PLX TECHNOLOGY

PCI 9080 Data Sheet

Version 1.02

July 24, 1997

Website: http://www.plxtech.com
Email: apps@plxtech.com
Phone: 408-774-9060
FAX: 408-774-2169

TABLE OF CONTENTS

REVISION HISTORY

Date	Revision	Comment
7/3/97	1.0	• Initial release. • Release timing diagrams. • Corrected typos and matched spec. • Changed Pin 170 to NC. • Changed LARBR (Local/Arbitration Register) to MARBR (Mode/Arbitration Register).
7/10/97	1.01	• Set up hold and output timings • Change mechanical package dimension. • Complete electrical tables in Section 6. • Correct timing diagrams. • Matched spec.
7/24/97	1.02	• Changed the title of Section 7. • Added READYo# value to Table 6-6. • Removed WR# and RD# signals from Timing Diagram 8-20. • Corrected titles of Timing Diagrams 8-20 and 8-68. • Corrected titles of Sections 8.3.3 and 8.4.3.

PLX TECHNOLOGY

PCI 9080

PCI I/O ACCELERATOR

JULY 24, 1997
VERSION 1.02

I_2O COMPATIBLE PCI BUS MASTER INTERFACE CHIP
FOR ADAPTERS AND EMBEDDED SYSTEMS

FEATURES

- PCI Version 2.1 compliant Bus Master Interface chip for adapters and embedded systems
- I_2O Compatible Messaging Unit
- 3.3 or 5 Volt PCI signaling, 5 volt core, low-power CMOS in 208-pin PQFP
- Two independent DMA channels for local bus memory to/from PCI host bus data transfers
- Eight programmable FIFOs for zero wait state burst operation
- PCI ⇔ Local data transfers up to 132 MB/sec
- Programmable local bus supports nonmultiplexed 32-bit address/data, multiplexed 32 or 16 bit, and slave accesses of 32, 16, or 8 bit local bus devices
- Local bus runs asynchronously to the PCI bus
- Eight 32 bit mailbox and two 32 bit doorbell registers
- Performs Big Endian/Little Endian conversion
- Upward compatibility with PCI 9060/9060ES/9060SD

1. GENERAL DESCRIPTION

PCI 9080 provides a compact, high performance PCI bus master interface for adapter boards and embedded systems. The programmable local bus of the chip can be configured to directly connect a wide variety of processors, controllers and memory subsystems.

PCI 9080 contains an Intelligent I/O (I_2O) messaging unit that allows high performance and compatible software implementations of the I_2O bus protocol specification. Users of the PCI 9060, 9060ES and 9060SD chips may upgrade their products to support I_2O, 3.3 Volts and other new features with little or no change to existing hardware and software.

PCI 9080 provides eight programmable FIFOs. Each operating mode—slave, direct, master, and DMA channel—have dedicated, independent read and write FIFOs. PCI 9080 also allows a local processor to configure other PCI devices in the system.

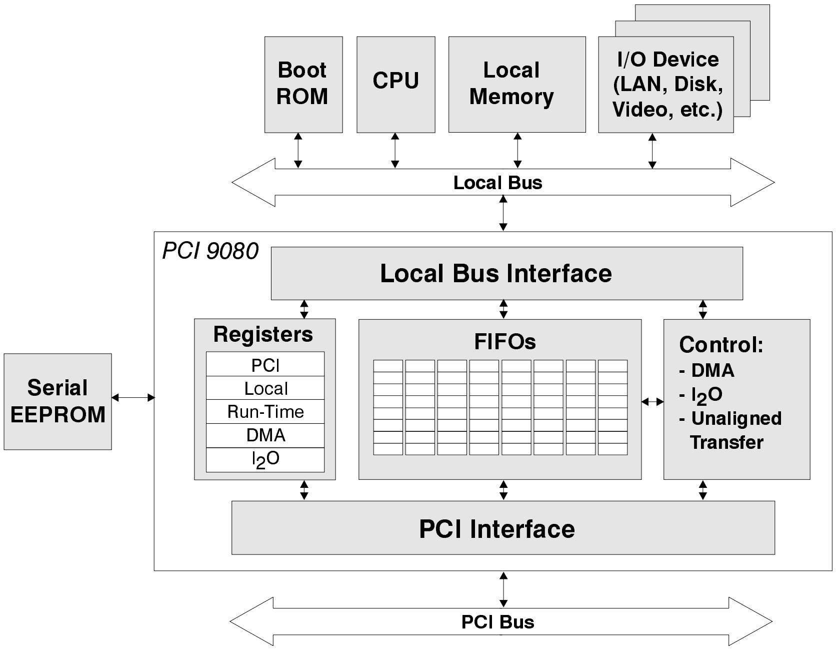


Figure 1-1. Typical Adapter or Block Diagram

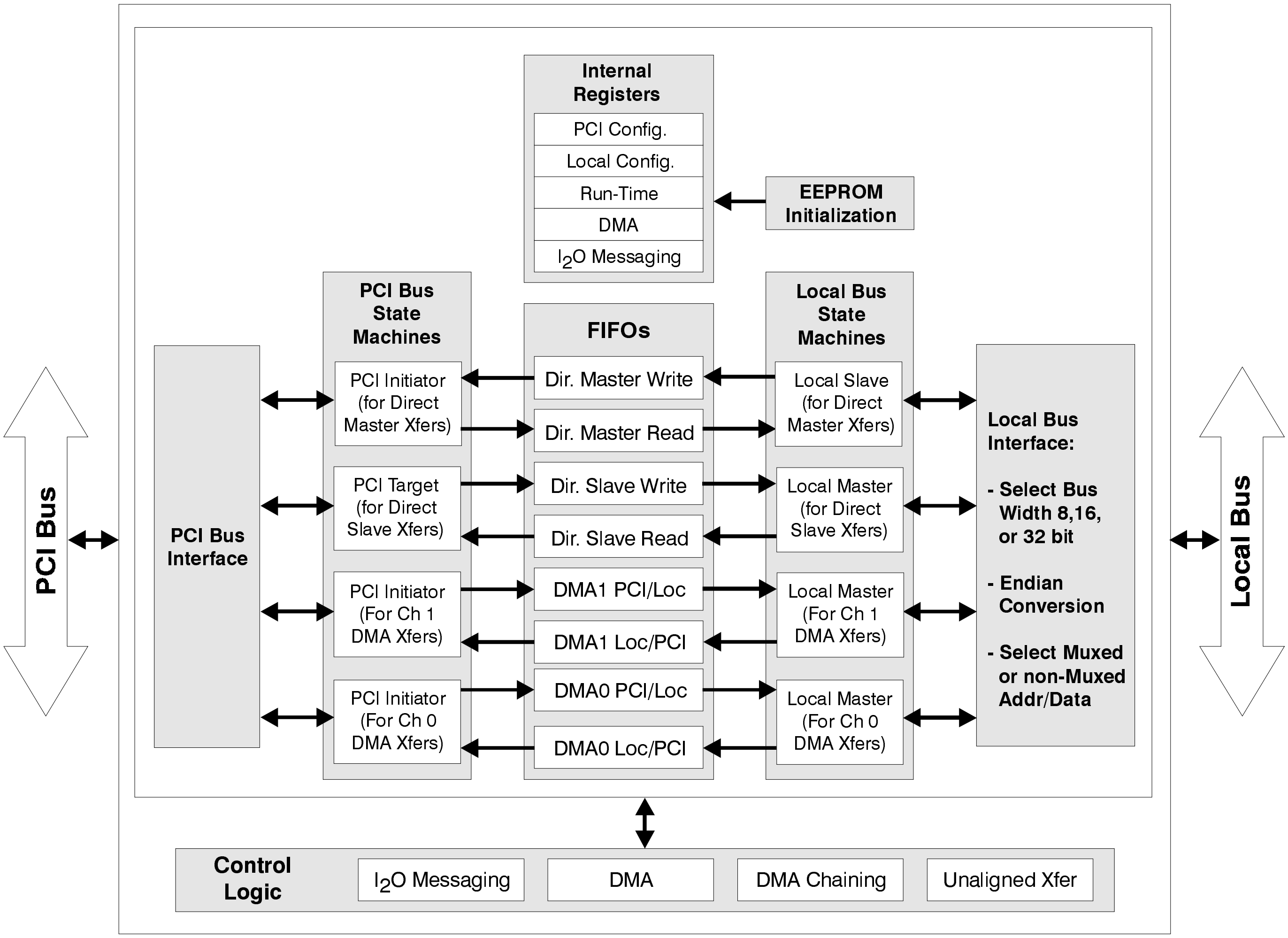


Figure 1-2. PCI 9080 Internal Block Diagram

1.1 APPLICATIONS FOR THE PCI 9080

1.1.1 PCI Adapter Cards

Major PCI adapter card applications for the PCI 9080 include high performance communications, networking, disk control, multimedia, and video adapters. PCI 9080 moves data between the host PCI bus and adapter local bus in several ways. First, the local CPU or host processor may program the DMA controller of the PCI 9080 to move data between the adapter memory and host PCI bus. Second, PCI 9080 can perform “Direct Master Transfers,” whereby a local CPU or controller accesses the PCI bus directly through a PCI master transfer. PCI 9080 also supports slave transfers in which another PCI device is the master. Finally, PCI 9080 contains a complete messaging unit with mailbox registers, doorbell registers, and queue management pointers that can be used for message passing under the I_2O protocol or a custom protocol.

1.1.2 Embedded Systems

Another application for PCI 9080 is in embedded systems, such as network hubs and routers, printer engines, and industrial equipment. In this configuration, all four of the above-mentioned data transfer modes are used. In addition, PCI 9080 supports Type 0 and Type 1 PCI configuration cycles, which allows embedded CPU to act as the embedded system host and to configure other PCI devices in the system.

1.2 MAJOR FEATURES

PCI 2.1 Compliant. PCI 9080 is compliant with all aspects of PCI specification v2.1.

I_2O Messaging Unit. PCI 9080 incorporates an I_2O messaging unit. This enables the adapter or embedded system to communicate with other I_2O-supported devices. I_2O messaging unit is fully compatible with the PCI extension of the I_2O specification v1.5.

Dual Independent Programmable DMA Controllers with Programmable FIFOs. PCI 9080 provides two independently programmable DMA controllers with programmable FIFOs for each channel. Each channel supports nonchaining and chaining DMA modes, demand mode DMA, and End of Transfer (EOT) mode.

Direct Bus Master. PCI 9080 supports memory-mapped bursts, transfer accesses, and I/O-mapped single-transfer accesses to the PCI bus from the Local Bus Master. PCI 9080 also supports PCI bus interlock (LOCK#) cycles. Read and write FIFOs enable high-performance bursting.

PCI Host Capability. In direct master mode, PCI 9080 can generate Type 0 or Type 1 PCI configuration cycles.

Direct Slave. PCI 9080 supports burst memory mapped and single I/O-mapped accesses to the local bus. Read and write FIFOs enable high-performance bursting.

Programmable Local Bus Modes. PCI 9080 is a PCI bus master interface chip that connects a PCI bus to one of three local bus types, selected through mode pins. PCI 9080 may be connected to any local bus with a similar design with little or no glue logic. Table 1-1 lists the three modes.

Table 1-1. Programmable Local Bus Modes

Mode	Description
C	32-bit address/32-bit data, nonmultiplexed
J	32-bit address/32-bit data, multiplexed
S	32-bit address/16-bit data, multiplexed

Interrupt Generator. PCI 9080 can generate PCI and local interrupts from several sources.

Clock. PCI 9080 local bus interface runs from a local TTL clock and generates the necessary internal clocks. This clock runs asynchronously to the PCI clock. There is a buffered PCI clock (BPCLKo) for the local side to use. BPCLKo may be connected to LCLK.

3.3 Volt and 5 Volt Operation. PCI 9080 core requires 5 V VCC. PCI 9080 provides 3.3 or 5 volt signaling on the PCI bus. The local bus operates at a 5V signaling level.

Serial EEPROM Interface. PCI 9080 contains an optional serial EEPROM interface that can be used to load configuration information. This is useful for loading information that is unique to a particular adapter (such as Network ID or Vendor ID).

Mailbox Registers. PCI 9080 contains eight 32 bit mailbox registers that may be accessed from the PCI or local bus.

Doorbell Registers. PCI 9080 includes two 32 bit doorbell registers. One generates interrupts from the PCI bus to local bus. The other generates interrupts from the local bus to the PCI bus.

Unaligned DMA Transfer Support. PCI 9080 can transfer data on any byte boundary.

Big/Little Endian Conversion. PCI 9080 supports dynamic switching between Big Endian and Little Endian operations for Direct Slave, Direct Master, DMA, and the internal register accesses on the local side.

PCI 9080 supports on-the-fly Endian conversion for Space 0, Space 1, and expansion ROM space. The local bus can be Big/Little Endian by using the BIGEND# input pin or programmable internal register configuration. When BIGEND# is asserted, it overwrites the internal register configuration.

Note: *PCI bus is always Little Endian.*

Read Ahead Mode. PCI 9080 supports read ahead mode, where prefetched data can be read from the PCI 9080 internal FIFO instead of the local side. Address must be subsequent to previous address and 32-bit aligned (next address = current address + 4).

Programmable Bus Wait States. PCI 9080 can be programmed to keep the PCI bus by generating a wait state(s), de-asserting TRDY#, if write FIFO becomes full. PCI 9080 can also be programmed to keep the local bus. LHOLD is asserted, if Direct Slave Write FIFO becomes empty or Direct Slave Read FIFO becomes full. The local bus is dropped in either case when Local Bus Latency Timer is enabled and expires.

1.3 COMPATIBILITY OF PCI 9080 WITH PCI 9060, 9060ES, AND 9060SD

PCI 9080 is upward compatible with PCI 9060, 9060ES and 9060SD, except as noted in Table 1-2 and Section 4.1, “New Register Definitions Summary.”

1.3.1 Pin Compatibility

When upgrading from the PCI 9060, 9060ES or 9060SD, observe the following new pin definitions listed in Table 1-2.

Table 1-2. Pin Compatibility

Pin #	9060/ES/SD		PCI 9080	
	Pin Name	Description	Pin Name	Description
170	CLKSEL	Serial EEPROM Clock Select	NC	–
175	EE1MC	Optional Serial EEPROM Clock Source	EESEL	Serial EEPROM Select 1=93CS46 (1K bit) 0=93CS56 (2K bit)

1.3.2 Register Compatibility

All registers implemented in the 9060/ES/SD are implemented in the PCI 9080. There are a limited number of new bit definitions and several new registers. Refer to Section 4.1, “New Register Definitions Summary.”

1.4 COMPARISON OF PCI 9060, PCI 9060ES, PCI 9060SD, AND PCI 9080

Table 1-3. Comparison of the PCI 9060, PCI 9060ES, PCI 9060SD, and PCI 9080

Feature	PCI 9060	PCI 9060ES	PCI 9060SD	PCI 9080
Number of DMA Channel(s)	2	0	1	2
Local Address Spaces	2	2	3	3
Direct Master Mode	Yes	Yes	No	Yes
Mailbox Registers	Eight 32 bit	Four 32 bit	Four 32 bit	Eight 32 bit
Doorbell Registers	Two 32 bit	Two 8 bit	Two 8 bit	Two 32 bit
FIFOs	8	4	4	8
FIFO Depth—Direct Slave Write, Direct Master Write, DMA 0 Read and DMA 0 Write	8 Lwords (32 bytes)	16 Lwords (64 bytes)	16 Lwords (64 bytes)	32 Lwords (128 bytes)
FIFO Depth —Direct Slave Read, Direct Master Read, DMA 1 Read and DMA 1 Write	8 Lwords (32 bytes)	16 Lwords (64 bytes)	16 Lwords (64 bytes)	16 Lwords (64 bytes)
LLOCKo# Pin for Lock Cycles	No	Yes	Yes	Yes
WAITI# Pin for Wait State Generation	No	Yes	Yes	Yes
BPCLKo Pin; Buffered PCI Clock	No	Yes	Yes	Yes
DREQ# and DACK# Pins for Demand Mode DMA Support	Yes	No	Yes (Channel 1 only)	Yes
Register Addresses	—	Identical except 9060ES has no DMA registers and Tables 25, 26, and 43 were added	Identical, except 9060SD has one DMA register and Tables 4-29 and 4-30 were added	Identical except PCI 9080 has additional I_2O related registers and 30h, 34h, 40h, and 44h were remapped
Pin Out ***Note:*** *PCI 9080 includes all changes made for PCI 9060, PCI 9060ES, and PCI 9060SD.*	—	Signals deleted: DREQ0# (pin 29) DACK0# (pin 30) Input signals added: WAITI# (pin 6) BIGEND# (pin 48) Output signals added: BPCLKo (pin 168) LLOCKo# (pin 7)	Signals deleted: BREQ (pin 21) DMPAF# (pin 8) DREQ0# (pin 29) DACK0# (pin 30) BTERMo# (pin 28) Input signals added: WAITI# (pin 6) BIGEND# (pin 48) EOT0# (pin 164 in C mode, Pin 5 in J and S modes) Output signals added: BPCLKo (pin 168) LLOCKo# (pin 7)	Input signal added: EOT1# (pin 163) Signal changed: EESEL (pin 175)
Big/Little Endian Conversion	No	Yes	Yes	Yes
Spec. 2.1 Deferred Reads	No	Yes	Yes	Yes
Programmable Prefetch Counter	No	Yes	Yes	Yes
Write and Invalidate Cycle	No	Yes	Yes	Yes
Additional Device and Vendor ID Register	No	Yes	Yes	Yes
I_2O Messaging Unit	No	No	No	Yes
3.3 V PCI Bus Signaling	No	No	No	Yes

2. BUS OPERATION

2.1 PCI BUS CYCLES

PCI 9080 is compliant with PCI Specification v2.1. Refer to the PCI 2.1 spec for any specific features of the PCI bus.

2.1.1 PCI Target Command Codes

As a target, PCI 9080 allows access to PCI 9080 internal registers and local bus, using the commands listed in Table 2-1.

Table 2-1. PCI Target Command Codes

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

All read or write accesses to PCI 9080 can be byte, word, or longword (Lword) accesses. All memory commands are aliased to the basic memory commands. All I/O accesses to PCI 9080 are decoded to an Lword boundary. The byte enables are used to determine which bytes are read or written. An I/O access with illegal byte enable combinations is terminated with a Target Abort.

2.1.2 PCI Master Command Codes

PCI 9080 can access the PCI bus to perform DMA transfers or Direct Master Local to PCI Bus transfers. During the Direct master or DMA transfer, the command code assigned to PCI 9080 internal register location (PCI:6Ch)(LOC:ECh) bits [15:0] is used as the PCI command code (refer to Table 4-59[15:0]). Table 2-2 through Table 2-5 list the various PCI Master Command codes.

Note: *Programmable internal registers determine PCI command codes when the PCI 9080 is master.*

Note: *DMA cannot perform I/O or configuration accesses.*

2.1.2.1 DMA Master Command Codes

DMA controllers of PCI 9080 can generate the memory cycles listed in Table 2-2.

Table 2-2. DMA Master Command Codes

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

2.1.2.2 Direct Local to PCI Command Codes

For direct local to PCI bus accesses, PCI 9080 generates the cycles listed in Table 2-3 through Table 2-5.

Table 2-3. Local to PCI Memory Access

Command Type	Code (C/BE[3:0]#)
Memory Read	0110 (6h)
Memory Write	0111 (7h)
Memory Read Multiple	1100 (Ch)
Memory Read Line	1110 (Eh)

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

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

Table 2-5. Local to PCI Configuration Access

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

2.1.3 PCI Arbitration

PCI 9080 asserts output REQ# to request the PCI bus. PCI 9080 can be programmed using bit 23 of (PCI:08h or PCI:ACh)(LOC:88h or LOC:12Ch) (refer to Table 4-35[23]) to de-assert REQ# when it asserts FRAME# during a bus master cycle, or to keep REQ# asserted for the entire bus master cycle. PCI 9080 always de-asserts

REQ# for a minimum of two PCI clocks between bus master ownership that includes a target disconnect.

The Direct Master Write Delay bits (bits [15:14]) of the DMPBAM Register (PCI:28h)(LOC:A8h) can be programmed to delay the PCI 9080's assertion of the PCI REQ# signal during a Direct Master write cycle. This register can be programmed to wait 0, 4, 8, or 16 PCI bus clocks after the PCI 9080 has received its first write data from the local master and is ready to begin the PCI write transaction. This feature is useful in applications where the local master is bursting and the local bus clock is slower than the PCI bus clock. This allows write data to accumulate in the PCI 9080's Direct Master Write FIFO, which provides for better PCI bus utilization.

2.2 LOCAL BUS CYCLES

PCI 9080 connects a PCI host bus to several local processor bus types, as listed in Table 2-6. It operates in one of three modes, selected through mode pins 9 and 10, corresponding to three bus types—C, J, and S.

Table 2-6. Local Processor Bus Types

Bit 9	Bit 10	Mode	Bus Type
0	0	C	32-bit nonmultiplexed
0	1	J	32-bit multiplexed
1	0	S	16-bit multiplexed
1	1	Reserved	—

2.2.1 Local Bus Arbitration

When PCI 9080 owns the local bus, both its LHOLD output and LHOLDA input are asserted. When PCI 9080 samples BREQ asserted during a DMA transfer or Direct Slave write transfer, it gives up the local bus within two Lword transfers by de-asserting LHOLD and floating its local bus outputs if BREQ is gated or disabled, or if gating is enabled and the Local Bus Latency Timer expires. The Local Arbiter can now grant the local bus to another local master. After PCI 9080 samples that its LHOLDA is de-asserted and its local pause timer is zero, it re-asserts LHOLD to request the local bus. When the PCI 9080 receives LHOLDA, it drives the bus and continues from where it left off.

2.2.2 Local Bus Direct Master

Local bus cycles can be continuous single or burst cycles (programmable by way of the PCI 9080 internal registers). As a local bus target, PCI 9080 allows access to PCI 9080 internal registers and PCI bus.

In C and J modes, local bus direct master accesses to the PCI 9080 must be for a 32 bit nonpipelined bus. In S mode, local bus direct master accesses to the PCI 9080 must be for a 16 bit nonpipelined bus.

2.2.3 Local Bus Direct Slave

PCI Bus Master read/write to local bus (PCI 9080 is a PCI bus target and local bus master).

2.2.3.1 Ready/Wait State Control

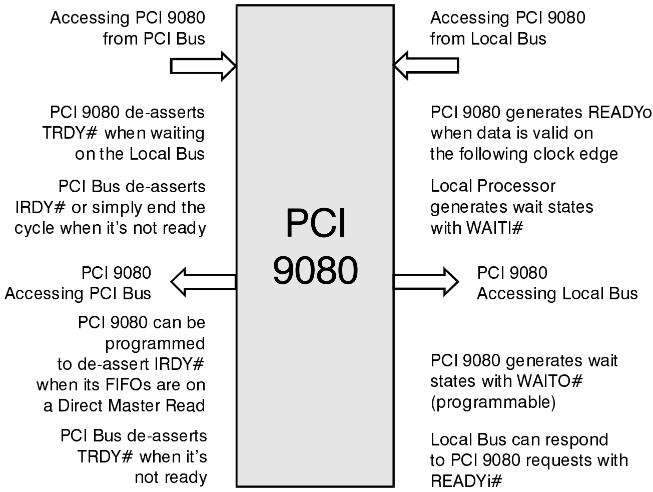

Accessing PCI 9080 from PCI Bus
Accessing PCI 9080 from Local Bus
PCI 9080 de-asserts TRDY# when waiting on the Local Bus
PCI 9080 generates READYo# when data is valid on the following clock edge
PCI Bus de-asserts IRDY# or simply end the cycle when it's not ready
Local Processor generates wait states with WAITI#
PCI 9080
PCI 9080 Accessing PCI Bus
PCI 9080 Accessing Local Bus
PCI 9080 can be programmed to de-assert IRDY# when its FIFOs are on a Direct Master Read
PCI 9080 generates wait states with WAITO# (programmable)
PCI Bus de-asserts TRDY# when it's not ready
Local Bus can respond to PCI 9080 requests with READYi#

Figure 2-1. Wait States

If READYi# input is disabled, external READYi# input has no effect on wait states for a local access. Wait states between data cycles are generated internally by a wait state counter. Wait state counter is initialized with its configuration register value at the start of each data access.

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

BTERM# input is not sampled until the wait state counter is 0. BTERM# overrides READYi# when BTERM# is asserted.

2.2.3.1.1 Wait State—Local Side

With Direct Master mode and accessing PCI 9080 registers (PCI 9080 local as slave):

- PCI 9080 generates wait states with READYo#
- Local processor generates wait states with WAITI#

With Direct Slave and DMA modes (PCI 9080 local as master):

- PCI 9080 generates wait states with WAITO#
- Local processor generates wait states with READYi#
- Use Table 4-39[21:18, 5:2], Table 4-62[5:2], and Table 4-67[5:2] to program the number of wait states

2.2.3.1.2 Wait State—PCI Side

When the wait state occurs on the PCI side, Master throttles IRDY# and Slave throttles TRDY#.

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

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

Mode	Burst	Bterm	Result
Single Cycle	0	0	One ADS# per data (default)
Single Cycle	0	1	Still one ADS# per data
Burst-4	1	0	One ADS# per four data (use this mode for i960)
Burst Forever	1	1	One ADS# per BTERM#

On the local side, BLAST# and BTERM# perform the following:

- If burst is enabled (Table 4-39[26,24] for non-DMA, Table 4-62[8] and Table 4-67[8] for DMA), but Bterm is disabled (Table 4-39[7], Table 4-62[7] and Table 4-67[7]), then PCI 9080 bursts four Lwords. BLAST# is generated at the fourth Lword (LA[3:2]=11), new ADS# at the first Lword (LA[3:2]=00) of the next burst.
- If BTERM# sampling is enabled and BTERM# is low, PCI 9080 forces a new ADS#, but does not generate a new BLAST#.
- BTERM# input is only valid when the PCI 9080 is the Master of the local bus (Direct Slave or DMA modes).
- BTERM# is generated by external logic. It is input to the PCI 9080 (and i960) and used to tell the PCI 9080 (and i960) to break up a burst cycle.
- BTERM# is used for example to signal memory access is crossing the page boundary.

On the PCI side, burst is always enabled.

Note: *If Bterm is disabled, the PCI 9080 performs the following:*

- *32 bit local bus—Burst up to four Lwords*
- *16 bit local bus—Burst up to two Lwords*
- *8 bit local bus—Burst up to one Lword*

In every case, it performs four transactions.

Note: *In the following sections, Bterm refers to PCI 9080 internal register bit. BTERM# refers to the PCI 9080 external signal.*

2.2.3.2.1 Burst Mode

If bursting is enabled and BTERM# input is not enabled, bursting can start on any boundary and continue up to an address boundary, as described in Table 2-8. After the data at the boundary is transferred, PCI 9080 generates a new address cycle (ADS#).

Table 2-8. Burst Mode

Bus Mode	Burst
C, J	32-bit bus—Four Lwords or up to a quad Lword boundary (LA3, LA2 = 11)
C, J	16-bit bus—Four words or up to a quad word boundary (LA2, LA1 = 11)
C, J	8-bit bus—Four bytes or up to a quad byte boundary (LA1, LA0 = 11)
S	16-bit bus—Eight words or up to a quad Lword boundary (LA3, LA2 = 11)

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

Bterm mode enables PCI 9080 to perform long bursts to devices that can accept longer than four Lword bursts. PCI 9080 generates one address cycle and continues to burst data. If a device requires a new address cycle after a certain address boundary, it can assert BTERM# input to cause the PCI 9080 to generate a new address cycle. BTERM# input acknowledges the current data transfer and requests that a new address cycle be generated (ADS#). The address will be for the next data transfer. If Bterm mode is enabled, PCI 9080 asserts BLAST# only if its FIFOs become FULL/EMPTY or if a transfer is complete.

Note: *If BTERM# is asserted, BLAST# does not assert until the previously described conditions are met.*

2.2.3.2.3 Partial Lword Accesses

Lword accesses in which not all byte enables are asserted are broken into single address and data cycles, as listed in Table 2-9.

Table 2-9. Partial Lword Accesses

Register Value (PCI:18h)(LOC:98h)		Result
Burst Enable	Bterm Enable	(Number of Transfers)
0	0	Single Cycle (Default)
0	1	Single Cycle
1	0	Burst four Lwords at a time
1	1	Continuous Burst Mode

2.2.3.3 Recovery States

In J and S modes, PCI 9080 inserts one recovery state between the last data transfer and next address cycle.

PCI 9080 does not support the 80960J feature of using READYi# input to add recovery states. No additional recovery states are added if READYi# input remains asserted during the last data cycle.

2.2.3.4 Local Bus Read Accesses

For all single cycle local bus read accesses, PCI 9080 reads only bytes corresponding to byte enables requested by the PCI initiator. For all burst cycle single cycle bus read accesses, PCI 9080 reads only Lwords.

2.2.3.5 Local Bus Write Accesses

For local bus writes, only the bytes specified by a PCI bus master or PCI 9080 DMA controller are written. Access to an 8- or 16-bit bus results in the PCI bus Lword being broken into multiple local bus transfers. For each transfer, the byte enables are encoded as in the 80960C to provide local address bits LA[1:0].

2.2.3.6 Direct Slave Write Accesses—8- and 16-Bit Buses

A Direct PCI access to an 8- or 16-bit bus results in the PCI bus Lword being broken into multiple local bus transfers. For each transfer, the byte enables are encoded as in the 80960C to provide local address bits LA[1:0].

2.2.3.7 Local Bus Data Parity

There is one data parity pin for each byte lane of the PCI 9080 data bus (DP[3:0]). Even data parity is generated for each lane during local bus reads from PCI 9080 and during PCI 9080 master writes to the local bus.

Even data parity is checked during local bus writes to PCI 9080 and during PCI 9080 reads from the local bus. Parity is checked for each byte lane with an asserted byte enable. PCHK# is asserted in the clock cycle following the data being checked if a parity error is detected.

Generation or use of local bus data parity is optional. The signals on data parity pins do not effect operation of PCI 9080. PCI bus parity checking and generation is independent of local bus parity checking and generation.

2.2.3.8 Local Bus Little/Big Endian

PCI bus is a Little Endian bus (that is, data is Lword aligned to the lowermost byte lane). Byte 0 (address 0) appears in AD[7:0], Byte 1 appears in AD[15:8], Byte 2 appears in AD[23:16] and Byte 3 appears in AD[31:24].

PCI 9080 local bus can be programmed to operate in Big or Little Endian mode, as listed in Table 2-10.

Table 2-10. Big / Little Endian Program Mode

BIGEND# Pin	Register 1=Big, 0=Little	Endian
0	0	Big
0	1	Big
1	0	Little
1	1	Big

For Configuration cycles, refer to Table 4-36[0]. For Direct Master, Memory, and I/O cycles, refer to Table 4-36[1]. For Direct Slave cycles, refer to Table 4-36[2], Space 0, and Table 4-36[3], expansion ROM.

In Big Endian mode, PCI 9080 transposes the data byte lanes. Data is transferred as listed in Table 2-11 through Table 2-15.

2.2.3.8.1 32 Bit Local Bus—Big Endian Mode

Data is Lword aligned to the uppermost byte lane. Byte lanes and burst orders are listed in Table 2-11 and illustrated in Figure 2-2.

Table 2-11. Upper Lword Lane Transfer

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

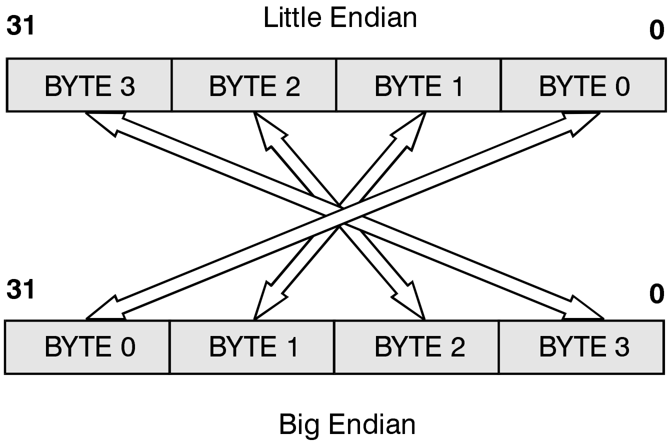


Figure 2-2. Big/Little Endian—32 Bit Local Bus

2.2.3.8.2 16 Bit Local Bus—Big Endian Mode

For a 16 bit local bus, PCI 9080 can be programmed to use the upper or lower word lane. Byte lanes and burst order are listed in Table 2-12 and Table 2-13 and illustrated in Figure 2-3.

Table 2-12. Upper Word Lane Transfer

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

Table 2-13. Lower Word Lane Transfer

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

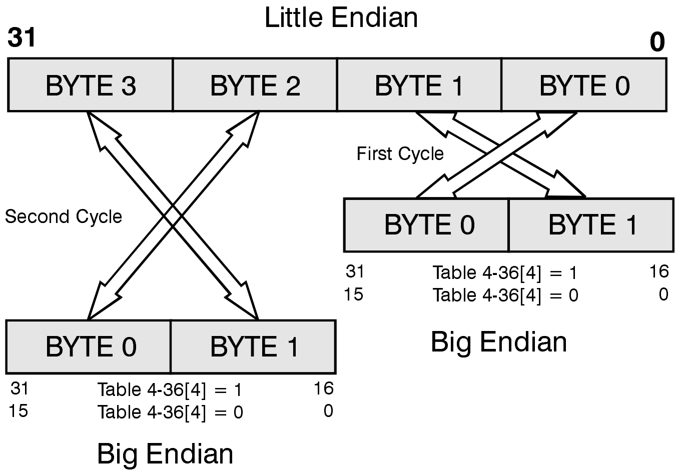


Figure 2-3. Big/Little Endian—16 Bit Local Bus

2.2.3.8.3 8 Bit Local Bus—Big Endian Mode

For an 8 bit local bus, PCI 9080 can be programmed to use the upper or lower byte lane. Byte lanes and burst order are listed in Table 2-14 and Table 2-15 and illustrated in Figure 2-4.

Table 2-14. Upper Byte Lane Transfer

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

Table 2-15. Lower Byte Lane Transfer

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

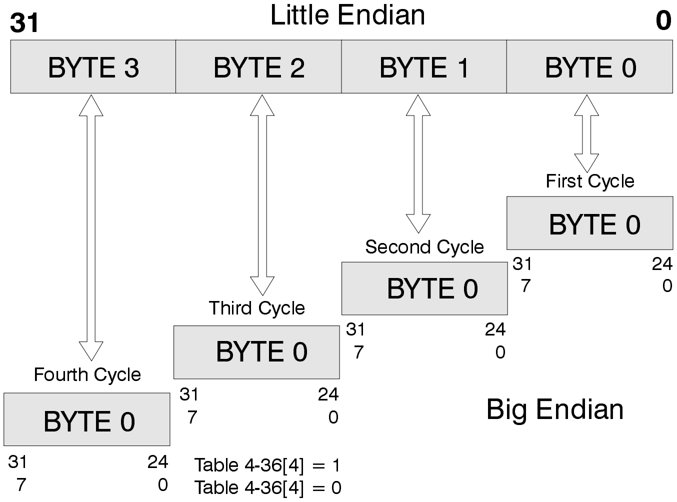


Figure 2-4. Big/Little Endian—8 Bit Local Bus

For each of the following transfer types, PCI 9080 local bus can be independently programmed to operate in Little Endian or Big Endian mode:

- Local bus accesses to PCI 9080 configuration registers
- Direct Slave PCI accesses to Local Address Space 0
- Direct Slave PCI accesses to Local Address Space 1
- Direct Slave PCI accesses to expansion ROM
- DMA Channel 0 accesses to the local bus
- DMA Channel 1 accesses to the local bus
- Direct Master Accesses to PCI bus

For local bus configuration accesses, an input pin can be used to dynamically change the Endian mode.

Notes: *PCI bus is always Little Endian mode.*

Only byte lanes are swapped, not individual bits.

3. FUNCTIONAL DESCRIPTION

Functional operation described can be changed or modified, depending on the register configuration.

3.1 RESET

3.1.1 PCI Bus Input RST#

PCI bus RST# input pin is a PCI host reset. It causes all PCI bus outputs to float, resets the entire PCI 9080 and causes the local reset output, LRESETo#, to be asserted. If you have a PCI host, Table 4-11[2:0] (Master Enable, Memory Space, I/O Space) is programmed by the host after initialization is complete (Table 4-59[31]=1). (Refer to Figure 3-1.)

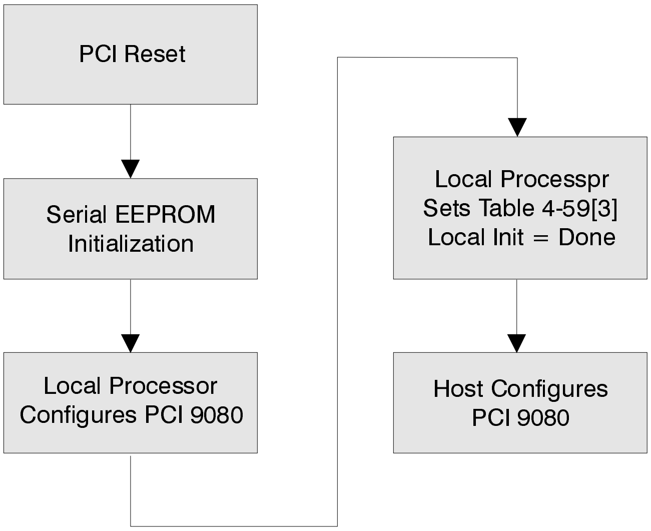

PCI Reset
Serial EEPROM Initialization
Local Processor Configures PCI 9080
Local Processpr Sets Table 4-59[3] Local Init = Done
Host Configures PCI 9080

Figure 3-1. Reset and Initialization Process

3.1.2 Software Reset LRESETo#

When asserted, the LRESETo# Software Reset bit (Table 4-59[30]) resets PCI 9080 Local Configuration and Local DMA Registers. However, it does not reset the PCI Configuration and Shared Runtime Registers. When the bit is set, PCI 9080 responds to PCI accesses, but not to local accesses. PCI 9080 remains in this condition until PCI host clears the bit. The serial EEPROM is reloaded if Table 4-59[29] is set.

3.1.3 Local Bus Input LRESETi#

When asserted, the LRESETi# input resets the local bus portion of PCI 9080, clears all local configuration and DMA registers and causes LRESETo# output to be asserted.

3.1.4 Local Bus Output LRESETo#

LRESETo# is asserted when PCI bus RST# input is asserted, the LRESETi# input is asserted, or the software reset bit in the Init Control Register is set to 1.

3.1.5 Software Reset

A host on the PCI bus can set the software reset bit in the Init Control Register to reset PCI 9080 and assert the LRESETo# output. All local configuration and DMA registers reset. PCI configuration registers do not reset. When the software reset bit is set, PCI 9080 responds to PCI accesses, but not to local bus accesses. PCI 9080 remains in this reset condition until the PCI host clears the bit.

Note: *The local side cannot clear this reset bit because the local bus is in a reset state.*

3.2 PCI 9080 INITIALIZATION

PCI 9080 configuration registers can be programmed by an optional serial EEPROM and/or by a local processor, as listed in Table 3-1. The serial EEPROM can be reloaded by setting bit 29 of (LOC:ECh), Serial EEPROM Control Register (refer to Table 4-59[29]).

In general, PCI 9080 retries all PCI cycles until the “Local Init Done bit” is set or NB# is low.

Note: *Internal configuration register can also be accessed by the PCI host processor after power-on.*

Table 3-1. NB# and Serial EEPROM Guidelines

NB#	Serial EEPROM	System Boot Condition
Low	No	Boot with PCI 9080 default values.
	Programmed	Boot with serial EEPROM values.
	Blank	Not recommended (uses default values).
High	No	Local processor programs PCI 9080 registers, then sets Local Init Status (Table 4-59[31] = done). ***Note:*** *Some systems hang if Direct Slave reads and writes take too long (during initialization, the PCI host also performs Direct Slave accesses). Value of PCI Target Retry Delay Clocks (Table 4-39[31:28]) may resolve this problem.*
	Programmed	Load serial EEPROM, but local processor can reprogram PCI 9080.
	Blank	Load serial EEPROM (default values), but local processor can reprogram PCI 9080. System can boot. ***Note:*** *Serial EEPROM can be programmed through PCI 9080 after system boots in this condition.*

3.2.1 Serial EEPROM Initialization

During serial EEPROM initialization, PCI 9080 response to PCI target accesses is RETRY. During serial EEPROM initialization, PCI 9080 response to a local processor is to hold off READYo#.

3.2.2 Local Initialization

PCI 9080 issues a RETRY to all PCI accesses until the "Local Init Done bit" in the Init Control Register is set. "Init Done bit" is programmable through local bus configuration accesses. If this bit is not going to be set by a local processor, then NB# input should be tied low. Holding NB# input low externally forces the Local Init Done bit to 1.

PCI 9080 default values are used if a serial EEPROM is not present and local Init Status bit is set to 1 by holding the NB# input low or set by the local processor.

3.3 SERIAL EEPROM

After reset, PCI 9080 attempts to read the serial EEPROM to determine its presence. An active low start bit indicates the serial EEPROM is present (PCI 9080 supports 93CS46 (1K) or 93CS56 (2K), selectable by way of the EESEL pin). (Refer to the manufacturer’s data sheet for the particular serial EEPROM being used.) The first word is then checked to verify the serial EEPROM is programmed. If the first word (16 bit) is all ones, a blank serial EEPROM PCI 9080 uses default values instead.

The 5 V serial EEPROM clock (EESK, pin 173) is derived from the PCI clock. PCI 9080 generates the serial EEPROM clock by internally dividing the PCI clock by 32.

The serial EEPROM can be read or programmed from the PCI or local bus. Bits [27:24] of the Serial EEPROM Control Register (refer to Table 4-59[27:24]) controls the PCI 9080 pins that enable the reading or writing of serial EEPROM data bits. (Refer to the manufacturer’s data sheet for the particular serial EEPROM being used.)

PCI 9080 has three serial EEPROM load options:

- **Short Load Mode**—SHORT# input pin is pulled down and PCI 9080 loads five Lwords from the serial EEPROM
- **Long Load Mode**—SHORT# input pin is pulled up, bit 25 of the Local Bus Region Descriptor Register (LOC:98h) is set to 0, and PCI 9080 loads 17 Lwords from the serial EEPROM (refer to Table 4-39)
- **Extra Long Load Mode**—SHORT# input pin is pulled up, bit 25 of the Local Bus Region Descriptor Register (LOC:98h) is set to 1 during Long Load from the serial EEPROM, and PCI 9080 loads 21 Lwords from the serial EEPROM (refer to Table 4-39)

3.3.1 Short Serial EEPROM Load

The registers listed in Table 3-2 are loaded from serial EEPROM after reset is de-asserted if SHORT# pin is low. The serial EEPROM is organized in words (16 bit). PCI 9080 first loads MSW (Most Significant Word; bits [31:16]), starting from the most significant bit (bit 31). PCI 9080 then loads LSW (Least Significant Word; bits [15:0]), starting again from the most significant bit (bit 15). Therefore, PCI 9080 loads Device ID, Vendor ID, class code, and so forth. The five 32-bit words are stored sequentially in the serial EEPROM.

Table 3-2. Short Serial EEPROM Load Registers

Serial EEPROM Offset	Description	Sample Serial EEPROM Value
0	Device ID	9080
2	Vendor ID	10B5
4	Class Code	0680
6	Class Code, Revision	0002
8	Maximum Latency, Minimum Grant	0000
A	Interrupt Pin, Interrupt Line Routing	0100
C	MSW of Mailbox 0 (User Defined)	xxxx
E	LSW of Mailbox 0 (User Defined)	xxxx
10	MSW of Mailbox 1 (User Defined)	xxxx
12	LSW of Mailbox 1 (User Defined)	xxxx

3.3.2 Long Serial EEPROM Load

The registers listed in Table 4-39 are loaded from serial EEPROM after reset is de-asserted if SHORT# pin is high. The serial EEPROM is organized in words (16 bit). PCI 9080 first loads MSW (Most Significant Word; bits [31:16]), starting from the most significant bit (bit 31). PCI 9080 then loads LSW (Least Significant Word; bits [15:0]), starting again from the most significant bit (bit 15). Therefore, PCI 9080 will load Device ID, Vendor ID, class code, and so forth.

The serial EEPROM value can be entered into a DATA I/O programmer in the order shown below. The values shown are examples and must be modified for each particular application. The 34 16-bit words listed in the table are stored sequentially in the serial EEPROM.

Table 3-3. Long Serial EEPROM Load Registers

Serial EEPROM Offset	Description
0	Device ID
2	Vendor ID
4	Class Code
6	Class Code, Revision
8	Maximum Latency, Minimum Grant
A	Interrupt Pin, Interrupt Line Routing
C	MSW of Mailbox 0 (User Defined)
E	LSW of Mailbox 0 (User Defined)
10	MSW of Mailbox 1 (User Defined)
12	LSW of Mailbox 1 (User Defined)
14	MSW of Range for PCI to Local Address Space 0
16	LSW of Range for PCI to Local Address Space 0
18	MSW of Local Base Address (Remap) for PCI to Local Address Space 0
1A	LSW of Local Base Address (Remap) for PCI to Local Address Space 0
1C	MSW of Local Arbitration Register
1E	LSW of Local Arbitration Register
20	MSW of Local Bus Big/Little Endian Descriptor Register
22	LSW of Local Bus Big/Little Endian Descriptor Register
24	MSW of Range for PCI to Local Expansion ROM
26	LSW of Range for PCI to Local Expansion ROM
28	MSW of Local Base Address (Remap) for PCI to Local Expansion ROM
2A	LSW of Local Base Address (Remap) for PCI to Local Expansion ROM
2C	MSW of Bus Region Descriptors for PCI to Local Accesses
2E	LSW of Bus Region Descriptors for PCI to Local Accesses
30	MSW of range for Direct Master to PCI
32	LSW of range for Direct Master to PCI
34	MSW of Local Base Address for Direct Master to PCI Memory
36	LSW of Local Base Address for Direct Master to PCI Memory
38	MSW of Local Bus Address for Direct Master to PCI IO/CFG
3A	LSW of Local Bus Address for Direct Master to PCI IO/CFG
3C	MSW of PCI Base Address (Remap) for Direct Master to PCI
3E	LSW of PCI Base Address (Remap) for Direct Master to PCI
40	MSW of PCI Configuration Address Register for Direct Master to PCI IO/CFG
42	LSW of PCI Configuration Address Register for Direct Master to PCI IO/CFG

3.3.3 Extra Long Serial EEPROM Load

An Extra Long Load mode is provided in the PCI 9080 (refer to Table 4-39) to load an additional five Lwords from the serial EEPROM. If bit 25 is set to 1 in the Local Bus Region Descriptor Register (LOC:98h) (refer to Table 4-39), the following five Lword registers are loaded in addition to normal Long Load process (refer to Section 3.3.2, “Long Serial EEPROM Load”). Bit 25 must be set to 1 during the Long Load Process. (Refer to Table 3-4.)

Table 3-4. Extra Long Serial EEPROM Load Registers

Serial EEPROM Offset	Description
44	Subsystem ID
46	Subsystem Vendor ID
48	MSW of Range for PCI to Local Address Space 1 (1 MB)
4A	LSW of Range for PCI to Local Address Space 1 (1 MB)
4C	MSW of Local Base Address (Remap) for PCI to Local Address Space 1
4E	LSW of Local Base Address (Remap) for PCI to Local Address Space 1
50	MSW of Bus Region Descriptors (Space 1) for PCI to local accesses
52	LSW of Bus Region Descriptors (Space 1) for PCI to local accesses
54	MSW of PCI Base Address for local expansion ROM
56	LSW of PCI Base Address for local expansion ROM

3.3.4 Recommended Serial EEPROMs

A 1K bit (National NM93CS46 or compatible) or 2K bit (National NM93CS56 or compatible) device can be used. Table 3-5 lists the recommended serial EEPROM loads. Refer also to Table 5-2 in Section 5, “Pin Description.”

Table 3-5. Recommended Serial EEPROM Loads

Load	Unused Bytes For CS46 (1K)	Unused Bytes For CS56 (2K)
Short	108	364
Long	60	316
Extra Long	40	296

Note: *PCI 9080 does not support serial EEPROMs that do not support sequential read and write (such as the NM93C46 or NM93C56).*

3.3.5 Programming the Serial EEPROM

The serial EEPROM can be written or read, using bits [28:24] of the Serial EEPROM Control Register (refer to Table 4-59[28:24]).

3.4 INTERNAL REGISTER ACCESS

PCI 9080 chip provides several internal registers, allowing for maximum flexibility in bus interface design and performance. The register types are accessible from both the PCI and local buses, including the following:

- PCI configuration registers
- Local configuration registers
- Mailbox registers
- Doorbell registers
- DMA registers
- Messaging queue registers (I_2O)

Figure 3-2 illustrates how these registers are accessed.

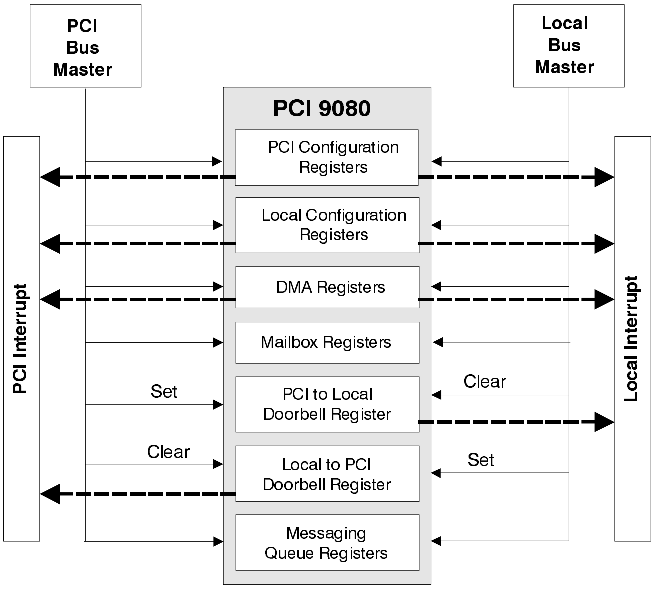

PCI Bus Master
Local Bus Master
PCI 9080
PCI Configuration Registers
Local Configuration Registers
DMA Registers
Mailbox Registers
Set
PCI to Local Doorbell Register
Clear
Clear
Local to PCI Doorbell Register
Set
Messaging Queue Registers
PCI Interrupt
Local Interrupt

Figure 3-2. PCI 9080 Internal Register Access

3.4.1 PCI Bus Access to Internal Registers

PCI 9080 “PCI configuration registers” can be accessed from the PCI bus with a configuration Type 0 cycle.

PCI 9080 internal registers can be accessed by a memory cycle, with the PCI bus address that matches the base address specified in the PCI Base Address 0 for Memory Mapped Configuration Register of PCI 9080. They can also be accessed by an I/O cycle, with the PCI bus address matching the base address specified in the PCI Base Address 1 for I/O Mapped Configuration Register of the PCI 9080.

All PCI read or write accesses to the PCI 9080 registers can be byte, word, or Lword accesses. All PCI memory accesses to the PCI 9080 registers can be burst or non-burst. PCI 9080 responds with a PCI Disconnect for all burst I/O accesses to PCI 9080 registers.

3.4.2 Local Bus Access to Internal Registers

The local processor can access all the internal registers of the PCI 9080 through either internal or external address decode logic. PCI 9080 provides an Address Decode Mode Pin (ADMODE) that selects whether the internal address decode logic is used or whether the designer will supply an external chip select from an external address decoder. Figure 3-3 illustrates how the dual address decode logic works.

If the Address Decode Mode pin is set to 1, internal PCI 9080 address decode logic is enabled. In this mode, PCI 9080 internal registers are selected when local address bits LA[31:29] match input address select pins S[2:0]. If the Address Decode Mode pin is set to 0, PCI 9080 responds to local bus access when S0 is asserted low through external chip select logic.

Notes: *S0 must be decoded while ADS# is low.*

If ADMODE is 1 LA[31:29], specify 512 MB of local memory space allocated for accessing internal registers.

All local read or write accesses to the PCI 9080 registers can be byte, word, or Lword accesses. All local accesses to the PCI 9080 registers can be burst or non-burst.

For C and J modes, accesses must be for a 32 bit nonpipelined bus. PCI 9080 READYo# indicates a data transfer is complete.

For S mode, accesses must be for a 16 bit nonpipelined bus. PCI 9080 READYo# indicates a data transfer is complete.

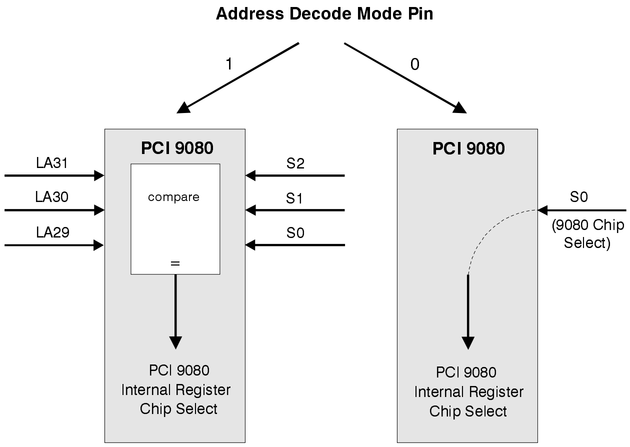


Figure 3-3. Dual Address Decode Mode

3.5 RESPONSE TO FIFO FULL/EMPTY

Table 3-6 lists the response of the PCI 9080 to full and empty FIFOs.

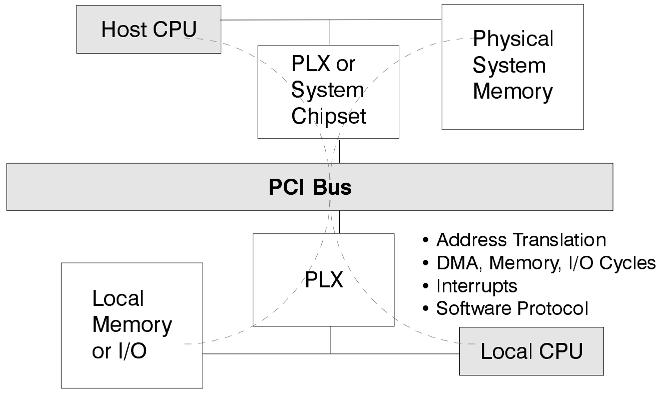
Table 3-6. Response to FIFO Full/Empty

Mode	Direction	FIFO	PCI Side	Local Side
Direct Master Write	Local to PCI	Full	No action	De-assert READYo#
		Empty	De-assert REQ# (off PCI bus)	No action
Direct Master Read	PCI to Local	Full	De-assert REQ# or throttle IRDY#	No action
		Empty	No action	De-assert READYo#
Direct Slave Write	PCI to Local	Full	Disconnect or throttle TRDY#	No action
		Empty	No action	De-assert LHOLD, assert BLAST# (see Note)
Direct Slave Read	Local to PCI	Full	No action	De-assert LHOLD, assert BLAST# (see Note)
		Empty	Throttle TRDY#	No action
DMA	Local to PCI	Full	No action	De-assert LHOLD, assert BLAST#
		Empty	De-assert REQ#	No action
	PCI to Local	Full	De-assert REQ#	No action
		Empty	No action	De-assert LHOLD, assert BLAST#

Note: *De-assert of LHOLD depends on MARBR[21].*

3.6 DIRECT DATA TRANSFER MODES

Figure 3-4 and Figure 3-5 illustrate the direct data transfer modes. Refer also to Table 3-6 for responses to full and empty FIFOs.



Host CPU accesses Local Memory or I/O = Direct Slave read/write
Local CPU accesses System Memory = Direct Master read/write
PCI 9080 read/write from System Memory and write/read to
local memory = DMA read/write

Figure 3-4. Direct Master, Direct Slave, and DMA

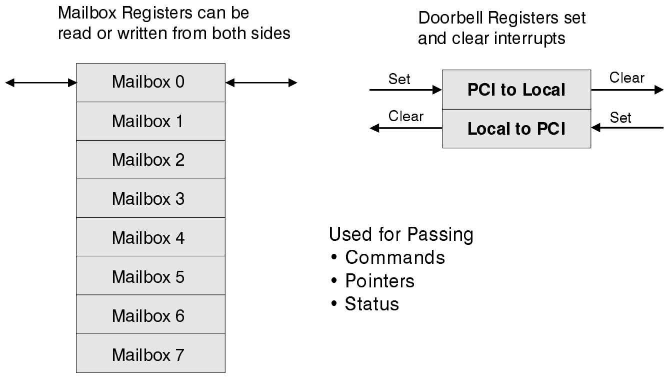


Figure 3-5. Mailbox/Doorbell Message Passing

3.6.1 Direct Master Operation (Local Master to PCI Target)

PCI 9080 supports the direct access of the PCI bus by the local processor or an intelligent controller. Master mode must be enabled in the PCI Command Register. Five registers are used to define local to PCI access:

- Range
- Local Base Address for Direct Master to PCI Memory Register
- Local Base Address for Direct Master to PCI IO/CFG Register

- PCI Configuration Address Register for Direct Master to PCI IO/CFG
- PCI Base Address

3.6.1.1 Decode

The Range register specifies the local address bits to use for decoding a Local to PCI access. The local processor can perform only memory cycles. Therefore, the Local Base Address for Direct Master to PCI Memory Register is used to decode an access to PCI memory space and the Local Base Address for Direct Master to PCI IO/CFG Register is used to decode an access to PCI I/O space or PCI bus configuration cycle access.

