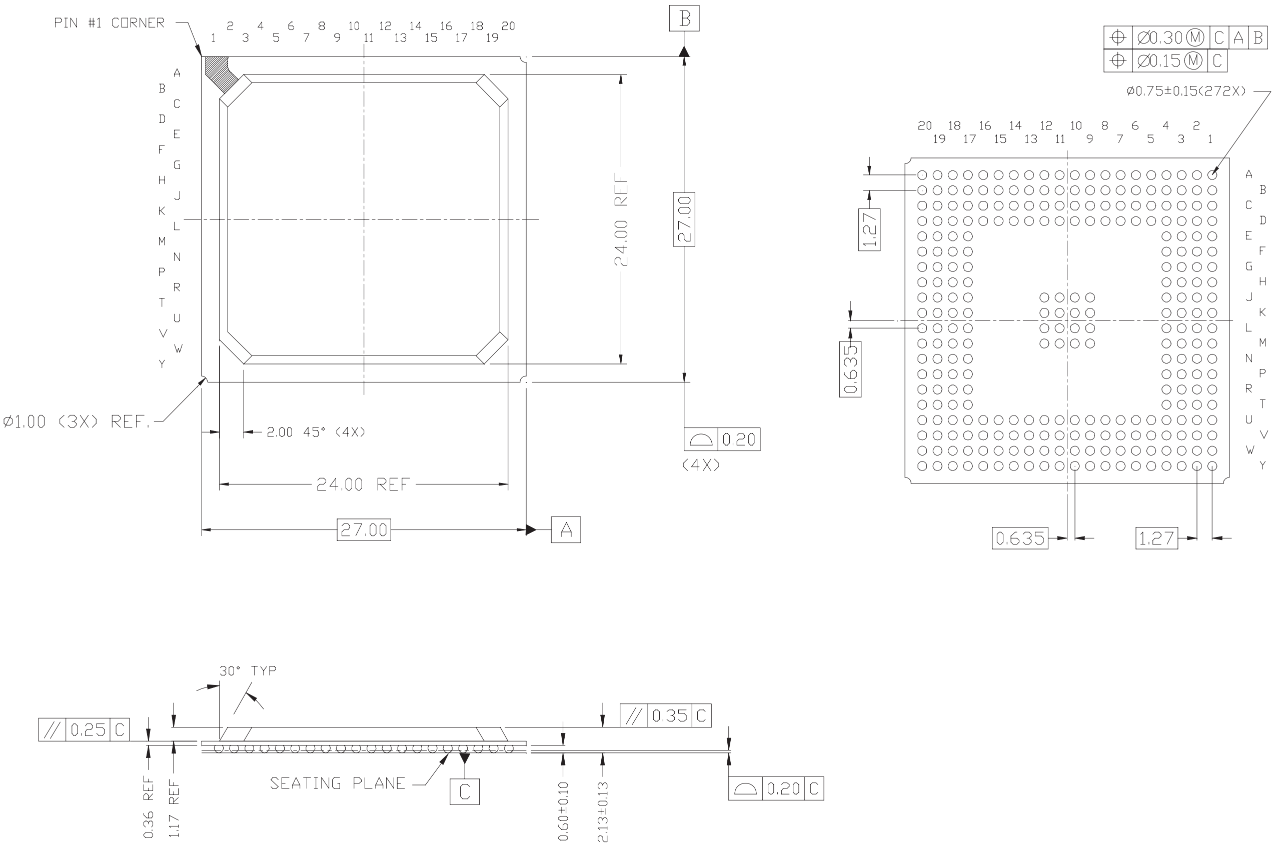


Figure 14-1. PCI 9656 Mechanical Dimensions

14.2 BALL GRID ASSIGNMENTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A	V_{RING}	AD21	AD24	AD27	AD31	RST#	TDI	TCK	V_{IO}	IDDQEN#	PRESENT_ DET	EESK	DACK0#	DREQ1#	LINTo#	LRESET#	RETRY# (M) BREQo (C,J)	BURST# (M) BLAST# (C,J)	MODE1	MODE0	A
B	AD19	AD22	AD23	AD25	AD28	REQ0# or GNT#	INTA#	TMS	PME#	PMEREQ#	EEDI/ EEDO	EECS	DREQ0#	USERi LLOCKi#	LINTi#	BDIP# (M) WAIT# (C,J)	BG# (M) LHOLDA (C,J)	BR# (M) LHOLD (C,J)	DP3 (M) DP0 (C,J)	DP1 (M) DP2 (C,J)	B
C	FRAME#	AD18	AD20	IDSEL	AD26	AD29	GNT0# or REQ#	TDO	BD_SEL#	Card_V_{AUX}	V_{CORE}	BIGEND# (all) WAIT# (M)	DACK1#	USERo LLOCKo#	HOSTEN#	BB# (M) BREQi (C,J)	TS# (M) ADS# (C,J)	DP2 (M) DP1 (C,J)	V_{CORE}	BI# (M) BTERM# (C,J)	C
D	IRDY#	AD17	AD16	V_{RING}	C/BE3#	V_{RING}	AD30	V_{RING}	TRST#	2.5V_{AUX}	V_{RING}	CCS#	V_{RING}	MDREQ# (M) DMPAF (all) EOT# (all)	V_{RING}	TEA# (M) LSERR# (C,J)	V_{RING}	DP0 (M) DP3 (C,J)	LD31 (M) LD0 (C) LAD0 (J)	LCLK	D
E	DEVSEL#	V_{CORE}	TRDY#	C/BE2#													TA# (M) READY# (C,J)	LD30 (M) LD1 (C) LAD1 (J)	LD29 (M) LD2 (C) LAD2 (J)	LD26 (M) LD5 (C) LAD5 (J)	E
F	V_{IO}	PERR#	STOP#	V_{RING}													V_{RING}	LD28 (M) LD3 (C) LAD3 (J)	LD25 (M) LD6 (C) LAD6 (J)	LD23 (M) LD8 (C) LAD8 (J)	F
G	C/BE1#	PAR	SERR#	LOCK#													LD27 (M) LD4 (C) LAD4 (J)	LD24 (M) LD7 (C) LAD7 (J)	LD22 (M) LD9 (C) LAD9 (J)	LD21 (M) LD10 (C) LAD10 (J)	G
H	AD13	AD14	AD15	V_{RING}													V_{RING}	LD20 (M) LD11 (C) LAD11 (J)	LD19 (M) LD12 (C) LAD12 (J)	LD18 (M) LD13 (C) LAD13 (J)	H
J	AD9	AD10	AD11	AD12					V_{SS}	V_{SS}	V_{SS}	V_{SS}					LD17 (M) LD14 (C) LAD14 (J)	LD16 (M) LD15 (C) LAD15 (J)	LD15 (M) LD16 (C) LAD16 (J)	LD14 (M) LD17 (C) LAD17 (J)	J