3.6.1.2 FIFOs

For Direct Master memory access to the PCI bus, PCI 9080 has a 32 Lword (128 byte) write FIFO and a 16 Lword (64 byte) read FIFO. FIFOs enable the local bus to operate independently of the PCI bus and allows high-performance bursting on local and PCI buses. In a Direct Master Write, the local processor (Master) writes data to PCI (Slave). In a Direct Master Read, the local processor (Master) reads data from PCI (Slave). Figure 3-6 and Figure 3-7 illustrate the FIFOs during a Direct Master Write and Read.

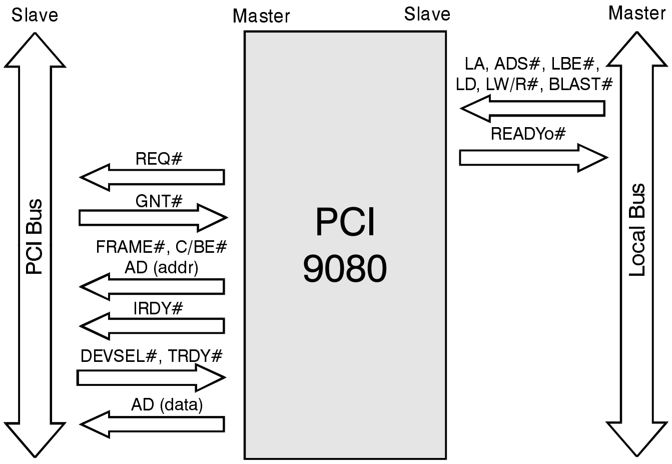


Figure 3-6. Direct Master Write

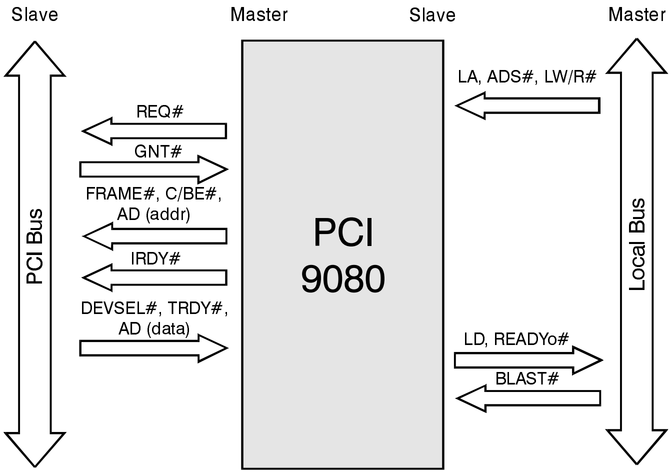


Figure 3-7. Direct Master Read

3.6.1.3 Memory Access

The local processor can read or write to the PCI memory. PCI 9080 converts the local read/write access. 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.

Writes—PCI 9080 continues to accept writes and return READYo# until write FIFO is full. It then holds off READYo# until space becomes available in the write FIFO. A programmable Direct Master FIFO “almost full” status output is provided (DMPAF#).

Reads—PCI 9080 holds off READYo# while gathering an Lword from the PCI bus. Programmable prefetch modes are available if prefetch is enabled: prefetch, 4, 8, 16, or continuous until Direct Master cycle ends. The read cycle is terminated when the local BLAST# input is asserted. Unused read data is flushed from the FIFO.

PCI 9080 does not prefetch read data for single cycle Direct Master reads (local BLAST# input asserted during first data phase). In this case, PCI 9080 reads a single PCI Lword.

For Direct Master single cycle reads, PCI 9080 asserts the same PCI bus byte enables as asserted on the local bus.

For multiple cycle reads, PCI 9080 reads entire Lwords (all PCI byte enables are asserted), regardless of local byte enables.

If the prefetch limit bit DMPBAM (PCI:28h)(LOC:A8h) (refer to Table 4-43[11]) is enabled, PCI 9080 does not prefetch past a 4K boundary. Also, the local side must not cross a 4K boundary during a burst read.

PCI 9080 never prefetches beyond the region specified for direct master accesses.

3.6.1.4 IO/CFG Access

When a Local Direct Master I/O access to the PCI bus is made, the PCI Configuration Address Register's Configuration Enable bit determines if an I/O or configuration access is to be made to the PCI bus.

Local burst accesses are broken into single PCI I/O address/data cycles. PCI 9080 does not prefetch read data for I/O and CFG reads.

For Direct Master I/O or Configuration cycles, PCI 9080 asserts the same PCI bus byte enables as asserted on the local bus.

3.6.1.5 I/O

If the Configuration Enable bit is clear, a single I/O access is made to the PCI bus. The local address, remapped decode address bits and the local byte enables are encoded to provide the address and is output with an I/O read or write command during the PCI address cycle.

For writes, data is loaded into the write FIFO and READYo# returned to the Local bus. For reads, PCI 9080 holds off READYo# while gathering an Lword from the PCI bus.

When the I/O remap select bit is set to a value of 1, these PCI address bits [31:16] are forced to a value of 0 (refer to Table 4-43[13]).

3.6.1.6 CFG (PCI Configuration Type 0 or Type 1 Cycles)

If the Configuration Enable bit is set, a CFG access is made to the PCI bus. In addition to enabling the configuration (bit 31) of (PCI:2Ch)(LOC:ACh) (refer to Table 4-44[31]), the user must provide all register information. The register number (bits [7:2]) or the device number (bits [15:11]) must be modified and a new CFT read/write cycle must be performed before other registers or devices can be accessed.

If the PCI Configuration Address Register selects a Type 0 command, bits [10:0] from the register are copied to address bits [10:0]. Bits [15:11] ("device number") are translated into a single bit being set in PCI address bits [31:11]. PCI address bits [31:11] can be used as a device select. For a Type 1 command, bits [23:0] are copied from the register to bits [23:0] of the PCI address. PCI address bits [31:24] are 0. A configuration read or write command code is output with the address during the PCI address cycle (refer to Table 4-44).

For writes, local data is loaded into the write FIFO and READYo# is returned. For reads, PCI 9080 holds off READYo# while gathering an Lword from the PCI bus.

Example 1—To perform a Type 0 configuration cycle to PCI device on AD[21].

1. PCI 9080 must be configured to allow Direct Master access to the PCI buses. PCI 9080 must also be set to respond to I/O space accesses. These bits must be set in (LOC:04h):
   - Field 0 = I/O Space = 1
   - Field 2 = Master Enable = 1
2. Direct Master Range is selected by the board designer. For this example, use a range of 1 MB:
   - 1 MB = 2^{20} = 000FFFFFh

   Value to program into the range register is the inverse of 000FFFFFh, which is FFF00000h:
   - (LOC:9Ch) = FFF00000h
3. Local Base Address for Direct Master to PCI IO/CFG is determined by the board designer. For this example, use 40000000h:
   - (LOC:A4h) = 40000000h
4. PCI Address (Remap) for Direct Master to PCI Memory Register must enable Direct Master I/O access. This bit must be set in (LOC:A8h):
   - Field 1 = Direct Master I/O Access Enable = 1
5. PCI must know which PCI device and PCI configuration register the PCI configuration cycle is accessing. For this example, access the PCI device on AD[21]. Also access the PCI Base Address 0 for Memory Mapped Configuration Register (PCI:10h). (PCI:10h) is the fourth register, counting from 0 (use Table 4-5, "PCI Configuration Registers," for reference). These bits must be set in (LOC:ACh):
   - Fields 1:0 = Configuration Type 0 = 00b
   - Fields 7:2 = Register number = The fourth register, and therefore must program a 4 into this field, beginning with bit 2 = 000100b
   - Fields 10:8 = Function Number = 000b
   - Fields 15:11 = Device Number = n-11, where n is the value in AD[n]=21-11 = 10 = 01010b
   - Fields 23:16 = Bus Number = 00000000b
   - Field 31 = Configuration Enable = 1

The register number (bits [7:2]) or device number (bits [15:11]) must be modified and a new CFT read/write cycle must be performed before other registers or devices can be accessed.

3.6.1.7 Direct Bus Master Lock

PCI 9080 supports direct local to PCI bus exclusive accesses (locked atomic operations). A locked operation must start with the local bus input LLOCK# being asserted during a Direct Master bus read cycle. Refer to the timing in Section 8, “Timing Diagrams.”

3.6.1.8 Master/Target Abort

PCI 9080 Master/Target abort logic enables a local bus master to perform a Direct Master bus poll of devices to determine whether the devices exist (typically when the local bus performs configuration cycles to the PCI bus).

If a PCI Master, Target Abort, or Retry Time-out is encountered during a transfer, PCI 9080 asserts LSERR# if enabled (refer to INTCSR[1:0], Table 4-58) (can be used as an NMI). If the local bus master is waiting for a READYo#, it is asserted along with BTERMo#. The local master’s interrupt handler can take the appropriate application specific action. It can then clear the abort bits in the PCI Status configuration register (refer to Table 4-12) of the PCI 9080 to clear the LSERR# interrupt and re-enable Direct Master transfers.

If a local bus master is attempting a burst read from a nonresponding PCI device (Master/Target abort), it receives the READYo# and BTERMo# for the first cycle only. 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 or by a local watchdog timer asserting RESETi#. If the local bus master cannot terminate its cycle with BTERMo#, it should not perform burst cycles when attempting to determine whether a PCI device exists.

3.6.1.9 Write and Invalidate

PCI 9080 can be programmed to perform write and invalidate cycles to the PCI for DMA and Direct Master transfers. PCI 9080 supports Write and Invalidate transfers for cache line sizes of 8 or 16 Lwords. The size is specified in the PCI Cache Line Size Register. If a size other than 8 or 16 is specified, PCI 9080 performs write transfers rather than Write and Invalidate transfers.

3.6.1.9.1 DMA Write and Invalidate

DMA Write and Invalidate transfers are enabled when the write and invalidate enable bit of a DMA controller is set in its Mode Register and the Memory Write and Invalidate enable bit is set in the PCI Command Register.

In Write and Invalidate mode, PCI 9080 waits until the number of Lwords required for the specified cache line size have been read from the local bus before starting the PCI access. This ensures that a complete cache line write can be completed in one PCI bus ownership. If a target disconnects before a cache line is completed, PCI 9080 completes the remainder of that cache line using normal writes before resuming write and invalidate transfers. If a write and invalidate cycle is in progress, PCI 9080 continues to burst if another cache line has been read from the local bus before the cycle completes. Otherwise, PCI 9080 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, PCI 9080 completes DMA transfer using normal writes.

3.6.1.9.2 Direct Master Write and Invalidate

Direct Master Write and Invalidate transfers are enabled when the invalidate enable bit is set in the PCI Base Address (Remap) Register for Direct Master to PCI Memory and the Memory Write and Invalidate enable bit is set in the PCI Command Register (refer to Table 4-11).

In Write and Invalidate mode, if the start address of the Direct Maser transfer is on a cache line boundary, PCI 9080 waits until the number of Lwords required for specified cache line size have been written from the local bus before starting PCI Write and Invalidate access. This ensures that a complete cache line write can be completed in one PCI bus ownership. If the start address is not on a cache line boundary, PCI 9080 starts a normal PCI write access. PCI 9080 terminates a cycle at a cache line boundary if it is performing a normal write or if it is performing a Write and Invalidate cycle and another cache line of data is not available. If an entire cache line is available by the time the PCI 9080 regains use of the PCI bus, PCI 9080 resumes Write and Invalidate cycles. Otherwise, it continues with a normal write. If a target disconnects before a cache line is completed, PCI 9080 completes the remainder of that cache line using normal writes.

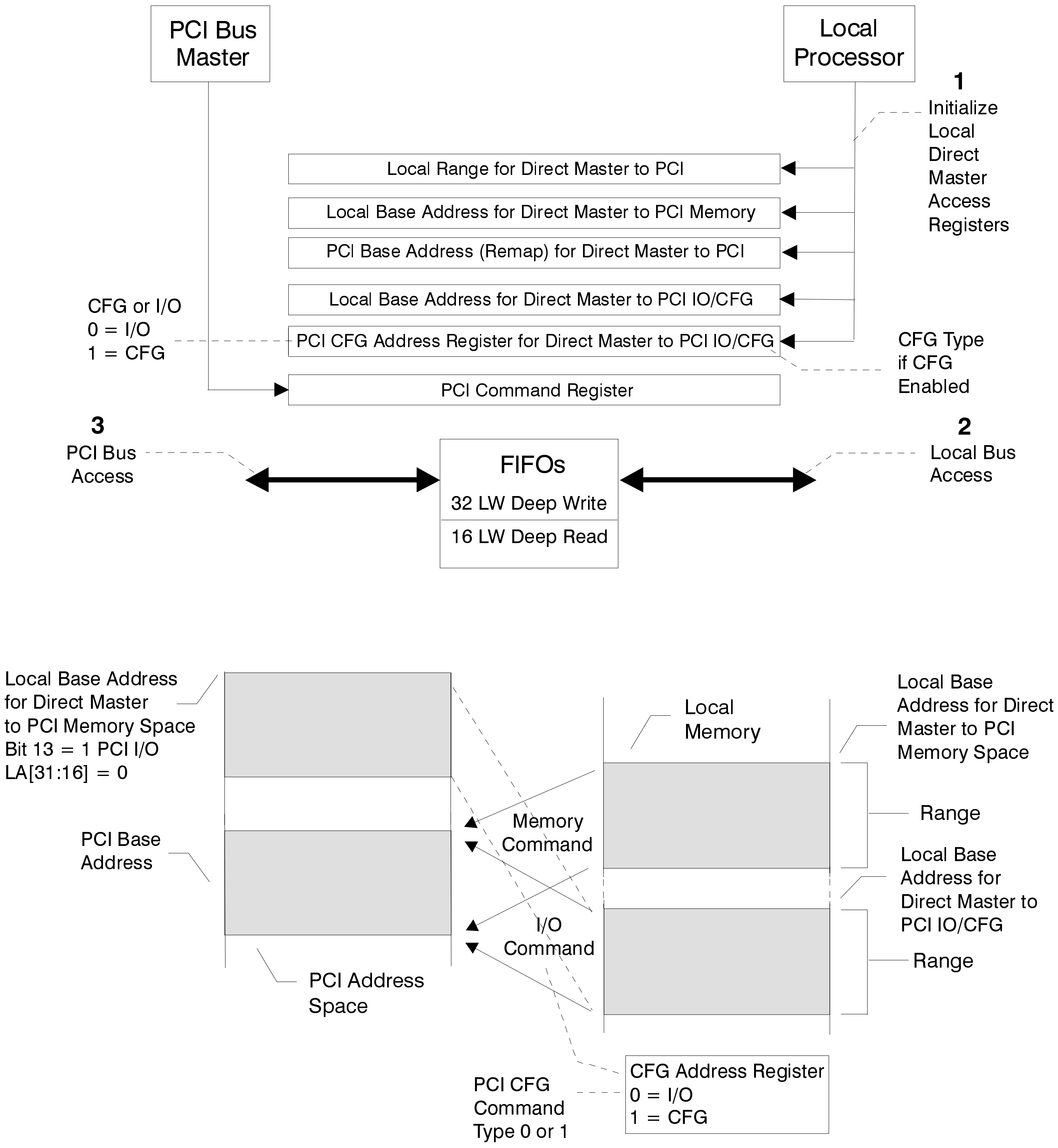

PCI Bus Master
Local Processor
1
Initialize Local Direct Master Access Registers
Local Range for Direct Master to PCI
Local Base Address for Direct Master to PCI Memory
PCI Base Address (Remap) for Direct Master to PCI
Local Base Address for Direct Master to PCI IO/CFG
PCI CFG Address Register for Direct Master to PCI IO/CFG
PCI Command Register
CFG or I/O
0 = I/O
1 = CFG
CFG Type if CFG Enabled
3
PCI Bus Access
2
Local Bus Access
FIFOs
32 LW Deep Write
16 LW Deep Read
Local Base Address for Direct Master to PCI Memory Space
Bit 13 = 1 PCI I/O
LA[31:16] = 0
PCI Base Address
PCI Address Space
Memory Command
I/O Command
Local Memory
Local Base Address for Direct Master to PCI Memory Space
Range
Local Base Address for Direct Master to PCI IO/CFG
Range
PCI CFG Command Type 0 or 1
CFG Address Register
0 = I/O
1 = CFG

Figure 3-8. Local Master Direct Master Access of PCI Bus

3.6.2 Direct Slave Operation (PCI Master to Local Bus Access)

PCI 9080 supports both burst memory mapped transfer accesses and I/O-mapped, single-transfer accesses to the local bus from the PCI bus. PCI Base Address registers are provided to set up the location of the adapter in PCI memory and I/O space. In addition, local mapping registers allow address translation from PCI address space to Local Address Space. There are three spaces available:

- Space 0
- Space 1
- Expansion ROM space

Expansion ROM space is intended to support a bootable ROM device for the host. Each local space can be programmed to operate 8 bit, 16 bit, or 32 bit local bus width. PCI 9080 has an internal wait state generator and external wait state input (READYi#). READYi# can be disabled or enabled with the internal configuration register. The local bus, independent of the PCI bus, can

- Burst as long as data is available (Continuous Burst mode)
- Burst four Lwords at a time
- Perform continuous single cycle, with or without wait state(s)

For single cycle Direct Slave reads, PCI 9080 reads a single local bus Lword or partial Lword. PCI 9080 disconnects after one transfer for all Direct Slave I/O accesses.

For the highest data transfer rate, PCI 9080 supports posted write and can be programmed to prefetch data during PCI Burst Read. The prefetch size, when enabled, can be one to 16 Lwords, or until the PCI stops requesting. PCI 9080 will prefetch if enabled and drop the local bus after the prefetch counter is reached. In a continuous prefetch mode, PCI 9080 prefetches as long as any FIFO space is available and terminates the prefetch when the PCI terminates the request. If read prefetching is disabled, PCI 9080 disconnects after one read transfer.

3.6.2.1 PCI 2.1 Mode

PCI 9080 can be programmed through the Local Arbitration and PCI Mode Register to perform delayed reads, as specified in PCI specification v2.1.

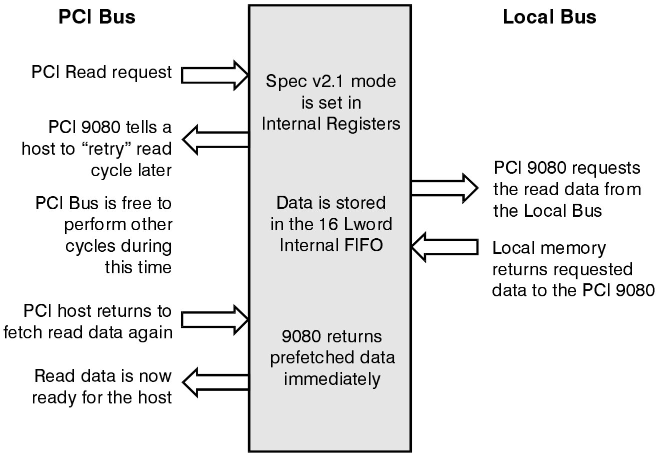

PCI Bus
Local Bus
PCI Read request
PCI 9080 tells a host to "retry" read cycle later
PCI Bus is free to perform other cycles during this time
PCI host returns to fetch read data again
Read data is now ready for the host
Spec v2.1 mode is set in Internal Registers
Data is stored in the 16 Lword Internal FIFO
9080 returns prefetched data immediately
PCI 9080 requests the read data from the Local Bus
Local memory returns requested data to the PCI 9080

Figure 3-9. PCI Specification v2.1 Delayed Reads

In addition to delayed read, PCI 9080 supports the following in PCI specification v2.1 features.

- No write while read is pending (RETRY for reads)
- Write and flush pending read

PCI 9080 also supports Read Ahead mode (refer to Figure 3-10), where prefetched data can be read from the PCI 9080 internal FIFO instead of from the local side. The address must be subsequent to the previous address and must be 32-bit aligned (next address = current address + 4).

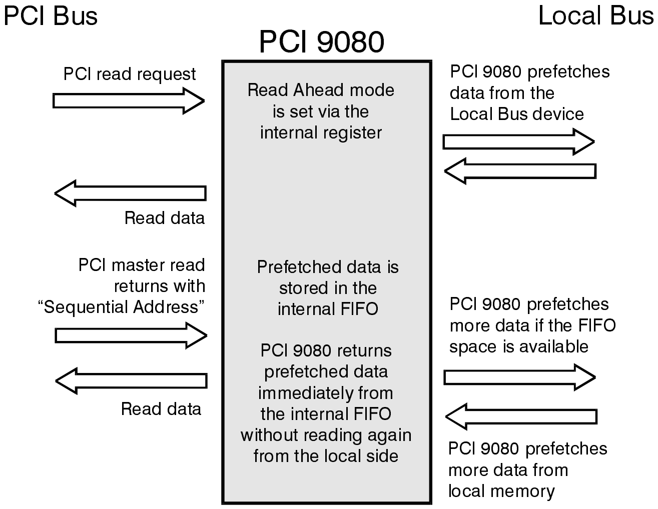

PCI Bus
Local Bus
PCI 9080
PCI read request
Read data
PCI master read returns with "Sequential Address"
Read data
Read Ahead mode is set via the internal register
Prefetched data is stored in the internal FIFO
PCI 9080 returns prefetched data immediately from the internal FIFO without reading again from the local side
PCI 9080 prefetches data from the Local Bus device
PCI 9080 prefetches more data if the FIFO space is available
PCI 9080 prefetches more data from local memory

Figure 3-10. PCI 9080 Read Ahead Mode

PCI 9080 can be programmed to keep the PCI bus by generating a wait state(s), de-asserting TRDY#, if write FIFO becomes full. PCI 9080 can also be programmed to keep the local bus, LHOLD is asserted, if Direct Slave Write FIFO becomes empty or the Direct Slave Read FIFO becomes full. The local bus is dropped in either case when the Local Bus Latency Timer is enabled and expires. (Refer to Figure 3-11 and Figure 3-12.)

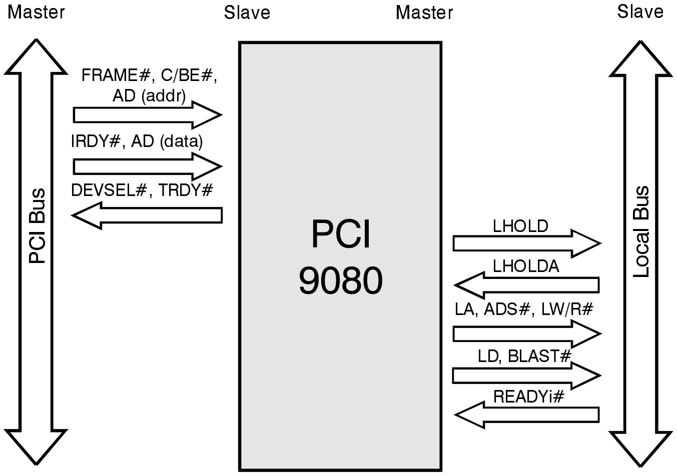


Figure 3-11. Direct Slave Write

For direct slave writes, the PCI (Master) writes data to the local bus (slave). Direct Slave is the “Command from the PCI host,” which has the highest priority. Direct Slave or Direct Master pre-empts DMA; however, Direct Slave does not pre-empt Direct Master (refer to Section 3.6.2.3.1, “Backoff”).

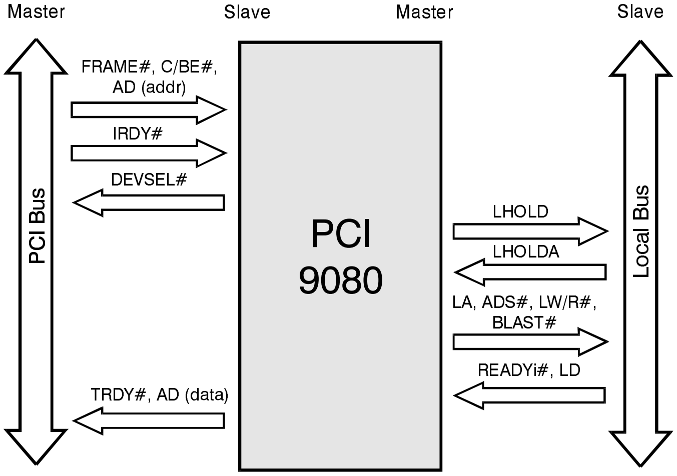


Figure 3-12. Direct Slave Read

For direct slave reads, the PCI (Master) reads data from the local bus (Slave).

PCI 9080 supports on-the-fly Endian conversion for Space 0, Space 1, and expansion ROM space. The local bus can be Big/Little Endian by either using the BIGEND# input pin or the programmable internal register configuration. When BIGEND# is asserted, it overwrites the internal register configuration.

Note: *PCI bus is always Little Endian.*

3.6.2.2 PCI to Local Address Mapping

Note: *Not applicable if I$_2$O 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:

- Local Address Range
- Local Base Address
- PCI Base Address

A fourth register, Bus Region Descriptor Register for PCI to local accesses, defines the local bus characteristics for both regions. (Refer to Figure 3-13.)

3.6.2.2.1 Byte Enables

LBE[3:0]# (pins 139-142) are encoded based on the configured bus width, as follows:

32-Bit Bus—For a 32-bit bus, the four byte enables indicate which of the four bytes are active during a data cycle.

- BE3# Byte Enable 3—LD[31:24]
- BE2# Byte Enable 2—LD[23:16]
- BE1# Byte Enable 1—LD[15:8]
- BE0# Byte Enable 0—LD[7:0]

16-Bit Bus—For a 16-bit bus, BE3#, BE1# and BE0# are encoded to provide BHE#, LA1, and BLE#, respectively.

- BE3# Byte High Enable (BHE#)—LD[15:8]
- BE2# not used
- BE1# Address bit 1 (LA1)
- BE0# Byte Low Enable (BLE#)— LD[7:0]

8-Bit Bus—For an 8-bit bus, BE1# and BE0# are encoded to provide LA1 and LA0, respectively.

- BE3# not used
- BE2# not used
- BE1# Address bit 1 (LA1)
- BE0# Address bit 0 (LA0)

Each PCI to Local Address space is defined as part of reset initialization as described in the next section.

3.6.2.2.2 Local Bus Initialization Software

Range—Specifies which PCI address bits to use for decoding a PCI access to local bus space. Each of the bits corresponds to a PCI address bit. Bit 31 corresponds to Address bit 31. Write a value of 1 to all bits that must be included in decode and a 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.

3.6.2.2.3 PCI Initialization Software

PCI reset software determines how much address space is required by writing a value of all ones (1) to a PCI Base Address register and then reading back the value. PCI 9080 return zeroes in “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 3-13.)

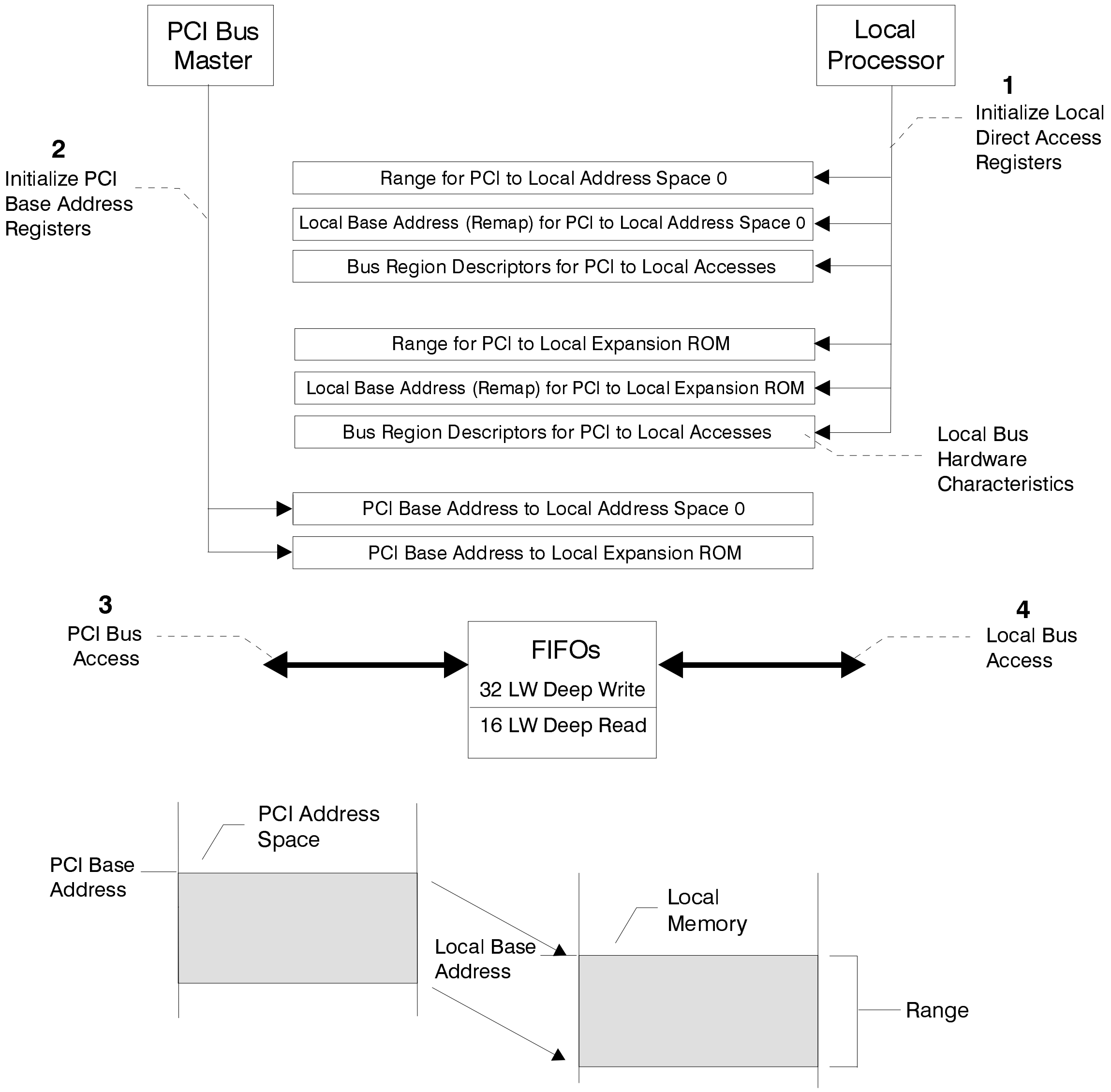

PCI Bus Master
Local Processor
1
Initialize Local Direct Access Registers
2
Initialize PCI Base Address Registers
Range for PCI to Local Address Space 0
Local Base Address (Remap) for PCI to Local Address Space 0
Bus Region Descriptors for PCI to Local Accesses
Range for PCI to Local Expansion ROM
Local Base Address (Remap) for PCI to Local Expansion ROM
Bus Region Descriptors for PCI to Local Accesses
Local Bus Hardware Characteristics
PCI Base Address to Local Address Space 0
PCI Base Address to Local Expansion ROM
3
PCI Bus Access
FIFOs
32 LW Deep Write
16 LW Deep Read
4
Local Bus Access
PCI Address Space
PCI Base Address
Local Memory
Local Base Address
Range

Figure 3-13. Direct Slave Access of Local Bus

Example 2—A 1 MB Local Address Space 12300000h through 123FFFFFh is accessible from the PCI bus at PCI addresses 78900000h through 789FFFFFh.

a. Local initialization software sets the Range and Local Base Address Registers as follows:

- **Range**—FFF00000h (1 MB, decode the upper 12 PCI address bits)
- **Local Base Address (remap)**—123XXXXXh (Local Base Address for PCI to local accesses) (bit 0, the Space Enable bit, must be set to 1 to be recognized by the host)

b. PCI Initialization software writes all ones to the PCI Base Address, then reads it back again.

- PCI 9080 returns a value FFF00000h. The PCI software then writes to PCI Base Address register
- **PCI Base Address**—789XXXXXh (PCI Base Address for access to Local Address space)

For PCI direct access to the local bus, PCI 9080 has a 32 Lword (128 byte) write FIFO and a 16 Lword (64 byte) read FIFO. FIFOs enable the local bus to operate independently of the PCI bus. PCI 9080 can be programmed to return a RETRY response or to throttle TRDY# for any PCI bus transaction attempting to write to the PCI 9080 local bus when the FIFO is full.

For PCI read transactions from the PCI 9080 local bus, PCI 9080 holds off TRDY# while gathering the local bus Lword to be returned. For read accesses mapped to the PCI memory space, PCI 9080 prefetches up to 16 Lwords (has continuous prefetch mode) from the local bus. Unused read data is flushed from the FIFO. For read accesses mapped to the PCI I/O space, PCI 9080 does not prefetch read data. Rather, it breaks each read of the burst cycle into a single address/data cycle on the local bus.

The period of time the PCI 9080 holds off TRDY# can be programmed, Target Retry Timer, in the Local Bus Region Descriptor Register (refer to Table 4-39). PCI 9080 issues a RETRY to the PCI bus transaction master when the programmed time period expires. This occurs when the PCI 9080 cannot gain control of the local bus and return TRDY# within the programmed time period.

3.6.2.3 Deadlock and BREQo

A deadlock situation can occur when a master on the PCI bus wants to access the PCI 9080 local bus at the same time a master on the local PCI 9080 bus wants to access the PCI bus. Two types of deadlock situations can occur:

- **Partial Deadlock**—A master on the local bus is performing a direct bus master access to a PCI bus device other than the PCI bus device that is concurrently trying to access the local bus.
- **Full Deadlock**—A master on the local bus is performing a direct bus master access to the same PCI bus device that is concurrently trying to access the local bus.

This applies only to direct (“pass through”) master and slave accesses through the PCI 9080. Deadlock will not occur in transfers through the PCI 9080 DMA controller or the mailboxes.

For partial deadlock, the PCI access to the local bus times out (the Target Retry Timer, which is programmable through the Local Bus Region Descriptor Register for PCI to local accesses) and the PCI 9080 responds with a PCI RETRY. PCI specification requires that a PCI master release its request for the PCI bus (de-asserts REQ#) for a minimum of two PCI clocks after receiving a RETRY. This allows the PCI bus arbiter to grant the PCI bus to the PCI 9080 so that it can complete its direct master access and free up the local bus. Possible solutions are described below for cases in which the PCI bus arbiter does not function as described (PCI bus architecture dependent), waiting for a time-out is undesirable, or a full deadlock condition exists.

For full deadlock, the only solution is to back off the local master.

3.6.2.3.1 Backoff

PCI 9080 contains a pin (BREQo) that indicates a possible deadlock condition exists. PCI 9080 starts the BREQo timer (programmable through registers) when it detects the following conditions:

- A master on the PCI bus is trying to access memory or an I/O device on the local bus and is not gaining access (for example, LHOLDA not received).
- A master on the local bus is performing a direct bus master read access to the PCI bus or a master on the local bus is performing a direct bus master write access to the PCI bus and the PCI 9080’s direct master write FIFO cannot accept another write cycle.

If timer expires and PCI 9080 has not received LHOLDA, PCI 9080 asserts BREQo. External bus logic can use this as a signal to perform backoff.

A backoff cycle is device/bus architecture dependent. External logic (arbiter) can assert the necessary signals to cause the local master to release the local bus (backoff). After backing off the local master, it can grant the bus to the PCI 9080 (by asserting LHOLDA).

Once BREQo is asserted, READYo# for the current data cycle will never be asserted (the local bus master must perform backoff). When the PCI 9080 detects LHOLDA, It proceeds with the PCI master to local bus access. When this access is complete and the PCI 9080 releases the local bus, the external logic can release backoff and the local master can resume the cycle that was interrupted by the backoff cycle. The write FIFO of the PCI 9080 retains all the data it has acknowledged (i.e. the last data for which READYo# was asserted).

After the backoff condition ends, the local master restarts the last cycle with ADS#. For writes, the data following this ADS# should be the data that was not acknowledged by the PCI 9080 prior to the backoff cycle (for instance, the last data for which there was no READYo# asserted).

If a PCI read cycle is completed when the local bus is backed off, the local bus master receives that data if local master restarts the same last cycle. (Data is not read twice). A new read is performed, if the resumed local bus cycle is not the same as the backed off cycle.

3.6.2.3.2 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 by PCI host software to prevent deadlock. USERO can be set to request that the external arbiter not grant the bus to any local bus master except the PCI 9080. A status output from the local arbiter can be connected to general purpose input USERI to indicate that no local bus master owns the local bus. The input can be read by the PCI host to determine that no local bus master currently owns the local bus. PCI host can then perform a direct slave access. When the host is done, it clears USERO. For devices that support preempt, USERO can be used to preempt the current bus master device. The current local bus master device completes its current cycle and gives up the local bus (de-asserts LHOLD).

3.6.2.3.3 Software Solutions to Deadlock

PCI host software 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.

3.6.2.4 Direct Slave Lock

PCI 9080 supports direct PCI to local bus exclusive accesses (locked atomic operations). A PCI locked operation to 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. PCI 9080 asserts LLOCKo# during the first clock of an atomic operation (address cycle) and de-asserts it a minimum of one clock, following the last bus access for the atomic operation. LLOCKo# is de-asserted after the PCI 9080 detects PCI FRAME# and PCI LOCK# de-asserted at the same time. Refer to the timing diagrams in Section 8, “Timing Diagrams.” Locked operations are enabled or disabled with the Local Bus Region Descriptor Register for PCI to local accesses.

It is the responsibility of external arbitration logic to monitor the LLOCKo# pin and enforce the meaning for an atomic operation. For example, if a local master initiates a locked operation, the local arbiter may choose to not grant use of the local bus to other masters until the locked operation is complete.

3.6.3 Direct Slave Priority

Direct Slave accesses have higher priority than DMA accesses.

Direct Slave accesses preempt DMA transfers. When the PCI 9080 DMA controller owns the local bus, its LHOLD output and LHOLDA input are asserted and its LDSHOLD output is de-asserted. When a Direct Slave access occurs, PCI 9080 gives up the local bus within two Lword transfers by de-asserting LHOLD and floating its local bus outputs. After the PCI 9080 samples its LHOLDA input de-asserted, it requests the local bus for a Direct Slave transfer by asserting LHOLD and LDSHOLD. When the PCI 9080 receives LHOLDA, it drives the bus and performs the Direct Slave transfer. Upon completion of the Direct Slave transfer, PCI 9080 gives up the local bus by de-asserting both LHOLD and LDSHOLD and floating its local bus outputs. After the PCI 9080 samples its LHOLDA de-asserted and its local pause timer is zero, it requests the local bus for a DMA transfer by re-asserting LHOLD. When it receives LHOLDA, it drives the bus and continues with the DMA transfer.

3.7 DMA OPERATION

PCI 9080 supports two independent DMA channels capable of transferring data from the local bus to the PCI bus or from the PCI bus to the local bus. Each channel consists of a DMA controller and a programmable FIFO. Both channels support chaining and non-chaining transfers, Demand Mode DMA, and End of Transfer (EOT) pins. Master mode must be enabled in the PCI Command register.

3.7.1 Non-Chaining Mode DMA

The host processor or the local processor sets the local address, PCI address, transfer count and transfer direction. The host or local processor then sets a control bit to initiate the transfer. PCI 9080 will arbitrate the PCI and local buses and transfer data. Once the transfer is complete, PCI 9080 sets the “channel done” bit to a value of 1 and generates an interrupt to the local processor or the PCI host (programmable). DMA done bit in the internal DMA register can be pooled to indicate the status of DMA transfer.

DMA registers are accessible from the PCI bus and local bus. (Refer to Figure 3-14.)

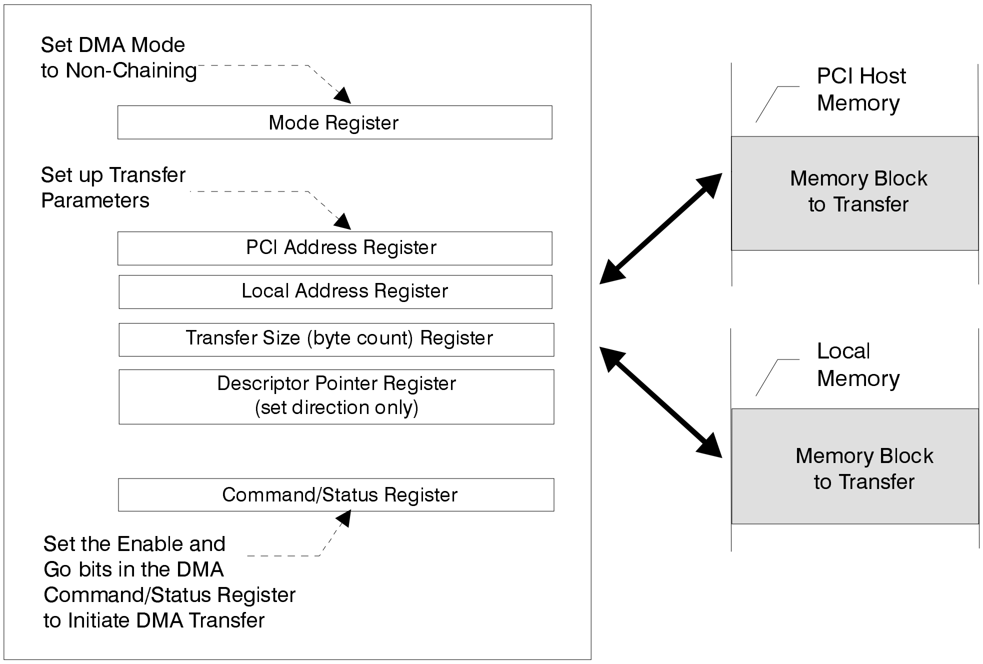


Figure 3-14. Non-Chaining DMA Initialization

The local processor or PCI requires DMA. PCI 9080 is master on both the PCI and local buses. Direct Slave or Direct Master pre-empts DMA.

PCI 9080 releases the PCI bus if one of the following occurs (refer to Figure 3-15):

- FIFO full
- Terminal count reached
- PCI Latency Timer (PCI:0Dh)(LOC:0Dh) expires (refer to Table 4-16[7:0])—normally programmed by the Host PCI BIOS— and PCI GNT# de-asserts
- PCI host asserts STOP
- Direct Master request pending

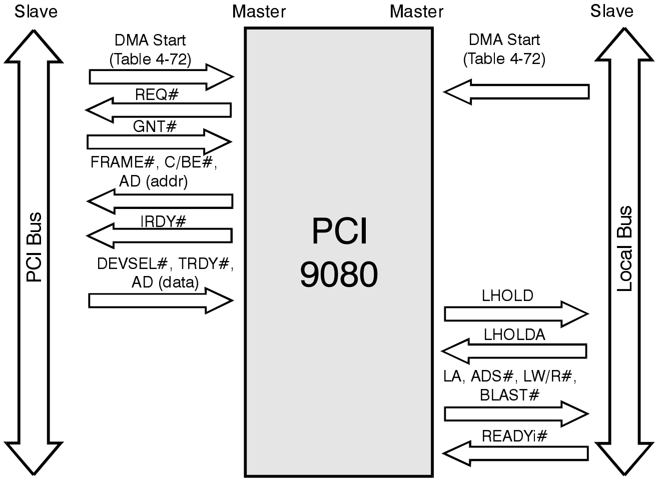


Figure 3-15. DMA, PCI to Local

PCI 9080 releases the local bus if one of the following occurs (refer to Figure 3-16):

- FIFO empty
- Terminal count reached
- Local Bus Latency Timer (PCI:08h or PCI:ACh) (LOC:88h or LOC:12Ch) expires (refer to Table 4-35[7:0])
- BREQ# input asserted
- Direct Slave request pending

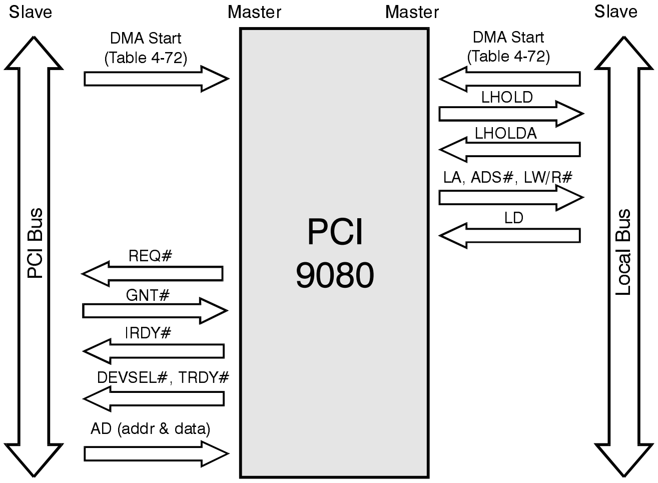


Figure 3-16. DMA, Local to PCI

3.7.2 Chaining Mode DMA

In Chaining mode DMA, the Host Processor or the Local Processor sets up descriptor blocks in local or host memory that are composed of a PCI address, local address, transfer count, transfer direction, and address of the next descriptor block (refer to Figure 3-18). Host or Local Processor then sets up the address of the initial descriptor block in the descriptor pointer register of the PCI 9080 and initiates the transfer by setting a control bit. PCI 9080 loads the first descriptor block and initiates the data transfer. PCI 9080 continues to load descriptor blocks and transfer data until it detects the end of chain bit is set in the next descriptor pointer register. PCI 9080 can be programmed to interrupt the local processor by setting the "Interrupt after Terminal Count" bit or PCI host upon completion of each block transfer and after all block transfers are complete (done) (refer to Figure 3-17). If chaining descriptors are located in local memory, the DMA controller can be programmed to clear the transfer size at the completion of each DMA. (Refer to DMA Clear Count mode, Table 4-62[16] and Table 4-67[16].)

Notes: *In Chaining mode DMA, the descriptor includes PCI address, local address, transfer size and the next descriptor pointer. (PCI:84h, location 104-PCI:90h, location 110D). The descriptor pointer register contains the end of chain bit, direction of transfer, next descriptor address, and next descriptor location.*

DMA descriptor can be on the local memory or PCI memory, or both (first descriptor on local memory, and second descriptor on PCI memory).

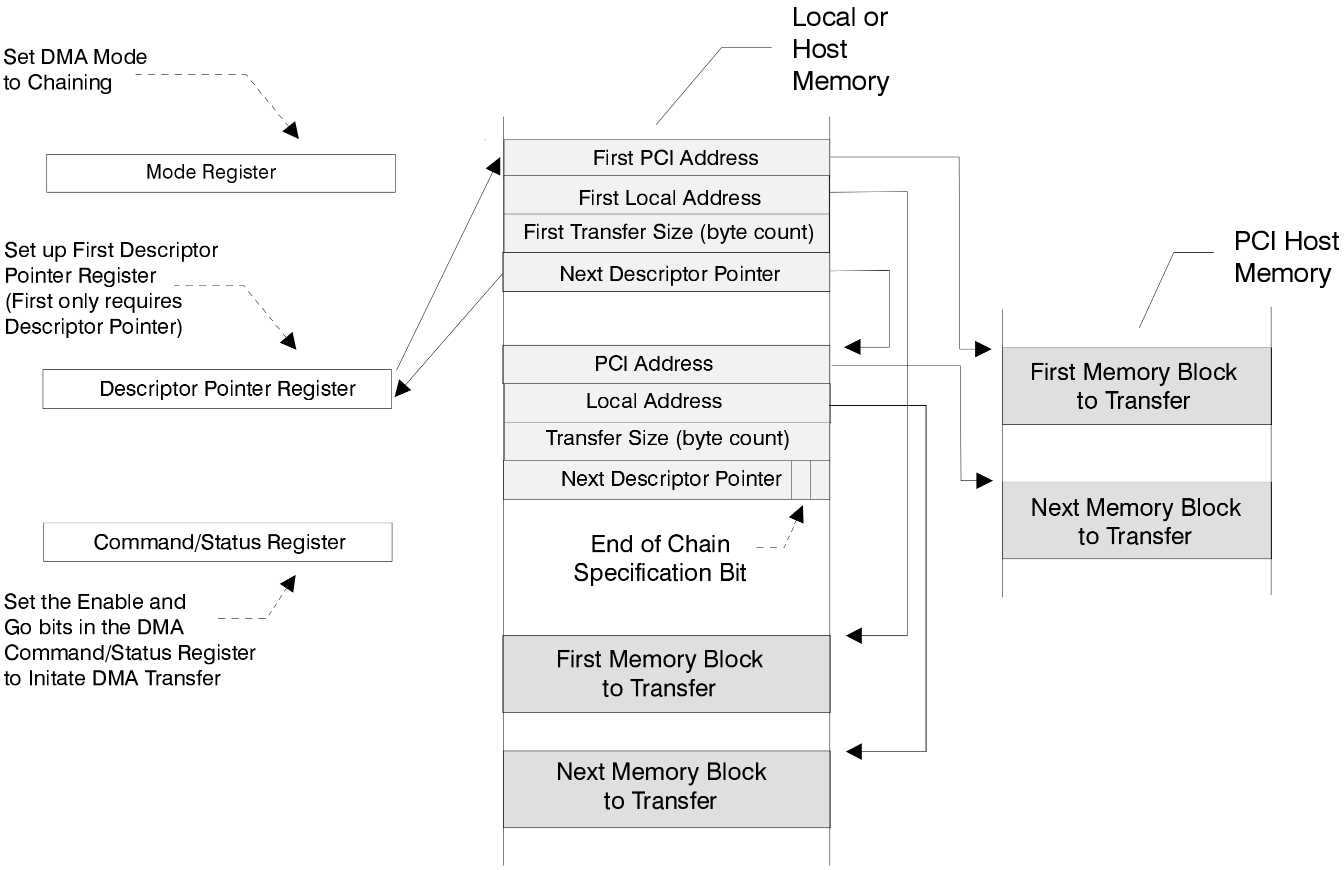

Local or Host Memory
Set DMA Mode to Chaining
Mode Register
Set up First Descriptor Pointer Register (First only requires Descriptor Pointer)
Descriptor Pointer Register
Command/Status Register
Set the Enable and Go bits in the DMA Command/Status Register to Initate DMA Transfer
First PCI Address
First Local Address
First Transfer Size (byte count)
Next Descriptor Pointer
PCI Address
Local Address
Transfer Size (byte count)
Next Descriptor Pointer
End of Chain Specification Bit
First Memory Block to Transfer
Next Memory Block to Transfer
PCI Host Memory
First Memory Block to Transfer
Next Memory Block to Transfer

Figure 3-17. Chaining DMA Initialization

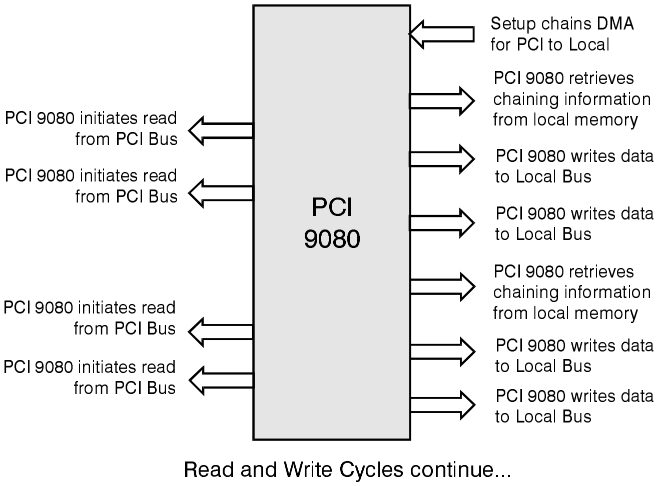


Figure 3-18. Chaining Mode DMA from PCI to Local

3.7.3 DMA Data Transfers

PCI 9080 DMA controller can be programmed to transfer data from the local bus side to the PCI bus side or from the PCI bus side to the local bus side. Refer to Figure 3-19 and Figure 3-20 for a description of the operation.

3.7.3.1 Local to PCI Bus DMA Transfer

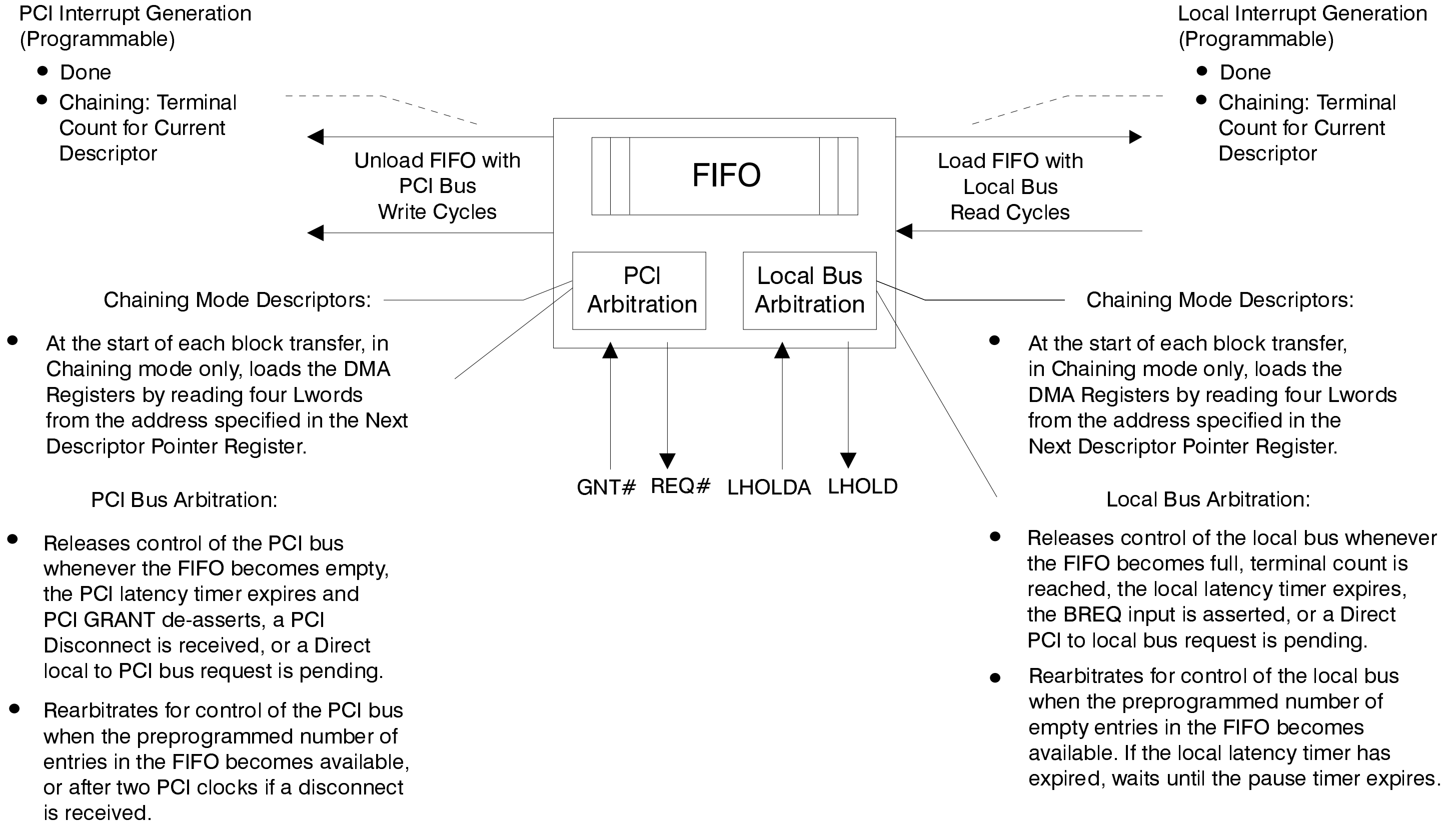
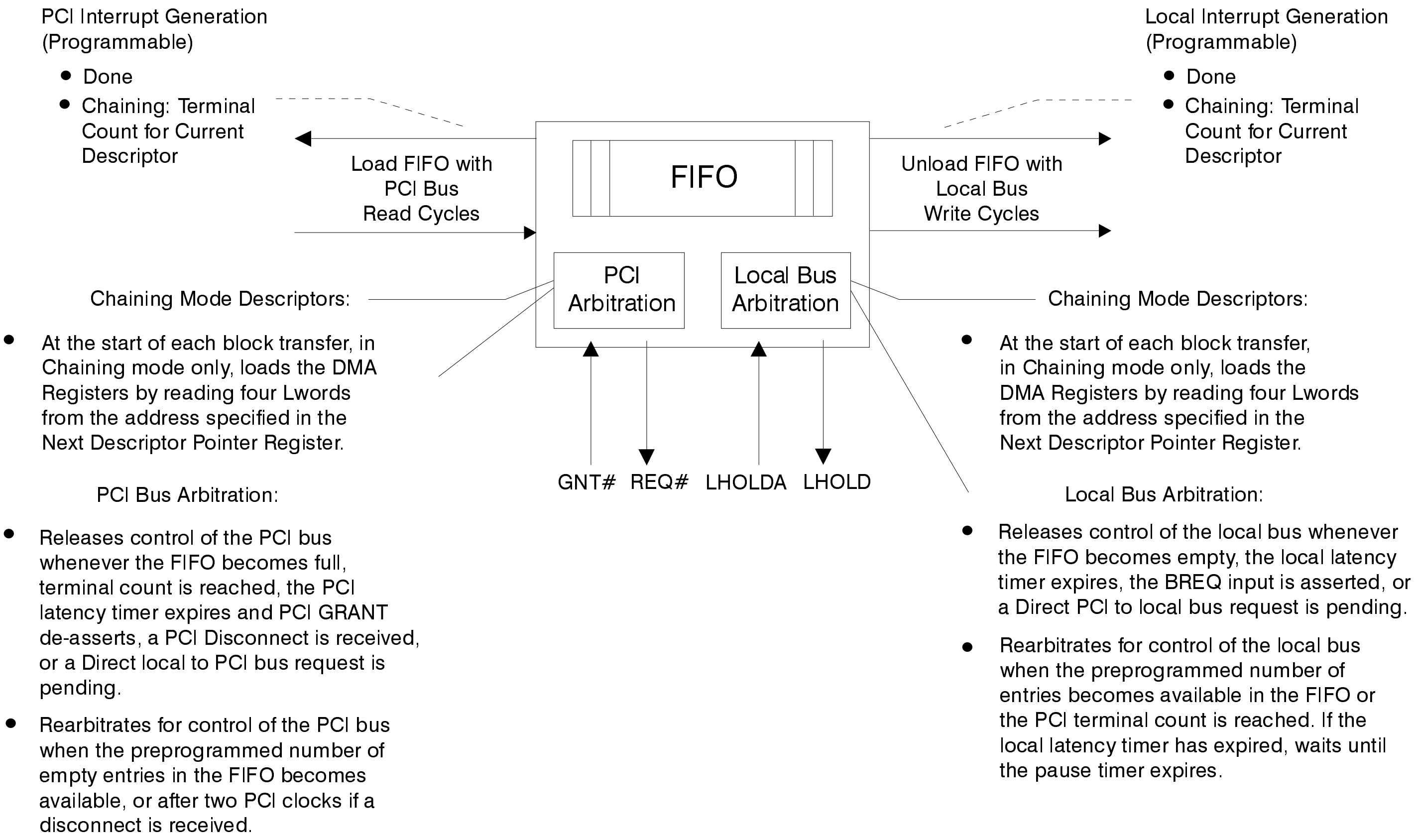

PCI Interrupt Generation
(Programmable)
• Done
• Chaining: Terminal
Count for Current
Descriptor
Local Interrupt Generation
(Programmable)
• Done
• Chaining: Terminal
Count for Current
Descriptor
FIFO
Unload FIFO with
PCI Bus
Write Cycles
Load FIFO with
Local Bus
Read Cycles
PCI
Arbitration
Local Bus
Arbitration
Chaining Mode Descriptors:
• At the start of each block transfer, in Chaining mode only, loads the DMA Registers by reading four Lwords from the address specified in the Next Descriptor Pointer Register.
PCI Bus Arbitration:
• Releases control of the PCI bus whenever the FIFO becomes empty, the PCI latency timer expires and PCI GRANT de-asserts, a PCI Disconnect is received, or a Direct local to PCI bus request is pending.
• Rearbitrates for control of the PCI bus when the preprogrammed number of entries in the FIFO becomes available, or after two PCI clocks if a disconnect is received.
GNT#
REQ#
LHOLDA
LHOLD
Chaining Mode Descriptors:
• At the start of each block transfer, in Chaining mode only, loads the DMA Registers by reading four Lwords from the address specified in the Next Descriptor Pointer Register.
Local Bus Arbitration:
• Releases control of the local bus whenever the FIFO becomes full, terminal count is reached, the local latency timer expires, the BREQ input is asserted, or a Direct PCI to local bus request is pending.
• Rearbitrates for control of the local bus when the preprogrammed number of empty entries in the FIFO becomes available. If the local latency timer has expired, waits until the pause timer expires.

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

3.7.3.2 PCI to Local Bus DMA Transfer


PCI Interrupt Generation (Programmable)
• Done
• Chaining: Terminal Count for Current Descriptor
Load FIFO with PCI Bus Read Cycles
FIFO
Unload FIFO with Local Bus Write Cycles
Local Interrupt Generation (Programmable)
• Done
• Chaining: Terminal Count for Current Descriptor
PCI Arbitration
Local Bus Arbitration
GNT#
REQ#
LHOLDA
LHOLD

Chaining Mode Descriptors:

- At the start of each block transfer, in Chaining mode only, loads the DMA Registers by reading four Lwords from the address specified in the Next Descriptor Pointer Register.

PCI Bus Arbitration:

- Releases control of the PCI bus whenever the FIFO becomes full, terminal count is reached, the PCI latency timer expires and PCI GRANT de-asserts, a PCI Disconnect is received, or a Direct local to PCI bus request is pending.
- Rearbitrates for control of the PCI bus when the preprogrammed number of empty entries in the FIFO becomes available, or after two PCI clocks if a disconnect is received.

Chaining Mode Descriptors:

- At the start of each block transfer, in Chaining mode only, loads the DMA Registers by reading four Lwords from the address specified in the Next Descriptor Pointer Register.

Local Bus Arbitration:

- Releases control of the local bus whenever the FIFO becomes empty, the local latency timer expires, the BREQ input is asserted, or a Direct PCI to local bus request is pending.
- Rearbitrates for control of the local bus when the preprogrammed number of entries becomes available in the FIFO or the PCI terminal count is reached. If the local latency timer has expired, waits until the pause timer expires.

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

3.7.3.3 Unaligned Transfers

For unaligned local to PCI transfers, PCI 9080 reads a partial Lword from the local bus. It continues to read Lwords from the local bus. Lwords are assembled, aligned to the PCI bus address and loaded into the FIFO.

For PCI to local transfers, Lwords are read from the PCI bus and loaded into the FIFO. On the local side, the Lwords are assembled from the FIFO, aligned to the local bus address and written to the local bus. On both the local and PCI buses, the byte enables for writes determine LA[1:0] for the start of a transfer. For the last transfer, the byte enables specify the bytes to be written. All reads are Lwords.

3.7.4 Demand Mode DMA

DMA Mode Register bit 15 (BLAST mode for demand mode DMA), determines the number of Lwords transferred after a DMA controllers DREQ[1:0]# input is de-asserted.

If BLAST# output is not required for the last Lword of the DMA transfer (bit 15 = 1), the DMA controller releases the data bus after it receives an external READYi# or the internal wait state counter decrements to a value of 0 for the current Lword. If DMA controller is currently bursting data, which is not the last data phase for the burst, BLAST# output will not be asserted.

If BLAST# output is required for the last Lword of the DMA transfer (bit 15 = 0), the DMA controller transfers one or two Lwords. If DREQ[1:0]# is de-asserted during the address phase of the first transfer in a PCI 9080 local bus ownership (ADS#, LHOLDA asserted), the DMA controller completes the current Lword. If DREQ[1:0]# is de-asserted during any phase other than the address phase of the first transfer in a PCI 9080 local bus ownership, the DMA controller completes the current Lword, and one additional Lword (this allows BLAST# output to be asserted during the final Lword). If DMA FIFO is full/empty after the data phase in which

DREQ[1:0]# is de-asserted, the second Lword is not transferred.

DREQ[1:0]# controls only the number of Lword transfers. For an 8-bit bus, PCI 9080 gives up the bus after the last byte for the Lword is transferred. For a 16-bit bus, PCI 9080 gives up the bus after the last word for the Lword is transferred.

3.7.5 DMA Priority

DMA Channel 0 priority, DMA Channel 1 priority, or rotating priority can be specified in the DMA Arbitration Register.

3.7.6 DMA Arbitration

PCI 9080 DMA controller releases control of the local bus (de-asserts LHOLD) when one of the following occurs:

- FIFOs are full in a local to PCI transfer
- FIFOs are empty in a PCI to local transfer
- Local Bus Latency Timer expires (if enabled)
- BREQ input is asserted (BREQ can be enabled or disabled, or gated with a latency timer before the PCI 9080 gives up the local bus)
- Direct Slave access is pending
- EOT input is received (if enabled)

DMA controller releases control of the PCI bus when one of the following occurs:

- FIFOs are full or empty
- PCI Latency Timer expires and loses the PCI grant signal
- Target Disconnect response received

DMA controller de-asserts its PCI bus request (REQ#) for a minimum of two PCI clocks.

3.7.6.1 End of Transfer (EOT0# or EOT1#) Input

DMA Mode Register bit 14 (BLAST mode for EOT), determines the number of Lwords transferred after a DMA controller EOT[1:0]# input is asserted.

If BLAST# output is not required for the last Lword of the DMA transfer (bit 14 = 1), the DMA controller releases the data bus and terminates DMA after it receives an external READYi# or the internal wait state counter decrements to a value of 0 for the current Lword. If the DMA controller is currently bursting data, which is not the last data phase for the burst, BLAST# output will not be asserted.

If BLAST# output is required for the last Lword of the DMA transfer (bit 14 = 0), the DMA controller transfers one or two Lwords. If EOT[1:0]# is asserted, the DMA controller completes the current Lword, and one additional Lword (this allows BLAST# output to be asserted during the final Lword). If DMA FIFO is full/empty after the data phase in which EOT[1:0]# is asserted, the second Lword is not transferred.

DMA controller terminates a transfer on an Lword boundary after EOT[1:0]# is asserted. For an 8-bit bus, PCI 9080 terminates after the last byte for the Lword is transferred. For a 16-bit bus, PCI 9080 terminates after the last word for the Lword is transferred.

3.7.6.2 DMA Abort

A DMA transfer can be aborted. The abort process is as follows:

1. DMA Channel must be enabled (Table 4-72[0]=1).
2. DMA Channel must be started (Table 4-72[1]=1).
3. Wait for the Channel Done bit to be set to zero (Table 4-72[4]=0).
4. Disable the DMA Channel (Table 4-72[0] =0).
5. Abort DMA by programming the Channel Abort bit (Table 4-72[2]=1).
6. Wait until the Channel Done bit is set (Table 4-72[4]=1).

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

3.7.6.3 Local Latency and Pause Timers

A Local Bus Latency Timer and Local Bus Pause Timer are programmable with the DMA Arbitration Register. If local latency timer expires, PCI 9080 completes the current Lword transfer and releases LHOLD. After its programmable Pause Timer expires, it reasserts LHOLD. When it receives LHOLDA, it continues the transfer. PCI bus transfer continues until the FIFO is empty for a local to PCI transfer or until it is full for a PCI to local transfer.

3.8 VENDOR AND DEVICE ID REGISTERS

Three Vendor and Device ID registers are supported:

- (LOC:00h), which contains the normal Device and Vendor IDs. This register can be loaded from the serial EEPROM or from local processors.
- (LOC:2Ch), which contains the Subsystem and Subvendor IDs. This register can be loaded from the serial EEPROM or from local processors.
- (LOC:F0h), which contains the hardcoded PLX Vendor and Device IDs.

3.9 DOORBELL REGISTERS

There are two 32 bit doorbell interrupt/status registers in the PCI 9080. One is assigned to the PCI bus interface and the other is assigned to the local bus interface.

The local processor can generate a PCI bus interrupt by writing any number other than all zeroes to the PCI to Local Doorbell Register (refer to Table 4-56).

A PCI host can generate a local bus interrupt by writing any number other than all zeroes to the Local to PCI Doorbell Register (refer to Table 4-57).

3.10 MAILBOX REGISTERS

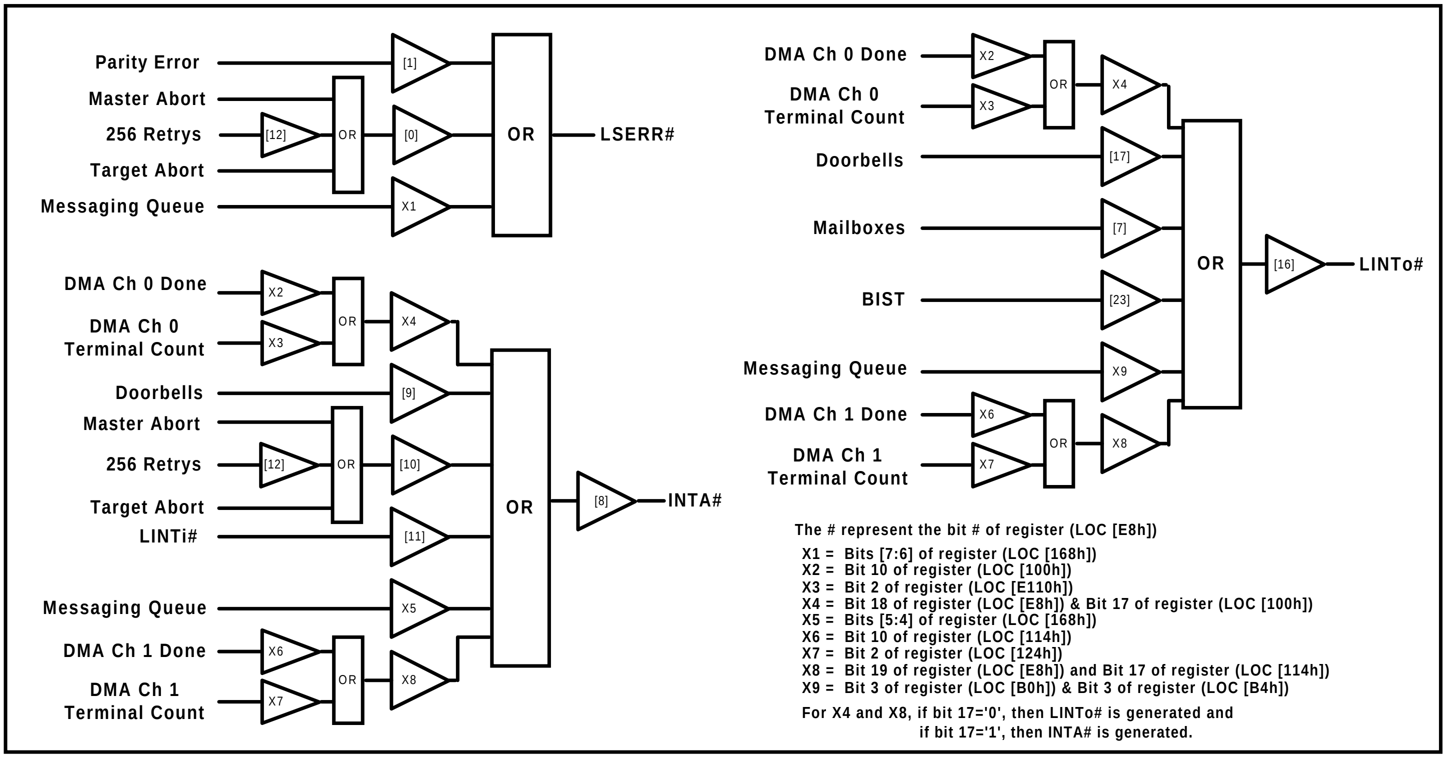
There are eight 32 bit mailbox registers in the PCI 9080 that can be written to and read from both buses. These registers can be used to pass command and status information directly between local and PCI bus devices.

A local interrupt can be generated, if enabled, when the PCI host writes to one of the first four mailbox registers.

3.11 USER INPUT AND OUTPUT

PCI 9080 supports user input and output pins, USERI (pin 31) and USERO (pin 27), respectively. User output data can be logged by writing to bit 16 of (LOC:ECh). User input data can be read from bit 17. (Refer to Table 4-59.)

3.12 INTERRUPTS



The # represent the bit # of register (LOC [E8h])

X1 = Bits [7:6] of register (LOC [168h])
X2 = Bit 10 of register (LOC [100h])
X3 = Bit 2 of register (LOC [E110h])
X4 = Bit 18 of register (LOC [E8h]) & Bit 17 of register (LOC [100h])
X5 = Bits [5:4] of register (LOC [168h])
X6 = Bit 10 of register (LOC [114h])
X7 = Bit 2 of register (LOC [124h])
X8 = Bit 19 of register (LOC [E8h]) and Bit 17 of register (LOC [114h])
X9 = Bit 3 of register (LOC [B0h]) & Bit 3 of register (LOC [B4h])

For X4 and X8, if bit 17='0', then LINTo# is generated and
if bit 17='1', then INTA# is generated.

Figure 3-21. Interrupt and Error Sources

3.12.1 PCI Interrupts (INTA#)

A PCI 9080 PCI Interrupt (INTA#) can be generated by one of the following:

- Local to PCI Doorbell Register
- Local interrupt input
- Master/target abort status condition
- DMA Ch 0/Ch 1 Done
- DMA Ch 0/Ch 1 Terminal Count reached
- Messaging Outbound Post Queue not empty

INTA#, or individual sources of an interrupt, can be enabled or disabled with the PCI 9080 Interrupt Control/Status Register (refer to Table 4-58). This register also provides interrupt status for each interrupt source.

PCI 9080 PCI bus interrupt is level output. An interrupt can be cleared by disabling an interrupt enable bit or clearing the cause(s) of the interrupt.

3.12.1.1 Local Interrupt Input

Asserting local bus input pin LINTi# can generate a PCI bus interrupt. PCI host processor can read PCI 9080 Interrupt Control/Status Register to determine that an interrupt is pending due to the LINTi# pin being asserted.

The interrupt remains asserted as long as the LINTi# pin is asserted and the Local Interrupt input is enabled. Adapter specific action can be taken by the PCI host processor to cause the local bus to release LINTi#.

3.12.1.2 Master/Target Abort Interrupt

PCI 9080 sets the master abort or target abort status bit in the PCI configuration register when it detects a master or target abort. These status bits cause PCI INTA# to be asserted if interrupts are enabled.

The interrupt remains asserted as long as the master or target abort bits remain set in the PCI Status configuration register (refer to Table 4-12) and master/target abort interrupt is enabled. Use a PCI Type 0 configuration access or a local access to clear the master abort and target abort interrupt bits in the PCI Status configuration register.

Bits [26:24] of the Interrupt Control/Status Register (refer to Table 4-58) are latched at the time of a target abort interrupt or master abort interrupt. They provide information as to who was master when an abort occurred. PCI 9080 updates these bits whenever an abort occurs.

3.12.2 Local Interrupts (LINTo#)

A PCI 9080 Local Interrupt (LINTo#) can be generated by one of the following:

- PCI to Local Doorbell/Mailboxes Register access
- PCI BIST interrupt, the DMA done interrupt
- DMA terminal count is reached
- DMA abort interrupt or messaging outbound post queue not empty

LINTo#, or individual sources of an interrupt, can be enabled or disabled with the PCI 9080 Interrupt Control/Status Register (refer to Table 4-58). Interrupt Control/Status Register also provides interrupt status for each source of the interrupt.

PCI 9080 local interrupt is a level output. An interrupt can be cleared by disabling the interrupt enable bit of a source or by clearing the cause of an interrupt.

3.12.2.1 Local to PCI Doorbell Interrupt

A local bus master can generate a PCI bus interrupt by writing to the Local to PCI Doorbell Register (refer to Table 4-57). PCI host processor can then read PCI 9080 Interrupt Control/Status Register (refer to Table 4-58) to determine that a doorbell interrupt is pending. It can then read the PCI 9080 Local to PCI Doorbell Register.

Each bit in the Local to PCI Doorbell Register is individually controlled. Bits in the Doorbell Register can only be set by the local side. From the local side, writing a 1 to any bit position sets that bit and writing a 0 to a bit position has no effect. Bits in the Local to PCI Doorbell Register can only be cleared from the PCI side. From the PCI side, writing a 1 to any bit position clears that bit and writing a 0 to a bit position has no effect.

The interrupt remains asserted as long as any of the Local to PCI Doorbell Register bits are set and PCI Doorbell interrupt is enabled.

To prevent race conditions when the PCI bus is accessing the Doorbell Register (or any configuration register), PCI 9080 automatically de-asserts READYo# to prevent local bus accesses.

3.12.2.2 PCI to Local Doorbell Interrupt

A PCI bus master can generate a local bus interrupt by writing to the PCI to Local Doorbell Register (refer to Table 4-56). Local processor can then read the PCI 9080 Interrupt Control/Status Register (refer to Table 4-58) to determine that a doorbell interrupt is pending. It can then read the PCI 9080 PCI to Local Doorbell Register.

Each bit in the PCI to Local Doorbell Register is individually controlled. Bits in the Doorbell Register can only be set by the PCI side. From the PCI side, writing 1 to any bit position sets that bit and writing 0 to a bit position has no effect. Bits in the PCI to Local Doorbell Register can only be cleared from the local side. From the local side, writing 1 to any bit position clears that bit and writing 0 to a bit position has no effect.

Note: *If local side cannot clear Doorbell Interrupt, do not use the PCI to Local Doorbell Register.*

The interrupt remains asserted as long any of the PCI to Local Doorbell Register bits are set and the Local Doorbell interrupt is enabled.

To prevent race conditions when the local bus is accessing the Doorbell Register (or any configuration register), PCI 9080 automatically issues a RETRY to the PCI bus.

3.12.2.3 Built-In Self Test Interrupt (BIST)

A PCI bus master can generate a local bus interrupt by performing a PCI Type 0 configuration write to a bit in the PCI BIST Register. The local processor can then read the PCI 9080 Interrupt Control/Status Register (refer to Table 4-58) to determine that a BIST interrupt is pending.

The interrupt remains asserted as long as the bit is set and the BIST interrupt is enabled. The local bus then resets the bit when BIST is complete. PCI Host software may fail the device if the bit is not reset after two seconds.

Note: *PCI 9080 does not have internal BIST.*

3.12.2.4 DMA Channel 0/1 Interrupts

A DMA channel can generate a PCI or local bus interrupt when done (transfer complete) or after a transfer is complete for a descriptor in chaining mode. A bit in the DMA mode register determines whether to generate a PCI or local interrupt. The local or PCI processor can then read the PCI 9080 Interrupt Control/Status Register (refer to Table 4-58) to determine whether a DMA channel interrupt is pending.

A Done Status Bit in the Control/Status Register can be used to determine whether the interrupt is

- A done interrupt
- The result of a transfer for a descriptor in a chain that is not yet complete

Mode Register of a channel enables a done interrupt. In chaining mode, a bit in the Next Descriptor Pointer Register of the channel (loaded from local memory) specifies whether to generate an interrupt at the end of the transfer for the current descriptor.

A DMA channel interrupt is cleared by writing a 1 to the Clear Interrupt bit in the DMA Command/Status Register (refer to Table 4-72[3] and Table 4-73[3]).

3.12.3 PCI SERR# (PCI NMI)

PCI 9080 generates an SERR# pulse if parity checking is enabled in the PCI Command Register and it detects an address parity error or the Generate SERR# Bit in the Interrupt Control/Status Register (refer to Table 4-58) is 0 and a 1 is written.

SERR# output can be enabled or disabled with PCI Command Register.

3.12.4 Local LSERR# (Local NMI)

LSERR# interrupt output is asserted if the following occurs:

- PCI Bus Target Abort or Master Abort status bit is set in the PCI Status configuration register
- Parity error status bit is set in the PCI Status configuration register
- Messaging outbound free queue overflows

If parity error checking is enabled in the PCI Command Register, PCI 9080 sets the Master Detected Parity Error Status bit in the PCI Status configuration register (refer to Table 4-12) if it detects one of the following:

- Parity error during a PCI 9080 master read
- PCI bus signal PERR# being asserted during a PCI 9080 master write

PCI 9080 sets a parity error bit in the PCI Status configuration register (refer to Table 4-12) if it detects one of the following:

- Data parity error during a PCI 9080 master read
- Data parity error during a slave write access to the PCI 9080
- Address parity error

The PCI 9080 Interrupt Control/Status Register (refer to Table 4-58) can be used to individually enable or disable LSERR# for an abort or parity error. LSERR# is a level output that remains asserted as long as the Abort or Parity Error Status bits are set.

3.13 $I_2$0 COMPATIBLE MESSAGE UNIT

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 Architecture Specification v1.5* for details.

Figure 3-22 and Figure 3-23 illustrate information about the $I_2$0 architecture.

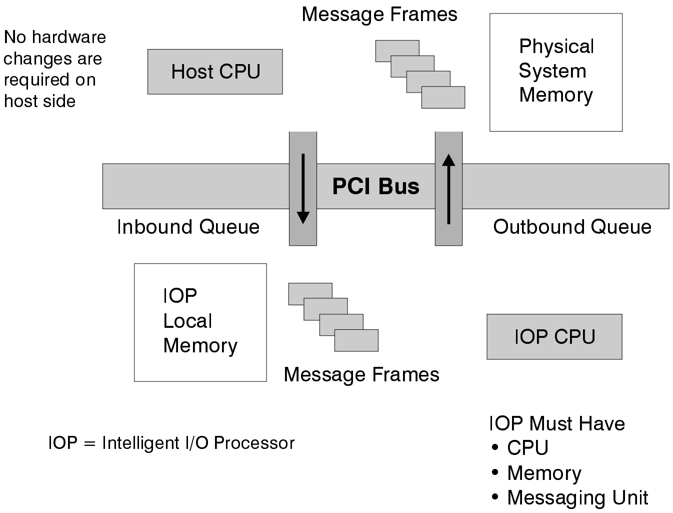


Figure 3-22. I_2O System Architecture

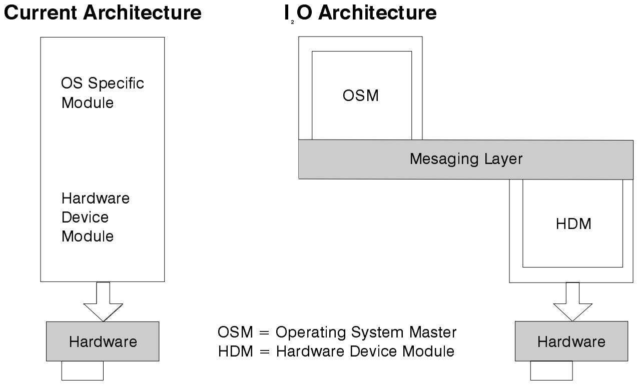


Figure 3-23. I_2O Software Architecture

3.13.1 Inbound Messages

Inbound messages reside in a pool of message frames (minimum 64-byte frames) allocated in shared local bus (IOP) memory. The inbound message queue is comprised of a pair of rotating FIFOs implemented in local memory. Inbound Free List FIFO holds the message frame addresses (MFA) of available message frames in local memory. Inbound Post List FIFO holds the MFA of all currently-posted messages.

The inbound circular FIFOs are accessed by external PCI agents through Inbound Queue Port location in the PCI address space. Inbound Queue Port, when read by an external PCI agent, returns the Inbound Free List FIFO MFA. An external PCI agent places a message frame into the Inbound Post List FIFO by writing its MFA to the inbound queue port location.

3.13.2 Outbound Messages

Outbound messages reside in a pool of message frames (minimum 64-byte frames) allocated in shared PCI bus (Host System) memory. The outbound message queue is comprised of a pair of rotating FIFOs implemented in local memory. Outbound Free List FIFO holds the message frame addresses (MFA) of available message frames in system memory. Outbound Post List FIFO holds the MFA of all currently posted messages.

The outbound circular FIFOs are accessed by external PCI agents through the Outbound Queue Port location in the PCI address space. Outbound Queue Port, when read by an external PCI agent, returns the Outbound Post List FIFO MFA. An 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 themselves must be allocated in local (IOP) memory. The queues base address is contained in Queue Base Address Register (QBAR). Each FIFO entry is a 32 bit data value. Each read and write of the queue must be a single 32-bit access.

The circular FIFOs range in size from 4K entries to 64K 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 Messaging Queue Configuration Register (MQCR) (refer to Table 4-79).

The starting address of each FIFO is based on the Queue base Address and the FIFO Size, as listed in Table 3-7.

Table 3-7. 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)

3.13.3 I_2O Pointer Management

FIFOs always reside in shared local (IOP) memory and are allocated and initialized by the IOP. Before enabling I_2O (Messaging Queue Configuration Register bit 0 set to 1), the local processor must initialize the Inbound Post and Free Head Pointer Registers, the Inbound Post and Free Tail Pointer Registers, the Outbound Post and Free Head Pointer Registers, and the Outbound Post and Free Tail Pointer Registers with the initial offset according to the configured FIFO size. Messaging Unit automatically adds the Queue Base Address to offset in each head and tail pointer register. The software can then enable I_2O. After initialization, the local software should not write to the pointers managed by the MU hardware.

The empty flags are set if the queues are disabled (MQCR bit 0 = 0) and 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 and pointers become not equal.

If an empty flag is cleared and the queues are enabled, the empty flag will only be set if the tail pointer is incremented and 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. Writes to a FIFO occur at the head of the FIFO and reads occur from the tail. The head and tail pointers are incremented by either the local processor or the MU hardware. The unit that writes to the FIFO also maintains the pointer. The pointers are incremented after 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 “chase” each other around and around in the circular FIFO. MU wraps the pointers automatically for the pointers that it maintains. IOP software must wrap the pointers that it maintains. Whenever they are equal, the FIFO is empty. To prevent overflow conditions, I_2O

specifies that the number of message frames allocated should be less than or equal to the number of entries in a FIFO. (Refer to Figure 3-24 for additional information.)

Each inbound MFA is specified by I_2O as the offset from the start of shared local (IOP) memory region 0 to the start of the message frame. Each outbound MFA is specified as the offset from Host memory location 0x00000000h to the start of the message frame in shared Host memory. Since the MFA is an actual address, the message frames need not be contiguous. IOP allocates and initializes inbound message frames in shared IOP memory using any suitable memory allocation technique. Host allocates and initializes outbound message frames in shared Host memory using any suitable memory allocation technique. Message frames are a minimum of 64 bytes in length.

I_2O uses a “push” (write preferred) memory model. That means that the IOP will write messages and data to the shared Host memory, and the Host will write messages and data to shared IOP memory. Software should make use of burst and DMA transfers whenever possible to ensure efficient use of the PCI bus for message passing.

Additional information on message passing implementation may be found in the *I_2O Architecture Specification v1.5*.

3.13.4 Inbound Free List FIFO

The local processor allocates inbound message frames in its shared memory and can place the address of a free (available) message frame into the Inbound Free List FIFO by writing its MFA into the FIFO location pointed to by the Queue Base Register + Inbound Free Head Pointer Register. The local processor must then increment the Inbound Free Head Pointer Register.

A PCI master (Host or another IOP) can obtain the MFA of a free message frame by reading the Inbound Queue Port Address (40h of the first PCI Memory Base Address Register). If FIFO is empty (no free inbound message frames are currently available, head and tail pointers are equal), the MU returns a value of -1 (FFFFFFFFh). If FIFO is not empty (head and tail pointers are not equal), the MU reads the MFA pointed to by the Queue Base Register + Inbound Free Tail Pointer Register, returns its value and increments the Inbound Free Tail Pointer Register. If Inbound Free Queue is not empty, and queue prefetching is enabled (QSR Register bit 3), 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.

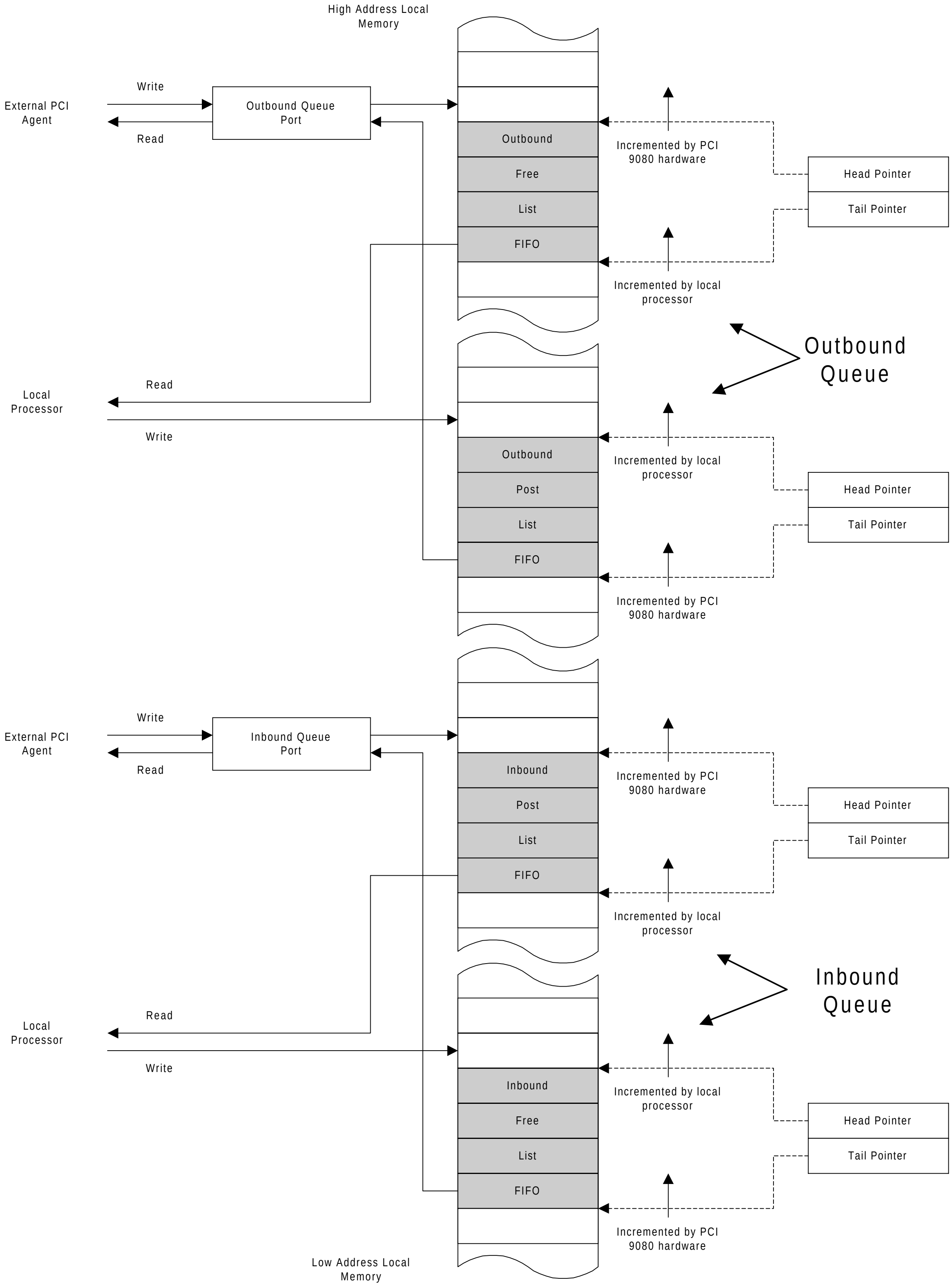


Figure 3-24. Circular FIFO Operation

3.13.5 Inbound Post List FIFO

A PCI master (Host or another IOP) can write a message into an available message frame in shared local (IOP) memory. It can then post that message by writing the message frame address (MFA) to the Inbound Queue Port Address (40h of the first PCI Memory Base Address Register). When the port is written, the MU writes the MFA to the Inbound Post List FIFO location pointed to by the Queue Base Register + FIFO Size + Inbound Post Head Pointer Register. After the MU writes the MFA to the Inbound Post List FIFO, it increments the Inbound Post Head Pointer Register.

Inbound Post Tail Pointer Register points to the Inbound Post List FIFO location which holds the MFA of the oldest posted message. The tail pointer is maintained by the local processor. After a local processor reads the oldest MFA, it can remove the MFA from the Inbound Post List FIFO by incrementing the Inbound Post Tail Pointer Register.

PCI 9080 generates a local Interrupt when the Inbound Post List FIFO is not empty. Inbound Post List FIFO Interrupt bit in the Queue Status/Control Register (QSR) indicates interrupt status. The interrupt clears when the Inbound Post List FIFO is empty. The interrupt can be masked by the Inbound Post List FIFO Interrupt Mask bit (refer to Table 4-89[4]).

To prevent race conditions from the time the PCI write transaction is received until the data is written in local memory and the Inbound Post Head Pointer Register is incremented, any PCI direct slave access to the PCI 9080 is issued a RETRY.

3.13.6 Outbound Post List FIFO

A local master (IOP) can write a message into an available message frame in shared Host memory. It can then post that message by writing the message frame address (MFA) to the Outbound Post List FIFO location pointed to by the Queue Base Register + Outbound Post Head Pointer Register + (2 * FIFO Size). The local processor should then increment the Outbound Post Head Pointer Register.

A PCI master can obtain the MFA of the oldest posted message by reading the Outbound Queue Port Address (44h of the first PCI Memory Base Address Register). If FIFO is empty (no more outbound messages are posted, head and tail pointers are equal), the MU returns a value of -1 (FFFFFFFFh). If Outbound Post List FIFO is not empty (head and tail pointers are not equal), the MU reads the MFA pointed to by the Queue Base Register + (2 * FIFO Size) + outbound Post Tail Pointer Register, returns its value and increments the Outbound Post Tail Pointer Register.

PCI 9080 generates a PCI Interrupt when the Outbound Post Head Pointer Register is not equal to the Outbound Post Tail Pointer Register. Outbound Post List FIFO Interrupt bit of the Outbound Post List FIFO Interrupt Status (OPLFIS) Register indicates interrupt status. When the pointers become equal, both the interrupt and the Outbound Post List FIFO interrupt bit are automatically cleared. The pointers become equal when a PCI master (Host or another IOP) reads enough FIFO entries to empty the FIFO. The interrupt can be masked by the Outbound Post List FIFO Interrupt Mask (OPLFIM) Register).

3.13.7 Outbound Post Queue

To reduce read latency, prefetching from the tail of the queue occurs whenever the queue is not empty and the tail pointer is incremented (queue has been read from), or when the queue is empty and the head pointer is incremented (queue has been written to). When the host CPU reads the Outbound Post Queue, the data is immediately available.

3.13.8 Inbound Free Queue

To reduce read latency, prefetching from the tail of the queue occurs whenever the queue is not empty and the tail pointer is incremented (queue has been read from), or when the queue is empty and the head pointer is incremented (queue has been written to). When the host CPU reads the Inbound Free Queue, the data is immediately available.

3.13.9 Outbound Free List FIFO

A PCI master (Host or another IOP) allocates outbound message frames in its shared memory and can place the address of a free (available) message frame into the Outbound Free List FIFO by writing the message frame address (MFA) to the Outbound Queue Port Address (44h of the first PCI Memory Base Address Register). When the port is written, the MU writes the MFA to the Outbound Free List FIFO location pointed to by the Queue Base Register + (3 * FIFO Size) + Outbound Free Head Pointer Register. After the MU writes the MFA to the Outbound Free List FIFO, it increments the Outbound Free Head Pointer Register.

When the IOP needs a free outbound message frame, it must first check whether any free frames are available. If Outbound Free List FIFO is empty (outbound free head and tail pointers are equal), the IOP must wait for the Host to place additional outbound free message frames in the Outbound Free List FIFO. If Outbound Free List FIFO is not empty (head and tail pointers are not equal), the IOP can obtain the MFA of the oldest free outbound message frame by reading the location pointed to by the Queue Base Register + (3 * FIFO Size) + Outbound Free Tail Pointer Register. After the IOP reads the MFA, it must increment the Outbound Free Tail Pointer Register. To prevent overflow conditions, I_2O specifies that the number of message frames allocated should be less than or equal to the number of entries in a FIFO. MU also checks for overflows of the Outbound Free List FIFO. When the head pointer is incremented and becomes equal to the tail pointer, the Outbound Free List FIFO is full, and the MU generates a local LSERR (NMI) interrupt. The interrupt is recorded in the Queue Status Control (QSR) Register.

From the time that the PCI write transaction is received until the data is written into local memory and the Outbound Free Head Pointer Register is incremented, any PCI direct slave access to the PCI 9080 is issued a RETRY.

Table 3-8. Circular FIFO Summary

FIFO Name	PCI Port	Generate PCI Interrupt?	Generate Local Interrupt	Head Pointer Maintained by	Tail Pointer Maintained by
Inbound Free List FIFO	Inbound Queue Port (Host read)	No	No	Local processor	MU hardware
Inbound Post List FIFO	Inbound Queue Port (Host write)	No	Yes, when Port is written	MU hardware	Local processor
Outbound Post List FIFO	Outbound Queue Port (Host read)	Yes, when FIFO is not empty	No	Local processor	MU hardware
Outbound Free List FIFO	Outbound Queue Port (Host write)	No	Yes, (LSERR) when FIFO full	MU hardware	Local processor

3.13.10 I_20 Enable Sequence

To enable I_2O, the local processor should perform the following:

- Initialize Space 1 address and range
- Initialize all FIFOs and message frame memory
- Set the PCI class code in Register (PCI:09h-0Bh) to be an I_2O device with programming interface 01h
- Set the I_2O enable bit
- Set the Local Init Done bit

Note: *NB# must be pulled up so the PCI 9080 issues retries to all PCI accesses until the Local Init Done bit is set in register (LOC:ECh) (refer to Table 4-59) by the local processor.*

The I_2O enable bit in the Queue Status Register (LOC: 168h; Table 4-89) causes remapping of resources for use in I_2O mode. When this bit is set, all memory mapped configuration registers (such as queue ports 40h and 44h) and Space 1 share PCIBAR0 (PCI:10h, LOC:10h; Table 4-19). PCI accesses to offset 00h-FFh of PCIBAR0 will result in accesses to the internal configuration registers of the PCI 9080. Accesses above offset FFh of PCIBAR0 will result in local space accesses beginning at offset 100h from the Local Space 1 Remap Register (LAS1BA, LOC:174h; Table 4-46). Therefore space located at offset 00h-FFh from LAS1BA is not addressable by way of PCIBAR0.

Programmer's Note: *Because PCI accesses to offset 00h-FFh of PCIBAR0 result in internal configuration accesses, Inbound Free MFAs must be greater than FFh.*

4. REGISTERS

4.1 NEW REGISTER DEFINITIONS SUMMARY

Refer to the descriptions in the following sections for a full explanation.

Table 4-1. New Registers Definitions Summary

PCI Offset	Local Offset	Register	Bits	Description
08h or ACh	88h or 12Ch	MARBR	23	Add PCIREQMODE output.
			28	Read Ahead mode.
18h	98h	LBRD0	15	Single read mode removed.
28h	A8h	DMPBAM	10	Extend almost full flag to five bits (fifth bit not contiguous).
			11	Add CDMPFLIMIT output; do not prefetch past 4K boundary for DM.
			12, 3	Direct master read prefetch size control.
			13	I/O Remap select.
			15:14	Direct master write delay.
30h	B0h	OPLFIS	all	New outbound post list FIFO Interrupt Status Register.
34h	B4h	OPLFIM	all	New outbound post list FIFO Interrupt Mask Register.
40h	N/A	IQP	all	New inbound queue port register.
44h	N/A	OQP	all	New outbound queue port register.
68h	E8h	INTCSR	4	Move DMA0INTSEL output to DMAMODE0. Change to reserved.
			5	Move DMA1INTSEL output to DMAMODE1. Change to reserved.
			3	Mailbox interrupt enable on F, not on 9060.
			31:28	Mailbox interrupts on SD, not on 9060.
80h	100h	DMAMODE0	16	Clear byte count in chaining descriptor.
			17	Add C0_INTSEL output. 0=local int., 1=PCI int.
94h	114h	DMAMODE1	16	Clear byte count in chaining descriptor.
			17	Add C1_INTSEL output. 0=local int., 1=PCI int.
C0h	140h	MQCR	all	New messaging queue configuration register.
C4h	144h	QBAR	all	New queue base address register.
C8h	148h	IFHPR	all	New inbound free head pointer.
CCh	14Ch	IFTPR	all	New inbound free tail pointer.
D0h	150h	IPHPR	all	New inbound post head pointer.
D4h	154h	IPTPR	all	New inbound post tail pointer.
D8h	158h	OFHPR	all	New outbound free head pointer.
DCh	15Ch	OFTPR	all	New outbound free tail pointer.
E0h	160h	OPHPR	all	New outbound post head pointer.
E4h	164h	OPTPR	all	New outbound post tail pointer.
E8h	168h	QSR	all	New I_2O queue status register.
F0h	170h	LAS1RR	all	New Local Address Space 1 Range Register for PCI to local.
F4h	174h	LAS1BA	all	New Local Address Space 1 Local Base Address (Remap).
F8h	178h	LBRD1	all	New Local Address Space 1 Bus Region Descriptor.

4.1.1 Register Differences between PCI 9080 and PCI 9060, PCI 9060ES, and PCI 9060SD

Table 4-2. Register Differences between PCI 9080 and PCI 9060

Register	PCI/Local Offset	Bits	Description
PCIIDR	00/00	31:16	Default changed from PCI 9060 to PCI 9080
PCICR	04/04	4	Memory Write and Invalidate now supported
PCISR	06/06	6	User definable bit added
PCICLSR	0C/0C	7:0	Cache line size is now used for Memory Write and Invalidate
PCIBAR0	10/10	8:6	Register Bank size changed from 128 to 256
PCIBAR1	14/14	8:6	Register Bank size changed from 128 to 256
PCIBAR3	1C/1C	31:0	Base address register for Local Address Space 1
PCISVID	2C/2C	15:0	Subsystem Vendor ID Register
PCISID	2E/2E	15:0	Subsystem ID Register
MARBR	08, AC/88, 12C	31:0	Mode/Arbitration Register now accessible from PCI bus
		21	Local Bus Direct Slave Give up Bus Mode
		22	Direct Slave Lock Enable
		23	PCI Request Mode
		24	PCI Rev 2.1 Mode
		25	PCI Read/No Write Mode
		26	PCI Read with Write Flush Mode
		27	Get the Local Bus Latency Timer with BREQ
		28	PCI Read/No Flush Mode
BIGEND	0C/8C	7:0	Big/Little Endian Descriptor Register
EROMBA	14/94	5	BREQo Timer Resolution control
LBRD0	18/98	1:0	Local bus width now programmable in S mode
		10	Read Prefetch Count Enable
		14:11	Read Prefetch Count
		17:16	Local bus width now programmable in S mode
		25	Extra long serial EEPROM load bit
DMPBAM	28/A8	12, 3	Direct Master Read Prefetch Size Control
		10, 8:5	Programmable Almost Full Flag increased by two bits
		11	Direct Master Prefetch Limit
		13	I/O Remap select
		15:14	Direct Master Write Delay
LAS1RR	F0/170	31:0	Local Address Space 1 Range Register
LAS1BA	F4/174	31:0	Local Address Space 1 Local Base Address Register (Remap)
LBRD1	F8/178	31:0	Local Address Space 1 Bus Region Descriptor Register
MBOX0	40, 78/C0	31:0	MBOX0 moved to PCI address 78 when Messaging Queue is enabled
MBOX1	44, 7C/C4	31:0	MBOX1 moved to PCI address 7C when Messaging Queue is enabled

Table 4-2. Register Differences between PCI 9080 and PCI 9060 (continued)

Register	PCI/Local Offset	Bits	Description
INTCSR	68/E8	3	Mailbox Interrupt Enable
		28	Mailbox 0 Interrupt Status
		29	Mailbox 1 Interrupt Status
		30	Mailbox 2 Interrupt Status
		31	Mailbox 3 Interrupt Status
PCIHIDR	70/F0	31:0	PCI Permanent Configuration ID Register
PCIHREV	74/F4	7:0	PCI Permanent Revision ID Register
DMAMODE0	80/100	13	Write and Invalidate Mode for DMA Channel 0 transfers
		13	DMA Write and Invalidate Mode
		14	DMA EOT[1:0]# (End of Transfer) Input Pin Enable
		15	DMA Stop Data Transfer Mode
		16	DMA Clear Count Mode
		17	DMA Interrupt Select
DMADPR0	90/110	0	DMA Descriptor Location Selector (PCI or Local)
DMAMODE1	94/114	13	DMA Write and Invalidate Mode
		14	DMA EOT[1:0]# (End of Transfer) Input Pin Enable
		15	DMA Stop Data Transfer Mode
		16	DMA Clear Count Mode
		17	DMA Interrupt Select
DMADPR1	A4/124	0	DMA Descriptor Location Selector (PCI or Local)
DMACSR0	A8/128	4	DMA Channel 0 Done
DMACSR1	A9/129	4	DMA Channel 1 Done
DMATHR	B0/130	15:0	Changed thresholds to accommodate 32 word write FIFOs
OPQIS	30/B0	31:0	Outbound Post Queue Interrupt Status Register
OPQIM	34/B4	31:0	Outbound Post Queue Interrupt Mask Register
IQP	40	31:0	Inbound Queue Port
OQP	44	31:0	Outbound Queue Port
MQCR	C0/140	31:0	Messaging Queue Configuration Register
QBAR	C4/144	31:0	Queue Base Address Register
IFHPR	C8/148	31:0	Inbound Free Head Pointer Register
IFTPR	CC/14C	31:0	Inbound Free Tail Pointer Register
IPHPR	D0/150	31:0	Inbound Post Head Pointer Register
IPTPR	D4/154	31:0	Inbound Post Tail Pointer Register
OFHPR	D8/158	31:0	Outbound Free Head Pointer Register
OFTPR	DC/15C	31:0	Outbound Free Tail Pointer Register
OFHPR	E0/160	31:0	Outbound Post Head Pointer Register
OPTPR	E4/164	31:0	Outbound Post Tail Pointer Register
QSR	E8/168	7:0	Queue Status/Control Register

Table 4-3. Register Differences between PCI 9080 and PCI 9060ES

Register	PCI/Local Offset	Bits	Description
PCIIDR	00/00	31:16	Default changed from PCI 906E to PCI 9080
PCISR	06/06	6	User definable bit added
PCICLSR	0C/0C	7:0	Cache line size is now used for Memory Write and Invalidate
PCIBAR0	10/10	8:6	Register Bank size changed from 128 to 256
PCIBAR1	14/14	8:6	Register Bank size changed from 128 to 256
PCIBAR3	1C/1C	31:0	Base address register for Local Address Space 1
PCISVID	2C/2C	15:0	Subsystem Vendor ID Register
PCISID	2E/2E	15:0	Subsystem ID Register
MARBR	08, AC/88, 12C	20:19	DMA Channel Priority
		23	PCI Request Mode
		25	PCI Read/No Write Mode
		26	PCI Read with Write Flush Mode
		27	Get the Local Bus Latency Timer with BREQ
		28	PCI Read/No Flush Mode
BIGEND	0C/8C	5	Direct Slave Big Endian Mode
		6	DMA Channel 1 Big Endian Mode
		7	DMA Channel 0 Big Endian Mode
EROMBA	14/94	5	BREQo Timer Resolution control
LBRD0	18/98	1:0	Local bus width now programmable in S mode
		15	Single Read Access Mode removed
		17:16	Local bus width now programmable in S mode
		25	Extra long serial EEPROM load bit
DMPBAM	28/A8	12, 3	Direct Master Read Prefetch Size Control
		10, 8:5	Programmable Almost Full Flag increased by one bit
		11	Direct Master Prefetch Limit
		13	I/O Remap select
		15:14	Direct Master Write Delay
LAS1RR	F0/170	31:0	Local Address Space 1 Range Register
LAS1BA	F4/174	31:0	Local Address Space 1 Local Base Address Register (Remap)
LBRD1	F8/178	31:0	Local Address Space 1 Bus Region Descriptor Register
MBOX0	40, 78/C0	31:0	MBOX0 moved to PCI address 78 when Messaging Queue is enabled
MBOX1	44, 7C/C4	31:0	MBOX1 moved to PCI address 7C when Messaging Queue is enabled
MBOX4	50/D0	31:0	MBOX4 added
MBOX5	54/D4	31:0	MBOX5 added
MBOX6	58/D8	31:0	MBOX6 added
MBOX7	5C/DC	31:0	MBOX7 added
P2LDBELL	60/E0	31:8	24 more doorbell bits added to PCI to Local Doorbell Register
L2PDBELL	64/E4	31:8	24 more doorbell bits added to Local to PCI Doorbell Register
INTCSR	68/E8	3	Mailbox Interrupt Enable

Table 4-3. Register Differences between PCI 9080 and PCI 9060ES (continued)

Register	PCI/Local Offset	Bits	Description
INTCSR	68/E8	18	DMA Channel 0 interrupt enable
		19	DMA Channel 1 interrupt enable
		21	DMA Channel 0 interrupt status
		22	DMA Channel 1 interrupt status
		25	DMA Channel 0 active during abort
		26	DMA Channel 1 active during abort
		28	Mailbox 0 Interrupt Status
		29	Mailbox 1 Interrupt Status
		30	Mailbox 2 Interrupt Status
		31	Mailbox 3 Interrupt Status
CNTRL	6C/EC	3:0	Read command for DMA
		7:4	Write command for DMA
PCIHREV	74/F4	7:0	PCI Permanent Revision ID Register
DMAMODE0	80/100	31:0	DMA Channel 0 Mode Register
DMAPADR0	84/104	31:0	DMA Channel 0 PCI Address Register
DMALADR0	88/108	31:0	DMA Channel 0 Local Address Register
DMASIZ0	8C/10C	31:0	DMA Channel 0 Size Register
DMADPR0	90/110	31:0	DMA Channel 0 Descriptor Pointer Register
DMAMODE1	94/114	31:0	DMA Channel 1 Mode Register
DMAPADR1	98/108	31:0	DMA Channel 1 PCI address Register
DMALADR1	9C/11C	31:0	DMA Channel 1 Local address Register
DMASIZ1	A0/120	31:0	DMA Channel 1 Size Register
DMADPR1	A4/124	31:0	DMA Channel 1 Descriptor Pointer Register
DMACSR0	A8/128	7:0	DMA Channel 0 Command/Status
DMACSR1	A9/129	7:0	DMA Channel 1 Command/Status
DMATHR	B0/130	31:0	DMA Threshold Register
OPQIS	30/B0	31:0	Outbound Post Queue Interrupt Status Register
OPQIM	34/B4	31:0	Outbound Post Queue Interrupt Mask Register
IQP	40	31:0	Inbound Queue Port
OQP	44	31:0	Outbound Queue Port
MQCR	C0/140	31:0	Messaging Queue Configuration Register
QBAR	C4/144	31:0	Queue Base Address Register
IFHPR	C8/148	31:0	Inbound Free Head Pointer Register
IFTPR	CC/14C	31:0	Inbound Free Tail Pointer Register
IPHPR	D0/150	31:0	Inbound Post Head Pointer Register
IPTPR	D4/154	31:0	Inbound Post Tail Pointer Register
OFHPR	D8/158	31:0	Outbound Free Head Pointer Register
OFTPR	DC/15C	31:0	Outbound Free Tail Pointer Register
OFHPR	E0/160	31:0	Outbound Post Head Pointer Register
OPTPR	E4/164	31:0	Outbound Post Tail Pointer Register
QSR	E8/168	7:0	Queue Status/Control Register

Table 4-4. Register Differences between PCI 9080 and PCI 9060SD

Register	PCI/Local Offset	Bits	Description
PCIIDR	00/00	31:16	Default changed from PCI 906D to PCI 9080
PCISR	06/06	6	User definable bit added
PCIBAR0	10/10	8:6	Register Bank size changed from 128 to 256
PCIBAR1	14/14	8:6	Register Bank size changed from 128 to 256
PCISVID	2C/2C	15:0	Subsystem Vendor ID Register
PCISID	2E/2E	15:0	Subsystem ID Register
MARBR	08, AC/88, 12C	31:0	Mode/Arbitration Register now accessible from PCI bus
		23	PCI Request Mode
		28	PCI Read/No Flush Mode
BIGEND	0C/8C	1	Direct Master Big Endian Mode
		7	DMA Channel 0 Big Endian Mode
EROMBA	14/94	3:0	Direct Slave BREQo Delay Clocks
		4	Local Bus BREQo Enable
		5	BREQo Timer Resolution control
LBRD0	18/98	1:0	Local bus width now programmable in S mode
		15	Single Read Access Mode removed
		17:16	Local bus width now programmable in S mode
DMRR	1C/9C	31:16	Local Range Register for Direct Master to PCI
DMLBAM	20/A0	31:0	Local Bus Base Address Register for Direct Master to PCI Memory
DMLBAI	24/A4	31:0	Local Bus Base Address Register for Direct Master to PCI IO/CFG
DMPBAM	28/A8	31:0	PCI Base Address (Remap) Register for Direct Master to PCI Memory
LAS1RR	F0/170	31:0	Local Address Space 1 Range Register was at 30/B0 in PCI 9060SD
LAS1BA	F4/174	31:0	Local Address Space 1 Local Base Address Register (Remap) was at 34/B4 in PCI 9060SD
LBRD1	F8/178	31:0	Local Address Space 1 Bus Region Descriptor Register was at 38/B8 in PCI 9060SD
LBRD1	F8/178	15	Single Read Access Mode removed
MBOX0	40,78/C0	31:0	MBOX0 moved to PCI address 78 when Messaging Queue is enabled
MBOX1	44, 7C/C4	31:0	MBOX1 moved to PCI address 7C when Messaging Queue is enabled
MBOX4	50/D0	31:0	MBOX4 added
MBOX5	54/D4	31:0	MBOX5 added
MBOX6	58/D8	31:0	MBOX6 added
MBOX7	5C/DC	31:0	MBOX7 added
INTCSR	68/E8	18	DMA Channel 0 interrupt enable
		21	DMA Channel 0 interrupt active
		24	Direct Master active during abort
		25	DMA Channel 0 active during abort
PCIHREV	74/F4	7:0	PCI Permanent Revision ID Register