K	AD7	AD8	C/BE0#	V_{RING}					V_{SS}	V_{SS}	V_{SS}	V_{SS}					LD13 (M) LD18 (C) LAD18 (J)	LD12 (M) LD19 (C) LAD19 (J)	LD11 (M) LD20 (C) LAD20 (J)	LD10 (M) LD21 (C) LAD21 (J)	K
L	PCLK	AD6	AD5	AD4					V_{SS}	V_{SS}	V_{SS}	V_{SS}					V_{RING}	LD8 (M) LD23 (C) LAD23 (J)	LD7 (M) LD24 (C) LAD24 (J)	LD9 (M) LD22 (C) LAD22 (J)	L
M	AD3	AD2	AD1	AD0					V_{SS}	V_{SS}	V_{SS}	V_{SS}					LD3 (M) LD28 (C) LAD28 (J)	LD4 (M) LD27 (C) LAD27 (J)	LD5 (M) LD26 (C) LAD26 (J)	LD6 (M) LD25 (C) LAD25 (J)	M
N	ACK64#	REQ64#	GNT1#	V_{RING}													V_{RING}	LD0 (M) LD31 (C) LAD31 (J)	LD1 (M) LD30 (C) LAD30 (J)	LD2 (M) LD29 (C) LAD29 (J)	N
P	REQ1#	REQ2#	V_{CORE}	GNT4#													LA29 (M) LA2 (C,J)	TSIZ1 (M) LBE2# (C,J)	LA30 (M) LBE1# (C,J)	LA31 (M) LBE0# (C,J)	P
R	GNT2#	GNT3#	REQ4#	V_{RING}													V_{RING}	LA28 (M) LA3 (C,J)	RD/WR# (M) LW/R# (C,J)	TSIZ0 (M) LBE3# (C,J)	R
T	REQ3#	C/BE7#	C/BE5#	AD63													LA22 (M) LA9 (C,J)	LA25 (M) LA6 (C,J)	LA26 (M) LA5 (C,J)	LA27 (M) LA4 (C,J)	T
U	C/BE6#	C/BE4#	AD62	V_{RING}	AD56	V_{RING}	AD49	V_{RING}	V_{CORE}	V_{RING}	GNT5#	64EN#	V_{RING}	LA5 (M) LA26 (C,J)	V_{RING}	LA12 (M) LA19 (C,J)	V_{RING}	LA21 (M) LA10 (C,J)	V_{CORE}	LA24 (M) LA7 (C,J)	U
V	PAR64	V_{IO}	AD60	AD57	AD53	AD50	AD46	AD43	AD40	AD36	AD32	REQ6#	LEDon#	LA2 (M) LA29 (C) ALE (J)	LA6 (M) LA25 (C,J)	LA9 (M) LA22 (C,J)	LA13 (M) LA18 (C,J)	LA16 (M) LA15 (C,J)	LA20 (M) LA11 (C,J)	LA23 (M) LA8 (C,J)	V
W	AD61	V_{RING}	V_{SS}	AD58	AD52	AD48	AD45	AD42	AD39	AD37	AD33	GNT6#	V_{IO}	LA0 (M) LA31 (C) DT/R# (J)	LA3 (M) LA28 (C,J)	LA7 (M) LA24 (C,J)	LA10 (M) LA21 (C,J)	LA14 (M) LA17 (C,J)	LA17 (M) LA14 (C,J)	LA19 (M) LA12 (C,J)	W
Y	V_{SS}	AD59	AD55	AD54	AD51	AD47	AD44	AD41	AD38	AD35	AD34	REQ5#	ENUM#	CPCISW	LA1 (M) LA30 (C) DEN# (J)	LA4 (M) LA27 (C,J)	LA8 (M) LA23 (C,J)	LA11 (M) LA20 (C,J)	LA15 (M) LA16 (C,J)	LA18 (M) LA13 (C,J)	Y
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