Table 4-4. Register Differences between PCI 9080 and PCI 9060SD (continued)

Register	PCI/Local Offset	Bits	Description
DMAMODE0	80/100	31:0	DMA Channel 0 Mode Register
DMAPADR0	84/104	31:0	DMA Channel 0 PCI Address Register
DMALADR0	88/108	31:0	DMA Channel 0 Local Address Register
DMASIZ0	8C/10C	31:0	DMA Channel 0 Transfer Size Register
DMADPR0	90/110	31:0	DMA Channel 0 Descriptor Pointer Register
DMACSR0	A8/128	7:0	DMA Channel 0 Command/Status Register
DMATHR	B0/130	15:0	DMA Channel 0 Thresholds
OPQIS	30/B0	31:0	Outbound Post Queue Interrupt Status Register
OPQIM	34/B4	31:0	Outbound Post Queue Interrupt Mask Register
IQP	40	31:0	Inbound Queue Port
OQP	44	31:0	Outbound Queue Port
MQCR	C0/140	31:0	Messaging Queue Configuration Register
QBAR	C4/144	31:0	Queue Base Address Register
IFHPR	C8/148	31:0	Inbound Free Head Pointer Register
IFTPR	CC/14C	31:0	Inbound Free Tail Pointer Register
IPHPR	D0/150	31:0	Inbound Post Head Pointer Register
IPTPR	D4/154	31:0	Inbound Post Tail Pointer Register
OFHPR	D8/158	31:0	Outbound Free Head Pointer Register
OFTPR	DC/15C	31:0	Outbound Free Tail Pointer Register
OFHPR	E0/160	31:0	Outbound Post Head Pointer Register
OPTPR	E4/164	31:0	Outbound Post Tail Pointer Register
QSR	E8/168	7:0	Queue Status/Control Register

4.2 REGISTER ADDRESS MAPPING

4.2.1 PCI Configuration Registers

Table 4-5. PCI Configuration Registers

PCI CFG Register Address	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9080 family and to ensure compatibility with future enhancements, write a zero to all unused bits.				PCI/Local Writable	Serial EEPROM Writable
		31 24	23 16	15 8	7 0		
00h	00h	Device ID		Vendor ID		Local	Y
04h	04h	Status		Command		Y	N
08h	08h	Class Code			Revision ID	Local	Y
0Ch	0Ch	BIST	Header Type	PCI Latency Timer	Cache Line Size	Y [15:0], Local	N
10h	10h	PCI Base Address 0 for Memory Mapped Configuration Registers (PCIBAR0)				Y	N
14h	14h	PCI Base Address 1 for I/O Mapped Configuration Registers (PCIBAR1)				Y	N
18h	18h	PCI Base Address 2 for Local Address Space 0 (PCIBAR2)				Y	N
1Ch	1Ch	PCI Base Address 3 for Local Address Space 1 (PCIBAR3)				Y	N
20h	20h	Unused Base Address (PCIBAR4)				N	N
24h	24h	Unused Base Address (PCIBAR5)				N	N
28h	28h	Cardbus CIS Pointer (Not Supported)				N	N
2Ch	2Ch	Subsystem ID		Subsystem Vendor ID		Local	Y
30h	30h	PCI Base Address for Local Expansion ROM				Y	N
34h	34h	Reserved				N	N
38h	38h	Reserved				N	N
3Ch	3Ch	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	Y [7:0], Local	Y

Note: *Refer to the PCI 2.1 spec for definitions of these registers.*

4.2.2 Local Configuration Registers

Table 4-6. Local Configuration Registers

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9080 family and to ensure compatibility with future enhancements, write a zero to all unused bits. 31 0	PCI/Local Writable	Serial EEPROM Writable
00h	80h	Range for PCI to Local Address Space 0	Y	Y
04h	84h	Local Base Address (Remap) for PCI to Local Address Space 0	Y	Y
08h	88h	Mode/Arbitration Register	Y	Y
0Ch	8Ch	Big/Little Endian Descriptor Register	Y	Y
10h	90h	Range for PCI to Local Expansion ROM	Y	Y
14h	94h	Local Base Address (Remap) for PCI to Local Expansion ROM and BREQo control	Y	Y
18h	98h	Local Bus Region Descriptors (Space 0 and Expansion ROM) for PCI to Local Accesses	Y	Y
1Ch	9Ch	Range for Direct Master to PCI	Y	Y
20h	A0h	Local Base Address for Direct Master to PCI Memory	Y	Y
24h	A4h	Local Base Address for Direct Master to PCI IO/CFG	Y	Y
28h	A8h	PCI Base Address (Remap) for Direct Master to PCI	Y	Y
2Ch	ACh	PCI Configuration Address Register for Direct Master to PCI IO/CFG	Y	Y
F0h	170h	Range for PCI to Local Address Space 1	Y	Y
F4h	174h	Local Base Address (Remap) for PCI to Local Address Space 1	Y	Y
F8h	178h	Local Bus Region Descriptor (Space 1) for PCI to Local Accesses	Y	Y

4.2.3 Runtime Registers

Table 4-7. Runtime Registers

PCI **(Offset from Base Address)**	**Local Access** **(Offset from Chip Select Address)**	**To ensure software compatibility with other versions of the PCI 9080 family and to ensure compatibility with future enhancements, write a zero to all unused bits.** **31**	**0**	**PCI/Local Writable**	**Serial EEPROM Writable**
40h	C0h	Mailbox Register 0 [1]		Y	Y
44h	C4h	Mailbox Register 1 [1]		Y	Y
48h	C8h	Mailbox Register 2		Y	N
4Ch	CCh	Mailbox Register 3		Y	N
50h	D0h	Mailbox Register 4		Y	N
54h	D4h	Mailbox Register 5		Y	N
58h	D8h	Mailbox Register 6		Y	N
5Ch	DCh	Mailbox Register 7		Y	N
60h	E0h	PCI to Local Doorbell Register		Y	N
64h	E4h	Local to PCI Doorbell Register		Y	N
68h	E8h	Interrupt Control / Status		Y	N
6Ch	ECh	Serial EEPROM Control, PCI Command Codes, User I/O Control, Init Control		Y	N
70h	F0h	Device ID	Vendor ID	N	N
74h	F4h	Unused	Revision ID	N	N
78h	C0h	Mailbox Register 0 [1]		Y	N
7Ch	C4h	Mailbox Register 1 [1]		Y	N

Note: *Mailbox registers 0 and 1 are always accessible at addresses 78h/C0h and 7Ch/C4. When the I_2O feature is disabled (bit 0 of QSR register = 0), mailbox registers 0 and 1 are also accessible at PCI addresses 40h and 44h for PCI 9060 compatibility. When the I_2O feature is enabled, the Inbound and Outbound Queue pointers are accessed at addresses 40h and 44h, replacing the mailbox registers in the PCI address space.*

4.2.4 DMA Registers

Table 4-8. DMA Registers

<table>
<tr><th>PCI
(Offset from Base Address)</th><th>Local Access
(Offset from Chip Select Address)</th><th colspan="3">To ensure software compatibility with other versions of the PCI 9080 family and to ensure compatibility with future enhancements, write a zero to all unused bits.
31 0</th><th>PCI/Local Writable</th><th>Serial EEPROM Writable</th></tr>
<tr><td>80h</td><td>100h</td><td colspan="3">DMA Ch 0 Mode</td><td>Y</td><td>N</td></tr>
<tr><td>84h</td><td>104h</td><td colspan="3">DMA Ch 0 PCI Address</td><td>Y</td><td>N</td></tr>
<tr><td>88h</td><td>108h</td><td colspan="3">DMA Ch 0 Local Address</td><td>Y</td><td>N</td></tr>
<tr><td>8Ch</td><td>10Ch</td><td colspan="3">DMA Ch 0 Transfer Byte Count</td><td>Y</td><td>N</td></tr>
<tr><td>90h</td><td>110h</td><td colspan="3">DMA Ch 0 Descriptor Pointer</td><td>Y</td><td>N</td></tr>
<tr><td>94h</td><td>114h</td><td colspan="3">DMA Ch 1 Mode</td><td>Y</td><td>N</td></tr>
<tr><td>98h</td><td>118h</td><td colspan="3">DMA Ch 1 PCI Address</td><td>Y</td><td>N</td></tr>
<tr><td>9Ch</td><td>11Ch</td><td colspan="3">DMA Ch 1 Local Address</td><td>Y</td><td>N</td></tr>
<tr><td>A0h</td><td>120h</td><td colspan="3">DMA Ch 1 Transfer Byte Count</td><td>Y</td><td>N</td></tr>
<tr><td>A4h</td><td>124h</td><td colspan="3">DMA Ch 1 Descriptor Pointer</td><td>Y</td><td>N</td></tr>
<tr><td>A8h</td><td>128h</td><td>Reserved</td><td>DMA Channel 1 Command/Status Register</td><td>DMA Channel 0 Command/Status Register</td><td>Y</td><td>N</td></tr>
<tr><td>ACh</td><td>12Ch</td><td colspan="3">Mode/Arbitration Register</td><td>Y</td><td>N</td></tr>
<tr><td>B0h</td><td>130h</td><td colspan="3">DMA Threshold Register</td><td>Y</td><td>N</td></tr>
</table>

4.2.5 Messaging Queue Registers

Table 4-9. Messaging Queue Registers

PCI (Offset from Base Address)	Local Access (Offset from Chip Select Address)	To ensure software compatibility with other versions of the PCI 9080 family and to ensure compatibility with future enhancements, write a zero to all unused bits. 31 0	PCI/Local Writable	Serial EEPROM Writable
30h	B0h	Outbound Post Queue Interrupt Status	N	N
34h	B4h	Outbound Post Queue Interrupt Mask	Y	N
40h	—	Inbound Queue Port	PCI	N
44h	—	Outbound Queue Port	PCI	N
C0h	140h	Messaging Unit Configuration Register	Y	N
C4h	144h	Queue Base Address Register	Y	N
C8h	148h	Inbound Free Head Pointer Register	Y	N
CCh	14Ch	Inbound Free Tail Pointer Register	Y	N
D0h	150h	Inbound Post Head Pointer Register	Y	N
D4h	154h	Inbound Post Tail Pointer Register	Y	N
D8h	158h	Outbound Free Head Pointer Register	Y	N
DCh	15Ch	Outbound Free Tail Pointer Register	Y	N
E0h	160h	Outbound Post Head Pointer Register	Y	N
E4h	164h	Outbound Post Tail Pointer Register	Y	N
E8h	168h	Queue Status/Control Register	Y	N

Notes: *When I_2O messaging is enabled (bit 0 of QSR register = 1), a PCI Master (Host or another IOP) uses the Inbound Queue Port to read MFAs from the Inbound Free List FIFO and to write MFAs to the Inbound Post List FIFO. It uses the Outbound Queue Port to read MFAs from the Outbound Post List FIFO and to write MFAs to the Outbound Free List FIFO.*

Each Inbound Message Frame Address (MFA) is specified by I_2O as the offset from the PCI Base Address 0 (programmed in register PCIBAR0 at offset 10H) to the start of the message frame. This means that all inbound message frames should reside in PCI Base Address 0 memory space.

Each Outbound Message Frame Address (MFA) is specified by I_2O as the offset from system address 0x00000000h. So the Outbound MFA is the physical 32-bit address of the frame in shared PCI system memory.

Inbound and Outbound Queues may reside in Local Address Space 0 or 1 by programming the QSR register. They need not be in shared memory.

4.3 PCI CONFIGURATION REGISTERS

All registers may be written to or read from in byte, word, or Lword accesses.

4.3.1 (PCIIDR; PCI:00h, LOC:00h) PCI Configuration ID Register

Table 4-10. (PCIIDR; PCI:00h, LOC:00h) PCI Configuration ID Register

Field	Description	Read	Write	Value after Reset
15:0	Vendor ID. Identifies device manufacturer. Defaults to the PCI SIG issued vendor ID of PLX (10B5h) if no serial EEPROM is present and pin NB# (no local bus initialization) is asserted low.	Yes	Local/ Serial EEPROM	10B5h or 0
31:16	Device ID. Identifies the particular device. Defaults to the PLX part number for PCI interface chip (PCI 9080) if no serial EEPROM is present and pin NB# (no local bus initialization) is asserted low.	Yes	Local/ Serial EEPROM	9080h or 0

4.3.2 (PCICR; PCI:04h, LOC:04h) PCI Command Register

Table 4-11. (PCICR; PCI:04h, LOC:04h) PCI Command Register

Field	Description	Read	Write	Value after Reset
0	I/O Space. Value of 1 allows device to respond to I/O space accesses. Value of 0 disables device from responding to I/O space accesses.	Yes	Yes	0
1	Memory Space. Value of 1 allows device to respond to memory space accesses. Value of 0 disables device from responding to memory space accesses.	Yes	Yes	0
2	Master Enable. Value of 1 allows device to behave as a bus master. Value of 0 disables device from generating bus master accesses. This bit must be set for the PCI 9080 to perform Direct Master or DMA cycles.	Yes	Yes	0
3	Special Cycle. (This bit is not supported.)	Yes	No	0
4	Memory Write/Invalidate Enable Bit. Value of 1 enables memory write/invalidate. Value of 0 disables memory write/invalidate. (Refer to the DMA Mode Registers (DM, DMAMODE0, and DMAMODE1) bit 13.) (Refer to Table 4-43[13], Table 4-62[13], and Table 4-67[13], respectively.)	Yes	Yes	0
5	VGA Palette Snoop. (This bit is not supported.)	Yes	No	0
6	Parity Error Response. Value of 0 indicates a parity error is ignored and operation continues. Value of 1 indicates parity checking is enabled.	Yes	Yes	0
7	Wait Cycle Control. Controls whether the device performs address/data stepping. Value of 0 indicates device never does stepping. Value of 1 indicates device always does stepping. ***Note:*** *Hardcoded to 0.*	Yes	No	0
8	SERR# Enable. Value of 1 enables SERR# driver. Value of 0 disables SERR# driver.	Yes	Yes	0
9	Fast Back-to-Back Enable. Indicates type of fast back-to-back transfers Master can perform on bus. Value of 1 indicates fast back-to-back transfers can occur to any agent on bus. Value of 0 indicates fast back-to-back transfers can only occur to same agent as previous cycle.	Yes	No	0
15:10	Reserved.	Yes	No	0

4.3.3 (PCISR; PCI:06h, LOC:06h) PCI Status Register

Table 4-12. (PCISR; PCI:06h, LOC:06h) PCI Status Register

Field	Description	Read	Write	Value after Reset
5:0	Reserved.	Yes	No	0
6	If high, supports User Definable Features. This bit can only be written from the local side. It is read-only from the PCI side.	Yes	Local	0
7	Fast Back-to-Back Capable. When this bit is set to 1, it indicates the adapter can accept fast back-to-back transactions. Value of 0 indicates adapter cannot.	Yes	No	1
8	Master Data Parity Error Detected. This bit is set to 1 when three conditions are met: 1) the PCI 9080 asserted PERR# itself or observed PERR# asserted; 2) the PCI 9080 was the bus master for the operation in which the error occurred; 3) the Parity Error Response bit in the Command Register is set. Writing 1 to this bit clears the bit (0).	Yes	Yes/Clr	0
10:9	DEVSEL Timing. Indicates timing for DEVSEL# assertion. Value of 01 indicates a medium decode. ***Note:*** *Hardcoded to 01.*	Yes	No	01
11	Target Abort. When this bit is set to 1, this bit indicates the PCI 9080 has signaled a target abort. Writing 1 to this bit clears the bit (0).	Yes	Yes/Clr	0
12	Received Target Abort. When set to 1, this bit indicates the PCI 9080 has received a target abort signal. Writing 1 to this bit clears the bit (0).	Yes	Yes/Clr	0
13	Master Abort. When set to 1, this bit indicates the PCI 9080 has generated a master abort signal. Writing 1 to this bit clears the bit (0).	Yes	Yes/Clr	0
14	Signaled System Error. When set to 1, this bit indicates the PCI 9080 has reported a system error on the SERR# signal. Writing 1 to this bit clears the bit (0).	Yes	Yes/Clr	0
15	Detected Parity Error. When set to 1, this bit indicates the PCI 9080 has detected a PCI bus parity error, even if parity error handling is disabled (the Parity Error Response bit in the Command Register is clear). One of three conditions can cause this bit to be set. 1) the PCI 9080 detected a parity error during a PCI address phase; 2) the PCI 9080 detected a data parity error when it was the target of a write; 3) the PCI 9080 detected a data parity error when performing a master read operation. Writing 1 to this bit clears the bit (0).	Yes	Yes/Clr	0

4.3.4 (PCIREV; PCI:08h, LOC:08h) PCI Revision ID Register

Table 4-13. (PCIREV; PCI:08h, LOC:08h) PCI Revision ID Register

Field	Description	Read	Write	Value after Reset
7:0	Revision ID. Silicon revision of PCI 9080.	Yes	Local/ Serial EEPROM	Current Rev #

4.3.5 (PCICCR; PCI:09-0Bh, LOC:09-0Bh) PCI Class Code Register

Table 4-14. (PCICCR; PCI:09-0Bh, LOC:09-0Bh) PCI Class Code Register

Field	Description	Read	Write	Value after Reset
7:0	Register Level Programming Interface. 00h = Queue Ports at 40h and 44h. 01h = Queue Ports at 40h and 44h, and Int Status and Int Mask at 30h and 34h, respectively.	Yes	Local/ Serial EEPROM	00
15:8	Subclass Code. 80h = Other Bridge Device, 00h = I_2O Device.	Yes	Local/ Serial EEPROM	80h
23:16	Base Class Code. 06h = Bridge Device, 0Eh = I_2O controller.	Yes	Local/ Serial EEPROM	06h

4.3.6 (PCICLSR; PCI:0Ch, LOC:0Ch) PCI Cache Line Size Register

Table 4-15. (PCICLSR; PCI:0Ch, LOC:0Ch) PCI Cache Line Size Register

Field	Description	Read	Write	Value after Reset
7:0	System cache line size in units of 32-bit words.	Yes	Yes	0

4.3.7 (PCILTR; PCI:0Dh, LOC:0Dh) PCI Latency Timer Register

Table 4-16. (PCILTR; PCI:0Dh, LOC:0Dh) PCI Latency Timer Register

Field	Description	Read	Write	Value after Reset
7:0	PCI Latency Timer. Units of PCI bus clocks that specify the amount of time the PCI 9080, as a bus master, can burst data on the PCI bus.	Yes	Yes	0

4.3.8 (PCIHTR; PCI:0Eh, LOC:0Eh) PCI Header Type Register

Table 4-17. (PCIHTR; PCI:0Eh, LOC:0Eh) PCI Header Type Register

Field	Description	Read	Write	Value after Reset
6:0	Configuration Layout Type. Specifies the layout of bits 10h through 3Fh in configuration space. Only one encoding 0 is defined. All other encodings are reserved.	Yes	Local	0
7	Header Type. Value of 1 indicates multiple functions. Value of 0 indicates a single function.	Yes	Local	0

4.3.9 (PCIBISTR; PCI:0Fh, LOC:0Fh) PCI Built-In Self Test (BIST) Register

Table 4-18. (PCIBISTR; PCI:0Fh, LOC:0Fh) PCI Built-In Self Test (BIST) Register

Field	Description	Read	Write	Value after Reset
3:0	Value of 0 indicates device passed its test. Nonzero values indicate the device failed. Device specific failure codes can be encoded in the nonzero value.	Yes	Local	0
5:4	Reserved. Device returns 0.	Yes	No	0
6	PCI writes a 1 to invoke BIST. Generates an interrupt to local bus. Local bus resets the bit when BIST is complete. Software should fail device if BIST is not complete after two seconds. Refer to Runtime registers for interrupt control/status.	Yes	Yes	0
7	Returns 1 if the device supports BIST. Returns 0 if the device is not BIST compatible.	Yes	Local	0

4.3.10 (PCIBAR0; PCI:10h, LOC:10h) PCI Base Address Register for Memory Accesses to Local, Runtime, and DMA Registers

Table 4-19. (PCIBAR0; PCI:10h, LOC:10h) PCI Base Address Register for Memory Accesses to Local, Runtime, and DMA Registers

Field	Description	Read	Write	Value after Reset
0	Memory Space Indicator. Value of 0 indicates register maps into memory space. Value of 1 indicates register maps into I/O space. ***Note:*** *Hardcoded to 0.*	Yes	No	0
2:1	Location of Register. Location values: 00—Locate anywhere in 32 bit memory address space 01—Locate below 1 MB memory address space 10—Locate anywhere in 64 bit memory address space 11—Reserved ***Note:*** *Hardcoded to 0.*	Yes	No	0
3	Prefetchable. Value of 1 indicates there are no side effects on reads. This bit has no effect on the operation of the PCI 9080. ***Note:*** *Hardcoded to 0.*	Yes	No	0
7:4	Memory Base Address. Memory base address for access to Local, Runtime, and DMA registers (default is 256 bytes). ***Note:*** *Hardcoded to 0.*	Yes	No	0
31:8	Memory Base Address. Memory base address for access to Local, Runtime, and DMA registers.	Yes	Yes	0

Note: *For I_2O, the Inbound message frame pool must reside in the address space pointed to by PCIBAR0. Message Frame Address (MFA) is defined by I_2O as the offset from this base address to the start of the message frame.*

4.3.11 (PCIBAR1; PCI:14h, LOC:14h) PCI Base Address Register for I/O Accesses to Local, Runtime, and DMA Registers

Table 4-20. (PCIBAR1; PCI:14h, LOC:14h) PCI Base Address Register for I/O Accesses to Local, Runtime, and DMA Registers

Field	Description	Read	Write	Value after Reset
0	Memory Space Indicator. Value of 0 indicates register maps into memory space. Value of 1 indicates register maps into I/O space. ***Note:*** *Hardcoded to 1.*	Yes	No	1
1	Reserved.	Yes	No	0
7:2	I/O Base Address. Base Address for I/O access to Local, Runtime, and DMA registers. (Default is 256 bytes) ***Note:*** *Hardcoded to 0.*	Yes	No	0
31:8	I/O Base Address. Base Address for I/O access to Local, Runtime, and DMA registers.	Yes	Yes	0

4.3.12 (PCIBAR2; PCI:18h, LOC:18h) PCI Base Address Register for Memory Accesses to Local Address Space 0

Table 4-21. (PCIBAR2; PCI:18h, LOC:18h) PCI Base Address Register for Memory Accesses to Local Address Space 0

Field	Description	Read	Write	Value after Reset
0	Memory Space Indicator. Value of 0 indicates register maps into memory space. Value of 1 indicates register maps into I/O space. (Specified in LAS0RR register.)	Yes	No	0
2:1	Location of Register (If memory space). Location values: 00—Locate anywhere in 32 bit memory address space 01—Locate below 1 MB memory address space 10—Locate anywhere in 64 bit memory address space 11—Reserved (Specified in LAS0RR register.) If I/O Space, bit 1 is always 0 and bit 2 is included in the base address.	Yes	Mem: No I/O: bit 1 no, bit 2 yes	0
3	Prefetchable (If memory space). Value of 1 indicates there are no side effects on reads. This bit reflects the value of bit 3 in the LAS0RR register and provides only status to the system. This bit has no effect on the operation of the PCI 9080. Prefetching features of this address space are controlled by the associated Bus Region Descriptor Register. (Specified in LAS0RR register.) If I/O Space, bit 3 is included in the base address.	Yes	Mem: No I/O: Yes	0
31:4	Memory Base Address. Memory base address for access to Local Address Space 0.	Yes	Yes	0

Note: *PCIBAR2 can be enabled or disabled by setting or clearing bit 0 in the LAS0BA register.*

4.3.13 (PCIBAR3; PCI:1Ch, LOC:1Ch) PCI Base Address Register for Memory Accesses to Local Address Space 1

Table 4-22. (PCIBAR3; PCI:1Ch, LOC:1Ch) PCI Base Address Register for Memory Accesses to Local Address Space 1

Field	Description	Read	Write	Value after Reset
0	Memory Space Indicator. Value of 0 indicates register maps into memory space. Value of 1 indicates register maps into I/O space. (Specified in LAS1RR register.)	Yes	No	0
2:1	Location of register. Location values: 00—Locate anywhere in 32 bit memory address space 01—Locate below 1 MB memory address space 10—Locate anywhere in 64 bit memory address space 11—Reserved (Specified in LAS1RR register.) If I/O Space, bit 1 is always 0 and bit 2 is included in the base address.	Yes	Mem: No I/O: bit 1 no, bit 2 yes	0
3	Prefetchable (If memory space). Value of 1 indicates there are no side effects on reads. This bit reflects the value of bit 3 in the LAS1RR register and only provides status to the system. This bit has no effect on the operation of the PCI 9080. Prefetching features of this address space are controlled by the associated Bus Region Descriptor Register. (Specified in LAS1RR register.) If I/O Space, bit 3 is included in the base address.	Yes	Mem: No I/O: Yes	0
31:4	Memory Base Address. Memory base address for access to Local Address Space 1.	Yes	Yes	0

Note: *PCIBAR3 can be enabled or disabled by setting or clearing bit 0 in the LAS1BA register. If QSR bit 0 is set, PCIBAR3 returns 0.*

4.3.14 (PCIBAR4; PCI:20h, LOC:20h) PCI Base Address Register

Table 4-23. (PCIBAR4; PCI:20h, LOC:20h) PCI Base Address Register

Field	Description	Read	Write	Value after Reset
31:0	Reserved.	Yes	No	0

4.3.15 (PCIBAR5; PCI:24h, LOC:24h) PCI Base Address Register

Table 4-24. (PCIBAR5; PCI:24h, LOC:24h) PCI Base Address Register

Field	Description	Read	Write	Value after Reset
31:0	Reserved.	Yes	No	0

4.3.16 (PCICIS; PCI:28h, LOC:28h) PCI Cardbus CIS Pointer Register

Table 4-25. (PCICIS; PCI:28h, LOC:28h) PCI Cardbus CIS Pointer Register

Field	Description	Read	Write	Value after Reset
31:0	Cardbus Information Structure Pointer for PCMCIA. (Not supported.)	Yes	No	0

4.3.17 (PCISVID; PCI:2Ch, LOC:2Ch) PCI Subsystem Vendor ID Register

Table 4-26. (PCISVID; PCI:2Ch, LOC:2Ch) PCI Subsystem Vendor ID Register

Field	Description	Read	Write	Value after Reset
15:0	Subsystem Vendor ID (unique add-in board Vendor ID).	Yes	Local/ Serial EEPROM	10B5

4.3.18 (PCISID; PCI:2Eh, LOC:2Eh) PCI Subsystem ID Register

Table 4-27. (PCISID; PCI:2Eh, LOC:2Eh) PCI Subsystem ID Register

Field	Description	Read	Write	Value after Reset
15:0	Subsystem ID (unique add-in board Device ID).	Yes	Local/ Serial EEPROM	9080h

4.3.19 (PCIERBAR; PCI:30h, LOC:30h) PCI Expansion ROM Base Register

Table 4-28. (PCIERBAR; PCI:30h, LOC:30h) PCI Expansion ROM Base Register

Field	Description	Read	Write	Value after Reset
0	Address Decode Enable. Value of 1 indicates device accepts accesses to the expansion ROM address. Value of 0 indicates device does not accept accesses to expansion ROM space. Should be set to 1 by PCI host if expansion ROM is present.	Yes	Yes	0
10:1	Reserved.	Yes	No	0
31:11	Expansion ROM Base Address (upper 21 bits).	Yes	Yes	0

4.3.20 (PCIILR; PCI:3Ch, LOC:3Ch) PCI Interrupt Line Register

Table 4-29. (PCIILR; PCI:3Ch, LOC:3Ch) PCI Interrupt Line Register

Field	Description	Read	Write	Value after Reset
7:0	Interrupt Line Routing Value. Indicates which input of the system interrupt controller(s) to which the interrupt line of the device is connected.	Yes	Yes	0

4.3.21 (PCIIPR; PCI:3Dh, LOC:3Dh) PCI Interrupt Pin Register

Table 4-30. (PCIIPR; PCI:3Dh, LOC:3Dh) PCI Interrupt Pin Register

Field	Description	Read	Write	Value after Reset
7:0	Interrupt Pin Register. Indicates which interrupt pin the device uses. The following values are decoded: 0 = No Interrupt Pin 1 = INTA# 2 = INTB# 3 = INTC# 4 = INTD# ***Note:*** *PCI 9080 supports only one PCI interrupt pin (INTA#).*	Yes	Local/ Serial EEPROM	1

4.3.22 (PCIMGR; PCI:3Eh, LOC:3Eh) PCI Min_Gnt Register

Table 4-31. (PCIMGR; PCI:3Eh, LOC:3Eh) PCI Min_Gnt Register

Field	Description	Read	Write	Value after Reset
7:0	Min_Gnt. Specifies how long a burst period the device needs, assuming a clock rate of 33 MHz. Value is multiple of 1/4 µsec increments.	Yes	Local/ Serial EEPROM	0

4.3.23 (PCIMLR; PCI:3Fh, LOC:3Fh) PCI Max_Lat Register

Table 4-32. (PCIMLR; PCI:3Fh, LOC:3Fh) PCI Max_Lat Register

Field	Description	Read	Write	Value after Reset
7:0	Max_Lat. Specifies how often the device must gain access to the PCI bus. Value is multiple of 1/4 µsec increments.	Yes	Local/ Serial EEPROM	0

4.4 LOCAL CONFIGURATION REGISTERS

4.4.1 (LAS0RR; PCI:00h, LOC:80h) Local Address Space 0 Range Register for PCI to Local Bus

Table 4-33. (LAS0RR; PCI:00h, LOC:80h) Local Address Space 0 Range Register for PCI to Local Bus

Field	Description	Read	Write	Value after Reset
0	Memory Space Indicator. Value of 0 indicates Local address space 0 maps into PCI memory space. Value of 1 indicates address space 0 maps into PCI I/O space.	Yes	Yes	0
2:1	If mapped into memory space, encoding is as follows: 2/1 Meaning 0 0 Locate anywhere in 32 bit PCI address space 0 1 Locate below 1 MB in PCI address space 1 0 Locate anywhere in 64 bit PCI address space 1 1 Reserved If mapped into I/O space, bit 1 must be set to 0. Bit 2 is included with bits [31:3] to indicate decoding range.	Yes	Yes	0
3	If mapped into memory space, a value of 1 indicates reads are prefetchable (bit has no effect on the operation of the PCI 9080, but is used for system status). If mapped into I/O space, bit is included with bits [31:2] to indicate decoding range.	Yes	Yes	0
31:4	Specifies which PCI address bits to use for decoding a PCI access to local bus space 0. Each bit corresponds to a PCI address bit. Bit 31 corresponds to Address bit 31. Write a value of 1 to all bits to be included in decode and a 0 to all others (used in conjunction with PCI Configuration register 18h). Default is 1 MB.	Yes	Yes	FFF0000h

Notes: *Range (**not** Range register) must be power of 2. “Range register value” is the inverse of range.*

User should limit all I/O spaces to 256 bytes per PCI v2.1 spec.

4.4.2 (LAS0BA; PCI:04h, LOC:84h) Local Address Space 0 Local Base Address (Remap) Register

Table 4-34. (LAS0BA; PCI:04h, LOC:84h) Local Address Space 0 Local Base Address (Remap) Register

Field	Description	Read	Write	Value after Reset
0	Space 0 Enable. Value of 1 enables decoding of PCI addresses for Direct Slave access to local space 0. Value of 0 disables decoding. If this bit is set to 0, the PCI BIOS may not allocate (assign) the base address for Space 0. ***Note:*** *Must be set to 1 for any Direct Slave access to Space 0.*	Yes	Yes	0
1	Reserved.	Yes	No	0
3:2	If local space 0 is mapped into memory space, bits are not used. If mapped into I/O space, bit is included with bits [31:4] for remapping.	Yes	Yes	0
31:4	Remap of PCI Address to Local Address Space 0 into a Local Address Space. The bits in this register remap (replace) the PCI Address bits used in decode as the Local Address bits. ***Note:*** *Remap Address value must be multiple of Range (**not** the Range register).*	Yes	Yes	0

4.4.3 (MARBR; PCI:08h or ACh, LOC:88h or 12Ch) Mode/Arbitration Register

Table 4-35. (MARBR; PCI:08h or ACh, LOC:88h or 12Ch) Mode/Arbitration Register

Field	Description	Read	Write	Value after Reset
7:0	Local Bus Latency Timer. Number of local bus clock cycles before negating HOLD and releasing the local bus. This timer is also used with bit 27 to delay BREQ input to give up the local bus only when this timer expires.	Yes	Yes	00
15:8	Local Bus Pause Timer. Number of local bus clock cycles before reasserting HOLD after releasing the local bus. ***Note:*** *Applicable only to DMA operation.*	Yes	Yes	00
16	Local Bus Latency Timer Enable. Value of 1 enables latency timer.	Yes	Yes	0
17	Local Bus Pause Timer Enable. Value of 1 enables pause timer.	Yes	Yes	0
18	Local Bus BREQ Enable. Value of 1 enables local bus BREQ input. When the BREQ input is active, PCI 9080 negates HOLD and releases the local bus.	Yes	Yes	0
20:19	DMA Channel Priority. Value of 00 indicates a rotational priority scheme. Value of 01 indicates Channel 0 has priority. Value of 10 indicates Channel 1 has priority. Value of 11 is reserved.	Yes	Yes	0
21	Local Bus Direct Slave Give up Bus Mode. When set to 1, PCI 9080 negates HOLD and releases the local bus when the Direct Slave write FIFO becomes empty during a Direct Slave write or when the Direct Slave read FIFO becomes full during a Direct Slave read.	Yes	Yes	1
22	Direct Slave LLOCKo# Enable. Value of 1 enables PCI Direct Slave locked sequences. Value of 0 disables Direct Slave locked sequences.	Yes	Yes	0
23	PCI Request Mode. Value of 1 causes PCI 9080 to negate REQ when it asserts FRAME during a master cycle. Value of 0 causes PCI 9080 to leave REQ asserted for the entire bus master cycle.	Yes	Yes	0
24	PCI Rev 2.1 Mode. When set to 1, PCI 9080 operates in Delayed Transaction mode for Direct Slave Reads. PCI 9080 issues a RETRY and prefetches the read data.	Yes	Yes	0
25	PCI Read No Write Mode. Value of 1 forces a retry on writes if read is pending. Value of 0 allows writes to occur while read is pending.	Yes	Yes	0
26	PCI Read with Write Flush Mode. Value of 1 submits a request to flush a pending read cycle if a write cycle is detected. Value of 0 submits a request to not effect pending reads when a write cycle occurs (PCI v2.1 compatible).	Yes	Yes	0
27	Gate the Local Bus Latency Timer with BREQ. If this bit is set to 0, PCI 9080 gives up the local bus during Direct Slave or DMA transfer after the current cycle (if enabled and BREQ is sampled). If this bit is set to 1, PCI 9080 gives up the local bus only (if BREQ is sampled) and the Local Bus Latency Timer is enabled and expired during Direct Slave or DMA transfer.	Yes	Yes	0
28	PCI Read No Flush Mode. Value of 1 submits request to not flush the read FIFO if PCI read cycle completes (Read Ahead mode). Value of 0 submits request to flush read FIFO if PCI read cycle completes.	Yes	Yes	0
29	If set to 0, reads from the PCI Configuration Register address 00h and returns the Device ID and Vendor ID. If set to 1, reads from the PCI Configuration Register address 00h and returns the Subsystem ID and Subsystem Vendor ID.	Yes	Yes	0
31:30	Reserved.	Yes	No	0

4.4.4 (BIGEND; PCI:0Ch, LOC:8Ch) Big/Little Endian Descriptor Register

Table 4-36. (BIGEND; PCI:0Ch, LOC:8Ch) Big/Little Endian Descriptor Register

Field	Description	Read	Write	Value after Reset
0	Configuration Register Big Endian Mode. Value of 1 specifies use of Big Endian data ordering for local accesses to the configuration registers. Value of 0 specifies Little Endian ordering. Big Endian mode can be specified for configuration register accesses by asserting the BIGEND# pin during the address phase of the access.	Yes	Yes	0
1	Direct Master Big Endian Mode. Value of 1 specifies use of Big Endian data ordering for Direct Master accesses. Value of 0 specifies Little Endian ordering. Big Endian mode can be specified for Direct Master accesses by asserting the BIGEND# input pin during the address phase of the access.	Yes	Yes	0
2	Direct Slave Address Space 0 Big Endian Mode. Value of 1 specifies use of Big Endian data ordering for Direct Slave accesses to Local Address space 0. Value of 0 specifies Little Endian ordering.	Yes	Yes	0
3	Direct Slave Address Expansion ROM 0 Big Endian Mode. Value of 1 specifies use of Big Endian data ordering for Direct Slave accesses to Expansion ROM. Value of 0 specifies Little Endian ordering.	Yes	Yes	0
4	Big Endian Byte Lane Mode. Value of 1 specifies that in Big Endian mode, use byte lanes 31:16 for a 16 bit local bus and byte lanes 31:24 for an 8 bit local bus. Value of 0 specifies that in Big Endian mode, byte lanes 15:0 be used for a 16 bit local bus and byte lanes 7:0 for an 8 bit local bus.	Yes	Yes	0
5	Direct Slave Address Space 1 Big Endian Mode. Value of 1 specifies use of Big Endian data ordering for Direct Slave accesses to local Address Space 1. Value of 0 specifies Little Endian ordering.	Yes	Yes	0
6	DMA Channel 1 Big Endian Mode. Value of 1 specifies use of Big Endian data ordering for DMA Channel 1 accesses to the local Address Space. Value of 0 specifies Little Endian ordering.	Yes	Yes	0
7	DMA Channel 0 Big Endian Mode. Value of 1 specifies use of Big Endian data ordering for DMA Channel 0 accesses to the Local Address space. Value of 0 specifies Little Endian ordering.	Yes	Yes	0
31:8	Reserved.	Yes	No	0

4.4.5 (EROMRR; PCI:10h, LOC:90h) Expansion ROM Range Register

Table 4-37. (EROMRR; PCI:10h, LOC:90h) Expansion ROM Range Register

Field	Description	Read	Write	Value after Reset
10:0	Reserved.	Yes	No	0
31:11	Specifies which PCI address bits to use for decoding a PCI to local bus expansion ROM. Each of the bits corresponds to a PCI address bit. Bit 31 corresponds to Address bit 31. Write a value of 1 to all bits to be included in decode and a 0 to all others (used in conjunction with PCI Configuration register 30h). Default is 64 KB.	Yes	Yes	FFFF00h

Note: *Range (**not** Range register) must be power of 2. “Range register value” is the inverse of range.*

4.4.6 (EROMBA; PCI:14h, LOC:94h) Expansion ROM Local Base Address (Remap) Register and BREQo Control

Table 4-38. (EROMBA; PCI:14h, LOC:94h) Expansion ROM Local Base Address (Remap) Register and BREQo Control

Field	Description	Read	Write	Value after Reset
3:0	Direct Slave BREQo (Backoff Request Out) Delay Clocks. Number of local bus clocks in which a Direct Slave HOLD request is pending and a Local Direct Master access is in progress and not being granted the bus (LHOLDA) before asserting BREQo. Once asserted, BREQo remains asserted until the PCI 9080 receives LHOLDA (LSB = 8 or 64 clocks).	Yes	Yes	0
4	Local Bus BREQo Enable. Value of 1 enables PCI 9080 to assert the BREQo output.	Yes	Yes	0
5	BREQo Timer-Resolution. Value of 1 changes the LSB of the BREQo timer from 8 to 64 clocks.	Yes	Yes	0
10:6	Reserved.	Yes	No	0
31:11	Remap of PCI Expansion ROM Space into a Local Address Space. The bits in this register remap (replace) the PCI address bits used in decode as the local address bits.	Yes	Yes	0

Note: *Remap Address value must be multiple of Range (**not** the Range register).*

4.4.7 (LBRD0; PCI:18h, LOC:98h) Local Address Space 0/Expansion ROM Bus Region Descriptor Register

Table 4-39. (LBRD0; PCI:18h, LOC:98h) Local Address Space 0/Expansion ROM Bus Region Descriptor Register

Field	Description	Read	Write	Value after Reset

1:0	Memory Space 0 Local Bus Width. Value of 00 indicates bus width of 8 bits, a value of 01 indicates bus width of 16 bits and a value of 10 or 11 indicates bus width of 32 bits.	Yes	Yes	S = 01
				J = 11
				C = 11

5:2	Memory Space 0 Internal Wait States (data to data; 0-15 wait states).	Yes	Yes	0
6	Memory Space 0 Ready Input Enable. Value of 1 enables Ready input. Value of 0 disables Ready input.	Yes	Yes	0
7	Memory Space 0 BTERM# Input Enable. Value of 1 enables BTERM# input. Value of 0 disables BTERM# input. If this bit is set to 0, PCI 9080 bursts four Lword maximum at a time.	Yes	Yes	0
8	Memory Space 0 Prefetch Disable. If mapped into memory space, a value of 0 enables read prefetching. Value of 1 disables prefetching. If prefetching is disabled, PCI 9080 disconnects after each memory read.	Yes	Yes	0
9	Expansion ROM Space Prefetch Disable. Value of 0 enables read prefetching. Value of 1 disables prefetching. If prefetching is disabled, PCI 9080 disconnects after each memory read.	Yes	Yes	0
10	Read Prefetch Count Enable. When set to 1 and memory prefetching is enabled, PCI 9080 prefetches up to the number of Lwords specified in the prefetch count. When set to 0, PCI 9080 ignores the count and continues prefetching until terminated by the PCI bus.	Yes	Yes	0
14:11	Prefetch Counter. Number of Lwords to prefetch during memory read cycles (0-15). A count of zero selects a prefetch of 16 Lwords.	Yes	Yes	0
15	Reserved.	Yes	No	0

17:16	Expansion ROM Space Local Bus Width. Value of 00 indicates bus width of 8 bits, a value of 01 indicates bus width of 16 bits and a value of 10 or 11 indicates bus width of 32 bits.	Yes	Yes	S = 01
				J = 11
				C = 11

21:18	Expansion ROM Space Internal Wait States (data to data; 0-15 wait states).	Yes	Yes	0
22	Expansion ROM Space Ready Input Enable. Value of 1 enables Ready input. Value of 0 disables Ready input.	Yes	Yes	0
23	Expansion ROM Space BTERM# Input Enable. Value of 1 enables BTERM# input. Value of 0 disables BTERM# input. If this bit is set to 1, PCI 9080 bursts four Lword maximum at a time.	Yes	Yes	0
24	Memory Space 0 Burst Enable. Value of 1 enables bursting. Value of 0 disables bursting. If burst is disabled, the local bus performs continuous single cycles for burst PCI read/write cycles.	Yes	Yes	0
25	Extra Long Load from serial EEPROM. Value of 1 loads the Subsystem ID and Local Address Space 1 registers. Value of 0 indicates not to load them.	Yes	No	0
26	Expansion ROM Space Burst Enable. Value of 1 enables bursting. Value of 0 disables bursting. If burst is disabled, the local bus performs continuous single cycles for burst PCI read/write cycles.	Yes	Yes	0
27	Direct Slave PCI Write Mode. Value of 0 indicates PCI 9080 should disconnect when the Direct Slave write FIFO is full. Value of 1 indicates PCI 9080 should de-assert TRDY# when the write FIFO is full.	Yes	Yes	0
31:28	PCI Target Retry Delay Clocks. Contains the value (multiplied by 8) of the number of PCI bus clocks after receiving a PCI local read or write access and not successfully completing a transfer. Only pertains to Direct Slave writes when bit 27 is set to 1.	Yes	Yes	4 (32 clocks)

4.4.8 (DMRR; PCI:1Ch, LOC:9Ch) Local Range Register for Direct Master to PCI

Table 4-40. (DMRR; PCI:1Ch, LOC:9Ch) Local Range Register for Direct Master to PCI

Field	Description	Read	Write	Value after Reset
15:0	Reserved (64 KB increments).	Yes	No	0
31:16	Specifies which local address bits to use for decoding a local to PCI bus access. Each of the bits corresponds to a PCI address bit. Bit 31 corresponds to Address bit 31. Write a value of 1 to all bits that must be included in decode and a 0 to all others. This range register is used for Direct Master memory, I/O, or configuration accesses.	Yes	Yes	0

Note: *Range (**not** Range register) must be power of 2. “Range register value” is the inverse of range.*

4.4.9 (DMLBAM; PCI:20h, LOC:A0h) Local Bus Base Address Register for Direct Master to PCI Memory

Table 4-41. (DMLBAM; PCI:20h, LOC:A0h) Local Bus Base Address Register for Direct Master to PCI Memory

Field	Description	Read	Write	Value after Reset
15:0	Reserved.	Yes	No	0
31:16	Assigns a value to the bits to use for decoding a local to PCI memory access.	Yes	Yes	0

Note: *Local Base Address value must be multiple of Range (**not** the Range register).*

4.4.10 (DMLBAI; PCI:24h, LOC:A4h) Local Base Address Register for Direct Master to PCI IO/CFG

Table 4-42. (DMLBAI; PCI:24h, LOC:A4h) Local Base Address Register for Direct Master to PCI IO/CFG

Field	Description	Read	Write	Value after Reset
15:0	Reserved.	Yes	No	0
31:16	Assigns a value to the bits to use for decoding a local to PCI I/O or configuration access. This base address is used for Direct Master I/O and configuration accesses.	Yes	Yes	0

Notes: *Local Base Address value must be multiple of Range (**not** the Range register).*

Refer to bit 13 of (LOC:A8h) for I/O Remap Address option (refer to Table 4-43).

4.4.11 (DMPBAM; PCI:28h, LOC:A8h) PCI Base Address (Remap) Register for Direct Master to PCI Memory

Table 4-43. (DMPBAM; PCI:28h, LOC:A8h) PCI Base Address (Remap) Register for Direct Master to PCI Memory

Field	Description	Read	Write	Value after Reset
0	Direct Master Memory Access Enable. Value of 1 enables decode of Direct Master Memory accesses. Value of 0 disables decode of Direct Master Memory accesses.	Yes	Yes	0
1	Direct Master I/O Access Enable. Value of 1 enables decode of Direct Master I/O accesses. Value of 0 disables decode of Direct Master I/O accesses.	Yes	Yes	0
2	LLOCK# Input Enable. Value of 1 enables LLOCK# input, enabling PCI-locked sequences. Value of 0 disables LLOCK# input.	Yes	Yes	0
12, 3	Direct Master Read Prefetch Size control. Values: 00 = PCI 9080 continues to prefetch read data from the PCI bus until Direct Master access is finished. This may result in an additional four unneeded Lwords being prefetched from the PCI bus. 01 = Prefetch up to four Lwords from the PCI bus 10 = Prefetch up to eight Lwords from the PCI bus 11 = Prefetch up to 16 Lwords from the PCI bus If PCI memory prefetch is not wanted, performs a Direct Master single cycle. The direct master burst reads must not exceed the programmed limit.	Yes	Yes	00
4	Direct Master PCI Read Mode. Value of 0 indicates PCI 9080 should release the PCI bus when the read FIFO becomes full. Value of 1 indicates PCI 9080 should keep the PCI bus and de-assert IRDY when the read FIFO becomes full.	Yes	Yes	0
10, 8:5	Programmable Almost Full Flag. When the number of entries in the 32 word direct master write FIFO exceeds this value, the output pin DMPAF# is asserted low.	Yes	Yes	000
9	Write and Invalidate Mode. When set to 1, PCI 9080 waits for 8 or 16 Lwords to be written from the local bus before starting PCI access. When set, all local Direct Master to PCI write accesses must be 8 or 16 Lword bursts. Use in conjunction with (PCI:04h)(LOC:04h) (refer to Table 4-11[4] and Section 3.6.1.9.2, “Direct Master Write and Invalidate”).	Yes	Yes	0
11	Direct Master Prefetch Limit. If set to 1, don’t prefetch past 4K (4098 bytes) boundaries.	Yes	Yes	0
13	I/O Remap Select. When set to 1, forces PCI address bits [31:16] to all zeros. When set to 0, uses bits [31:16] of this register as PCI address bits [31:16].	Yes	Yes	0
15:14	Direct Master Write Delay. This register is used to delay the PCI bus request after direct master burst write cycle has started. Values: 00 = No delay; start the cycle immediately 01 = Delay 4 PCI clocks 10 = Delay 8 PCI clocks 11 = Delay 16 PCI clocks	Yes	Yes	00
31:16	Remap of Local to PCI Space into a PCI Address Space. The bits in this register remap (replace) the local address bits used in decode as the PCI address bits. This PCI Remap address is used for Direct Master memory and I/O accesses.	Yes	Yes	0

Note: *Remap Address value must be multiple of Range (**not** the Range register).*

4.4.12 (DMCFGA; PCI:2Ch, LOC:ACh) PCI Configuration Address Register for Direct Master to PCI IO/CFG

Table 4-44. (DMCFGA; PCI:2Ch, LOC:ACh) PCI Configuration Address Register for Direct Master to PCI IO/CFG

Field	Description	Read	Write	Value after Reset
1:0	Configuration Type (00=Type 0, 01=Type 1).	Yes	Yes	0
7:2	Register Number. If different register read/write is needed, this register value must be programmed and a new PCI configuration cycle must be generated.	Yes	Yes	0
10:8	Function Number.	Yes	Yes	0
15:11	Device Number.	Yes	Yes	0
23:16	Bus Number.	Yes	Yes	0
30:24	Reserved.	Yes	No	0
31	Configuration Enable. Value of 1 allows local to PCI I/O accesses to be converted to a PCI configuration cycle. The parameters in this table are used to generate the PCI configuration address.	Yes	Yes	0

Note: *Refer to the Configuration Cycle Generation example in Section 3.6.1.6, "CFG (PCI Configuration Type 0 or Type 1 Cycles)."*

4.4.13 (LAS1RR; PCI:F0h, LOC:170h) Local Address Space 1 Range Register for PCI to Local Bus

Table 4-45. (LAS1RR; PCI:F0h, LOC:170h) Local Address Space 1 Range Register for PCI to Local Bus

Field	Description	Read	Write	Value after Reset
0	Memory Space Indicator. Value of 0 indicates Local Address Space 1 maps into PCI memory space. Value of 1 indicates address space 1 maps into PCI I/O space.	Yes	Yes	0
2:1	If mapped into memory space, encoding is as follows: 2/1 Meaning 0 0 Locate anywhere in 32 bit PCI address space 0 1 Locate below 1 MB in PCI address space 1 0 Locate anywhere in 64 bit PCI address space 1 1 Reserved If mapped into I/O space, bit 1 must be set to 0. Bit 2 is included with bits [31:3] to indicate decoding range.	Yes	Yes	0
3	If mapped into memory space, a value of 1 indicates reads are prefetchable (bit has no effect on the operation of the PCI 9080, but is for system status). If mapped into I/O space, bit is included with bits [31:2] to indicate decoding range.	Yes	Yes	0
31:4	Specifies which PCI address bits to use for decoding a PCI access to local bus space 1. Each of the bits corresponds to a PCI address bit. Bit 31 corresponds to Address bit 31. Write a value of 1 to all bits that must be included in decode and a 0 to all others (Used in conjunction with PCI Configuration Register Ch 1). Default is 1 MB.	Yes	Yes	FFF0000h

Note: *Range (**not** Range register) must be power of 2. "Range register value" is the inverse of range.*

User should limit all I/O spaces to 256 bytes per PCI 2.1 spec.

If QSR bit 0 is set, defines PCI Base Address 0.