Figure 14-2. Physical Layout with Pinout – Topside View

Note: *Image is not to scale.*

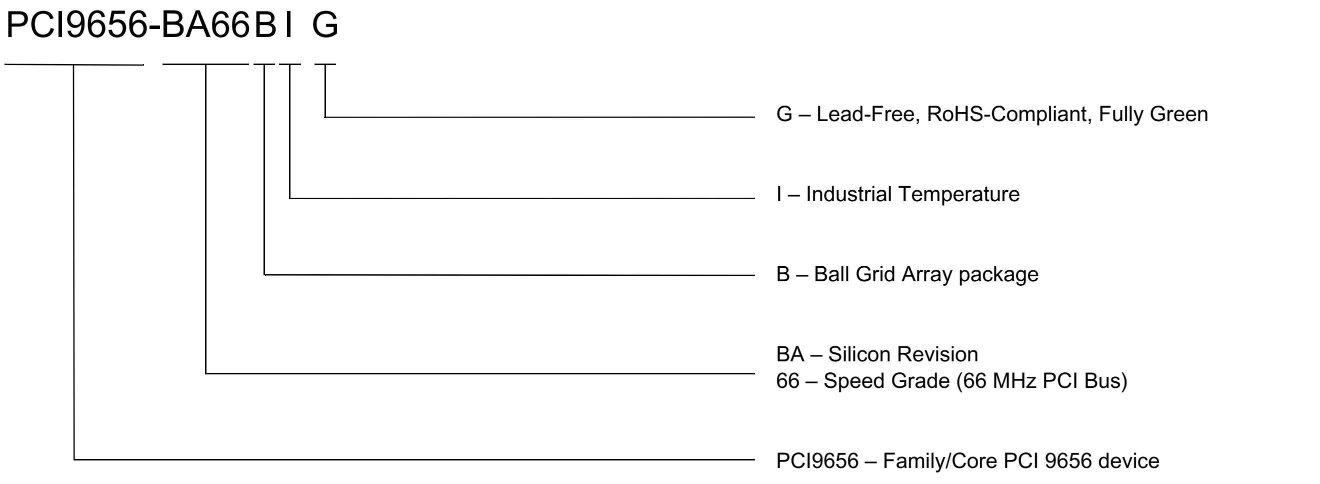
A GENERAL INFORMATION

A.1 PRODUCT ORDERING INFORMATION

Contact your local PLX Sales Representative for ordering information.

Table A-1. Product Ordering Information

Part Number	Description
PCI 9656-BA66BI	PCI 9656 64-Bit, 66 MHz PCI Bus Mastering I/O Accelerator (272-pin, 27 x 27 mm^2, 1.27 mm ball pitch PBGA) Leaded Package
PCI 9656-BA66BI G	PCI 9656 64-Bit, 66 MHz PCI Bus Mastering I/O Accelerator (272-pin, 27 x 27 mm^2, 1.27 mm ball pitch PBGA) Lead-Free Package


PCI9656-BA66B I G
G – Lead-Free, RoHS-Compliant, Fully Green
I – Industrial Temperature
B – Ball Grid Array package
BA – Silicon Revision
66 – Speed Grade (66 MHz PCI Bus)
PCI9656 – Family/Core PCI 9656 device

A.2 UNITED STATES AND INTERNATIONAL REPRESENTATIVES, AND DISTRIBUTORS

PLX Technology, Inc., representatives and distributors are listed at www.plxtech.com.

A.3 TECHNICAL SUPPORT

PLX Technology, Inc., technical support information is listed at www.plxtech.com/support, or call 800 759-3735 (domestic only) or 408 774-9060.