4.4.14 (LAS1BA; PCI:F4h, LOC:174h) Local Address Space 1 Local Base Address (Remap) Register

Table 4-46. (LAS1BA; PCI:F4h, LOC:174h) Local Address Space 1 Local Base Address (Remap) Register

Field	Description	Read	Write	Value after Reset
0	Space 1 Enable. Value of 1 enables decoding of PCI addresses for Direct Slave access to local space 1. Value of 0 disables decoding. If this bit is set to 0, the PCI BIOS may not allocate (assign) the base address for Space 1. ***Note:*** *Must be set to 1 for any Direct Slave access to Space 1.*	Yes	Yes	0
1	Reserved.	Yes	No	0
3:2	If local space 1 is mapped into memory space, bits are not used. If mapped into I/O space, bit is included with bits [31:4] for remapping.	Yes	Yes	0
31:4	Remap of PCI Address to Local Address Space 1 into a Local Address Space. The bits in this register remap (replace) the PCI Address bits used in decode as the Local Address bits.	Yes	Yes	0

Note: *Remap Address value must be multiple of Range (**not** the Range register).*

4.4.15 (LBRD1; PCI:F8h, LOC:178h) Local Address Space 1 Bus Region Descriptor Register

Table 4-47. (LBRD1; PCI:F8h, LOC:178h) Local Address Space 1 Bus Region Descriptor Register

Field	Description	Read	Write	Value after Reset

1:0	Memory Space 1 Local Bus Width. Value of 00 indicates bus width of 8 bits, a value of 01 indicates bus width of 16 bits and a value of 10 or 11 indicates bus width of 32 bits.	Yes	Yes	S = 01
				J = 11
				C = 11

5:2	Memory Space 1 Internal Wait States (data to data; 0-15 wait states).	Yes	Yes	0
6	Memory Space 1 Ready Input Enable. Value of 1 enables Ready input. Value of 0 disables Ready input.	Yes	Yes	0
7	Memory Space 1 BTERM# Input Enable. Value of 1 enables BTERM# input. Value of 0 disables BTERM# input. If this bit is set to 0, PCI 9080 bursts four Lword maximum at a time.	Yes	Yes	0
8	Memory Space 1 Burst Enable. Value of 1 enables bursting. Value of 0 disables bursting. If burst is disabled, the local bus performs continuous single cycles for burst PCI read/write cycles.	Yes	Yes	0
9	Memory Space 1 Prefetch Disable. If mapped into memory space, 0 enables read prefetching. Value of 1 disables prefetching. If prefetching is disabled, PCI 9080 disconnects after each memory read.	Yes	Yes	0
10	Read Prefetch Count Enable. When set to 1 and memory prefetching is enabled, PCI 9080 prefetches up to the number of Lwords specified in the prefetch count. When set to 0, PCI 9080 ignores the count is ignored and continues prefetching until terminated by the PCI bus.	Yes	Yes	0
14:11	Prefetch Counter. Number of Lwords to prefetch during memory read cycles (0-15).	Yes	Yes	0
31:15	Reserved.	Yes	No	0

4.5 RUNTIME REGISTERS

4.5.1 (MBOX0; PCI:40h or 78h, LOC:C0h) Mailbox Register 0

Table 4-48. (MBOX0; PCI:40h or 78h, LOC:C0h) Mailbox Register 0

Field	Description	Read	Write	Value after Reset
31:0	32 Bit Mailbox Register.	Yes	Yes	0

Note: *Mailbox Register 0 is replaced by the Inbound Queue Port when the I_2O feature is enabled (bit 0 in the QSR register is set). Mailbox Register 0 is always accessible at PCI address 78h and local address C0h.*

4.5.2 (MBOX1; PCI:44h or 7Ch, LOC:C4h) Mailbox Register 1

Table 4-49. (MBOX1; PCI:44h or 7Ch, LOC:C4h) Mailbox Register 1

Field	Description	Read	Write	Value after Reset
31:0	32 Bit Mailbox Register.	Yes	Yes	0

Note: *Mailbox Register 1 is replaced by the Outbound Queue Port when the I_2O feature is enabled (bit 0 in the QSR register is set). Mailbox Register 1 is always accessible at PCI address 7Ch and local address C4h.*

4.5.3 (MBOX2; PCI:48h, LOC:C8h) Mailbox Register 2

Table 4-50. (MBOX2; PCI:48h, LOC:C8h) Mailbox Register 2

Field	Description	Read	Write	Value after Reset
31:0	32 Bit Mailbox Register.	Yes	Yes	0

4.5.4 (MBOX3; PCI:4Ch, LOC:CCh) Mailbox Register 3

Table 4-51. (MBOX3; PCI:4Ch, LOC:CCh) Mailbox Register 3

Field	Description	Read	Write	Value after Reset
31:0	32 Bit Mailbox Register.	Yes	Yes	0

4.5.5 (MBOX4; PCI:50h, LOC:D0h) Mailbox Register 4

Table 4-52. (MBOX4; PCI:50h, LOC:D0h) Mailbox Register 4

Field	Description	Read	Write	Value after Reset
31:0	32 Bit Mailbox Register.	Yes	Yes	0

4.5.6 (MBOX5; PCI:54h, LOC:D4h) Mailbox Register 5

Table 4-53. (MBOX5; PCI:54h, LOC:D4h) Mailbox Register 5

Field	Description	Read	Write	Value after Reset
31:0	32 Bit Mailbox Register.	Yes	Yes	0

4.5.7 (MBOX6; PCI:58h, LOC:D8h) Mailbox Register 6

Table 4-54. (MBOX6; PCI:58h, LOC:D8h) Mailbox Register 6

Field	Description	Read	Write	Value after Reset
31:0	32 Bit Mailbox Register.	Yes	Yes	0

4.5.8 (MBOX7; PCI:5Ch, LOC:DCh) Mailbox Register 7

Table 4-55. (MBOX7; PCI:5Ch, LOC:DCh) Mailbox Register 7

Field	Description	Read	Write	Value after Reset
31:0	32 Bit Mailbox Register.	Yes	Yes	0

4.5.9 (P2LDBELL; PCI:60h, LOC:E0h) PCI to Local Doorbell Register

Table 4-56. (P2LDBELL; PCI:60h, LOC:E0h) PCI to Local Doorbell Register

Field	Description	Read	Write	Value after Reset
31:0	Doorbell Register. A PCI master can write to this register and generate a local interrupt to the local processor. The local processor can then read this register to determine which doorbell bit was asserted. PCI master sets a 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	0

4.5.10 (L2PDBELL; PCI:64h, LOC:E4h) Local to PCI Doorbell Register

Table 4-57. (L2PDBELL; PCI:64h, LOC:E4h) Local to PCI Doorbell Register

Field	Description	Read	Write	Value after Reset
31:0	Doorbell Register. The local processor can write to this register and generate a PCI interrupt. A PCI master can then read this register to determine which doorbell bit was asserted. The local processor sets a doorbell by writing 1 to a particular bit. PCI master can clear a doorbell bit by writing 1 to that bit position.	Yes	Yes/Clr	0

4.5.11 (INTCSR; PCI:68h, LOC:E8h) Interrupt Control/Status Register

Table 4-58. (INTCSR; PCI:68h, LOC:E8h) Interrupt Control/Status Register

Field	Description	Read	Write	Value after Reset
0	Enable Local Bus LSERR#. Value of 1 enables PCI 9080 to assert LSERR# interrupt output when PCI bus Target Abort or Master Abort status bit is set in the PCI Status configuration register.	Yes	Yes	0
1	Enable Local Bus LSERR# when PCI parity error occurs during PCI 9080 Master Transfer or PCI 9080 Slave access or Outbound Free List FIFO Overflow Init.	Yes	Yes	0
2	Generate PCI Bus SERR#. When this bit is set to 0, writing a 1 generates a PCI bus SERR#.	Yes	Yes	0
3	Mailbox Interrupt Enable. Value of 1 enables a Local Interrupt to be generated when the PCI bus writes to Mailbox registers 0-3. To clear the Local Interrupt, the Local master must read the Mailbox. Used in conjunction with Local interrupt enable.	Yes	Yes	0
7:4	Reserved.	Yes	No	0
8	PCI Interrupt Enable. Value of 1 enables PCI interrupts.	Yes	Yes	1
9	PCI Doorbell Interrupt Enable. Value of 1 enables doorbell interrupts. Used in conjunction with PCI interrupt enable. Clearing the doorbell interrupt 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 detect of a target abort to generate a PCI interrupt. Used in conjunction with PCI interrupt enable. Clearing the abort status bits also clears the PCI interrupt.	Yes	Yes	0
11	PCI Local Interrupt Enable. Value of 1 enables a local interrupt input to generate a PCI interrupt. Use in conjunction with PCI interrupt enable. Clearing the local bus cause of the interrupt also clears the interrupt.	Yes	Yes	0
12	Retry Abort Enable. Value of 1 enables PCI 9080 to treat 256 Master consecutive retries to a Target as a Target Abort. Value of 0 enables PCI 9080 to attempt Master Retries indefinitely. ***Note:*** *For diagnostic purposes only.*	Yes	Yes	0
13	Value of 1 indicates PCI doorbell interrupt is active.	Yes	No	0
14	Value of 1 indicates PCI abort interrupt is active.	Yes	No	0
15	Value of 1 indicates local interrupt is active (LINTi#).	Yes	No	0
16	Local Interrupt Output Enable. Value of 1 enables local interrupt output.	Yes	Yes	1
17	Local Doorbell Interrupt Enable. Value of 1 enables doorbell interrupts. Used in conjunction with Local interrupt enable. Clearing the local doorbell interrupt bits that caused the interrupt also clears the interrupt.	Yes	Yes	0
18	Local DMA Channel 0 Interrupt Enable. Value of 1 enables DMA Channel 0 interrupts. Used in conjunction with Local interrupt enable. Clearing the DMA status bits also clears the interrupt.	Yes	Yes	0
19	Local DMA Channel 1 Interrupt Enable. Value of 1 enables DMA Channel 1 interrupts. Used in conjunction with Local interrupt enable. Clearing the DMA status bits also clears the interrupt.	Yes	Yes	0
20	Value of 1 indicates local doorbell interrupt is active.	Yes	No	0
21	Value of 1 indicates DMA Ch 0 interrupt is active.	Yes	No	0
22	Value of 1 indicates DMA Ch 1 interrupt is active.	Yes	No	0
23	Value of 1 indicates BIST interrupt is active. BIST (Built-In Self Test) interrupt is generated by writing 1 to bit 6 of the PCI Configuration BIST register. Clearing bit 6 clears the interrupt. Refer to the BIST Register for a description of self test. (Refer to Table 4-18.)	Yes	No	0

Table 4-58. (INTCSR; PCI:68h, LOC:E8h) Interrupt Control/Status Register (continued)

Field	Description	Read	Write	Value after Reset
24	Value of 0 indicates a Direct Master was the bus master during a Master or Target abort. (Not valid until abort occurs.)	Yes	No	1
25	Value of 0 indicates DMA CH 0 was the bus master during a Master or Target abort. (Not valid until abort occurs.)	Yes	No	1
26	Value of 0 indicates DMA CH 1 was the bus master during a Master or Target abort. (Not valid until abort occurs.)	Yes	No	1
27	Value of 0 indicates a Target Abort was generated by the PCI 9080 after 256 consecutive Master retries to a Target. (Not valid until abort occurs.)	Yes	No	1
28	Value of 1 indicates PCI wrote data to the MailBox #0. Enabled only if MBOXINTENB is enabled (bit 3 high).	Yes	No	0
29	Value of 1 indicates PCI wrote data to the MailBox #1. Enabled only if MBOXINTENB is enabled (bit 3 high).	Yes	No	0
30	Value of 1 indicates PCI wrote data to the MailBox #2. Enabled only if MBOXINTENB is enabled (bit 3 high).	Yes	No	0
31	Value of 1 indicates PCI wrote data to the MailBox #3. Enabled only if MBOXINTENB is enabled (bit 3 high).	Yes	No	0

4.5.12 (CNTRL; PCI:6Ch, LOC:ECh) Serial EEPROM Control, PCI Command Codes, User I/O Control, Init Control Register

Table 4-59. (CNTRL; PCI:6Ch, LOC:ECh) Serial EEPROM Control, PCI Command Codes, User I/O Control, Init Control Register

Field	Description	Read	Write	Value after Reset
3:0	PCI Read Command Code for DMA. This PCI command is sent out during DMA read cycles.	Yes	Yes	1110
7:4	PCI Write Command Code for DMA. This PCI command is sent out during DMA write cycles.	Yes	Yes	0111
11:8	PCI Memory Read Command Code for Direct Master. This PCI command is sent out during Direct Master read cycles.	Yes	Yes	0110
15:12	PCI Memory Write Command Code for Direct Master. This PCI command is sent out during Direct Master write cycles.	Yes	Yes	0111
16	General Purpose Output. Value of 1 causes USERO output to go high. Value of 0 causes USER0 output to go low.	Yes	Yes	1
17	General Purpose Input. Value of 1 indicates USERI input pin is high. Value of 0 indicates USERI pin is low.	Yes	No	—
23:18	Reserved.	Yes	No	0
24	Serial EEPROM Clock for Local or PCI Bus Reads or Writes to serial EEPROM. Toggling this bit generates a serial EEPROM clock. (Refer to the manufacturer's data sheet for the particular serial EEPROM being used.)	Yes	Yes	0
25	Serial EEPROM Chip Select. For local or PCI bus reads or writes to serial EEPROM, setting this bit to 1 provides the serial EEPROM chip select.	Yes	Yes	0
26	Write Bit to serial EEPROM. For writes, this output bit is the input to the serial EEPROM. Clocked into the serial EEPROM by the serial EEPROM clock.	Yes	Yes	0
27	Read serial EEPROM Data Bit. For reads, this input bit is the output of the serial EEPROM. Clocked out of the serial EEPROM by the serial EEPROM clock.	Yes	No	—
28	Serial EEPROM Present. Value of 1 indicates a serial EEPROM is present.	Yes	No	0
29	Reload Configuration Registers. When this bit is set to 0, writing a 1 causes PCI 9080 to reload the local configuration registers from serial EEPROM.	Yes	Yes	0
30	PCI Adapter Software Reset. Value of 1 holds the local bus logic in the PCI 9080 reset and LRESETo# asserted. The contents of the PCI configuration registers and Shared Run Time registers will not be reset. Software Reset can only be cleared from the PCI bus. (Local bus remains reset until this bit is cleared.)	Yes	Yes	0
31	Local Init Status. Value of 1 indicates Local Init done. Responses to PCI accesses are RETRYs until this bit is set. While input pin NB# is asserted low, this bit is forced to 1.	Yes	Yes	0

4.5.13 (PCIHIDR; PCI:70h, LOC:F0h) PCI Permanent Configuration ID Register

Table 4-60. (PCIHIDR; PCI:70h, LOC:F0h) PCI Permanent Configuration ID Register

Field	Description	Read	Write	Value after Reset
15:0	Permanent Vendor ID. Identifies device manufacturer. ***Note:*** *Hardcoded to the PCI SIG issued vendor ID of PLX (10B5h).*	Yes	No	10B5h
31:16	Permanent Device ID. Identifies the particular device. ***Note:*** *Hardcoded to the PLX part number for PCI interface chip PCI 9080.*	Yes	No	9080h

4.5.14 (PCIHREV; PCI:74h, LOC:F4h) PCI Permanent Revision ID Register

Table 4-61. (PCIHREV; PCI:74h, LOC:F4h) PCI Permanent Revision ID Register

Field	Description	Read	Write	Value after Reset
7:0	Permanent Revision ID. ***Note:*** *Hardcoded to the silicon revision of the PCI 9080.*	Yes	No	Current Rev #

4.6 DMA REGISTERS

4.6.1 (DMAMODE0; PCI:80h, LOC:100h) DMA Channel 0 Mode Register

Table 4-62. (DMAMODE0; PCI:80h, LOC:100h) DMA Channel 0 Mode Register

Field	Description	Read	Write	Value after Reset

1:0	Local Bus Width. Value of 00 indicates bus width of 8 bits, a value of 01 indicates bus width of 16 bits and a value of 10 or 11 indicates bus width of 32 bits.	Yes	Yes	S = 01
				J = 11
				C = 11

5:2	Internal Wait States (data to data).	Yes	Yes	0
6	Ready Input Enable. Value of 1 enables Ready input. Value of 0 disables Ready input.	Yes	Yes	0
7	BTERM# Input Enable. Value of 1 enables BTERM# input. Value of 0 disables BTERM# input. If this bit is set to 0, PCI 9080 bursts four Lword maximum at a time.	Yes	Yes	0
8	Local Burst Enable. Value of 1 enables bursting. Value of 0 disables local bursting. If burst is disabled, the local bus performs continuous single cycles for burst PCI read/write cycles.	Yes	Yes	0
9	Chaining. Value of 1 indicates chaining mode is enabled. For chaining mode, the DMA source address, destination address and byte count are loaded from memory in PCI or Local Address Spaces. Value of 0 indicates nonchaining mode is enabled.	Yes	Yes	0
10	Done Interrupt Enable. Value of 1 enables interrupt when done. Value of 0 disables interrupt when done. If DMA Clear Count mode is enabled, the interrupt won't occur until the byte count is cleared.	Yes	Yes	0
11	Local Addressing Mode. Value of 1 indicates local address LA[31:2] to be held constant. Value of 0 indicates local address is incremented.	Yes	Yes	0
12	Demand Mode. Value of 1 causes DMA controller to operate in Demand mode. In Demand mode, the DMA controller transfers data when its DREQ[1:0]# input is asserted. It asserts DACK[1:0]# to indicate the current local bus transfer is in response to the DREQ[1:0]# input. DMA controller transfers Lwords (32 bits) of data. This may result in multiple transfers for an 8- or 16-bit bus.	Yes	Yes	0
13	Write and Invalidate Mode for DMA Transfers. When set to 1, PCI 9080 performs Write and Invalidate cycles to the PCI bus. PCI 9080 supports Write and Invalidate sizes of 8 or 16 Lwords. The size is specified in the PCI Cache Line Size Register. If a size other than 8 or 16 is specified, PCI 9080 performs write transfers rather than Write and Invalidate transfers. Transfers must start and end at the Cache Line Boundaries.	Yes	Yes	0
14	DMA EOT (End of Transfer) Enable. Value of 1 enables EOT[1:0]# input pin. Value of 0 disables EOT[1:0]# input pin. (Refer to Section 3.7.6.1, "End of Transfer (EOT0# or EOT1#) Input.")	Yes	Yes	0
15	DMA Stop Data Transfer Mode. Value of 0 sends a BLAST to terminate DMA transfer. Value of 1 indicates an EOT asserted or DREQ[1:0]# negated during demand mode DMA terminates the DMA transfer. (Refer to Section 3.7.6.1, "End of Transfer (EOT0# or EOT1#) Input.")	Yes	Yes	0
16	DMA Clear Count Mode. When set to 1, the byte count in each chaining descriptor, if it is in local memory, is cleared when the corresponding DMA transfer is complete. ***Note:*** *If chaining descriptor is in PCI memory, the count is not cleared.*	Yes	Yes	0
17	DMA Channel 0 Interrupt Select. Value of 1 routes the DMA Channel 0 interrupt to the PCI interrupt. Value of 0 routes the DMA Channel 0 interrupt to the local bus interrupt.	Yes	Yes	0
31:18	Reserved.	Yes	No	0

4.6.2 (DMAPADR0; PCI:84h, LOC:104h) DMA Channel 0 PCI Address Register

Table 4-63. (DMAPADR0; PCI:84h, LOC:104h) DMA Channel 0 PCI Address Register

Field	Description	Read	Write	Value after Reset
31:0	PCI Address Register. Indicates from where in the PCI memory space the DMA transfers (reads or writes) start.	Yes	Yes	0

4.6.3 (DMALADR0; PCI:88h, LOC:108h) DMA Channel 0 Local Address Register

Table 4-64. (DMALADR0; PCI:88h, LOC:108h) DMA Channel 0 Local Address Register

Field	Description	Read	Write	Value after Reset
31:0	Local Address Register. Indicates from where in the local memory space the DMA transfers (reads or writes) start.	Yes	Yes	0

4.6.4 (DMASIZ0; PCI:8Ch, LOC:10Ch) DMA Channel 0 Transfer Size (Bytes) Register

Table 4-65. (DMASIZ0; PCI:8Ch, LOC:10Ch) DMA Channel 0 Transfer Size (Bytes) Register

Field	Description	Read	Write	Value after Reset
22:0	DMA Transfer Size (Bytes). Indicates number of bytes to be transferred during DMA operation.	Yes	Yes	0
31:23	Reserved.	Yes	No	0

4.6.5 (DMADPR0; PCI:90h, LOC:110h) DMA Channel 0 Descriptor Pointer Register

Table 4-66. (DMADPR0; PCI:90h, LOC:110h) DMA Channel 0 Descriptor Pointer Register

Field	Description	Read	Write	Value after Reset
0	Descriptor Location. Value of 1 indicates PCI address space. Value of 0 indicates Local Address Space.	Yes	Yes	0
1	End of Chain. Value of 1 indicates end of chain. Value of 0 indicates not end of chain descriptor. (Same as Nonchaining Mode.)	Yes	Yes	0
2	Interrupt after Terminal Count. Value of 1 causes an interrupt to be generated after the terminal count for this descriptor is reached. Value of 0 disables interrupts from being generated.	Yes	Yes	0
3	Direction of Transfer. Value of 1 indicates transfers from local bus to PCI bus. Value of 0 indicates transfers from PCI bus to local bus.	Yes	Yes	0
31:4	Next Descriptor Address. Quad word aligned (bits [3:0] = 0000).	Yes	Yes	0

4.6.6 (DMAMODE1; PCI:94h, LOC:114h) DMA Channel 1 Mode Register

Table 4-67. (DMAMODE1; PCI:94h, LOC:114h) DMA Channel 1 Mode Register

Field	Description	Read	Write	Value after Reset

1:0	Local Bus Width. Value of 00 indicates bus width of 8 bits, a value of 01 indicates bus width of 16 bits and a value of 10 or 11 indicates bus width of 32 bits.	Yes	Yes	S = 01
				J = 11
				C = 11

5:2	Internal Wait States (data to data).	Yes	Yes	0
6	Ready Input Enable. Value of 1 enables Ready input. Value of 0 disables Ready input.	Yes	Yes	0
7	BTERM# Input Enable. Value of 1 enables BTERM# input. Value of 0 disables BTERM# input. If this bit is set to 0, PCI 9080 bursts four Lword maximum at a time.	Yes	Yes	0
8	Local Burst Enable. Value of 1 enables bursting. Value of 0 disables local bursting. If burst is disabled, the local bus performs continuous single cycles for burst PCI read/write cycles.	Yes	Yes	0
9	Chaining. Value of 1 indicates chaining mode enabled. For chaining mode, the DMA source address, destination address and byte count are loaded from memory in PCI or Local address spaces. Value of 0 indicates nonchaining mode enabled.	Yes	Yes	0
10	Done Interrupt Enable. Value of 1 enables interrupt when done. Value of 0 disables interrupt when done. If DMA Clear Count mode is enabled, the interrupt won't occur until the byte count is cleared.	Yes	Yes	0
11	Local Addressing Mode. Value of 1 indicates local address LA[31:2] to be held constant. Value of 0 indicates local address is incremented.	Yes	Yes	0
12	Demand Mode. Value of 1 causes DMA controller to operate in Demand mode. In Demand mode, the DMA controller transfers data when its DREQ[1:0]# input is asserted. It asserts DACK[1:0]# to indicate the current local bus transfer is in response to the DREQ[1:0]# input. DMA controller transfers Lwords (32 bits) of data. This may result in multiple transfers for an 8- or 16-bit bus.	Yes	Yes	0
13	Write and Invalidate Mode for DMA Transfers. When set to 1, PCI 9080 performs Write and Invalidate cycles to the PCI bus. PCI 9080 supports Write and Invalidate sizes of 8 or 16 Lwords. The size is specified in the PCI Cache Line Size Register. If a size other than 8 or 16 is specified, PCI 9080 performs write transfers rather than Write and Invalidate transfers. Transfers must start and end at the Cache Line Boundaries.	Yes	Yes	0
14	DMA EOT (End of Transfer) Enable. Value of 1 enables EOT[1:0]# input pin. Value of 0 disables EOT[1:0]# input pin. (Refer to Section 3.7.6.1, "End of Transfer (EOT0# or EOT1#) Input.")	Yes	Yes	0
15	DMA Stop Data Transfer Mode. Value of 0 BLAST terminates DMA transfer. Value of 1 indicates EOT. In demand DMA mode, if this bit is set to a value of 1, assertion of EOT causes DMA controller to terminate following the current data phase (blast may or may not be asserted). If bit is not set, and EOT asserted, the DMA controller completes current data phase and potentially a following data phase in which blast is asserted. (Refer to Section 3.7.6.1, "End of Transfer (EOT0# or EOT1#) Input.")	Yes	Yes	0
16	DMA Clear Count Mode. When set to 1, the byte count in each chaining descriptor, if it is in local memory, is cleared when the corresponding DMA transfer is complete. ***Note:*** *If chaining descriptor is in PCI memory, the count is not cleared.*	Yes	Yes	0
17	DMA Channel 1 Interrupt Select. Value of 1 routes the DMA Channel 1 interrupt to the PCI interrupt. Value of 0 routes the DMA Channel 1 interrupt to the local bus interrupt.	Yes	Yes	0
31:18	Reserved.	Yes	No	0

4.6.7 (DMAPADR1; PCI:98h, LOC:118h) DMA Channel 1 PCI Address Register

Table 4-68. (DMAPADR1; PCI:98h, LOC:118h) DMA Channel 1 PCI Address Register

Field	Description	Read	Write	Value after Reset
31:0	PCI Data Address Register. Indicates from where in the PCI memory space the DMA transfers (reads or writes) start.	Yes	Yes	0

4.6.8 (DMALADR1; PCI:9Ch, LOC:11Ch) DMA Channel 1 Local Address Register

Table 4-69. (DMALADR1; PCI:9Ch, LOC:11Ch) DMA Channel 1 Local Address Register

Field	Description	Read	Write	Value after Reset
31:0	Local Data Address Register. Indicates from where in the local memory space the DMA transfers (reads or writes) start.	Yes	Yes	0

4.6.9 (DMASIZ1; PCI:A0h, LOC:120h) DMA Channel 1 Transfer Size (Bytes) Register

Table 4-70. (DMASIZ1; PCI:A0h, LOC:120h) DMA Channel 1 Transfer Size (Bytes) Register

Field	Description	Read	Write	Value after Reset
22:0	DMA Transfer Size (Bytes). Indicates the number of bytes to transfer during a DMA operation.	Yes	Yes	0
31:23	Reserved.	Yes	No	0

4.6.10 (DMADPR1; PCI:A4h, LOC:124h) DMA Channel 1 Descriptor Pointer Register

Table 4-71. (DMADPR1; PCI:A4h, LOC:124h) DMA Channel 1 Descriptor Pointer Register

Field	Description	Read	Write	Value after Reset
0	Descriptor Location. Value of 1 indicates PCI address space. Value of 0 indicates Local Address Space.	Yes	Yes	0
1	End of Chain. Value of 1 indicates end of chain. Value of 0 indicates not end of chain descriptor. (Same as Nonchaining Mode.)	Yes	Yes	0
2	Interrupt after Terminal Count. Value of 1 causes an interrupt to be generated after the terminal count for this descriptor is reached. Value of 0 disables interrupts from being generated.	Yes	Yes	0
3	Direction of Transfer. Value of 1 indicates transfers from local bus to PCI bus. Value of 0 indicates transfers from PCI bus to local bus.	Yes	Yes	0
31:4	Next Descriptor Address. Quad word aligned (bits [3:0] = 0000).	Yes	Yes	0

4.6.11 (DMACSR0; PCI:A8h, LOC:128h) DMA Channel 0 Command/Status Register

Table 4-72. (DMACSR0; PCI:A8h, LOC:128h) DMA Channel 0 Command/Status Register

Field	Description	Read	Write	Value after Reset
0	Channel 0 Enable. Value of 1 enables channel to transfer data. Value of 0 disables channel from starting a DMA transfer and if in the process of transferring data suspend transfer (pause).	Yes	Yes	0
1	Channel 0 Start. Writing 1 to this bit causes channel to start transferring data if the channel is enabled.	No	Yes/Set	0
2	Channel 0 Abort. Writing 1 to this bit causes channel to abort the current transfer. Channel enable bit must be cleared. Channel complete bit is set when the abort is complete.	No	Yes/Set	0
3	Clear Interrupt. Writing 1 to this bit clears Channel 0 interrupts.	No	Yes/Clr	0
4	Channel 0 Done. Value of 1 indicates channel’s transfer is complete. Value of 0 indicates channel’s transfer is not complete.	Yes	No	1
7:5	Reserved.	Yes	No	0

4.6.12 (DMACSR1; PCI:A9h, LOC:129h) DMA Channel 1 Command/Status Register

Table 4-73. (DMACSR1; PCI:A9h, LOC:129h) DMA Channel 1 Command/Status Register

Field	Description	Read	Write	Value after Reset
0	Channel 1 Enable. Value of 1 enables channel to transfer data. Value of 0 disables channel from starting a DMA transfer and if in the process of transferring data suspend transfer (Pause).	Yes	Yes	0
1	Channel 1 Start. Writing 1 to this bit causes channel to start transferring data if channel is enabled.	No	Yes/Set	0
2	Channel 1 Abort. Writing 1 to this bit causes channel to abort the current transfer. Channel enable bit must be cleared. Channel complete bit is set when the abort is complete.	No	Yes/Set	0
3	Clear Interrupt. Writing 1 to this bit clears Channel 1 interrupts.	No	Yes/Clr	0
4	Channel 1 Done. Value of 1 indicates this channel’s transfer is complete. Value of 0 indicates channel’s transfer is not complete.	Yes	No	1
7:5	Reserved.	Yes	No	0

4.6.13 (DMAARB; PCI:ACh, LOC:12Ch) DMA Arbitration Register

Same as Mode/Arbitration Register (MARBR) at address (PCI:08h)(LOC:88h) (refer to Table 4-35).

4.6.14 (DMATHR; PCI:B0h, LOC:130h) DMA Threshold Register

Table 4-74. (DMATHR; PCI:B0h, LOC:130h) DMA Threshold Register

Field	Description	Read	Write	Value after Reset
3:0	DMA Channel 0 PCI to Local Almost Full (C0PLAF). Number of full entries (divided by two, minus one) in FIFO before requesting local bus for writes. (C0PLAF+1) + (C0PLAE+1) should be ≤ FIFO Depth of 32.	Yes	Yes	0
7:4	DMA Channel 0 Local to PCI Almost Empty (C0LPAE). Number of empty entries (divided by two, minus one) in FIFO before requesting local bus for reads. (C0LPAF+1) + (C0LPAE+1) should be ≤ FIFO depth of 32.	Yes	Yes	0
11:8	DMA Channel 0 Local to PCI Almost Full (C0LPAF). Number of full entries (divided by two, minus one) in FIFO before requesting local bus for writes.	Yes	Yes	0
15:12	DMA Channel 0 PCI to Local Almost Empty (C0PLAE). Number of empty entries (divided by two, minus one) in FIFO before requesting local bus for reads.	Yes	Yes	0
19:16	DMA Channel 1 PCI to Local Almost Full (C1PLAF). Number of full entries (divided by two, minus one) in FIFO before requesting local bus for writes. (C1PLAF+1) + (C1PLAE+1) should be ≤ FIFO depth of 16.	Yes	Yes	0
23:20	DMA Channel 1 Local to PCI Almost Empty (C1LPAE). Number of empty entries (divided by two, minus one) in FIFO before requesting local bus for reads. (C1PLAF+1) + (C1PLAE+1) should be ≤ FIFO depth of 16.	Yes	Yes	0
27:24	DMA Channel 1 Local to PCI Almost Full (C1LPAF). Number of full entries (divided by two, minus one) in FIFO before requesting local bus for writes.	Yes	Yes	0
31:28	DMA Channel 1 PCI to Local Almost Empty (C1PLAE). Number of empty entries (divided by two, minus one) in FIFO before requesting local bus for reads.	Yes	Yes	0

Note: *If the number of entries needed is x, then value is one less than half the number of entries (that is, x/2 – 1).*

4.7 MESSAGING QUEUE REGISTERS

4.7.1 (OPLFIS; PCI:30h, LOC:B0) Outbound Post List FIFO Interrupt Status Register

Table 4-75. (OPLFIS; PCI:30h, LOC:B0) Outbound Post List FIFO Interrupt Status Register

Field	Description	Read	Write	Value after Reset
2:0	Reserved.	Yes	No	0
3	Outbound Post List FIFO Interrupt. This bit is set when the Outbound Post List FIFO is not empty. This bit is not effected by the interrupt mask bit.	Yes	No	0
31:4	Reserved.	Yes	No	0

4.7.2 (OPLFIM; PCI:34h, LOC:B4) Outbound Post List FIFO Interrupt Mask Register

Table 4-76. (OPLFIM; PCI:34h, LOC:B4) Outbound Post List FIFO Interrupt Mask Register

Field	Description	Read	Write	Value after Reset
2:0	Reserved.	Yes	No	0
3	Outbound Post List FIFO Interrupt Mask. Interrupt is masked when this bit is set.	Yes	Yes	1
31:4	Reserved.	Yes	No	0

4.7.3 (IQP; PCI:40h) Inbound Queue Port Register

Table 4-77. (IQP; PCI:40h) Inbound Queue Port Register

Field	Description	Read	Write	Value after Reset
31:0	Value written by PCI master is stored into the Inbound Post List FIFO, which is located in local memory at the address pointed to by the Queue Base Address + FIFO Size + Inbound Post Head Pointer. From the time of the PCI write until the local memory write and update of the Inbound Post Queue Head Pointer, further accesses to this register result in a retry. A local interrupt is generated when the Inbound Post List FIFO is not empty. When the port is read by the PCI master, the value is read from the Inbound Free List FIFO, which is located in local memory at the address pointed to by the Queue Base Address + Inbound Free Tail Pointer. If FIFO is empty, a value of FFFFFFFh is returned.	PCI	PCI	0

4.7.4 (OQP; PCI:44h) Outbound Queue Port Register

Table 4-78. (OQP; PCI:44h) Outbound Queue Port Register

Field	Description	Read	Write	Value after Reset
31:0	Value written by PCI master is stored into the Outbound Free List FIFO, which is located in local memory at the address pointed to by the Queue Base Address + (3*FIFO Size) + Outbound Free Head Pointer. From the time of the PCI write until the local memory write and update of the Outbound Free Head Pointer, further accesses to this register result in a retry. If FIFO fills up, a local LSERR interrupt is generated. When the port is read by the PCI master, the value is read from the Outbound Post List FIFO, which is located in local memory at the address pointed to by the Queue Base Address + (2*FIFO Size) + Outbound Post Tail Pointer. If FIFO is empty, a value of FFFFFFFh is returned. A PCI interrupt is generated if Outbound Post List FIFO is not empty.	PCI	PCI	0

4.7.5 (MQCR; PCI:C0h, LOC:140h) Messaging Queue Configuration Register

Table 4-79. (MQCR; PCI:C0h, LOC:140h) Messaging Queue Configuration Register

<table>
<tr><th>Field</th><th>Description</th><th>Read</th><th>Write</th><th>Value after Reset</th></tr>
<tr><td>0</td><td>Queue Enable. Value of 1 allows accesses to the Inbound and Outbound Queue ports. If cleared to 0, writes are accepted but ignored and reads return FFFFFFFF. All pointer initialization and frame allocation should be completed before enabling this bit.</td><td>Yes</td><td>Yes</td><td>0</td></tr>
<tr><td>5:1</td><td>Circular FIFO Size. Defines the size of one of the circular FIFOs. Each of the four FIFOs are the same size. Each FIFO entry is one 32 bit word.
FIFO Size Encoding
<table>
<tr><th>5:1</th><th>Max entries per FIFO</th><th>FIFO Size</th><th>Total FIFO Memory</th></tr>
<tr><td>00001</td><td>4K entries</td><td>16 KB</td><td>64 KB</td></tr>
<tr><td>00010</td><td>8K entries</td><td>32 KB</td><td>128 KB</td></tr>
<tr><td>00100</td><td>16K entries</td><td>64 KB</td><td>256 KB</td></tr>
<tr><td>01000</td><td>32K entries</td><td>128 KB</td><td>512 KB</td></tr>
<tr><td>10000</td><td>64K entries</td><td>256 KB</td><td>1 MB</td></tr>
</table></td><td>Yes</td><td>Yes</td><td>00001</td></tr>
<tr><td>31:6</td><td>Reserved.</td><td>Yes</td><td>No</td><td>0</td></tr>
</table>

4.7.6 (QBAR; PCI:C4h, LOC:144h) Queue Base Address Register

Table 4-80. (QBAR; PCI:C4h, LOC:144h) Queue Base Address Register

Field	Description	Read	Write	Value after Reset
19:0	Reserved.	Yes	No	0
31:20	Queue Base Address. Local memory base address of the Inbound and Outbound Queues (four contiguous and equal size FIFOs). Queue base address must be aligned on a 1 MB boundary.	Yes	Yes	0

4.7.7 (IFHPR; PCI:C8h, LOC:148h) Inbound Free Head Pointer Register

Table 4-81. (IFHPR; PCI:C8h, LOC:148h) Inbound Free Head Pointer Register

Field	Description	Read	Write	Value after Reset
1:0	Reserved.	Yes	No	0
19:2	Inbound Free Head Pointer. Local Memory Offset for Inbound Free List FIFO. This register is initialized as (0*FIFO Size) and maintained by the local CPU software.	Yes	Yes	0
31:20	Queue Base Address.	Yes	No	0

4.7.8 (IFTPR; PCI:CCh, LOC:14Ch) Inbound Free Tail Pointer Register

Table 4-82. (IFTPR; PCI:CCh, LOC:14Ch) Inbound Free Tail Pointer Register

Field	Description	Read	Write	Value after Reset
1:0	Reserved.	Yes	No	0
19:2	Inbound Free Tail Pointer. Local Memory Offset for Inbound Free List FIFO. This register is initialized as (0*FIFO Size) by the local CPU software. It is maintained by the MU hardware and is incremented modulo the FIFO size.	Yes	Yes	0
31:20	Queue Base Address.	Yes	No	0

4.7.9 (IPHPR; PCI:D0h, LOC:150h) Inbound Post Head Pointer Register

Table 4-83. (IPHPR; PCI:D0h, LOC:150h) Inbound Post Head Pointer Register

Field	Description	Read	Write	Value after Reset
1:0	Reserved.	Yes	No	0
19:2	Inbound Post Head Pointer. Local Memory Offset for Inbound Post List FIFO. This register is initialized as (1*FIFO Size) by the local CPU software. It is maintained by the MU hardware and is incremented modulo the FIFO size.	Yes	Yes	0
31:20	Queue Base Address.	Yes	No	0

4.7.10 (IPTPR; PCI:D4h, LOC:154h) Inbound Post Tail Pointer Register

Table 4-84. (IPTPR; PCI:D4h, LOC:154h) Inbound Post Tail Pointer Register

Field	Description	Read	Write	Value after Reset
1:0	Reserved.	Yes	No	0
19:2	Inbound Post Tail Pointer. Local Memory Offset for Inbound Post List FIFO. This register is initialized as (1*FIFO Size) and maintained by the local CPU software.	Yes	Yes	0
31:20	Queue Base Address.	Yes	No	0

4.7.11 (OFHPR; PCI:D8h, LOC:158h) Outbound Free Head Pointer Register

Table 4-85. (OFHPR; PCI:D8h, LOC:158h) Outbound Free Head Pointer Register

Field	Description	Read	Write	Value after Reset
1:0	Reserved.	Yes	No	0
19:2	Outbound Free Head Pointer. Local Memory Offset for Outbound Free List FIFO. This register is initialized as (3*FIFO Size) by the local CPU software and is maintained by the MU hardware and is incremented modulo the FIFO size.	Yes	Yes	0
31:20	Queue Base Address.	Yes	No	0

4.7.12 (OFTPR; PCI:DCh, LOC:15Ch) Outbound Free Tail Pointer Register

Table 4-86. (OFTPR; PCI:DCh, LOC:15Ch) Outbound Free Tail Pointer Register

Field	Description	Read	Write	Value after Reset
1:0	Reserved.	Yes	No	0
19:2	Outbound Free Tail Pointer. Local Memory Offset for Outbound Free List FIFO. This register is initialized as (3*FIFO Size) and maintained by the local CPU software.	Yes	Yes	0
31:20	Queue Base Address.	Yes	No	0

4.7.13 (OPHPR; PCI:E0h, LOC:160h) Outbound Post Head Pointer Register

Table 4-87. (OPHPR; PCI:E0h, LOC:160h) Outbound Post Head Pointer Register

Field	Description	Read	Write	Value after Reset
1:0	Reserved.	Yes	No	0
19:2	Outbound Post Head Pointer. Local Memory Offset for Outbound Post List FIFO. This register is initialized as (2*FIFO Size) and maintained by the local CPU software.	Yes	Yes	0
31:20	Queue Base Address.	Yes	No	0

4.7.14 (OPTPR; PCI:E4h, LOC:164h) Outbound Post Tail Pointer Register

Table 4-88. (OPTPR; PCI:E4h, LOC:164h) Outbound Post Tail Pointer Register

Field	Description	Read	Write	Value after Reset
1:0	Reserved.	Yes	No	0
19:2	Outbound Post Tail Pointer. Local Memory Offset for Outbound Post List FIFO. This register is initialized as (2*FIFO Size) and maintained by the MU hardware and is incremented modulo the FIFO size.	Yes	Yes	0
31:20	Queue Base Address.	Yes	No	0

4.7.15 (QSR; PCI:E8h, LOC:168h) Queue Status/Control Register

Table 4-89. (QSR; PCI:E8h, LOC:168h) Queue Status/Control Register

Field	Description	Read	Write	Value after Reset
0	$I_2$0 Decode Enable. When this bit is set, Mailbox registers 0 and 1 are replaced by the Inbound and Outbound Queue Port Registers and redefines Space 1 as PCI Base Address 0 to be accessed by PCIBAR0. Former Space 1 registers F0, F4, and F8 should be programmed to configure their shared $I_2$0 memory space, defined as PCI Base Address 0.	Yes	Yes	0
1	Queue Local Space Select. When this bit is set to 0, use Local Address Space 0 bus region descriptor for queue accesses. When this bit is set to 1, use Local Address Space 1 bus region descriptor for queue accesses.	Yes	Yes	0
2	Outbound Post List FIFO Prefetch Enable. When this bit is set, prefetching occurs from the Outbound Post List FIFO if not empty.	Yes	Yes	0
3	Inbound Free List FIFO Prefetch Enable. When this bit is set, prefetching occurs from the Inbound Free List FIFO if not empty.	Yes	Yes	0
4	Inbound Post List FIFO Interrupt Mask. Interrupt is masked when bit is set.	Yes	Yes	1
5	Inbound Post List FIFO Interrupt. This bit is set when the Inbound Post List FIFO is not empty. This bit is not affected by the Interrupt Mask bit.	Yes	No	0
6	Outbound Free List FIFO Overflow Interrupt Mask. Interrupt is masked when bit is set.	Yes	Yes	1
7	Outbound Free List FIFO Overflow Interrupt. This bit is set when the Outbound Free List FIFO becomes full. A local LSERR (NMI) interrupt is generated if enabled in the Interrupt Control/Status Register. Writing 1 clears the interrupt.	Yes	Yes/Clr	0
31:8	Unused.	Yes	No	0

5. PIN DESCRIPTION

5.1 PIN SUMMARY

The tables in this section describe PCI 9080 pins. Table 5-2 through Table 5-5 provide pin information common to all three local bus modes of operation (that is, C, J, and S modes):

- Power and Ground Pin Description
- Serial EEPROM Interface Pin Description
- PCI System Bus Interface Pin Description
- Local Bus Mode and Processor Independent Interface Pin Description

The pins in Table 5-6 through Table 5-8 correspond to the local bus modes of the PCI 9080:

- C Bus Mode Interface Pin Description (32-bit address/32-bit data, nonmultiplexed)
- J Bus Mode Interface Pin Description (32-bit address/32-bit data, multiplexed)
- S Bus Mode Interface Pin Description (32-bit address/16-bit data, multiplexed)

The following pins have internal pull-ups:

ADMODE, BIGEND#, BTERM#, DREQ[1:0]#, EEDO, EESEL, LINTi#, LLOCK#, LRESETi#, NB#, READYi#, S[2:0], SHORT#, and WAITI#.

The following pins have internal pull-downs:

BREQ, LHOLDA, TEST, and USERI.

For a visual view of the chip pin out, refer to Figure 7-3 in Section 7.3, “9080 Pin Out (S, J, and C Modes).”

Table 5-1 lists the abbreviations used in this section to represent the various pin types.

Table 5-1. Pin Type Abbreviations

Abbreviation	Pin Type
I/O	Input and output pin
I	Input pin only
O	Output pin only
TS	Tri-state pin
OC	Open collector pin
TP	Totem pole pin
STS	Sustained tri-state pin, driven high for one CLK before float
DTS	Driven tri-state pin, driven high for one-half CLK before float

All local bus internal pull-ups go through a 2 kΩ resistor. All local bus internal pull-downs go through a 100 kΩ resistor.

All local tri-state I/O pins should have external pull-ups (use 3 kΩ - 10 kΩ).

Unspecified pins are not connected.

Note: *For PCI Pins, DO NOT pull up or down any pins unless the PCI 9080 is being used in an embedded design. Refer to the* PCI Local Bus Specification*, v2.1, page 123.*

5.2 PIN OUT COMMON TO ALL BUS MODES

Table 5-2. Power and Ground Pin Description

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
TEST	Test	1	I	49	Test Pin. Pull high for test, low for normal operation. When TEST is pulled high, all outputs except USERO (pin 27) are placed in tri-state. USERO provides a NAND-TREE output when TEST is pulled high.
VDDL (Core)	Power (+5 V)	6	I	53, 68, 105, 144, 157, 167	Five volt power supply pins for core. Liberal .01 µF to .1 µF decoupling capacitors should be placed near the PCI 9080.
VDDH (PCI)	Power (+5 V or +3.3 V)	3	I	38, 60, 83	Power supply pins for PCI bus pins. Liberal .01 µF to .1 µF decoupling capacitors should be placed near the PCI 9080.
VDDH (Local)	Power (+5 V)	3	I	1, 124, 184	Power supply pins for local bus pins. Liberal .01 µF to .1 µF decoupling capacitors should be placed near the PCI 9080.
VSS	Ground	20	I	22, 37, 45, 52, 59, 67, 75, 82, 90, 98, 104, 114, 123, 134, 143, 156, 166, 183, 193, 208	Ground pins.

Table 5-3. Serial EEPROM Interface Pin Description

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
EECS	Serial EEPROM Chip Select	1	O TP 8 mA	176	Serial EEPROM chip select.
EEDI	Serial EEPROM Data IN	1	O TP 8 mA	172	Write data to serial EEPROM.
EEDO	Serial EEPROM Data OUT	1	I	171	Read data from serial EEPROM.
EESK	Serial Data Clock	1	O TP 8 mA	173	Serial EEPROM clock.
SHORT#	Load Short	1	I	174	When active low, only five 32-bit registers are loaded from the serial EEPROM. When active high, all local configuration registers are also loaded from serial EEPROM.
EESEL	Serial EEPROM Select	1	I	175	When high, use 93CS46 (1K bit) serial EEPROM. When low, use 93CS56 (2K bit) serial EEPROM.

Note: *Serial EEPROM interface operates at the core voltage (+5 V). PCI 9080 requires the use of a serial EEPROM that can operate up to 1 MHz.*

Table 5-4. PCI System Bus Interface Pin Description

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
AD[31:0]	Address and Data	32	I/O TS PCI	32-36, 39-44, 46-47, 76-81, 84-89, 91-97	All multiplexed on the same PCI pins. A bus transaction consists of an address phase followed by one or more data phases. PCI 9080 supports both read and write bursts.
C/BE[3:0]#	Bus Command and Byte Enables	4	I/O TS PCI	70-73	All multiplexed on the same PCI pins. During the address phase of a transaction, C/BE[3:0]# defines the bus command. During the data phase C/BE[3:0]# are used as Byte Enables. Refer to PCI spec for further detail if needed.
CLK	Clock	1	I	54	Provides timing for all transactions on PCI and is an input to every PCI device. PCI operates up to 33 MHz.
DEVSEL#	Device Select	1	I/O STS PCI	64	When actively driven, indicates the driving device has decoded its address as the target of the current access. As an input, indicates whether any device on the bus is selected.
FRAME#	Cycle Frame	1	I/O STS PCI	57	Driven by the current master to indicate the beginning and duration of an access. FRAME# is asserted to indicate a bus transaction is beginning. While FRAME# is asserted, data transfers continue. When FRAME# is negated, the transaction is in the final data phase.
GNT#	Grant	1	I	51	Indicates to the agent that access to the bus is granted. Every master has its own REQ# and GNT#.
IDSEL	Initialization Device Select	1	I	63	Used as a chip select during configuration read and write transactions.
INTA#	Interrupt A	1	O OC PCI	55	Used to request an interrupt.
IRDY#	Initiator Ready	1	I/O STS PCI	61	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	69	Indicates an atomic operation that may require multiple transactions to complete.
PAR	Parity	1	I/O TS PCI	74	Even parity across AD[31:0] and C/BE[3:0]#. Parity generation is required by all PCI agents. 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 completion of the current data phase.
PERR#	Parity Error	1	I/O STS PCI	65	Reporting of data parity errors during all PCI transactions, except during a Special Cycle.
REQ#	Request	1	O PCI	50	Indicates to the arbiter that this agent needs to use the bus. Every master has its own GNT# and REQ#.
RST#	Reset	1	I	56	Used to bring PCI-specific registers, sequencers and signals to a consistent state.
SERR#	Systems Error	1	O OC PCI	66	Reports address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic.
STOP#	Stop	1	I/O STS PCI	62	Indicates the current target is requesting the master to stop the current transaction.
TRDY#	Target Ready	1	I/O STS PCI	58	Indicates the ability of the target agent (selected device) to complete the current data phase of the transaction.

Table 5-5. Local Bus Mode and Processor Independent Interface Pin Description

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
ADMODE	Address Decode Mode	1	I	20	Determines how S[2:0] are used to access the PCI 9080 internal registers.
BIGEND#	Big Endian Select	1	I	48	Can be asserted during the local bus address phase of a Direct Master transfer or a 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 configuration registers.
BPCLKo	Buffered PCI Clock Output	1	O TP 8 mA	168	Provides a buffered PCI clock output.
BREQ	Bus Request	1	I	169	Asserted to indicate a local bus master requires the bus. If enabled through the PCI 9080 configuration registers, PCI 9080 releases the bus during a DMA transfer if this signal is asserted.
BREQo	Bus Request Out	1	O TP 8 mA	21	Asserted to indicate the PCI 9080 requires the bus to perform a direct PCI to local bus access while a Direct Master access is pending on the local bus. It can be used with external logic to generate backoff to a local bus master. Its operational parameters are set up through the PCI 9080 configuration registers.
BTERMo#	Burst Terminate Out	1	O DTS 8 mA	28	Asserted, along with READYo#, to request the break up of a burst and the start of a new address cycle (Abort only).
DACK[1:0]#	DMA Acknowledge Outputs	2	O TP 8 mA	25, 30	When a channel is programmed through the configuration registers to operate in demand mode, its DACK output indicates a DMA transfer is being executed. DACK0# corresponds to PCI 9080 DMA Ch 0 and DACK1# to DMA Ch 1.
DMPAF#	Direct Master Programmable Almost Full	1	O TP 8 mA	8	Direct Master write FIFO almost full status output. Programmable through a configuration register.
DP[3:0]	Data Parity	4	I/O TS 8 mA	12-15	Parity is even for each of up to 4 byte lanes on the local bus. Parity is checked for writes to the PCI 9080 or reads by the PCI 9080. Parity is generated for reads from the PCI 9080 or writes by the PCI 9080.
DREQ[1:0]#	DMA Request Inputs	2	I	24, 29	When a channel is programmed through the configuration registers to operate in demand mode, its DREQ input serves as a DMA request. DREQ0# corresponds to PCI 9080 DMA Ch 0 and DREQ1# to DMA Ch 1.
LDSHOLD	Direct Slave HOLD Request	1	O TP 8 mA	165	Asserted concurrent with LHOLD to indicate the PCI 9080 is requesting use of the Local Bus in order to perform a Direct Slave transfer.
LINTi#	Local Interrupt In	1	I	151	When asserted low, causes a PCI interrupt.
LINTo#	Local Interrupt Out	1	O TP 8 mA	152	Synchronous level output that remains asserted as long as an interrupt condition exists. If an edge level interrupt is required, disabling and then enabling local interrupts though the Interrupt Control/Status Register (refer to Table 4-58) creates an edge if an interrupt condition still exists or a new interrupt condition occurs.
LLOCKo#	Bus Lock	1	O TP 8 mA	7	Indicates an atomic operation for a Direct Slave PCI to local bus access may require multiple transactions to complete.
LRESETi#	Local Reset Input	1	I	150	Resets the local bus portion of the PCI 9080, the local configuration registers and the DMA configuration registers. Also causes local reset output to be asserted.

Table 5-5. Local Bus Mode and Processor Independent Interface Pin Description (continued)

Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LSERR#	System Error Interrupt Output	1	O TP 8 mA	23	Synchronous level output that is asserted when the PCI bus Target Abort or Master Abort status bit is set in the PCI Status configuration register. If an edge level interrupt is required, disabling and then enabling LSERR# interrupts through the interrupt/control status creates an edge if an interrupt condition still exists or a new interrupt condition occurs.
MODE[1:0]	Bus Mode	2	I	9, 10	Selects the bus operation mode of the PCI 9080: Bit 1 Bit 0 Bus Mode 0 0 C 0 1 J 1 0 S 1 1 Reserved
NB#	No Local Bus Initialization	1	I	26	Externally forces Local Init Done bit in the Init Control Register to 1. Init Done bit is also programmable through local bus configuration accesses. PCI 9080 issues RETRYs to all PCI accesses until Local Init Done bit is set. If this bit is not going to be set by a local processor, tie NB# low.
PCHK#	Data Parity Check	1	O TP 8 mA	16	Parity is checked for writes to the PCI 9080 or reads by the PCI 9080. Parity is checked for each byte lane with its byte enable asserted. Asserted in the clock cycle following the data being checked if a parity error is detected.
S[2:0]	Address Select	3	I	17-19	If ADMODE is high, internal PCI 9080 registers are selected when LA[31:29] match S[2:0]. If ADMODE is low, the internal PCI 9080 registers are selected when S0 is asserted low.
USERI	User Input	1	1	31	General purpose input that can be read from the PCI 9080 configuration registers.
USERO	User Output	1	O TP 12 mA	27	General purpose output controlled from the PCI 9080 configuration registers.
WAITI#	Wait Input	1	I	6	Can be asserted to cause the PCI 9080 to insert wait states for local direct master accesses to the PCI bus. Can be thought of as a ready input for direct master accesses.
WAITO#	Wait Out	1	O TS 8 mA	149	Indicates the PCI 9080 programmable wait state generator status. WAITO# is asserted when wait states are being caused by the internal wait state generator. Can be thought of as an output providing ready out status.

5.3 C BUS MODE PIN OUT

Table 5-6. C Bus Mode Interface Pin Description

C Mode Bus Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
ADS#	Address Strobe	1	I/O TS 12 mA	154	Indicates a valid address and the start of a new bus access. Asserted for the first clock of a bus access.
BLAST#	Burst Last	1	I/O TS 8 mA	155	Signal driven by the current local bus master to indicate the last transfer in a bus access.
BTERM#	Burst Terminate	1	I	146	For processors that burst up to four Lwords. If Bterm is disabled through the PCI 9080 configuration registers, PCI 9080 also bursts up to four Lwords. If enabled, PCI 9080 continues to burst until a BTERM# input is asserted. BTERM# is a ready input that breaks up a burst cycle and causes another address cycle to occur. Used in conjunction with the PCI 9080 programmable wait state generator.
DEN#	Data Enable	1	O TS 12 mA	145	Used in conjunction with DT/R# to provide control for data transceivers attached to the local bus.
DT/R#	Data Transmit/Receive	1	O TS 12 mA	138	Used in conjunction with DEN# to provide control for data transceivers attached to the local bus. When asserted, the signal indicates the PCI 9080 receives data.
LW/R#	Write/Read	1	I/O TS 12 mA	137	Asserted low for reads and high for writes.
LLOCK#	Bus Lock	1	I	153	Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9080 for direct local access to the PCI bus.
LA[31:2]	Address Bus	30	I/O TS 8 mA	136, 135, 133-125, 122-115, 113-106, 103-101	Carries the upper 30 bits of the physical address bus. During bursts, LA[31:2] increment to indicate successive data cycles.
LD[31:0]	Data Bus	32	I/O TS 8 mA	177-182, 185-192, 194-207, 2-5	Carries 32, 16, or 8 bit data quantities depending on bus width configuration.

Table 5-6. C Bus Mode Interface Pin Description (continued)

C Mode Bus Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LBE[3:0]#	Byte Enables	4	I/O TS 12 mA	139-142	Encoded, based on configured bus width, as follows: **32-bit bus**: For a 32-bit bus, the four byte enables indicate which of the four bytes are active during a data cycle: BE3# Byte Enable 3—LD[31:24] BE2# Byte Enable 2—LD[23:16] BE1# Byte Enable 1—LD[15:8] BE0# Byte Enable 0—LD[7:0] **16-bit bus:** For a 16-bit bus, BE3#, BE1# and BE0# are encoded to provide BHE#, LA1, and BLE#, respectively: BE3# Byte High Enable (BHE#)—LD[15:8] BE2# not used BE1# Address bit 1 (LA1) BE0# Byte Low Enable (BLE#)—LD[7:0] **8-bit bus:** For an 8-bit bus, BE1# and BE0# are encoded to provide LA1and LA0, respectively: BE3# not used BE2# not used BE1# Address bit 1 (LA1) BE0# Address bit 0 (LA0)
LCLK	Local Processor Clock	1	I	160	Local clock input.
LHOLD	Hold Request	1	O TP 8 mA	158	Asserted to request use of the local bus. The local bus arbiter asserts LHOLDA when control is granted.
LHOLDA	Hold Acknowledge	1	I	159	Asserted by the local bus arbiter when control is granted in response to LHOLD. The bus should not be granted to PCI 9080 unless requested by LHOLD.
LRESETo#	Local Bus Reset Out	1	O TP 8 mA	11	Asserted when the PCI 9080 chip is reset. Used to drive the RESET# input of the local processor.
READYi#	Ready In	1	I	147	When the PCI 9080 is a bus master, indicates that read data on the bus is valid or that a write data transfer is complete. Used in conjunction with the PCI 9080 programmable wait state generator.
READYo#	Ready Out	1	O DTS 8 mA	148	When a local bus access is made to the PCI 9080, indicates read data on the bus is valid or a write data transfer is complete. READYo# can be connected to READYi#.
EOT0#	End of Transfer for DMA Ch 0	1	I	163	Terminates the current DMA Ch 0 transfer.
EOT1#	End of Transfer for DMA Ch 1	1	I	164	Terminates the current DMA Ch 1 transfer.

5.4 J BUS MODE PIN OUT

Table 5-7. J Bus Mode Interface Pin Description

J Bus Mode Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
ALE	Address Latch Enable	1	O TS 8 mA	161	Asserted during the address phase and negated before the data phase.
ADS#	Address Strobe	1	I/O TS 12 mA	154	Indicates the valid address and the start of a new bus access. Asserted for the first clock of a bus access.
BLAST#	Burst Last	1	I/O TS 8 mA	155	Signal driven by the current local bus master to indicate the last transfer in a bus access.
BTERM#	Burst Terminate	1	I	146	For processors that burst up to four Lwords. If Bterm is disabled through the PCI 9080 configuration registers, PCI 9080 also bursts up to four Lwords. If enabled, PCI 9080 continues to burst until a BTERM# input is asserted. BTERM# is a ready input that breaks up a burst cycle and causes another address cycle to occur. Used in conjunction with the PCI 9080 programmable wait state generator.
DEN#	Data Enable	1	I/O TS 12 mA	145	As an input, DEN# must only be asserted during data phases. For processor systems in which ADS# is not asserted during the data phase, DEN# can be pulled high. As an output, DT/R# is used in conjunction with DEN# to provide control for data transceivers attached to the local bus.
DT/R#	Data Transmit/Receive	1	O TS 12 mA	138	Used in conjunction with DEN# to provide control for data transceivers attached to the local bus. When asserted the signal indicates the PCI 9080 receives data.
LW/R#	Write/Read	1	I/O TS 12 mA	137	Asserted low for reads and high for writes.
LABS[3:2]	Address Bus Burst	2	I/O TS 8 mA	162,163	Carries the word address of the 32 bit memory address. These bits are incremented during a burst access.
LAD[31:0]	Address/Data Bus	32	I/O TS 8 mA	136, 135, 133-125, 122-115, 113-106, 103-99	During the address phase, the bus carries the upper 30 bits of the physical address bus. During the data phase, the bus carries 32 bits of data.

Table 5-7. J Bus Mode Interface Pin Description (continued)

J Mode Bus Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LBE[3:0]#	Byte Enables	4	I/O TS 12 mA	139-142	Byte enables are encoded based on configured bus width as follows: **32-Bit Bus:** For a 32-bit bus, the four byte enables indicate which of the four bytes are active during a data cycle: BE3# Byte Enable 3—LAD[31:24] BE2# Byte Enable 2—LAD[23:16] BE1# Byte Enable 1—LAD[15:8] BE0# Byte Enable 0—LAD[7:0] **16-Bit Bus:** For a 16-bit bus, BE3#, BE1# and BE0# are encoded to provide BHE#, LA1, and BLE#, respectively: BE3# Byte High Enable (BHE#)—LAD[15:8] BE2# not used BE1# Address bit 1 (LA1) BE0# Byte Low Enable (BLE#)—LAD[7:0] **8-Bit Bus:** For an 8-bit bus, BE1# and BE0# are encoded to provide LA1and LA0, respectively: BE3# not used BE2# not used BE1# Address bit 1 (LA1) BE0# Address bit 0 (LA0)
LCLK	System Clock	1	I	160	Local clock input.
LHOLD	Hold Request	1	O TP 8 mA	158	Asserted to request use of the local bus. The local bus arbiter asserts LHOLDA when control is granted.
LHOLDA	Hold Acknowledge	1	I	159	Asserted by the local bus arbiter when control is granted in response to LHOLD. The bus should not be granted to PCI 9080 unless requested by LHOLD.
LLOCK#	Bus Lock	1	I	153	Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9080 for direct local access to the PCI bus.
LRESETo#	Local Bus Reset Out	1	O TP 8 mA	11	Asserted when the PCI 9080 chip is reset.
READYi#	Ready In	1	I	147	When the PCI 9080 is a bus master, READYi# is used to indicate read data on the bus is valid or a write data transfer is complete. READYi# is used in conjunction with the PCI 9080 programmable wait state generator.
READYo#	Ready Out	1	O DTS 8 mA	148	When a local bus access is made to the PCI 9080, indicates that read data on the bus is valid or that a write data transfer is complete. READYo# can be connected to READYi#.
EOT0#	End of Transfer for DMA Ch 0	1	I	4	Terminates the current DMA Ch 0 transfer.
EOT1#	End of Transfer for DMA Ch 1	1	I	5	Terminates the current DMA Ch 1 transfer.

5.5 S BUS MODE PIN OUT

Table 5-8. S Bus Mode Interface Pin Description

S Bus Mode Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
ALE	Address Latch Enable	1	O TS 8 mA	161	Asserted during the address phase and negated before the data phase.
AS#	Address Strobe	1	I/O TS 12 mA	154	Indicates valid address and the start of a new bus access. Asserted for the first clock of a bus access.
BLAST#	Burst Last	1	I/O TS 8 mA	155	Signal driven by the current local bus master to indicate the last transfer in a bus access.
BTERM#	Burst Terminate	1	I	146	For processors that burst up to eight words and do not use BTERM# input. If Bterm is disabled through the PCI 9080 configuration registers, PCI 9080 also bursts up to eight words. If enabled, PCI 9080 continues to burst until a BTERM# input is asserted. BTERM# breaks up a burst cycle and causes another address cycle to occur. Used in conjunction with the PCI 9080 programmable wait state generator.
DEN#	Data Enable	1	O TS 12 mA	145	Used in conjunction with DT/R# to provide control for data transceivers attached to the local bus.
DT/R#	Data Transmit/Receive	1	O TS 12 mA	138	Used in conjunction with DEN# to provide control for data transceivers attached to the local bus. When asserted, the signal indicates the PCI 9080 is receiving data.
LA[31:16]	Address Bus	16	I/O TS 8 mA	136, 135, 133-125, 122-118	Carries the upper 16 bits of the address.
LABS[3:1]	Address Bus Burst	3	I/O TS 8 mA	162-164	Carries the word address of the 32 bit memory address. These bits are incremented during a burst access.
LAD[15:1],D0	Address/Data Bus	16	I/O TS 8 mA	117-115, 113-106, 103-99	During the address phase, carries the lower physical address bits. During the data phase, carries 16 bits of data.
LBE[1:0]#	Byte Enables	2	I/O TS 12 mA	141,142	Indicate which of the two bytes are active during a data cycle.
LCLK	Local Clock	1	I	160	Local clock input. ***Note:*** *For i960®S processor systems, CLK2 input. i960®S processor's RESET# input must be connected to PCI 9080 LRESETo# output. This enables PCI 9080 to determine the phase of the 2x clock processor.*
LHOLD	Hold Request	1	O TP 8 mA	158	Asserted to request use of the local bus. The local bus arbiter asserts LHOLDA when control is granted.
LHOLDA	Hold Acknowledge	1	I	159	Asserted by the local bus arbiter when control is granted in response to LHOLD. The bus should not be granted to the PCI 9080 unless requested by LHOLD.

Table 5-8. S Bus Mode Interface Pin Description (continued)

S Bus Mode Symbol	Signal Name	Total Pins	Pin Type	Pin Number	Function
LLOCK#	Bus Lock	1	I	153	Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9080 for direct local access to the PCI bus.
LRESETo#	Local Bus Reset Out	1	O TP 8 mA	11	Asserted when the PCI 9080 chip is reset. ***Note:*** *For i960®S processors, this output must be used to drive the Reset Input of the i960®S processor. This enables PCI 9080 to determine the phase of the 2x clock processor.*
LW/R#	Write/Read	1	I/O TS 12 mA	137	Asserted low for reads and high for writes.
READYi#	Ready In	1	I	147	When the PCI 9080 is a bus master, READYi# is used to indicate read data on the bus is valid or a write data transfer is complete. READYi# is used in conjunction with the PCI 9080 programmable wait state generator.
READYo#	Ready Out	1	O DTS 8 mA	148	When a local bus access is made to the PCI 9080, indicates that read data on the bus is valid or that a write data transfer is complete. READYo# can be connected to READYi#.
EOT0#	End of Transfer for DMA Ch 0	1	I	4	Terminates the current DMA Ch 0 transfer.
EOT1#	End of Transfer for DMA Ch 1	1	I	5	Terminates the current DMA Ch 1 transfer.

6. ELECTRICAL SPECIFICATIONS

Table 6-1. Absolute Maximum Ratings

Specification	Maximum Rating
Storage Temperature	-65 °C to +150 °C
Ambient Temperature with Power Applied	-55 °C to +125 °C
Supply Voltage to Ground	-0.5 V to +7.0 V
Input Voltage (VIN)	VSS -0.5 V, VDD +0.5 V
Output Voltage (VOUT)	VSS -0.5 V, VDD +0.5 V

Table 6-2. Operating Ranges

Ambient Temperature	Supply Voltage (VDD)	Input Voltage (VIN)	
		Min	Max
0 °C to +70 °C	5 V ±5%	VSS	VDD

Table 6-3. Operating Ranges

Parameter	Test Conditions	Pin Type	Typical Value	Units
CIN	VIN = 2.0 V, f = 1 MHz	Input	5	pF
COUT	VOUT = 2.0 V, f = 1 MHz	Output	10	pF

Table 6-4. Electrical Characteristics Estimated over Operating Range

Parameter	Description	Test Conditions		Min	Max	Units
VOH	Output High Voltage	VDD = Min VIN = VIH or VIL	IOH = -4.0 mA	2.4	VDD	V
VOL	Output Low Voltage		IOL per Tables	VSS	0.4	V
VIH	Input High Level	—	—	2.0	5.5	V
VIL	Input Low Level	—	—	-0.5	0.3	V
VOH3	PCI 3.3 V Output High Voltage	VDD = Min	IOH = -4.0 mA	—	—	V
VOL3	PCI 3.3 V Output Low Voltage	VIN = VIH or VIL	IOL per Tables	—	—	V
VIH3	PCI 3.3 V Input High Level	—	—	—	—	V
VIL3	PCI 3.3 V Input Low Level	—	—	—	—	V
ILI	Input Leakage Current	VSS ≤ VIN ≤ VDD, VDD = Max		-10	+10	µA
IOZ	Tri-State Output Leakage Current	VDD = Max, VSS ≤ VIN ≤ VDD		-10	+10	µA
ICC	Power Supply Current	VDD=5.25 V, PCLK=LCLK=33 MHz		—	130	mA

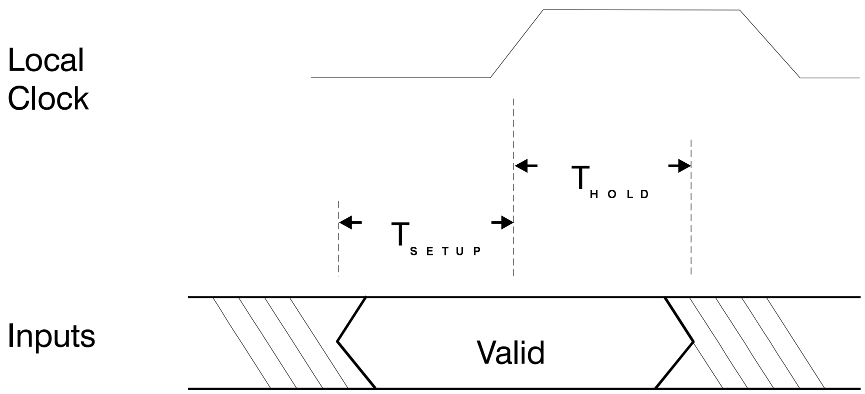


Figure 6-1. PCI 9080 Local Input Setup and Hold Waveform

Table 6-5. AC Electrical Characteristics (Local Inputs) Estimated over Operating Range

Signals (Synchronous Inputs) **C_L = 50 pF, VCC = 5.0 ± 5%**	**T_{SETUP}** **(nsec)**	**T_{HOLD} (nsec)** **(WORST CASE)**
ADS#	7	1.5
BLAST#	5	2.3
BTERM#	7	1.5
DP[3:0]	9	1.5
DREQ[1:0]#	6	1.5
LA[31:0]	10	1.5
LBE[3:0]#	8	1.5
LD[31:0]	9	1.5
LW/R#	9	1.5
READYi#	8	1.5
Input Clocks	**Min**	**Max**
Local Clock Input Frequency	0	40 MHz
PCI Clock Input Frequency	0	33 MHz

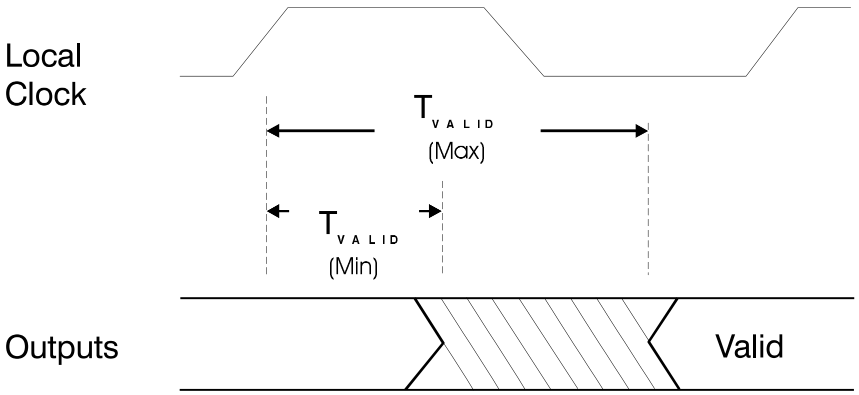


Figure 6-2. PCI 9080 Local Output Delay

Table 6-6. AC Electrical Characteristics (Local Outputs) Estimated over Operating Range

Signals (Synchronous Outputs) C_L = 50 pF, VCC = 5.0 ± 5%	Output T_{VALID} (Max)
ADS#	14.5
ALE (J and S modes) (address setup and hold relative to ALE negative edge)	—
BLAST#	16
BREQo	13
BTERMo#	15
DACK[1:0]#	14
DEN#	13
DMPAF#	—
DP[3:0]	20
DT/R#	16
LA[31:2]	20
LABS[3:1] (S mode), LABS[3:2] (J mode)	12
LBE[3:0]#	16
LD[31:0]	20
LHOLD	14
LINTo#	13
LSERR#	14
LW/R#	14.5
PCHK#	13
READYo#	14
WAITO#	20

Note: *All T_{VALID} (Mins) values are greater than 5 nsec.*

7. PACKAGE, SIGNAL, AND PIN OUT SPECS

7.1 PACKAGE MECHANICAL DIMENSIONS

For 208 PQFP, θJC = 5 °C/watt

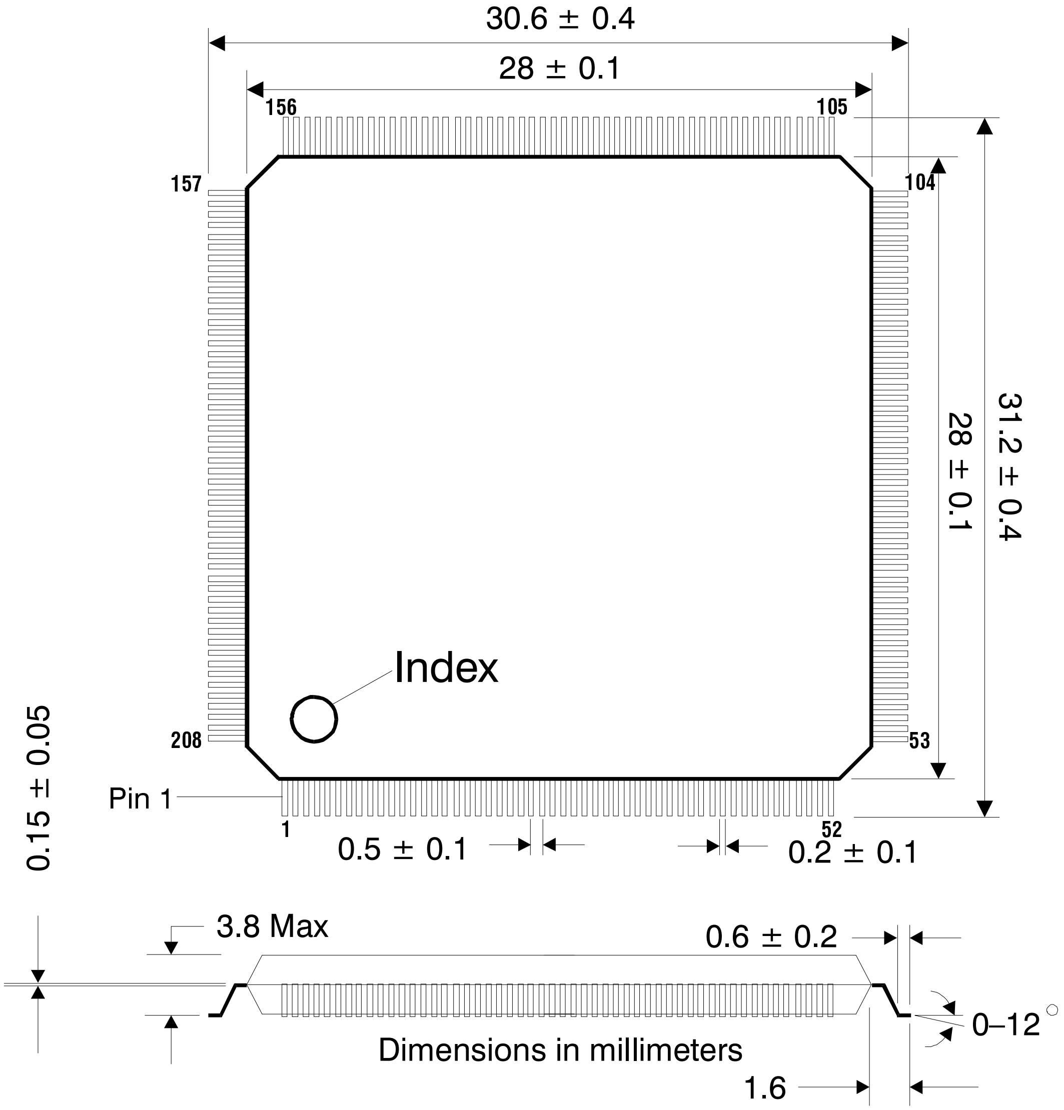


Figure 7-1. Package Mechanical Dimensions

7.2 TYPICAL PCI BUS MASTER ADAPTER

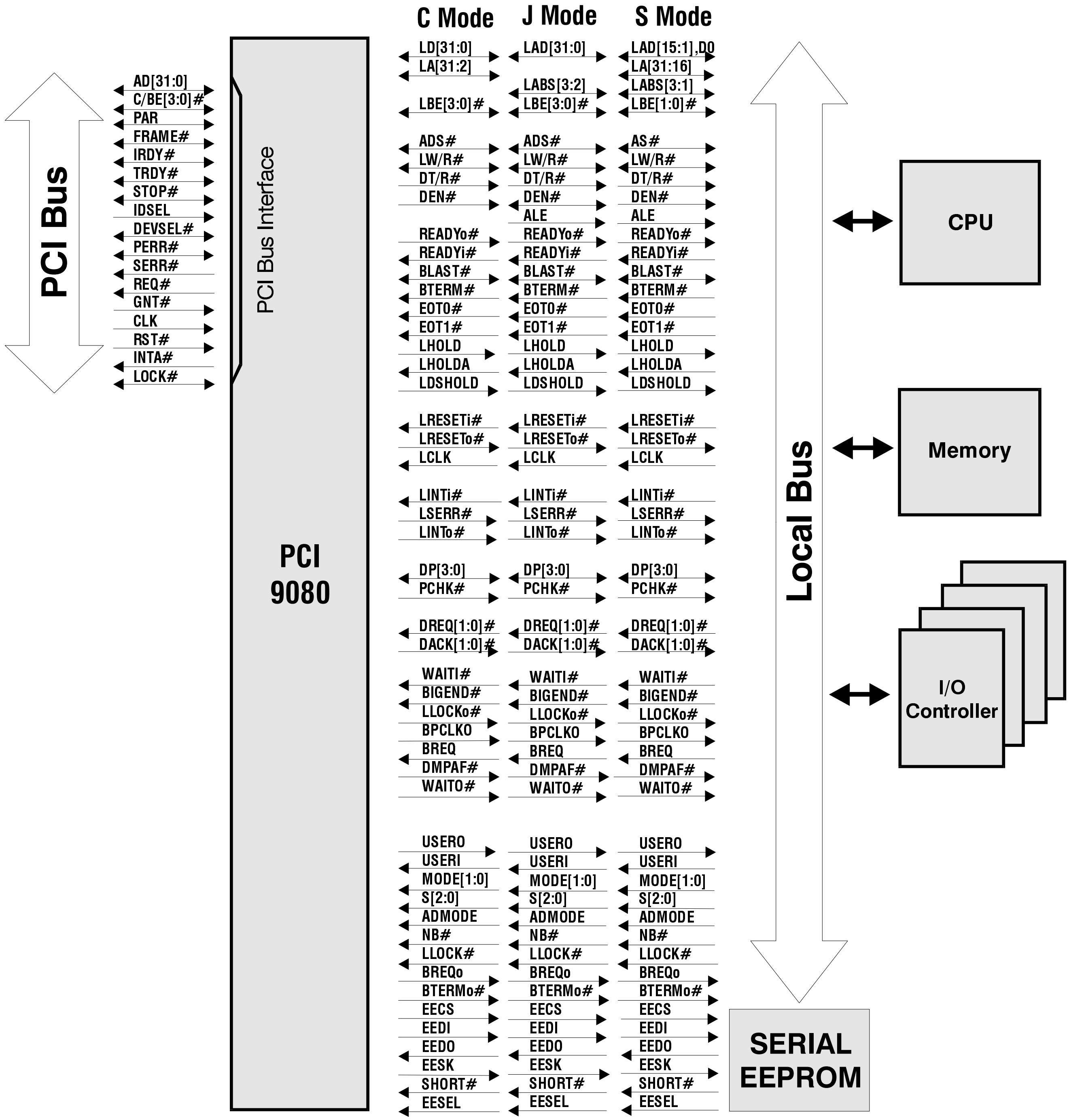
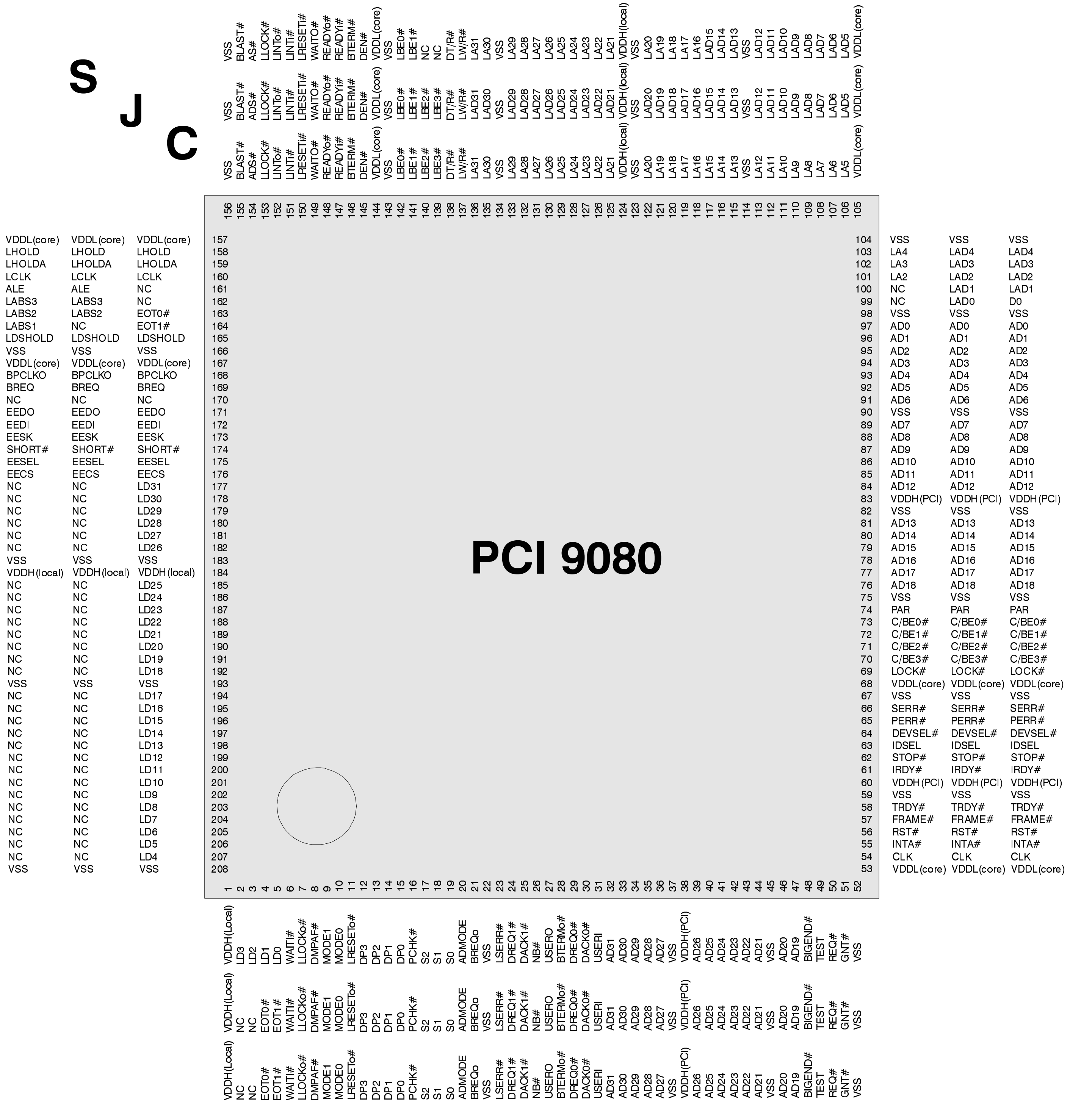


Figure 7-2. Typical PCI Bus Master Adapter

7.3 9080 PIN OUT (S, J, AND C MODES)

Refer to Section 5, “Pin Description,” for a complete description of each pin used in S, J, and C modes.



	S	J	C
1	VDDH(Local)	VDDH(Local)	VDDH(Local)
2	LD3	NC	NC
3	LD2	NC	NC
4	LD1	EOT0#	EOT0#
5	LD0	EOT1#	EOT1#
6	WAITI#	WAITI#	WAITI#
7	LLOCKo#	LLOCKo#	LLOCKo#
8	DMPAF#	DMPAF#	DMPAF#
9	MODE1	MODE1	MODE1
10	MODE0	MODE0	MODE0
11	LRESETo#	LRESETo#	LRESETo#
12	DP3	DP3	DP3
13	DP2	DP2	DP2
14	DP1	DP1	DP1
15	DP0	DP0	DP0
16	PCHK#	PCHK#	PCHK#
17	S2	S2	S2
18	S1	S1	S1
19	S0	S0	S0
20	ADMODE	ADMODE	ADMODE
21	BREQo	BREQo	BREQo
22	VSS	VSS	VSS
23	LSERR#	LSERR#	LSERR#
24	DREQ1#	DREQ1#	DREQ1#
25	DACK1#	DACK1#	DACK1#
26	NB#	NB#	NB#
27	USERO	USERO	USERO
28	BTERMo#	BTERMo#	BTERMo#
29	DREQ0#	DREQ0#	DREQ0#
30	DACK0#	DACK0#	DACK0#
31	USERI	USERI	USERI
32	AD31	AD31	AD31
33	AD30	AD30	AD30
34	AD29	AD29	AD29
35	AD28	AD28	AD28
36	AD27	AD27	AD27
37	VSS	VSS	VSS
38	VDDH(PCI)	VDDH(PCI)	VDDH(PCI)
39	AD26	AD26	AD26
40	AD25	AD25	AD25
41	AD24	AD24	AD24
42	AD23	AD23	AD23
43	AD22	AD22	AD22
44	AD21	AD21	AD21
45	VSS	VSS	VSS
46	AD20	AD20	AD20
47	AD19	AD19	AD19
48	BIGEND#	BIGEND#	BIGEND#
49	TEST	TEST	TEST
50	REQ#	REQ#	REQ#
51	GNT#	GNT#	GNT#
52	VSS	VSS	VSS
53	VDDL(core)	VDDL(core)	VDDL(core)
54	CLK	CLK	CLK
55	INTA#	INTA#	INTA#
56	RST#	RST#	RST#
57	FRAME#	FRAME#	FRAME#
58	TRDY#	TRDY#	TRDY#
59	VSS	VSS	VSS
60	VDDH(PCI)	VDDH(PCI)	VDDH(PCI)
61	IRDY#	IRDY#	IRDY#
62	STOP#	STOP#	STOP#
63	IDSEL	IDSEL	IDSEL
64	DEVSEL#	DEVSEL#	DEVSEL#
65	PERR#	PERR#	PERR#
66	SERR#	SERR#	SERR#
67	VSS	VSS	VSS
68	VDDL(core)	VDDL(core)	VDDL(core)
69	LOCK#	LOCK#	LOCK#
70	C/BE3#	C/BE3#	C/BE3#
71	C/BE2#	C/BE2#	C/BE2#
72	C/BE1#	C/BE1#	C/BE1#
73	C/BE0#	C/BE0#	C/BE0#
74	PAR	PAR	PAR
75	VSS	VSS	VSS
76	AD18	AD18	AD18
77	AD17	AD17	AD17
78	AD16	AD16	AD16
79	AD15	AD15	AD15
80	AD14	AD14	AD14
81	AD13	AD13	AD13
82	VSS	VSS	VSS
83	VDDH(PCI)	VDDH(PCI)	VDDH(PCI)
84	AD12	AD12	AD12
85	AD11	AD11	AD11
86	AD10	AD10	AD10
87	AD9	AD9	AD9
88	AD8	AD8	AD8
89	AD7	AD7	AD7
90	VSS	VSS	VSS
91	AD6	AD6	AD6
92	AD5	AD5	AD5
93	AD4	AD4	AD4
94	AD3	AD3	AD3
95	AD2	AD2	AD2
96	AD1	AD1	AD1
97	AD0	AD0	AD0
98	VSS	VSS	VSS
99	NC	LAD0	D0
100	NC	LAD1	LAD1
101	LA2	LAD2	LAD2
102	LA3	LAD3	LAD3
103	LA4	LAD4	LAD4
104	VSS	VSS	VSS
105	VDDL(core)	VDDL(core)	VDDL(core)
106	LAD5	LAD5	LA5
107	LAD6	LAD6	LA6
108	LAD7	LAD7	LA7
109	LAD8	LAD8	LA8
110	LAD9	LAD9	LA9
111	LAD10	LAD10	LA10
112	LAD11	LAD11	LA11
113	LAD12	LAD12	LA12
114	VSS	VSS	VSS
115	LAD13	LAD13	LA13
116	LAD14	LAD14	LA14
117	LAD15	LAD15	LA15
118	LA16	LAD16	LA16
119	LA17	LAD17	LA17
120	LA18	LAD18	LA18
121	LA19	LAD19	LA19
122	LA20	LAD20	LA20
123	VSS	VSS	VSS
124	VDDH(local)	VDDH(local)	VDDH(local)
125	LA21	LAD21	LA21
126	LA22	LAD22	LA22
127	LA23	LAD23	LA23
128	LA24	LAD24	LA24
129	LA25	LAD25	LA25
130	LA26	LAD26	LA26
131	LA27	LAD27	LA27
132	LA28	LAD28	LA28
133	LA29	LAD29	LA29
134	VSS	VSS	VSS
135	LA30	LAD30	LA30
136	LA31	LAD31	LA31
137	LW/R#	LW/R#	LW/R#
138	DT/R#	DT/R#	DT/R#
139	NC	LBE3#	LBE3#
140	NC	LBE2#	LBE2#
141	LBE1#	LBE1#	LBE1#
142	LBE0#	LBE0#	LBE0#
143	VSS	VSS	VSS
144	VDDL(core)	VDDL(core)	VDDL(core)
145	DEN#	DEN#	DEN#
146	BTERM#	BTERM#	BTERM#
147	READYi#	READYi#	READYi#
148	READYo#	READYo#	READYo#
149	WAITO#	WAITO#	WAITO#
150	LRESETi#	LRESETi#	LRESETi#
151	LINTi#	LINTi#	LINTi#
152	LINTo#	LINTo#	LINTo#
153	LLOCK#	LLOCK#	LLOCK#
154	AS#	ADS#	ADS#
155	BLAST#	BLAST#	BLAST#
156	VSS	VSS	VSS
157	VDDL(core)	VDDL(core)	VDDL(core)
158	LHOLD	LHOLD	LHOLD
159	LHOLDA	LHOLDA	LHOLDA
160	LCLK	LCLK	LCLK
161	ALE	ALE	NC
162	LABS3	LABS3	NC
163	LABS2	LABS2	EOT0#
164	LABS1	NC	EOT1#
165	LDSHOLD	LDSHOLD	LDSHOLD
166	VSS	VSS	VSS
167	VDDL(core)	VDDL(core)	VDDL(core)
168	BPCLKO	BPCLKO	BPCLKO
169	BREQ	BREQ	BREQ
170	NC	NC	NC
171	EEDO	EEDO	EEDO
172	EEDI	EEDI	EEDI
173	EESK	EESK	EESK
174	SHORT#	SHORT#	SHORT#
175	EESEL	EESEL	EESEL
176	EECS	EECS	EECS
177	NC	NC	LD31
178	NC	NC	LD30
179	NC	NC	LD29
180	NC	NC	LD28
181	NC	NC	LD27
182	NC	NC	LD26
183	VSS	VSS	VSS
184	VDDH(local)	VDDH(local)	VDDH(local)
185	NC	NC	LD25
186	NC	NC	LD24
187	NC	NC	LD23
188	NC	NC	LD22
189	NC	NC	LD21
190	NC	NC	LD20
191	NC	NC	LD19
192	NC	NC	LD18
193	VSS	VSS	VSS
194	NC	NC	LD17
195	NC	NC	LD16
196	NC	NC	LD15
197	NC	NC	LD14
198	NC	NC	LD13
199	NC	NC	LD12
200	NC	NC	LD11
201	NC	NC	LD10
202	NC	NC	LD9
203	NC	NC	LD8
204	NC	NC	LD7
205	NC	NC	LD6
206	NC	NC	LD5
207	NC	NC	LD4
208	VSS	VSS	VSS

Figure 7-3. PCI 9080 Pin Out (S, J, and C Modes)

8. TIMING DIAGRAMS

The PCI 9080 operates in three modes, selected through mode pins, corresponding to three bus types—C, J, and S. Timing Diagrams are provided for the three operating modes. For some functions, a timing diagram may only be provided for one mode of operation. Even though a different mode is used, the timing diagram can be used to determine functionality.

8.1 LIST OF TIMING DIAGRAMS

Initialization

C Mode Direct Slave

C Mode Direct Master

C Mode DMA

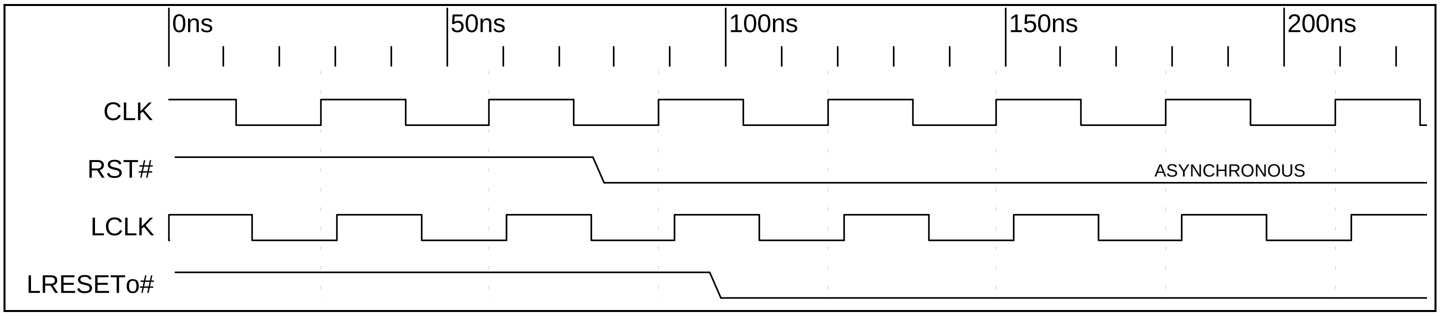
J Mode Direct Slave

J Mode Direct Master

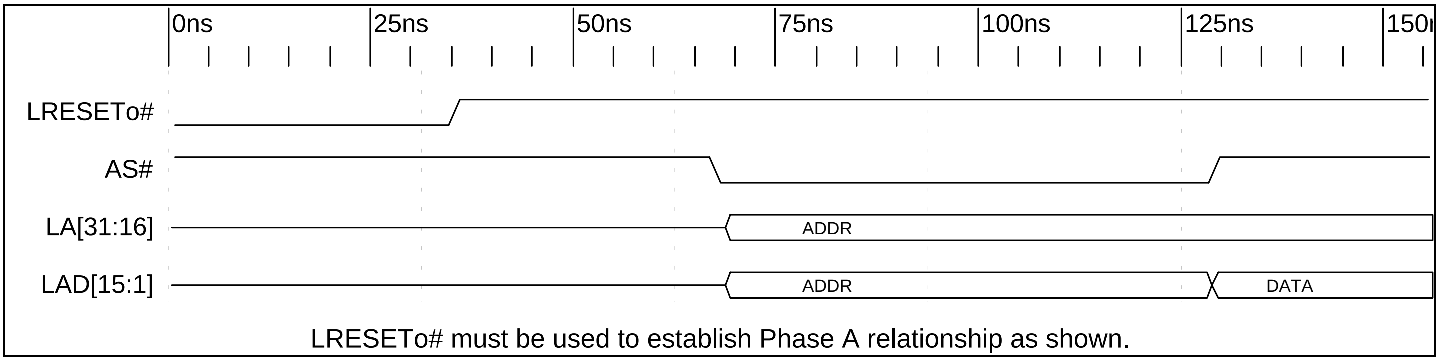
J Mode DMA

S Mode

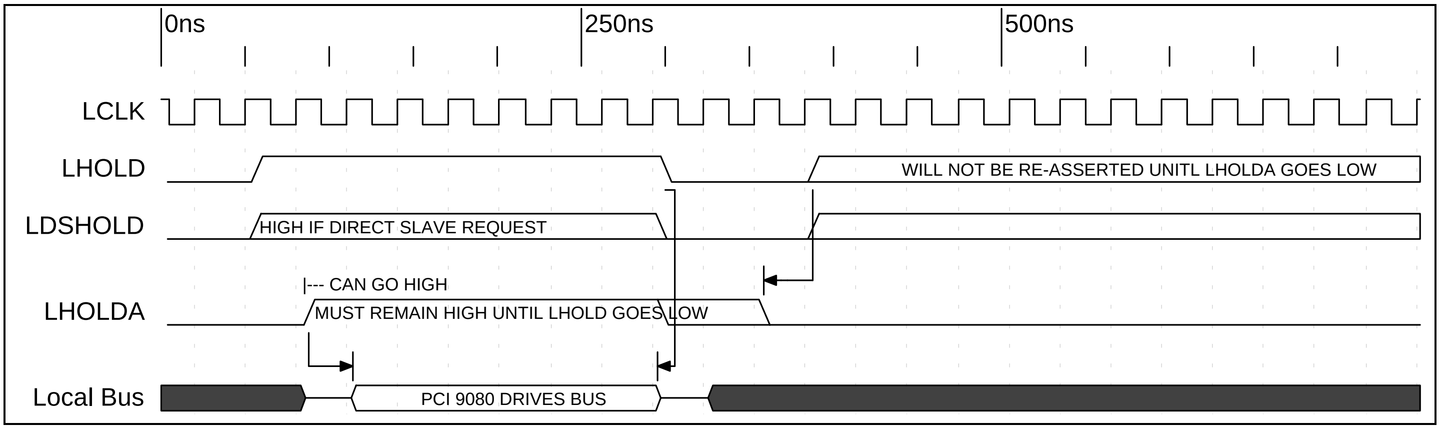
8.2 INITIALIZATION



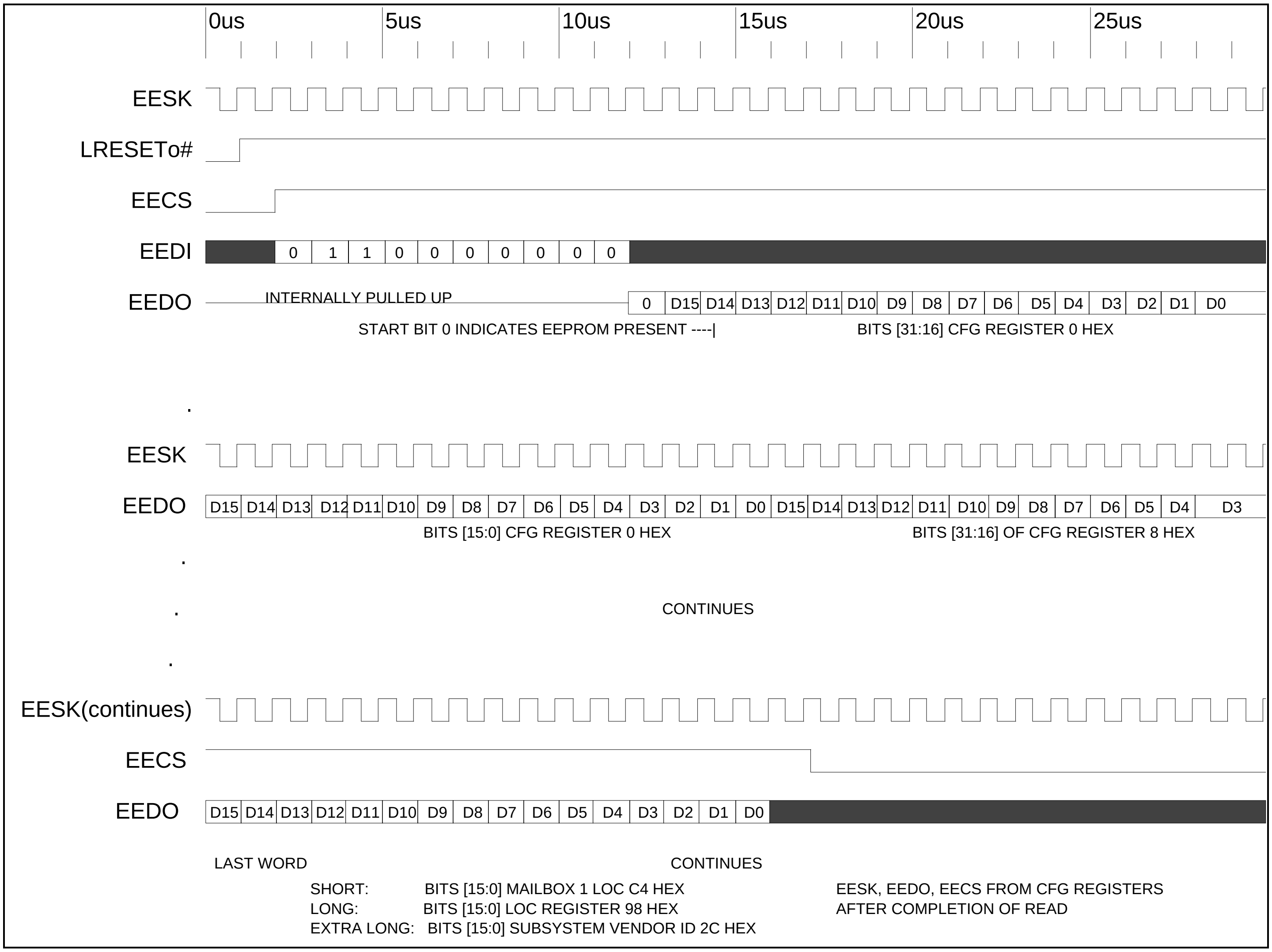
Timing Diagram 8-1. (C, J Modes) PCI RST# Asserting Local Output LRESETo#



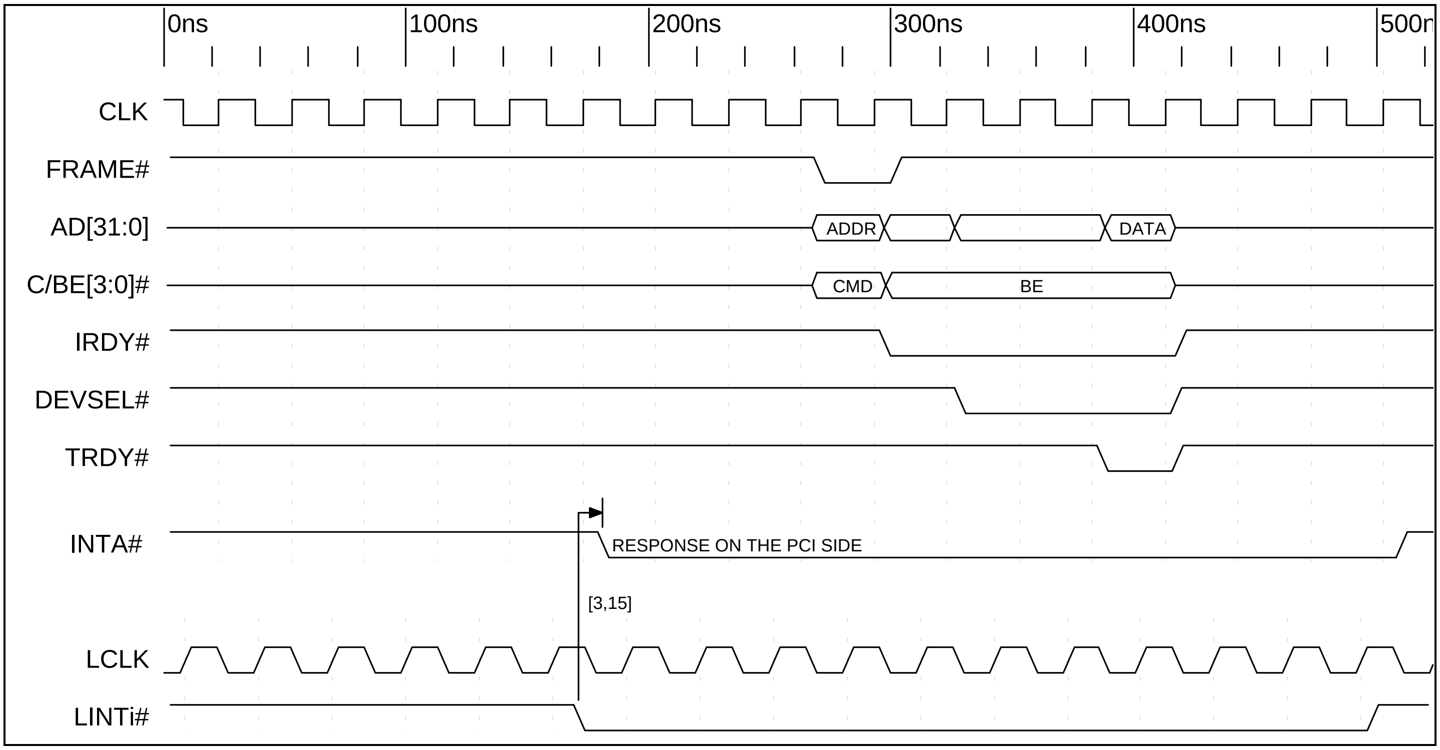
Timing Diagram 8-2. (S Mode) Two Phase Clock Synchronization Using LRESETo#


0ns
250ns
500ns
LCLK
LHOLD
WILL NOT BE RE-ASSERTED UNITL LHOLDA GOES LOW
LDSHOLD
HIGH IF DIRECT SLAVE REQUEST
|--- CAN GO HIGH
LHOLDA
MUST REMAIN HIGH UNTIL LHOLD GOES LOW
Local Bus
PCI 9080 DRIVES BUS

Timing Diagram 8-3. PCI 9080 Local Bus Arbitration


0us
5us
10us
15us
20us
25us
EESK
LRESETo#
EECS
EEDI
0 1 1 0 0 0 0 0 0 0
EEDO
INTERNALLY PULLED UP
0 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
START BIT 0 INDICATES EEPROM PRESENT ----|
BITS [31:16] CFG REGISTER 0 HEX
EESK
EEDO
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3
BITS [15:0] CFG REGISTER 0 HEX
BITS [31:16] OF CFG REGISTER 8 HEX
CONTINUES
EESK(continues)
EECS
EEDO
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
LAST WORD
CONTINUES
SHORT: BITS [15:0] MAILBOX 1 LOC C4 HEX
LONG: BITS [15:0] LOC REGISTER 98 HEX
EXTRA LONG: BITS [15:0] SUBSYSTEM VENDOR ID 2C HEX
EESK, EEDO, EECS FROM CFG REGISTERS
AFTER COMPLETION OF READ

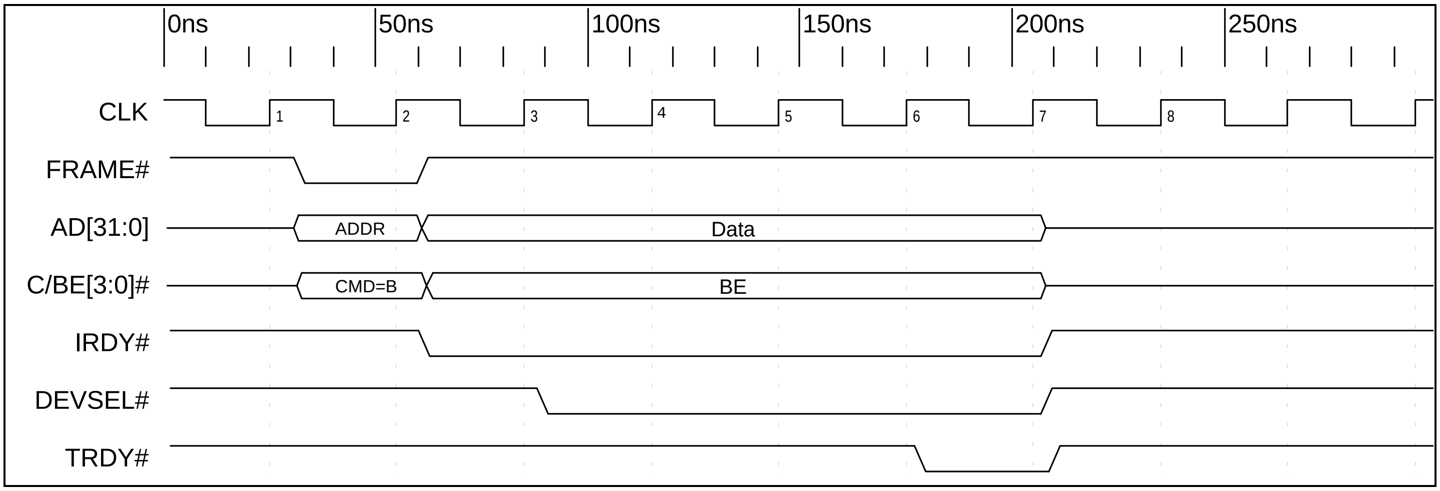
Timing Diagram 8-4. PCI 9080 1K Serial EEPROM PCI Initialization



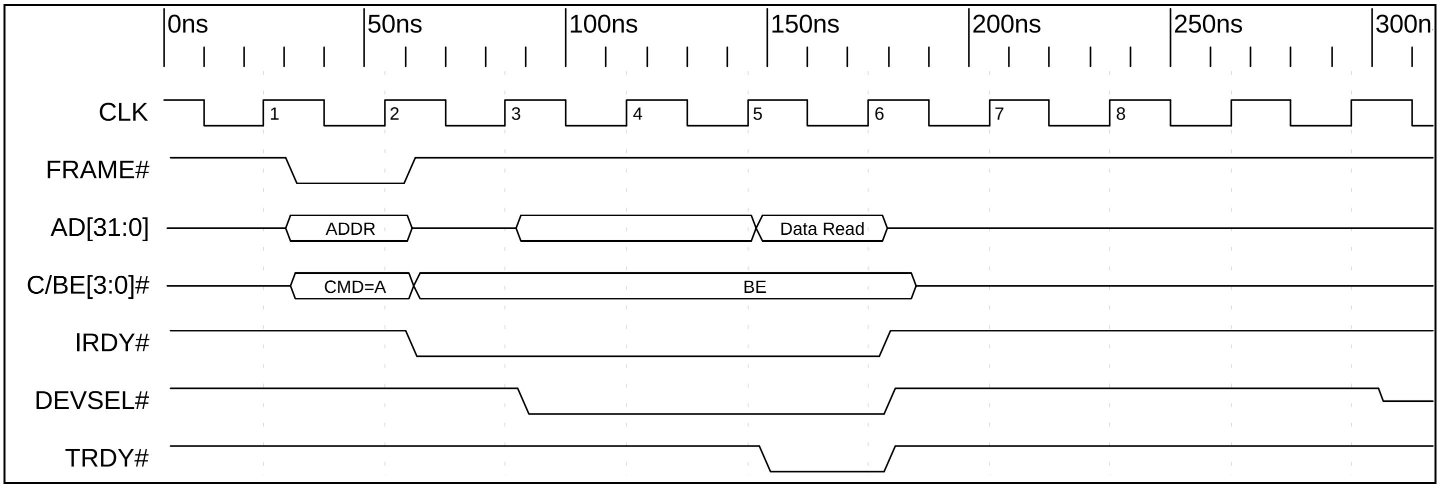
Timing Diagram 8-5. Local Interrupt (LINTi#) Input Asserting PCI Output INTA#

8.3 C MODE

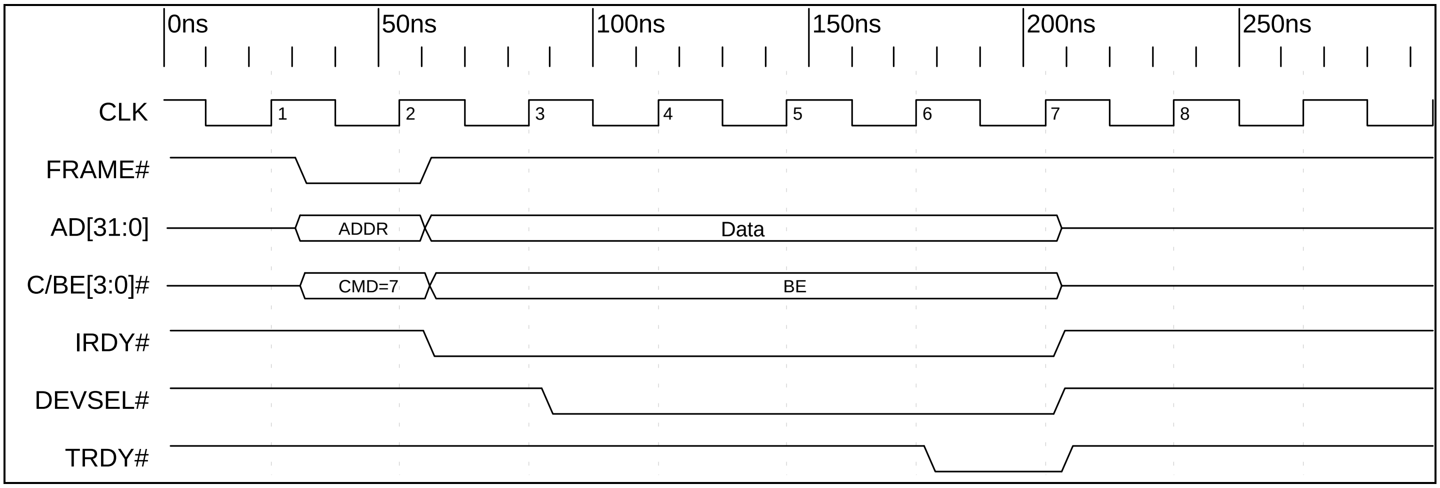
8.3.1 C Mode Direct Slave



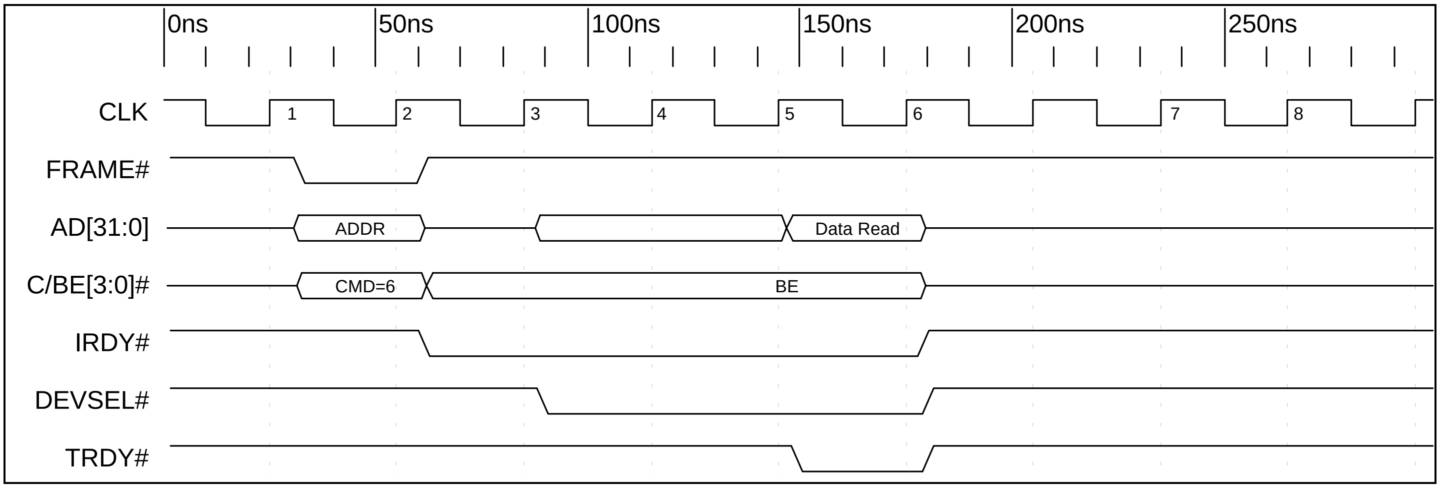
Timing Diagram 8-6. (C Mode) PCI Configuration Write to PCI 9080 PCI Configuration Register



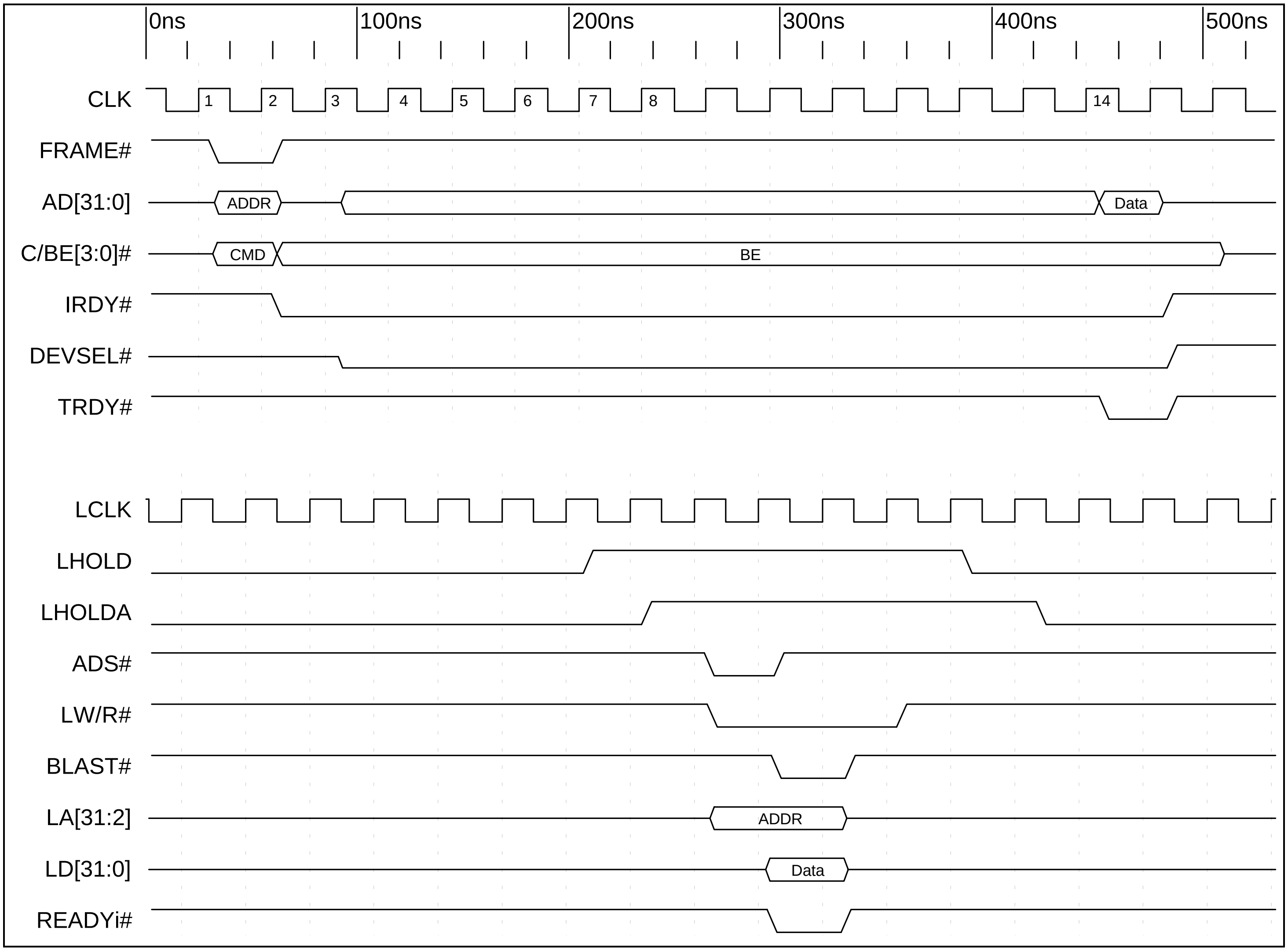
Timing Diagram 8-7. (C Mode) PCI Configuration Read to PCI 9080 PCI Configuration Register



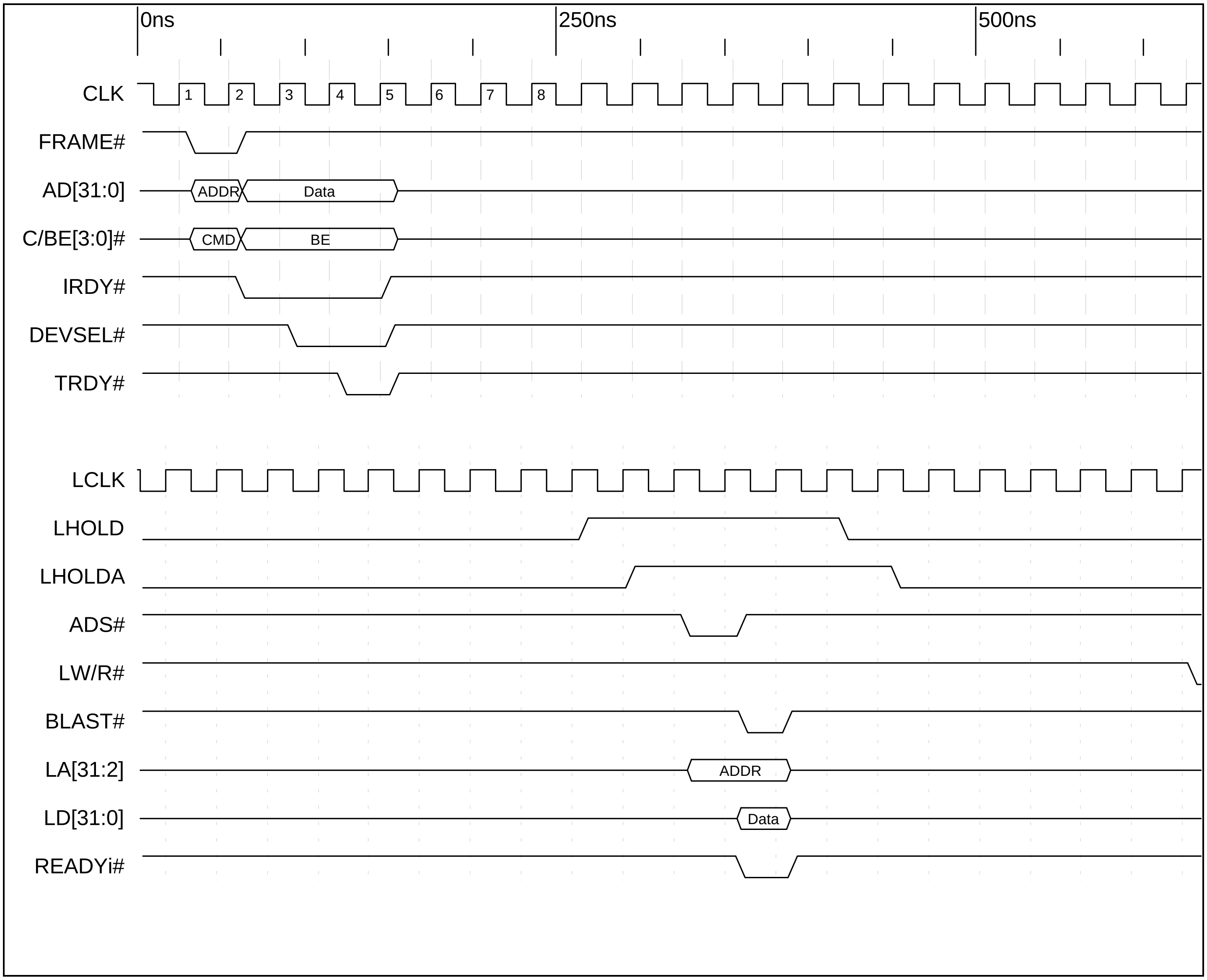
Timing Diagram 8-8. (C Mode) PCI Configuration Write to PCI 9080 Local Configuration Register



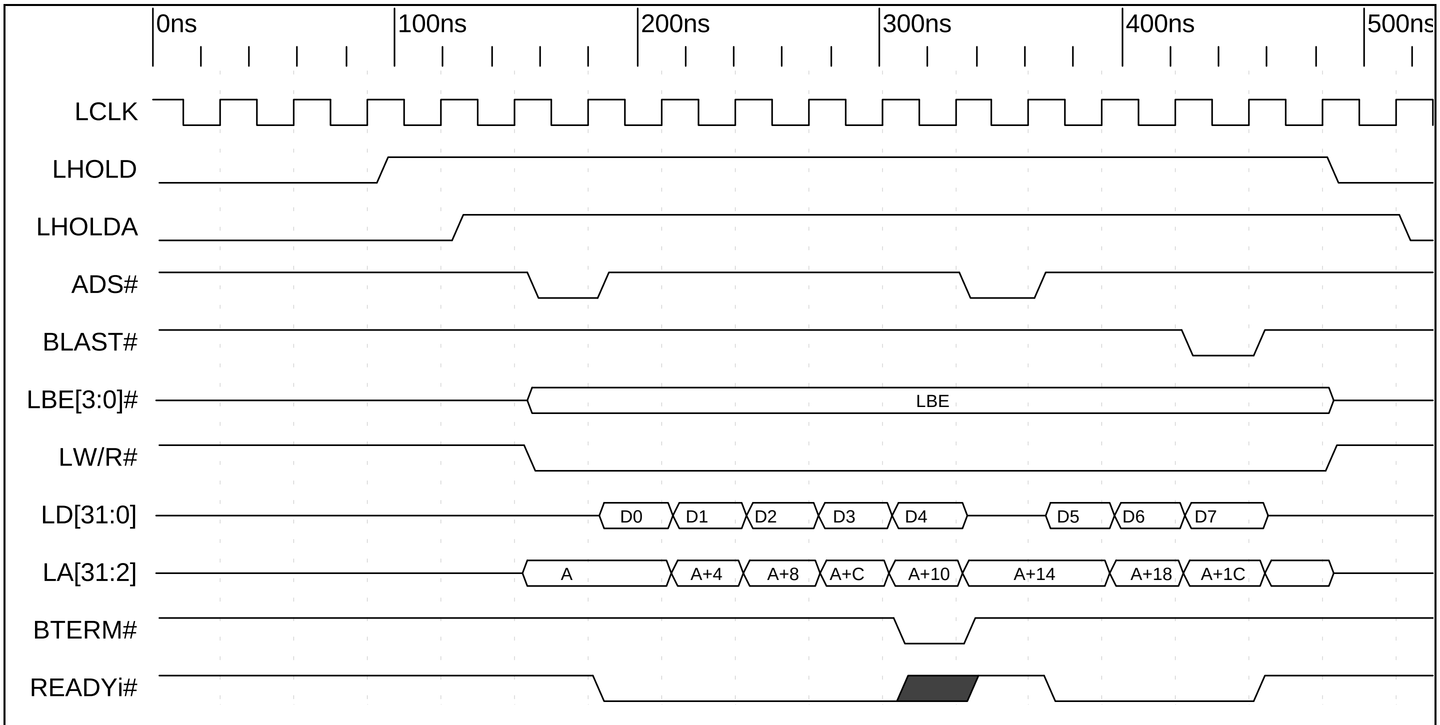
Timing Diagram 8-9. (C Mode) PCI Configuration Read to PCI 9080 Local Configuration Register



Timing Diagram 8-10. (C Mode) Direct Slave Single Cycle Read



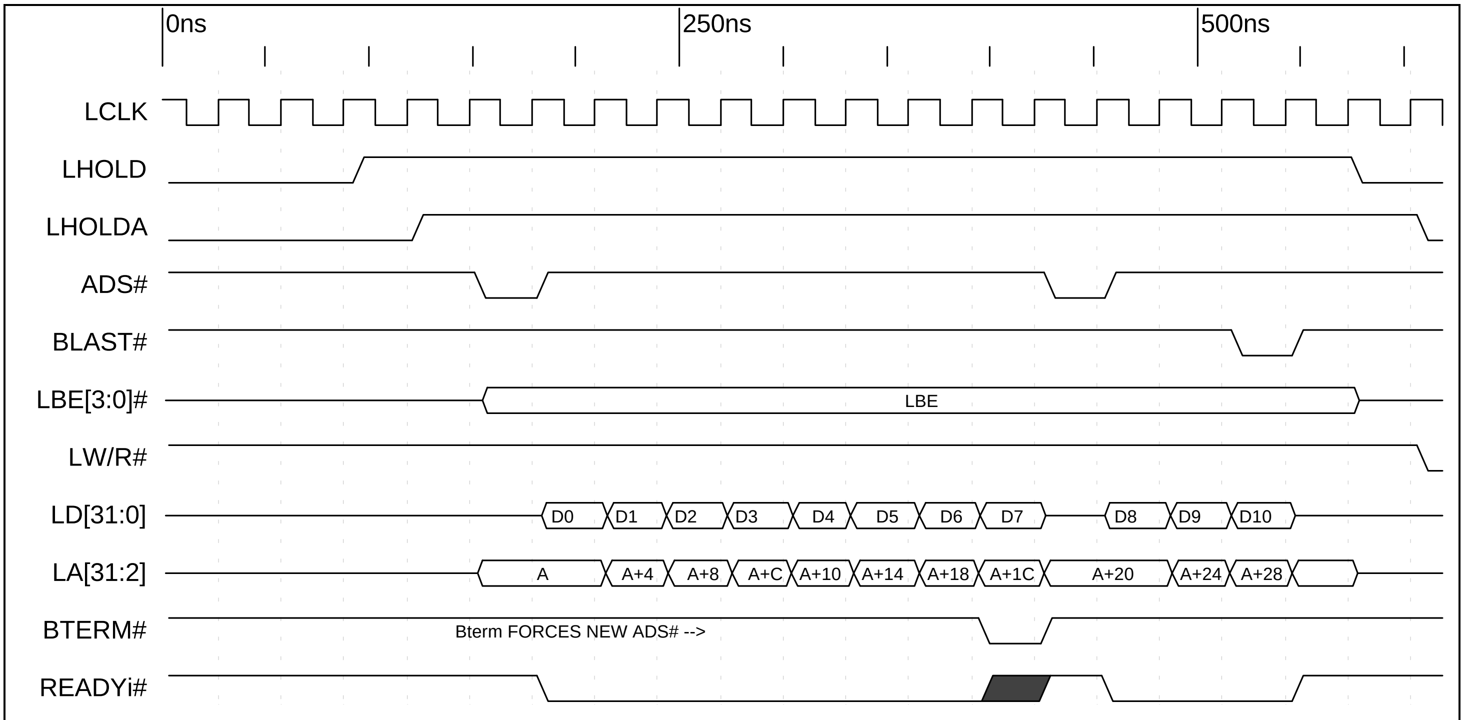
Timing Diagram 8-11. (C Mode) Direct Slave Single Cycle Write



Eight Lword burst, no wait states, burst enabled, Bterm enabled, 32 bit local bus.

Note: *If Bterm is disabled, a new ADS cycle starts every quad Lword boundary.*

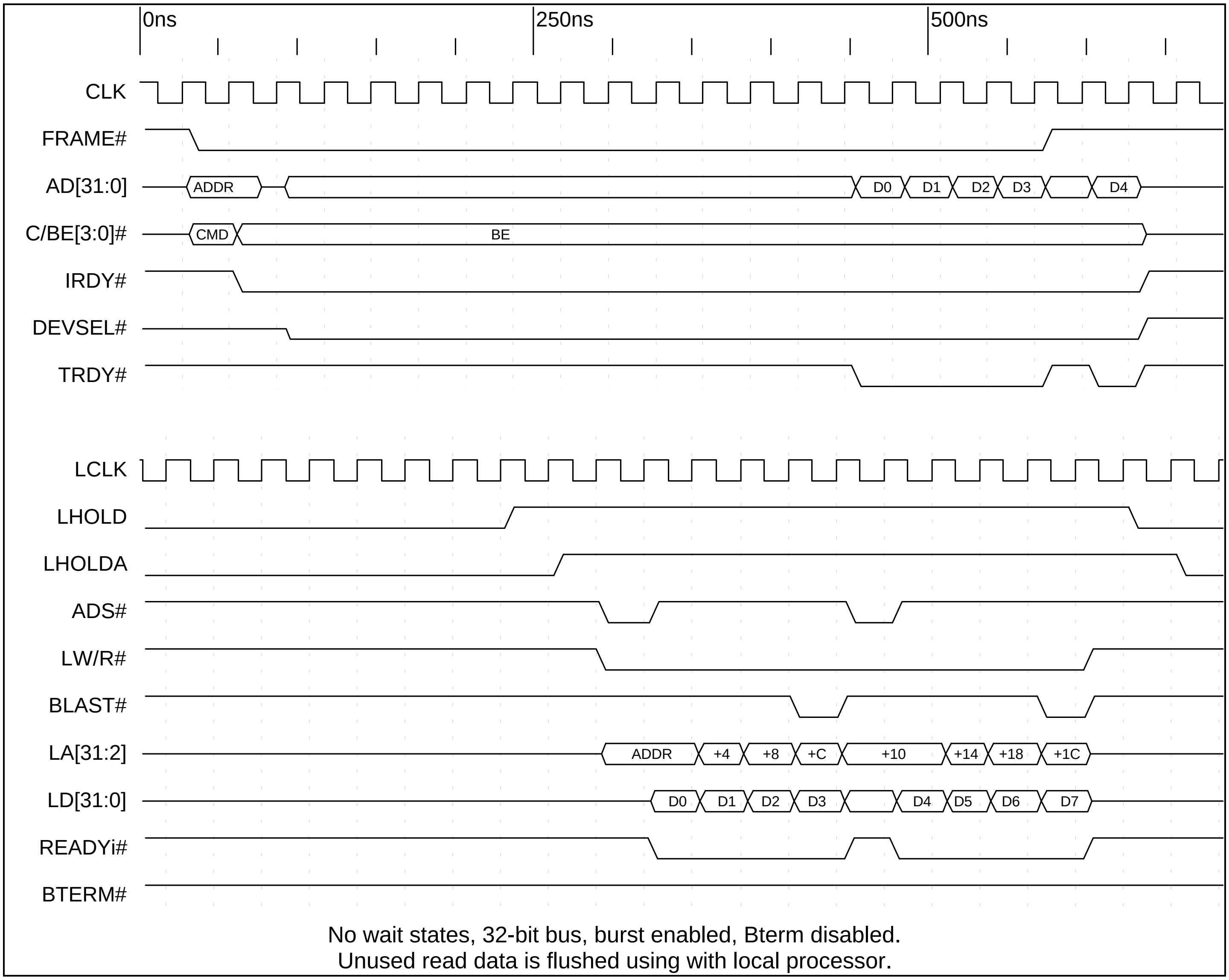
Timing Diagram 8-12. (C Mode) PCI 9080 DMA or Direct Slave Burst Read from Local Bus, Bterm Enabled



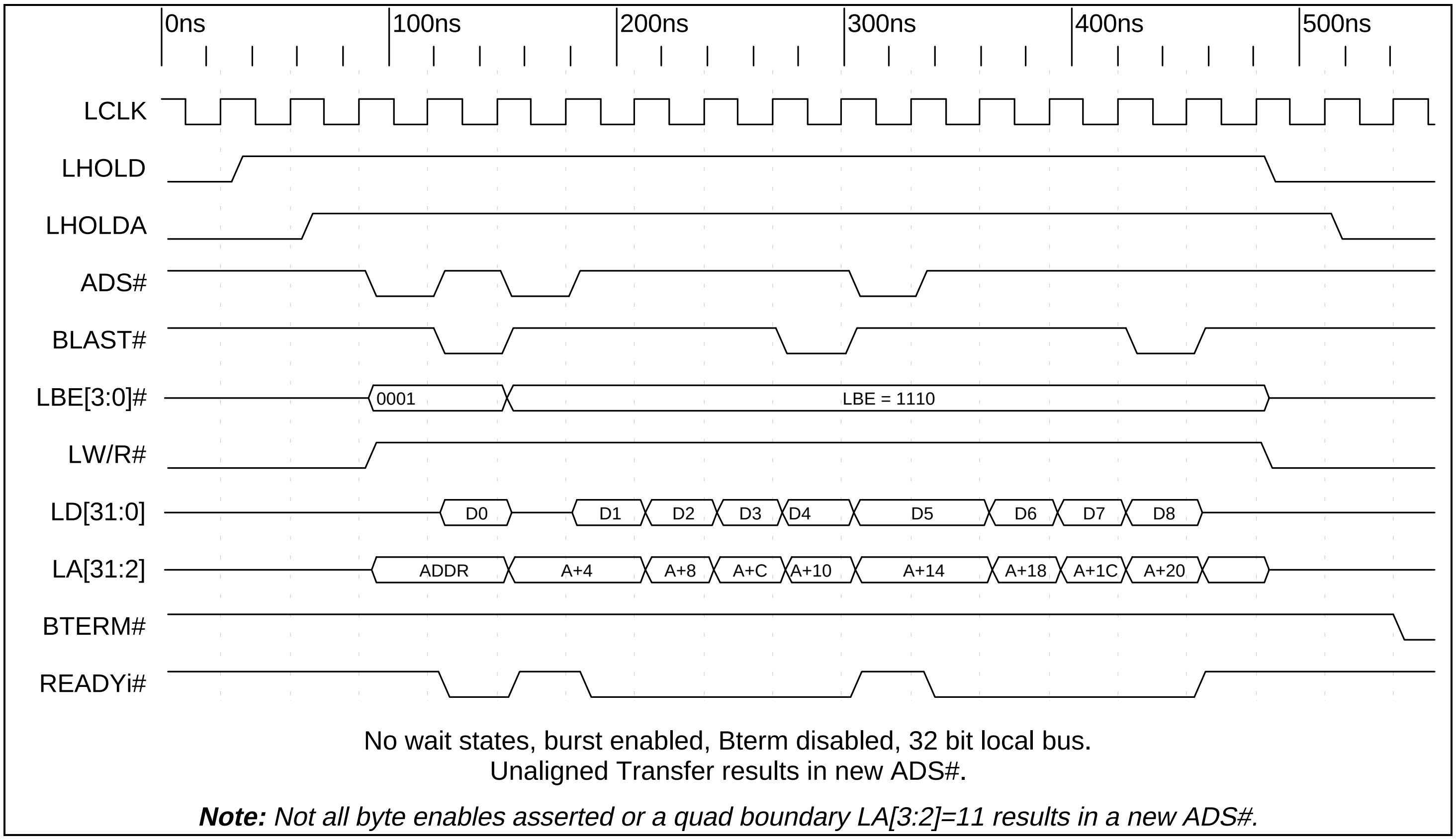
Eight Lword burst, no wait states, burst enabled, Bterm enabled, 32 bit local bus.

Note: *If Bterm is disabled, a new ADS# cycle starts every quad Lword boundary.*

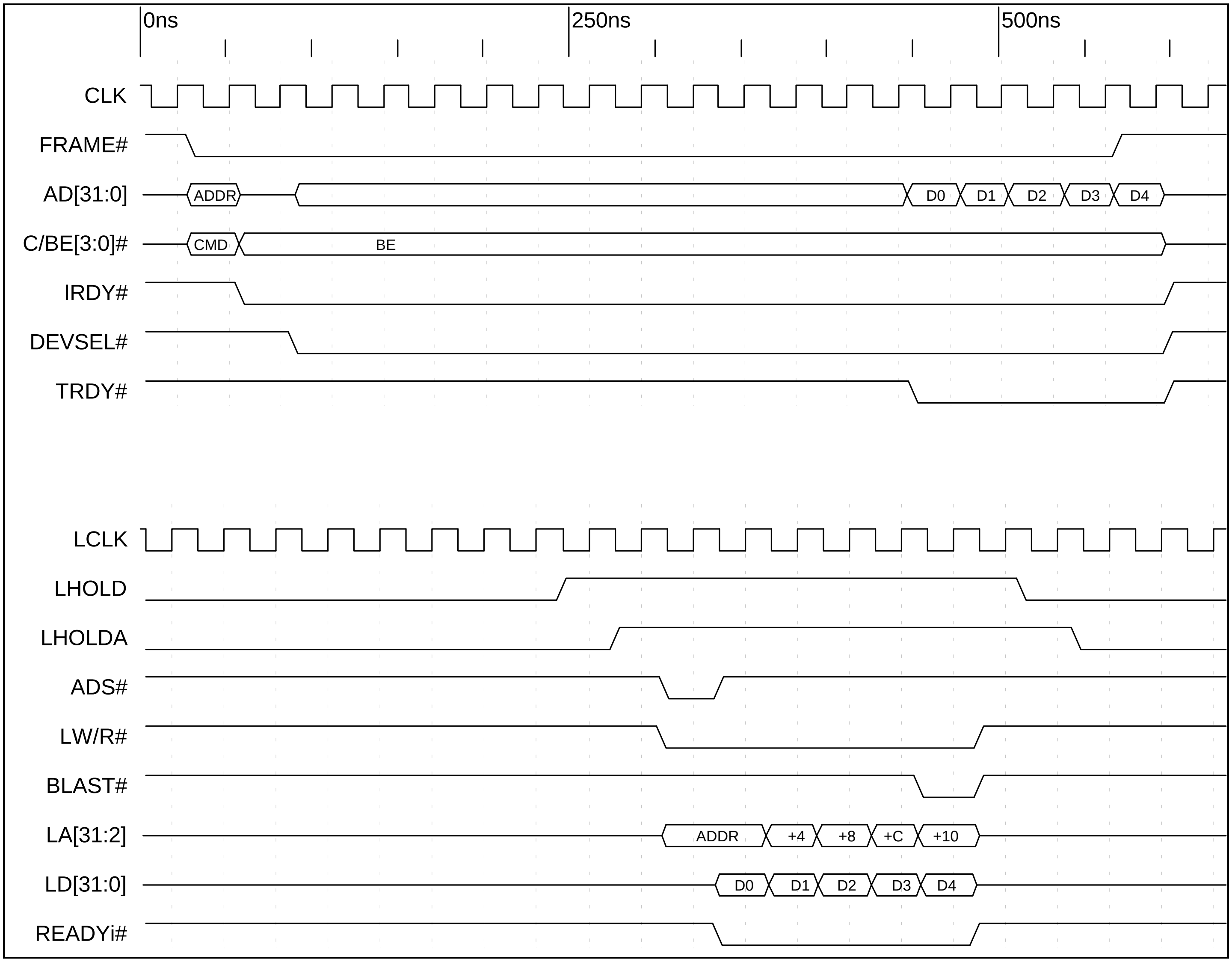
Timing Diagram 8-13. (C Mode) DMA or Direct Slave PCI 9080 Burst Write to Local Bus, Bterm Enabled



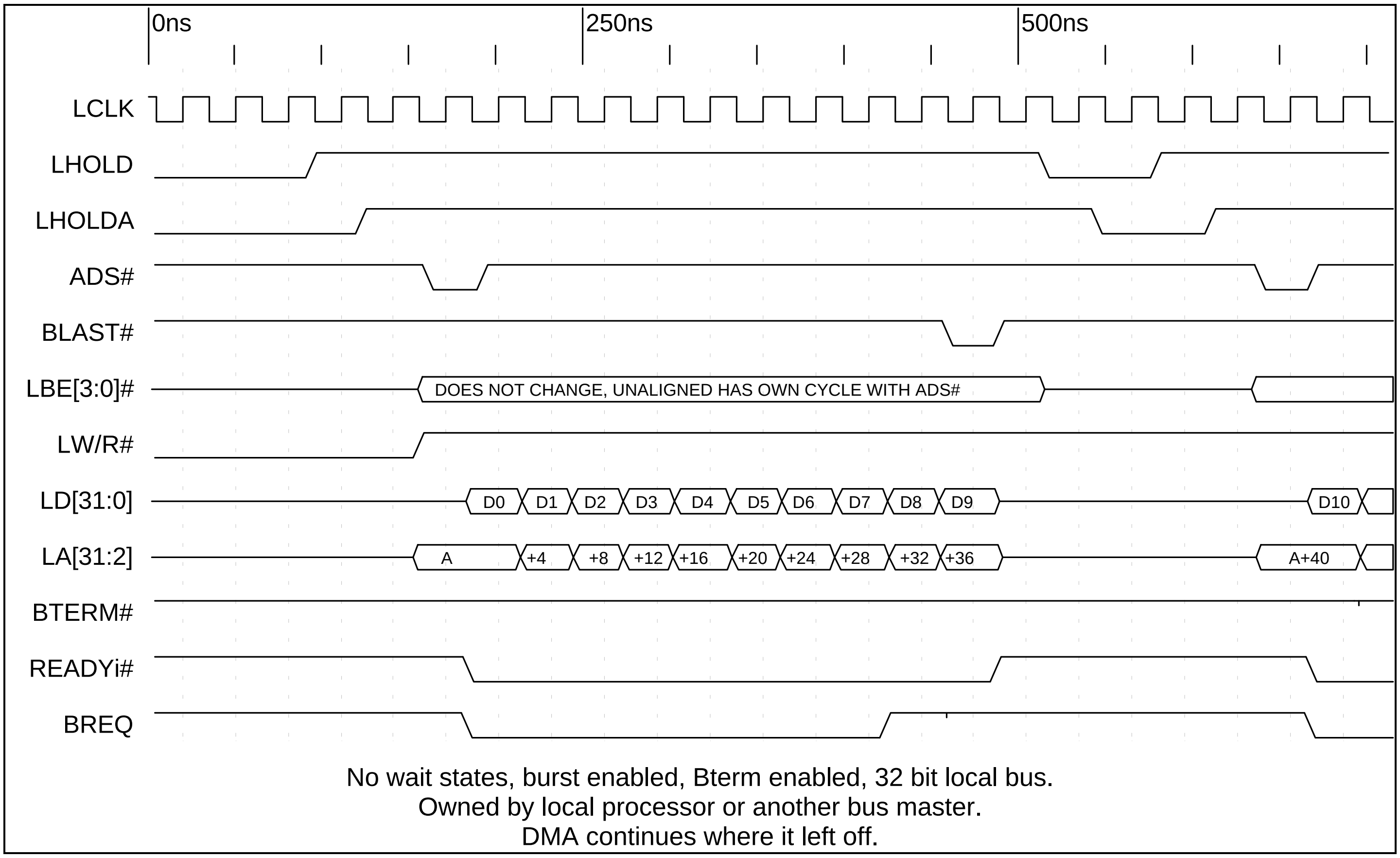
Timing Diagram 8-14. (C Mode) Direct Slave PCI to Local Burst Read, Bterm Disabled


0ns
100ns
200ns
300ns
400ns
500ns
LCLK
LHOLD
LHOLDA
ADS#
BLAST#
LBE[3:0]#
0001
LBE = 1110
LW/R#
LD[31:0]
D0
D1
D2
D3
D4
D5
D6
D7
D8
LA[31:2]
ADDR
A+4
A+8
A+C
A+10
A+14
A+18
A+1C
A+20
BTERM#
READYi#
No wait states, burst enabled, Bterm disabled, 32 bit local bus.
Unaligned Transfer results in new ADS#.
Note: Not all byte enables asserted or a quad boundary LA[3:2]=11 results in a new ADS#.

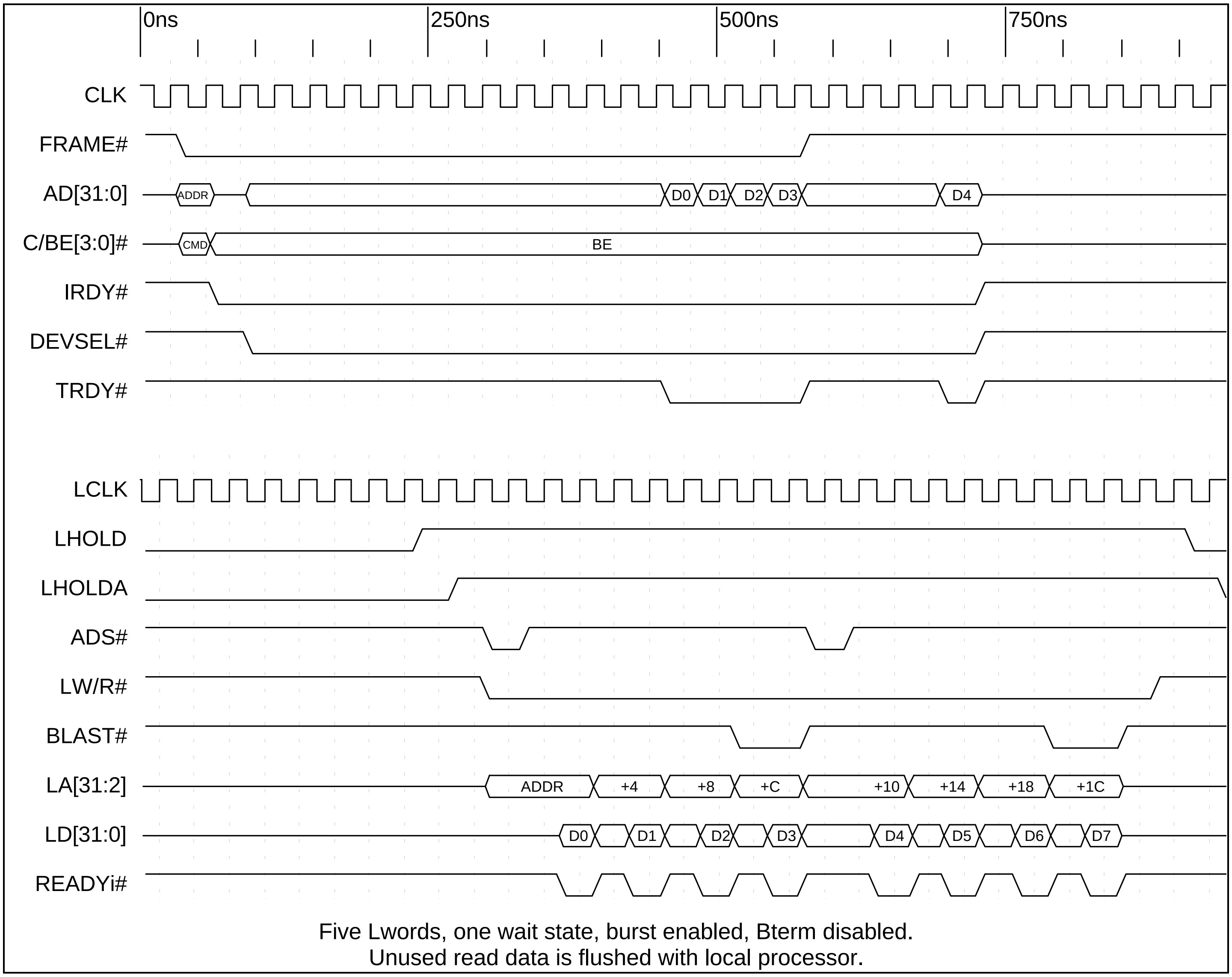
Timing Diagram 8-15. (C Mode) PCI 9080 DMA or Direct Slave Burst Write, Bterm Disabled



Timing Diagram 8-16. (C Mode) Direct Slave Read with Prefetch Counter Set to 5

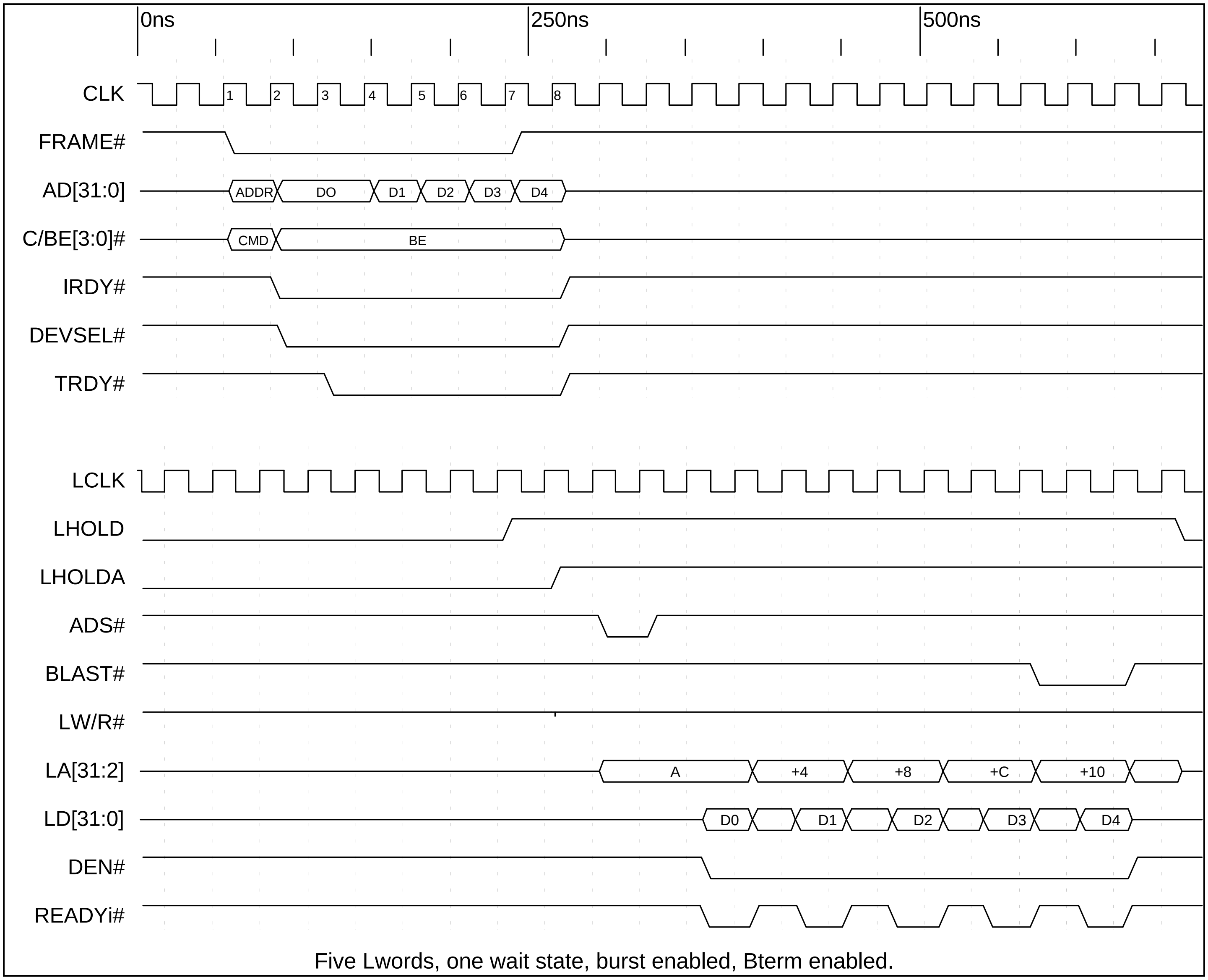

0ns
250ns
500ns
LCLK
LHOLD
LHOLDA
ADS#
BLAST#
LBE[3:0]#
DOES NOT CHANGE, UNALIGNED HAS OWN CYCLE WITH ADS#
LW/R#
LD[31:0]
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
LA[31:2]
A
+4
+8
+12
+16
+20
+24
+28
+32
+36
A+40
BTERM#
READYi#
BREQ
No wait states, burst enabled, Bterm enabled, 32 bit local bus.
Owned by local processor or another bus master.
DMA continues where it left off.

Timing Diagram 8-17. (C Mode) Direct Slave or DMA Burst Write to 32 bit local bus Suspended by BREQ Input

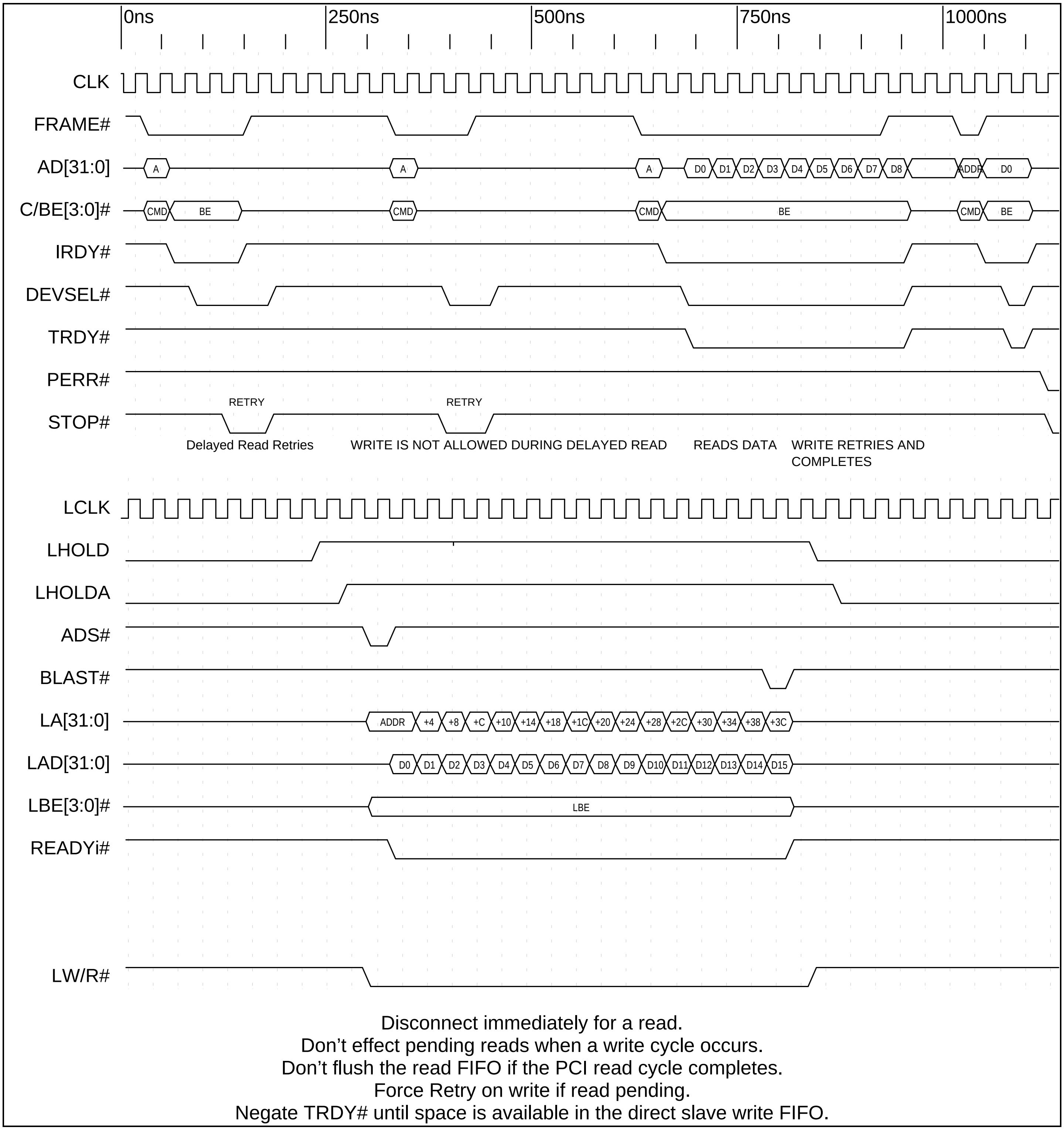


Five Lwords, one wait state, burst enabled, Bterm disabled.
Unused read data is flushed with local processor.

Timing Diagram 8-18. (C Mode) Direct Slave Burst Read of Five Lwords with One Wait State

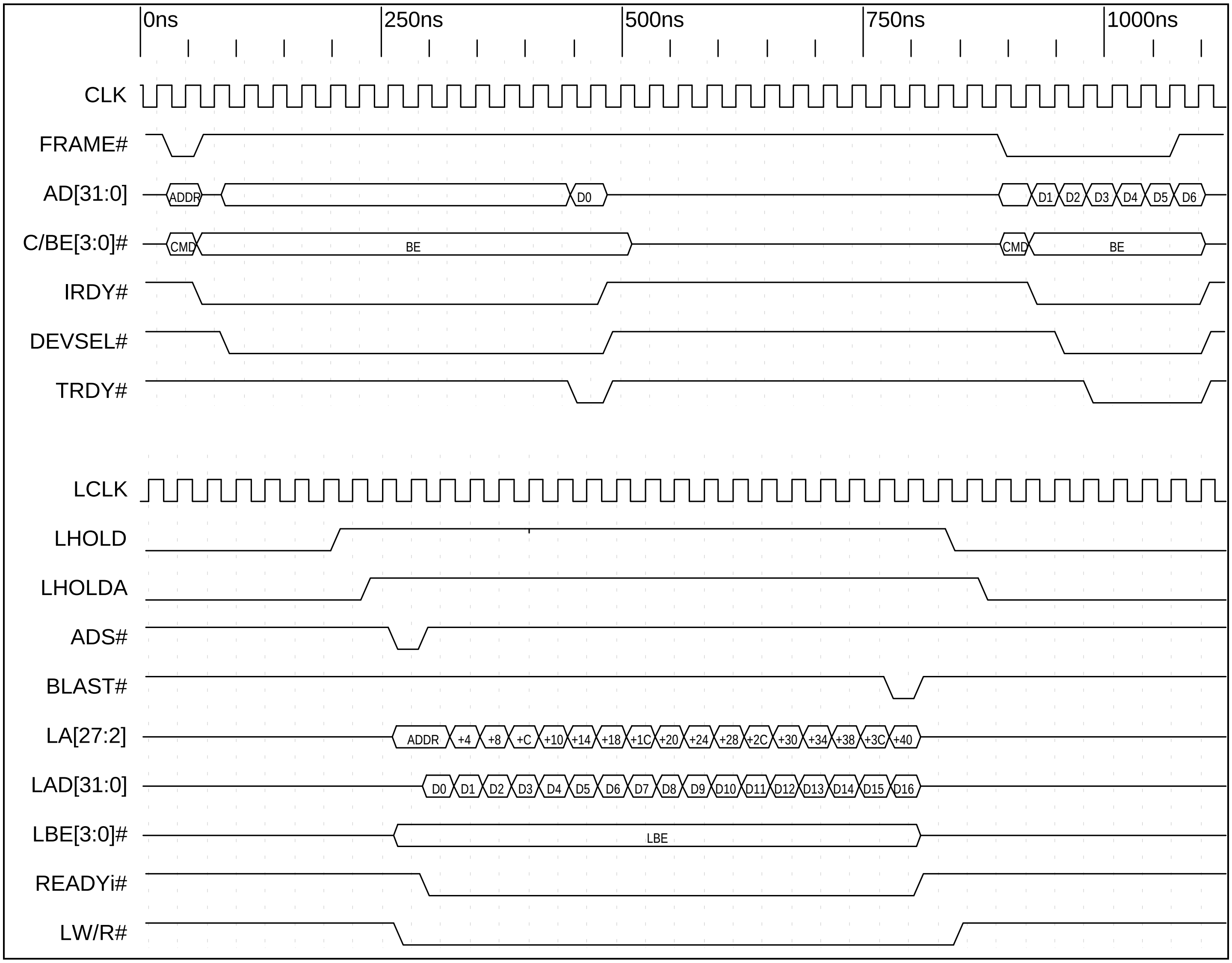


Timing Diagram 8-19. (C Mode) Direct Slave Burst of Five Lwords with One Wait State

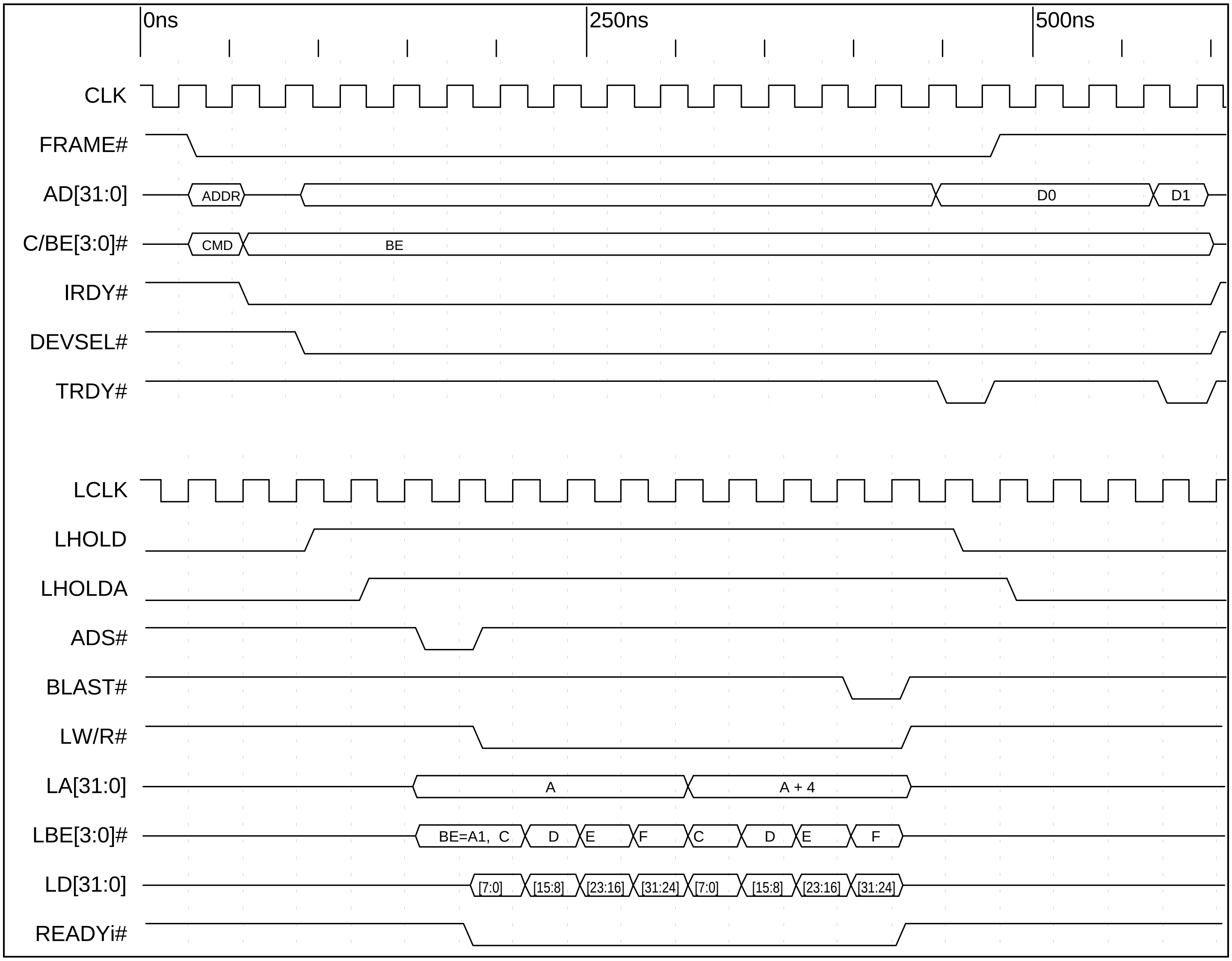


Disconnect immediately for a read.
Don’t effect pending reads when a write cycle occurs.
Don’t flush the read FIFO if the PCI read cycle completes.
Force Retry on write if read pending.
Negate TRDY# until space is available in the direct slave write FIFO.

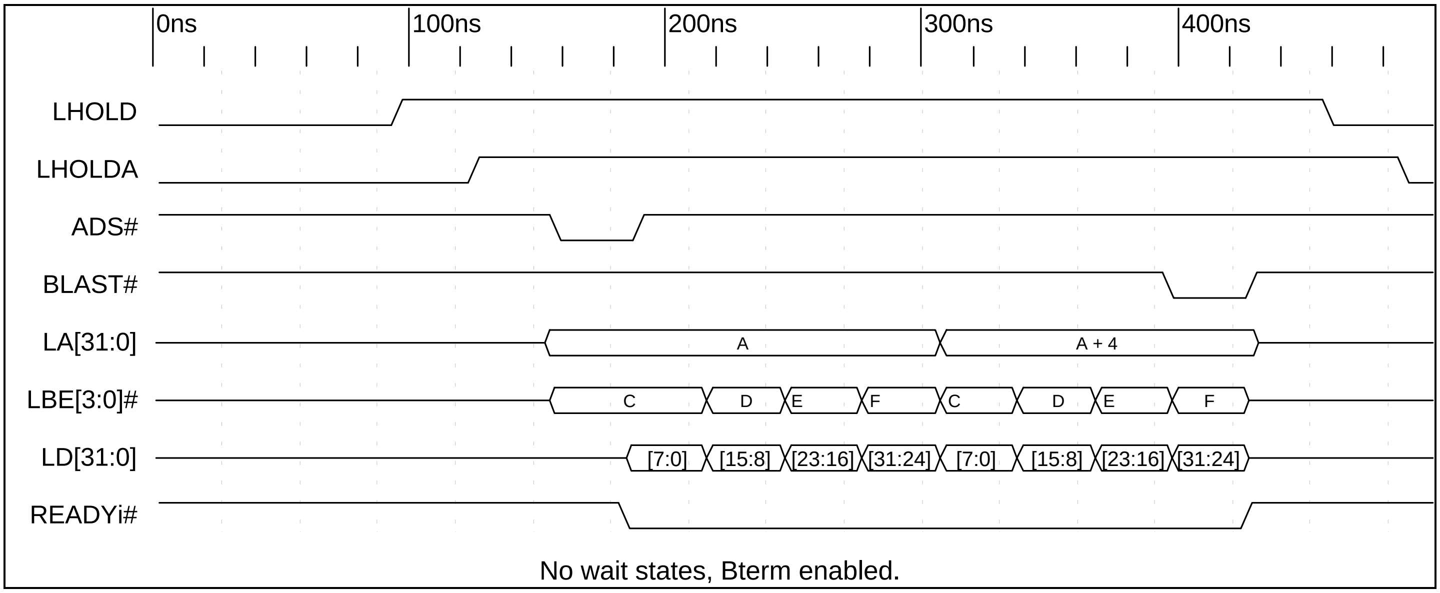
Timing Diagram 8-20. (C Mode) Direct Slave Read 2.1 Spec



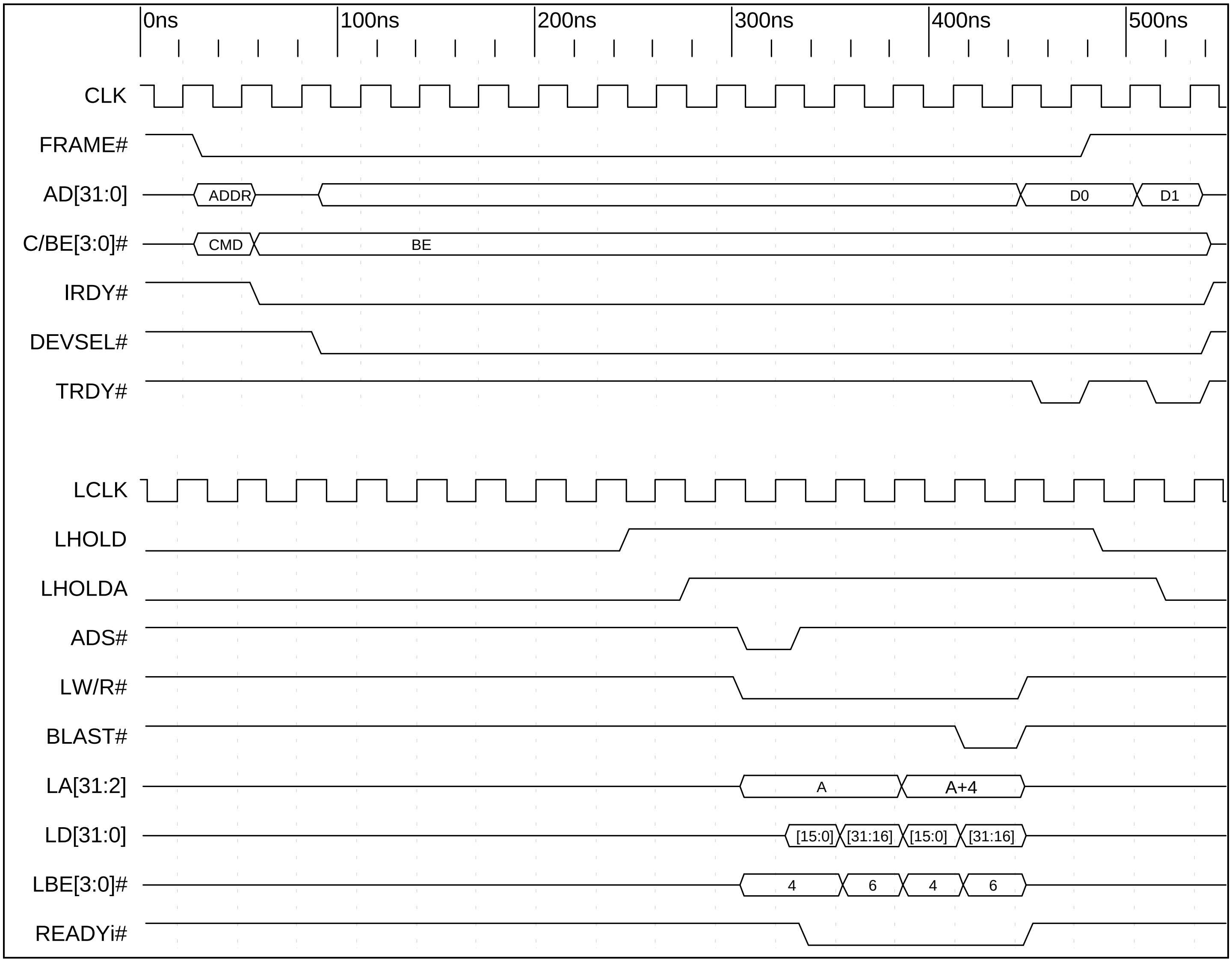
Timing Diagram 8-21. (C Mode) Direct Slave Read No Flush Mode (Read Ahead Mode)



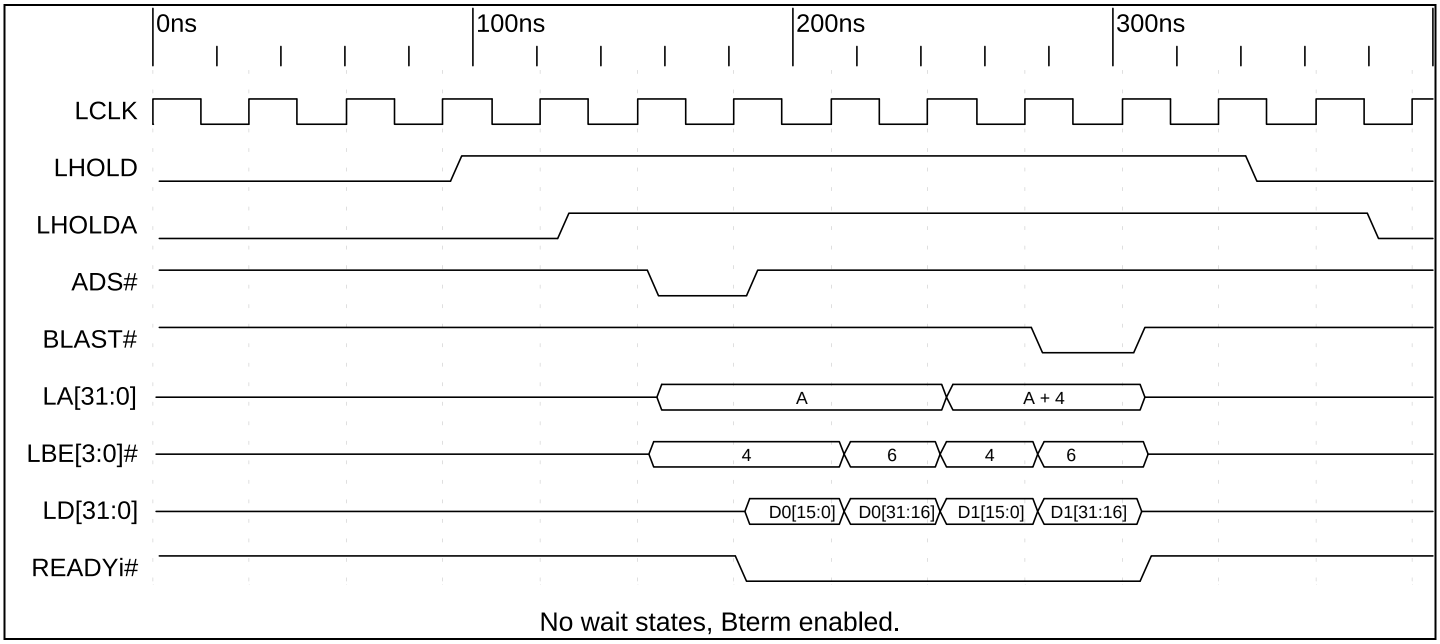
Timing Diagram 8-22. (C Mode) Direct Slave Read of Two Lwords from 8-Bit Bus



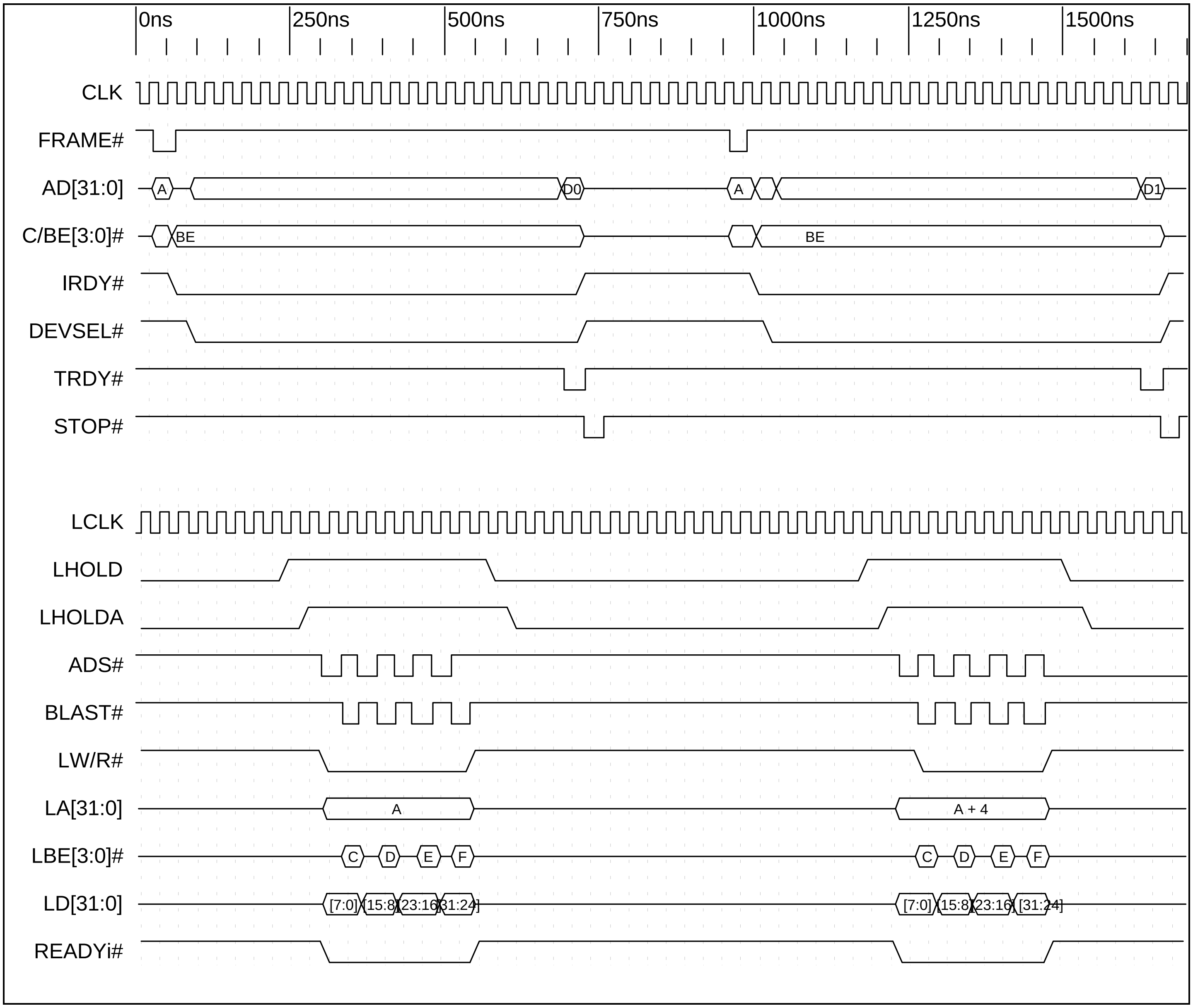
Timing Diagram 8-23. (C Mode) PCI 9080 DMA or Direct Slave Two Lword Burst Write to 8 Bit Local Bus



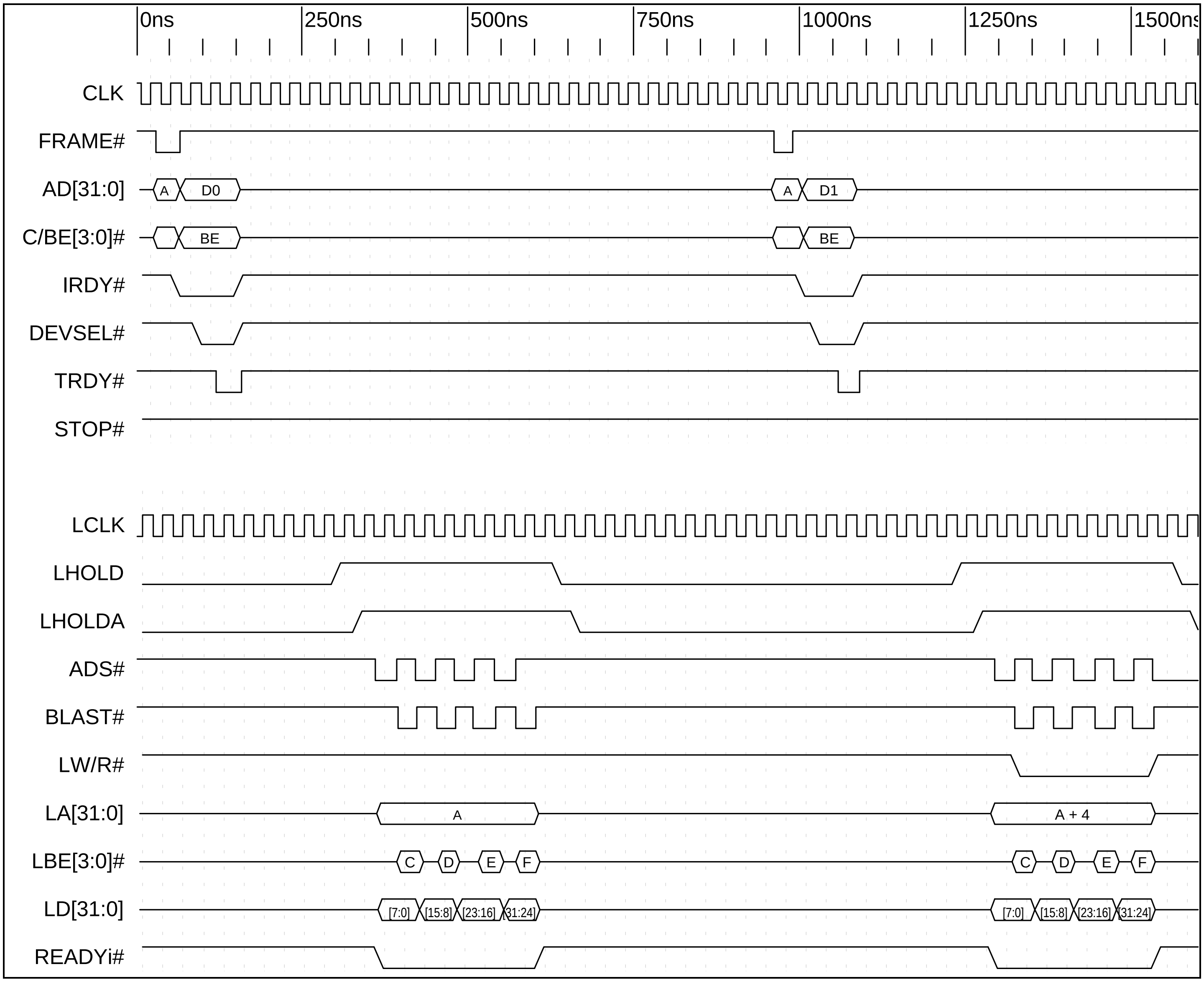
Timing Diagram 8-24. (C Mode) Direct Slave Read of Two Lwords from 16-Bit Bus



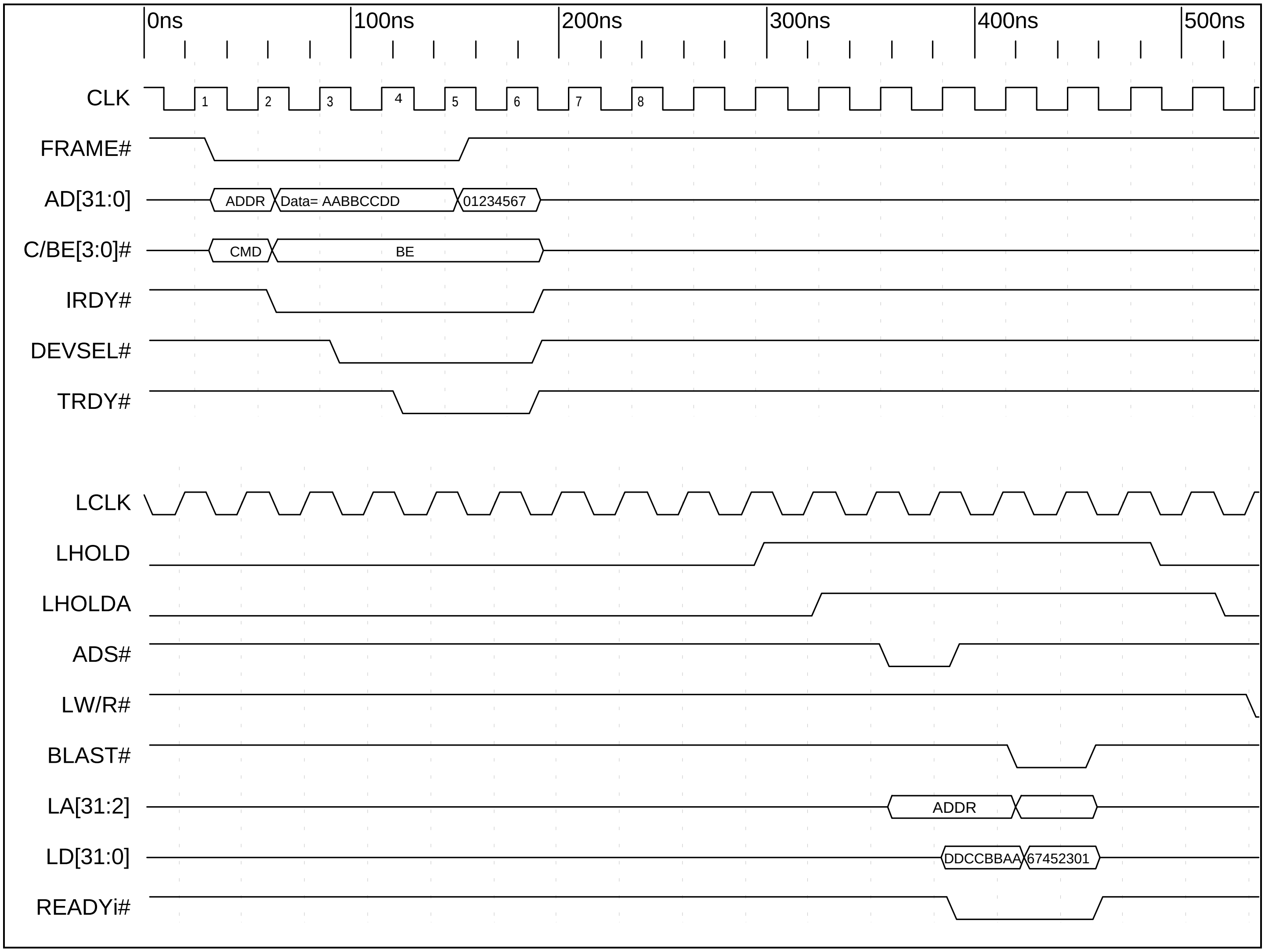
Timing Diagram 8-25. (C Mode) PCI 9080 DMA or Direct Slave Two Lword Burst Write to 16 Bit Local Bus



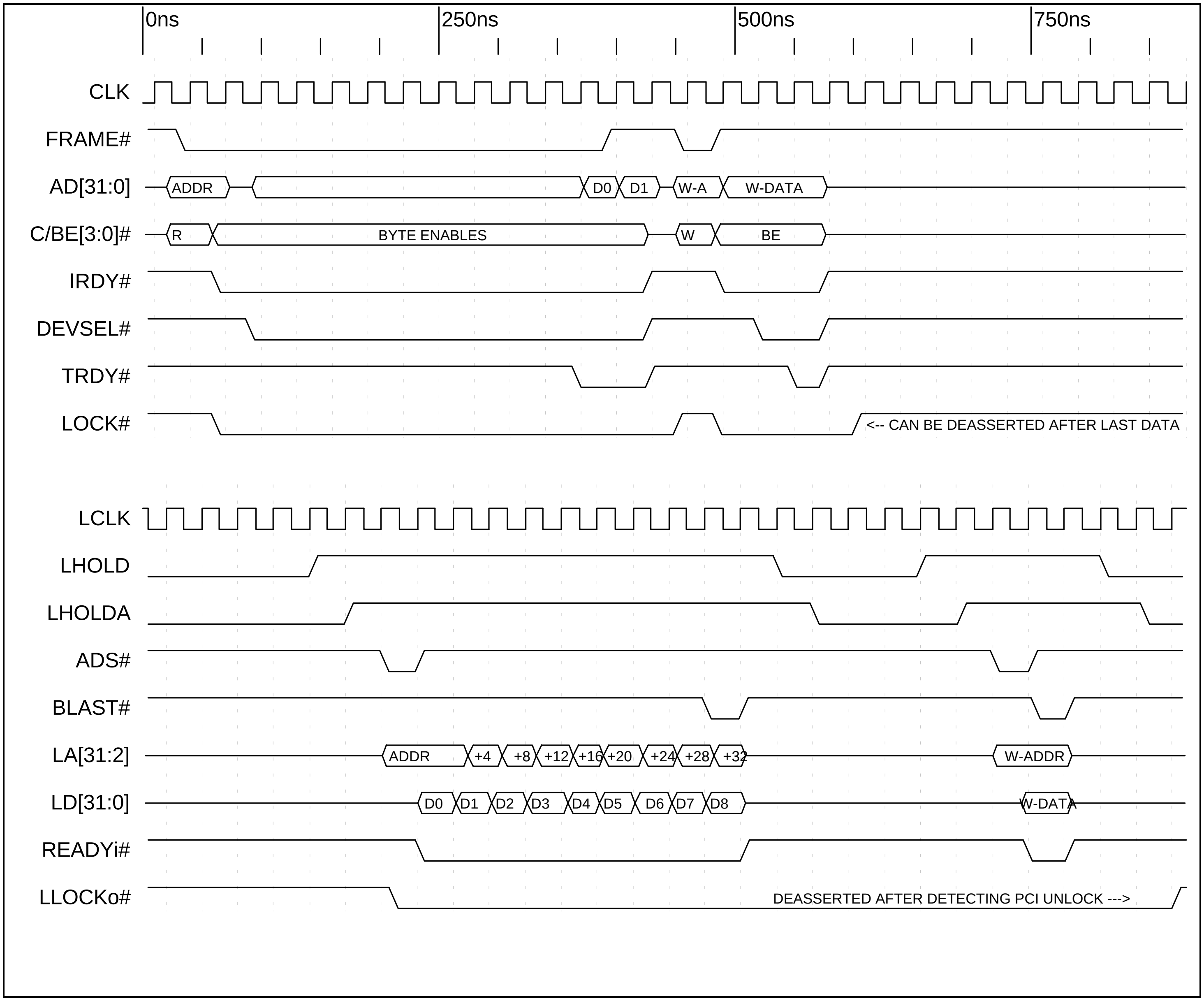
Timing Diagram 8-26. (C Mode) Direct Slave Read of Two Lwords from 8 Bit I/O Local Bus



Timing Diagram 8-27. (C Mode) Direct Slave Write of Two Lwords to 8 Bit I/O Local Bus

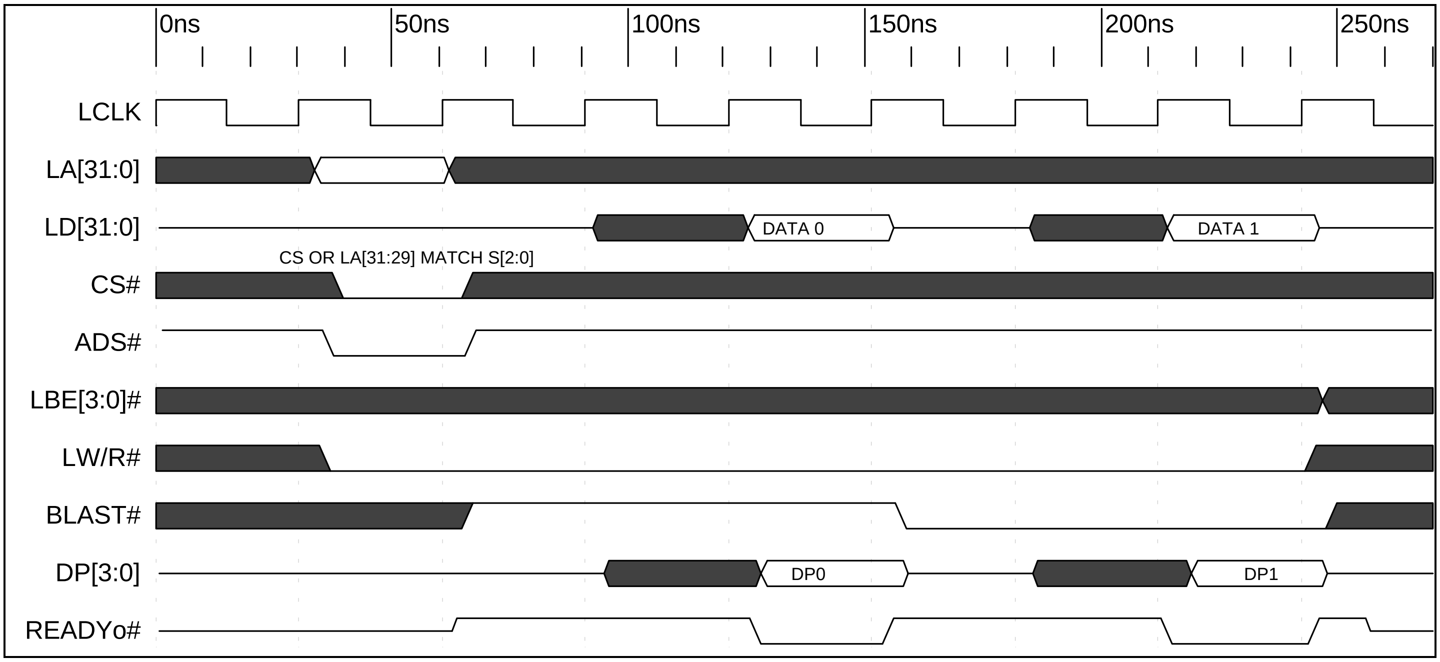


Timing Diagram 8-28. (C Mode) Direct Slave in BIGEND Local Bus with BIGEND# Input or Internal Register Setting

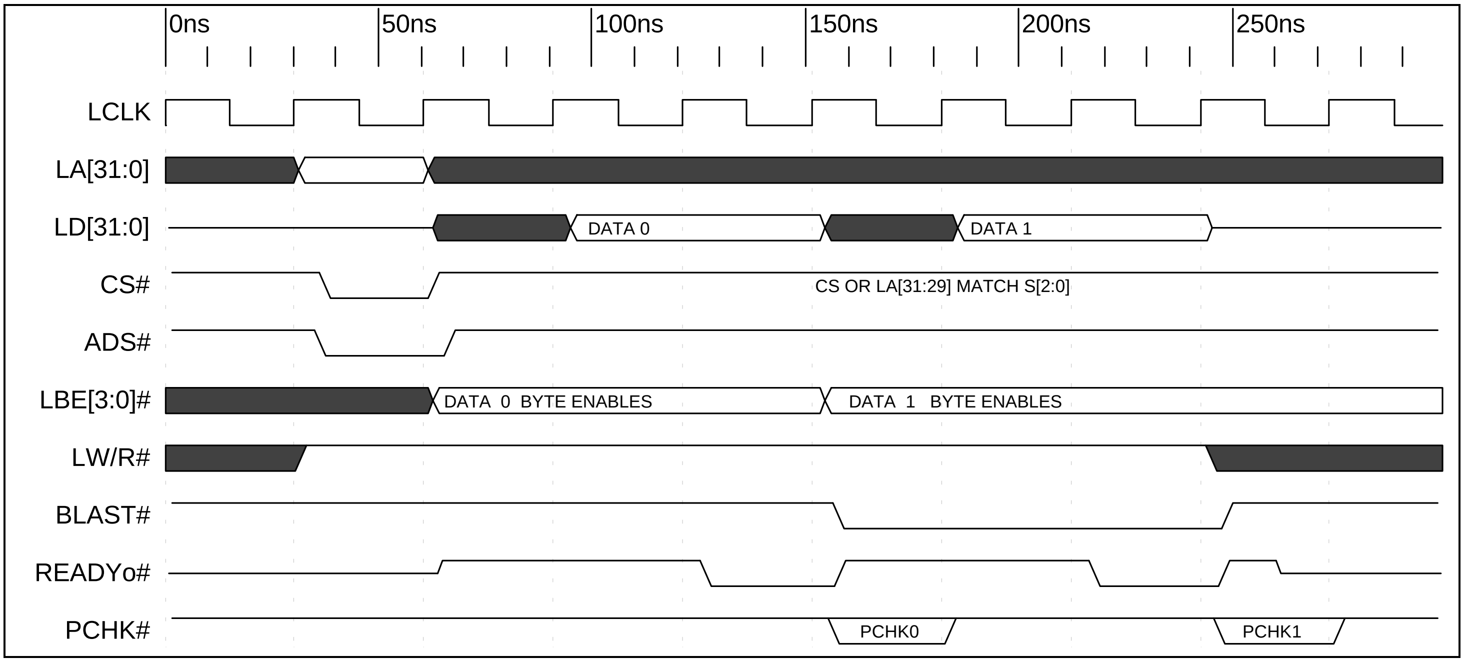


Timing Diagram 8-29. (C Mode) Locked Direct Slave Read Followed by Write and Release (LLOCKo#)

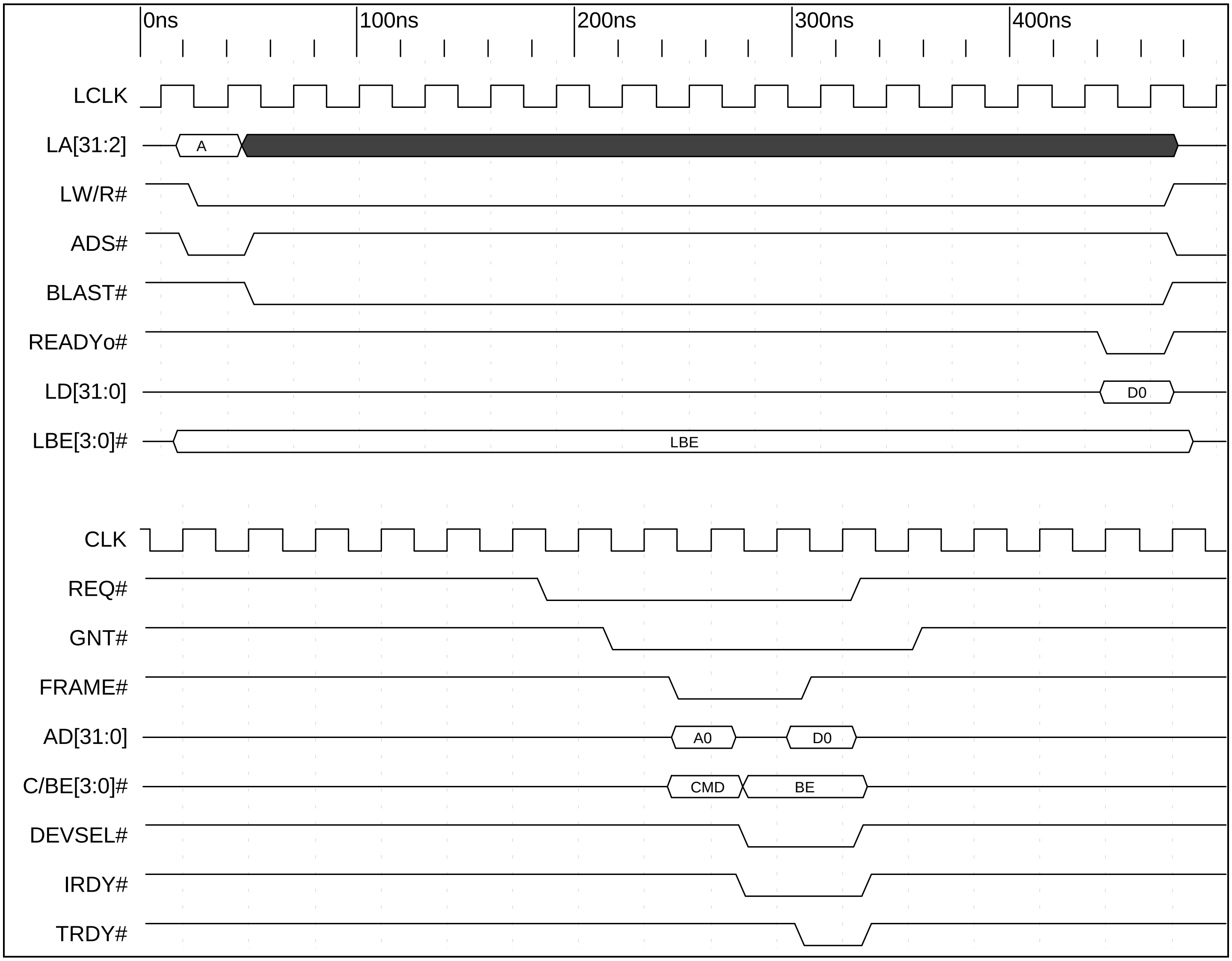
8.3.2 C Mode Direct Master



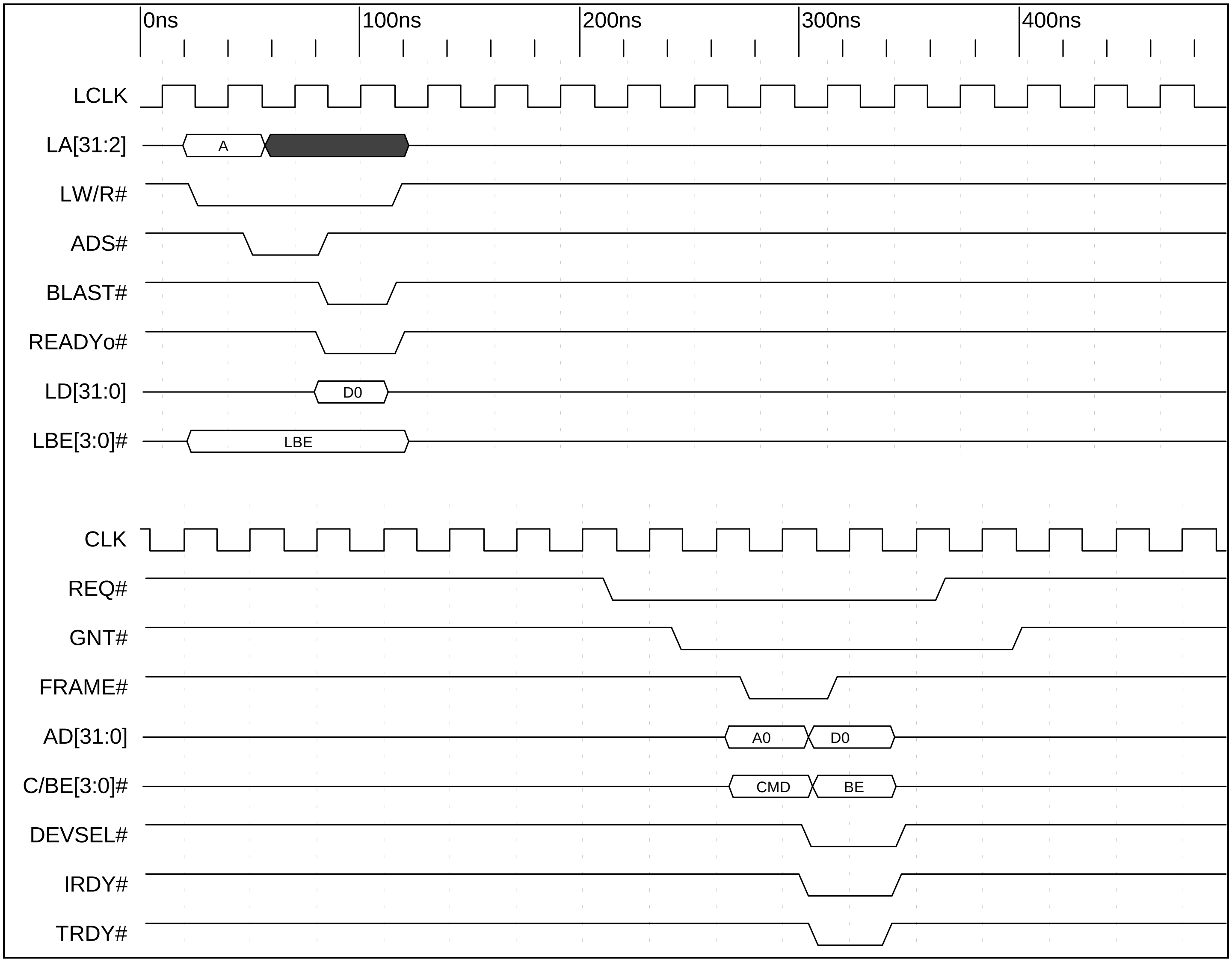
Timing Diagram 8-30. (C Mode) Local Bus Read from PCI 9080 CFG Registers



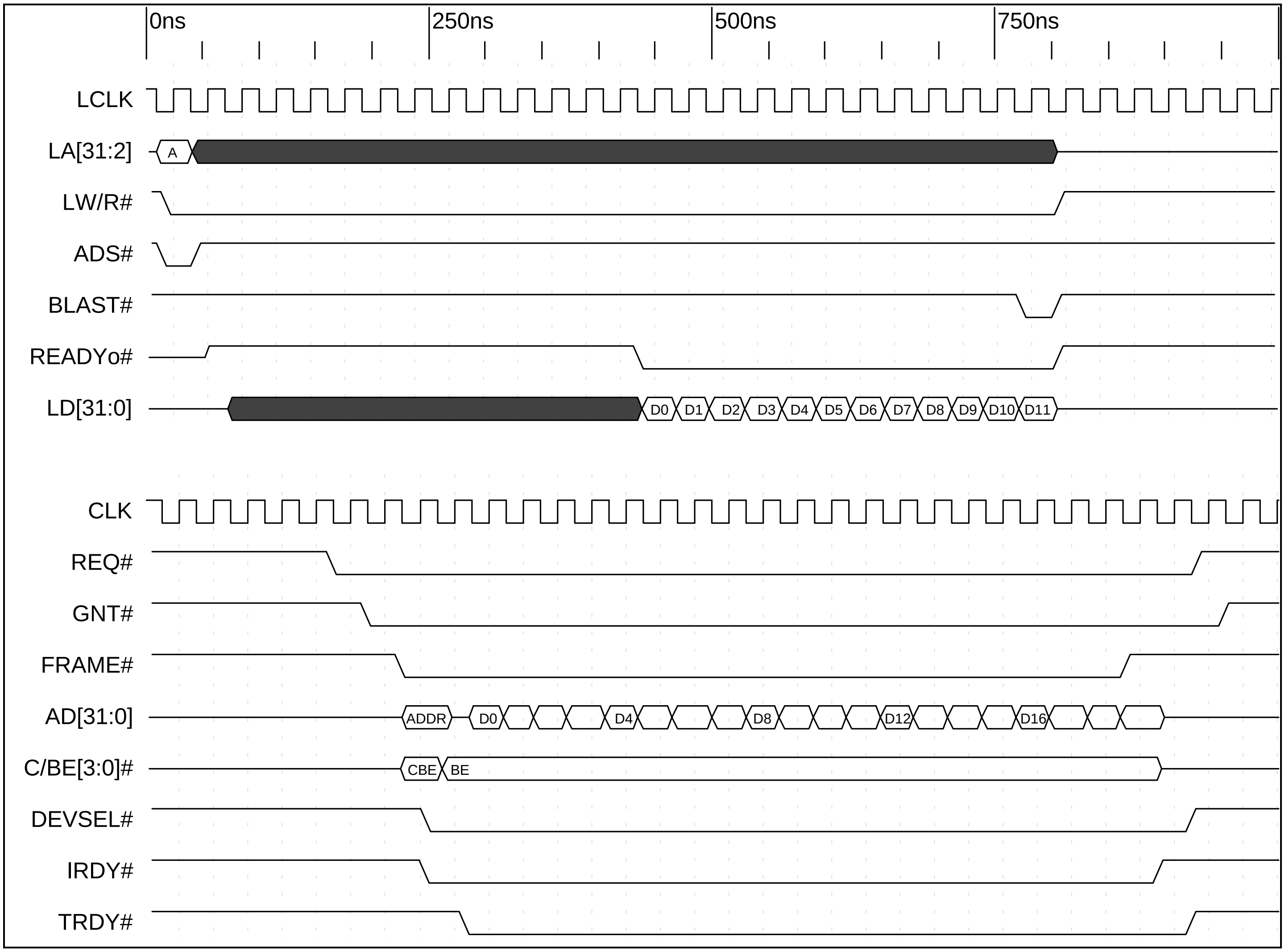
Timing Diagram 8-31. (C Mode) Local Bus Write to PCI 9080 CFG Registers



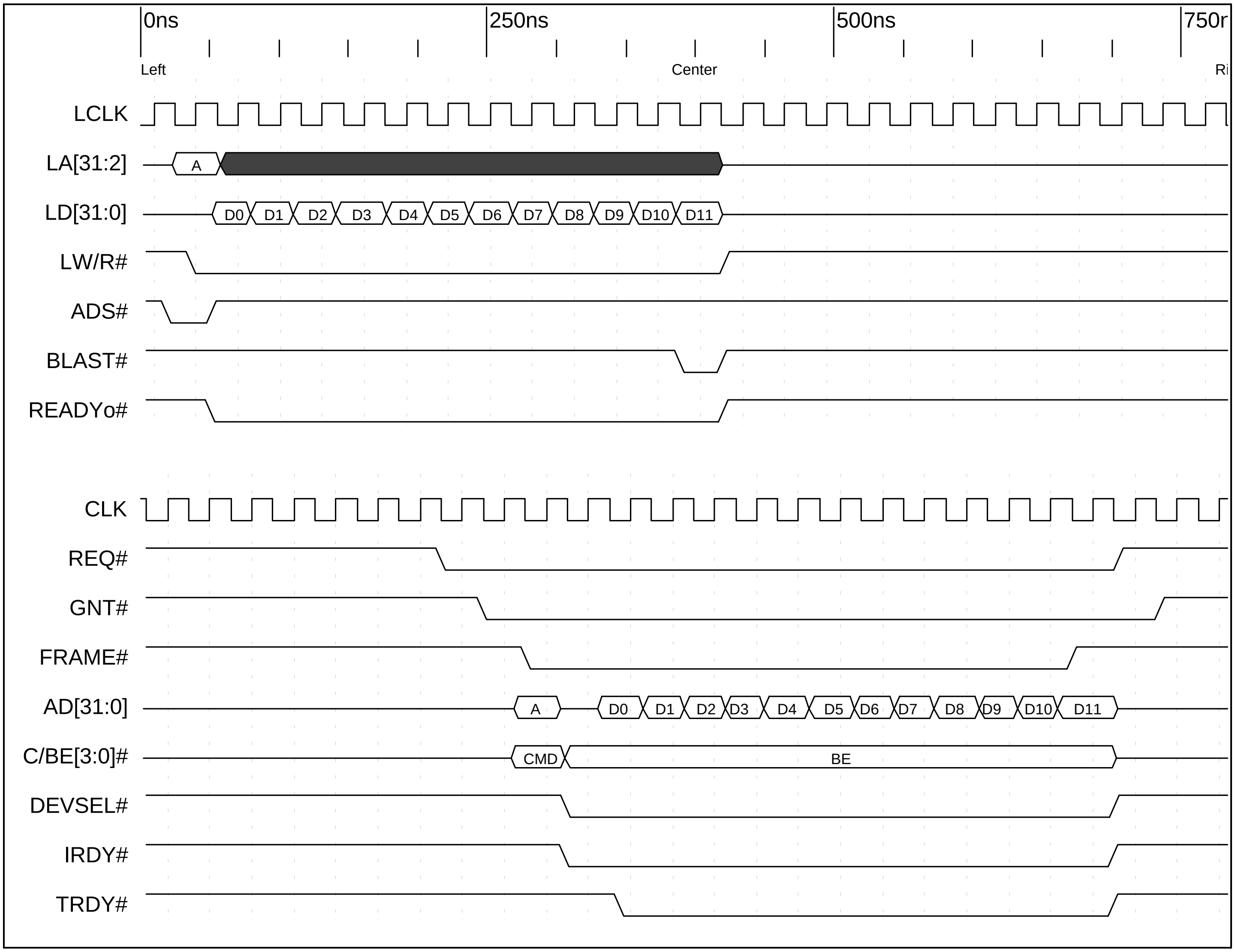
Timing Diagram 8-32. (C Mode) Local Bus Direct Master Single Memory Read



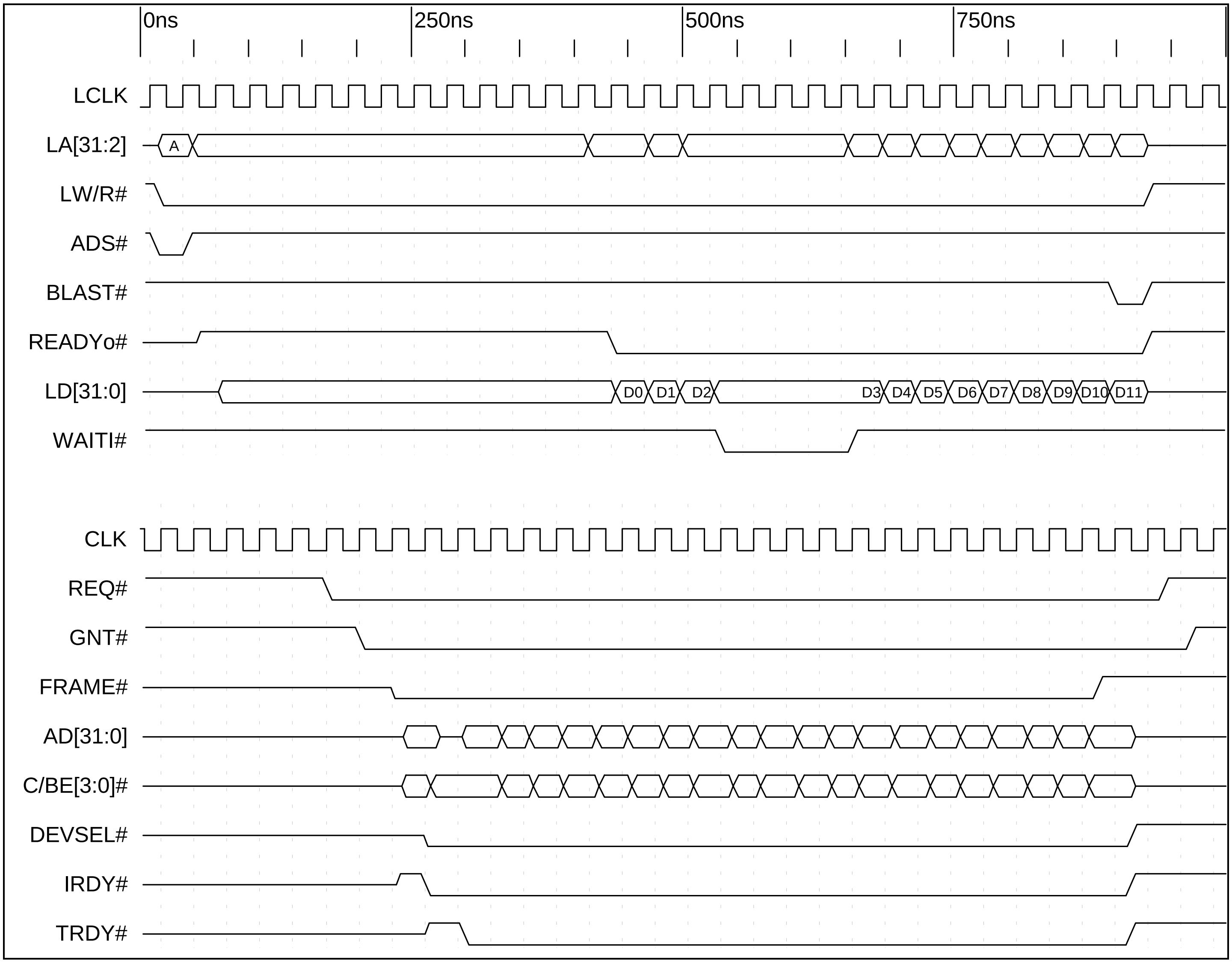
Timing Diagram 8-33. (C Mode) Local Bus Direct Master Single Memory Write Cycle



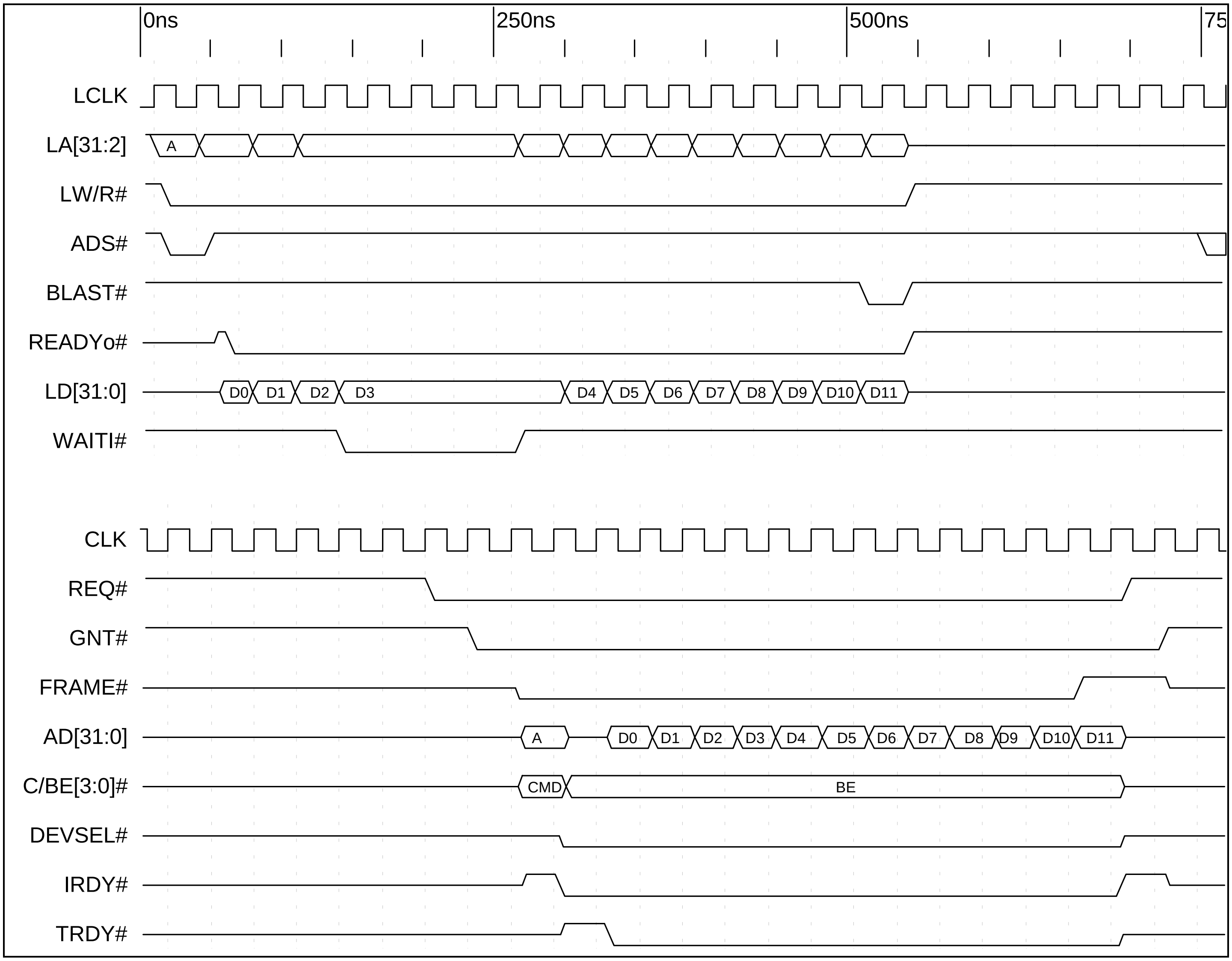
Timing Diagram 8-34. (C Mode) PCI 9080 Direct Master Memory Read, 12 Lword Burst



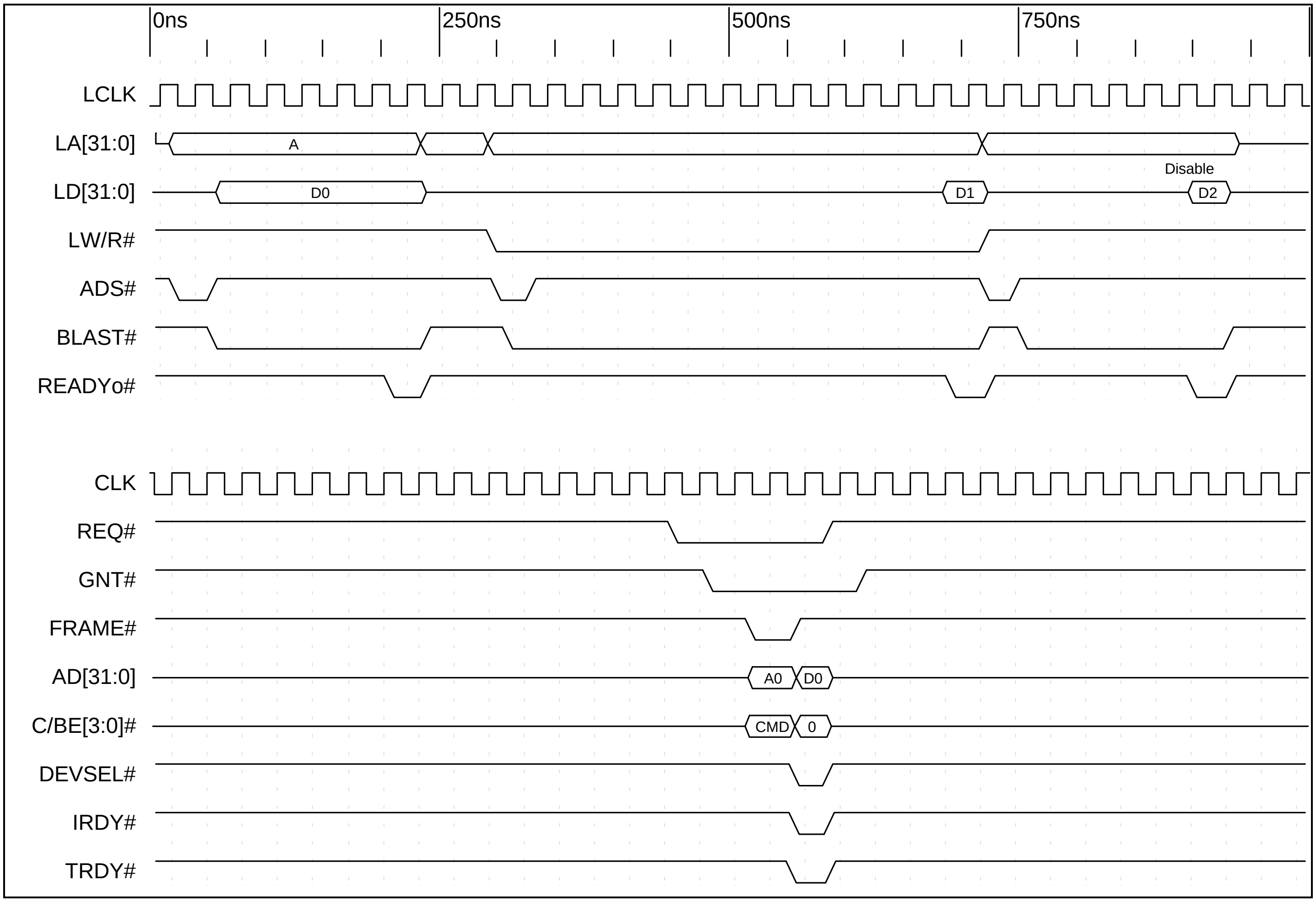
Timing Diagram 8-35. (C Mode) PCI 9080 Direct Master Memory Write of 12 Lwords



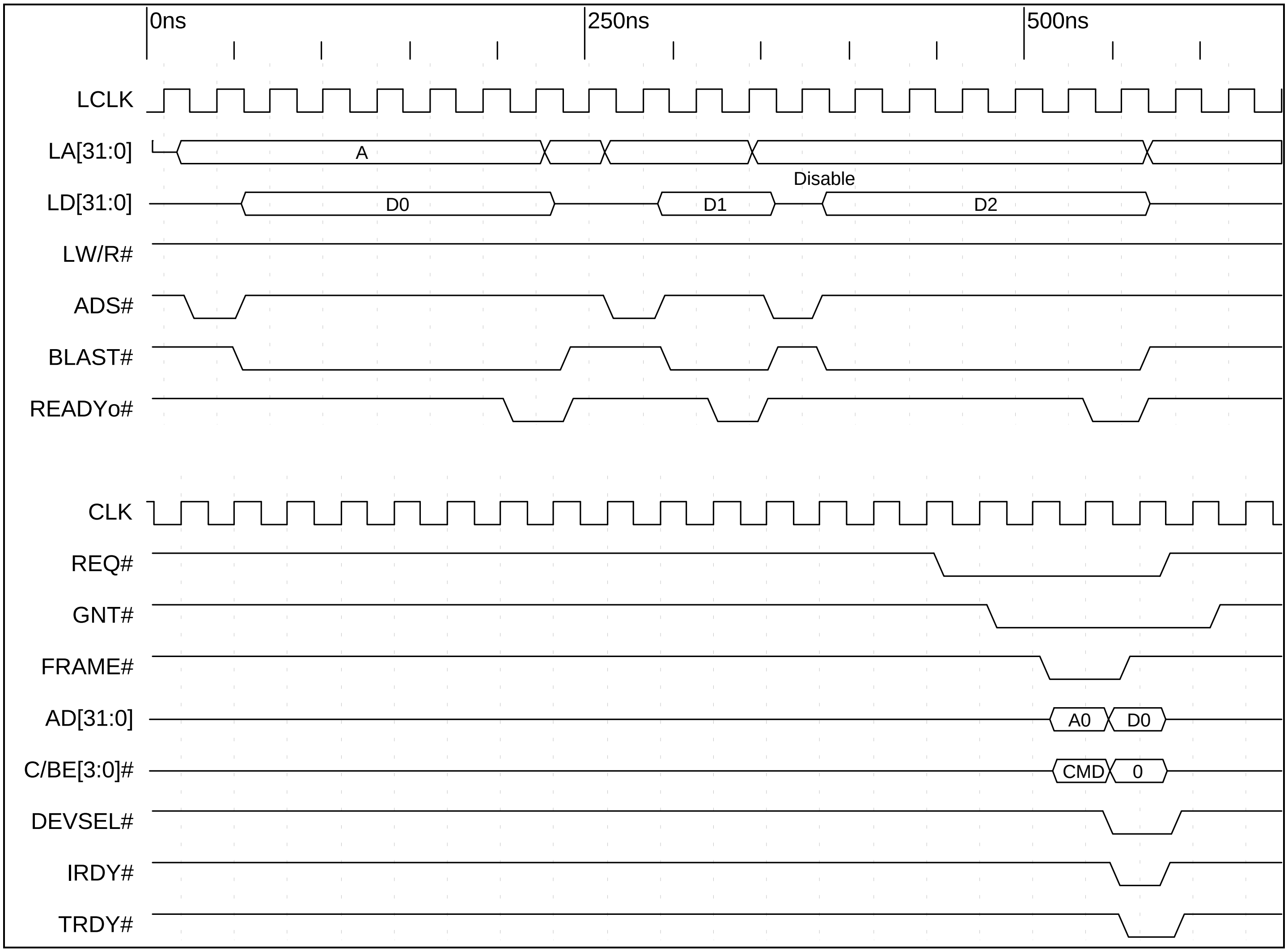
Timing Diagram 8-36. (C Mode) PCI 9080 Direct Master Memory Read with WAITI#



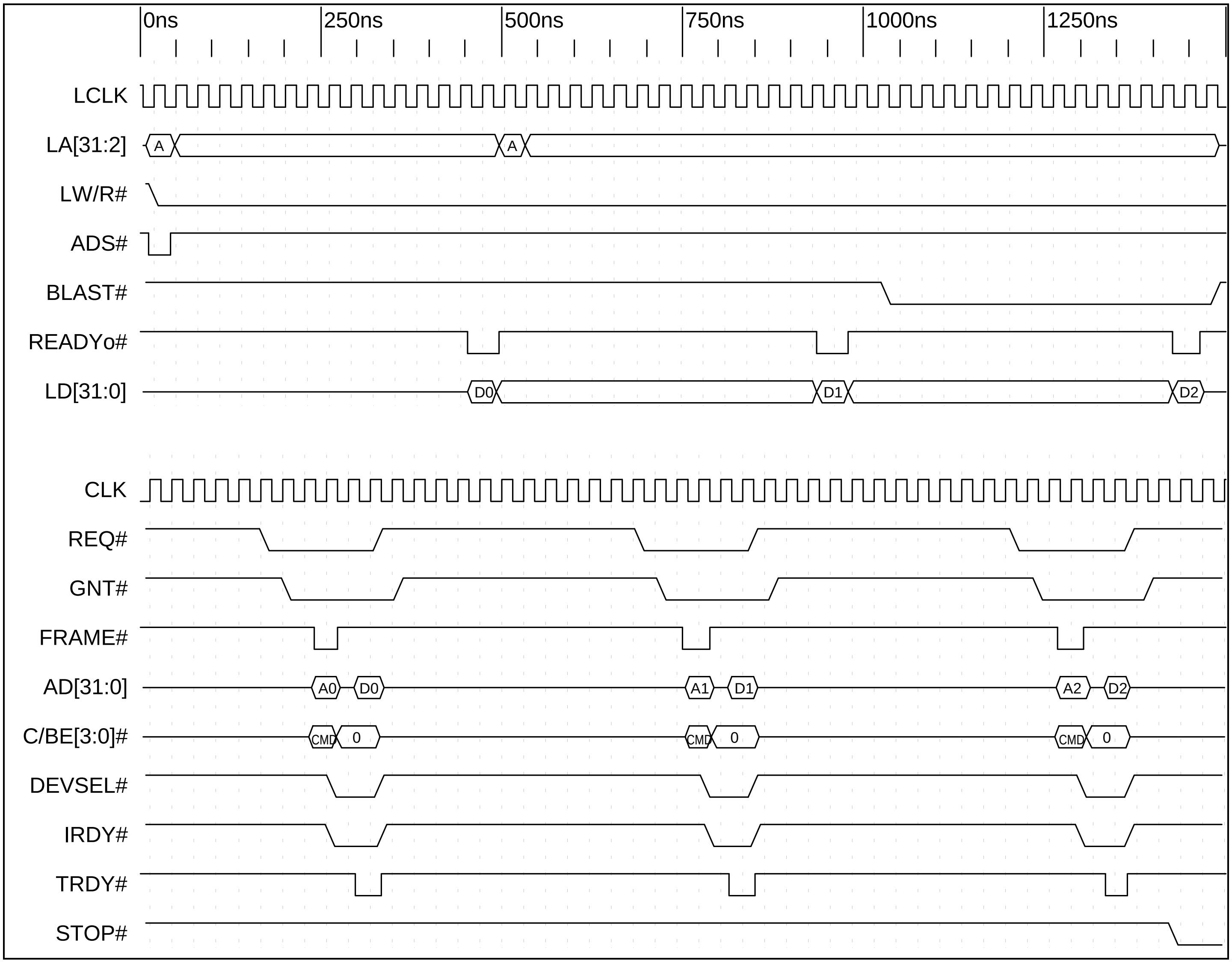
Timing Diagram 8-37. (C Mode) PCI 9080 Direct Master Memory Write with WAITI#



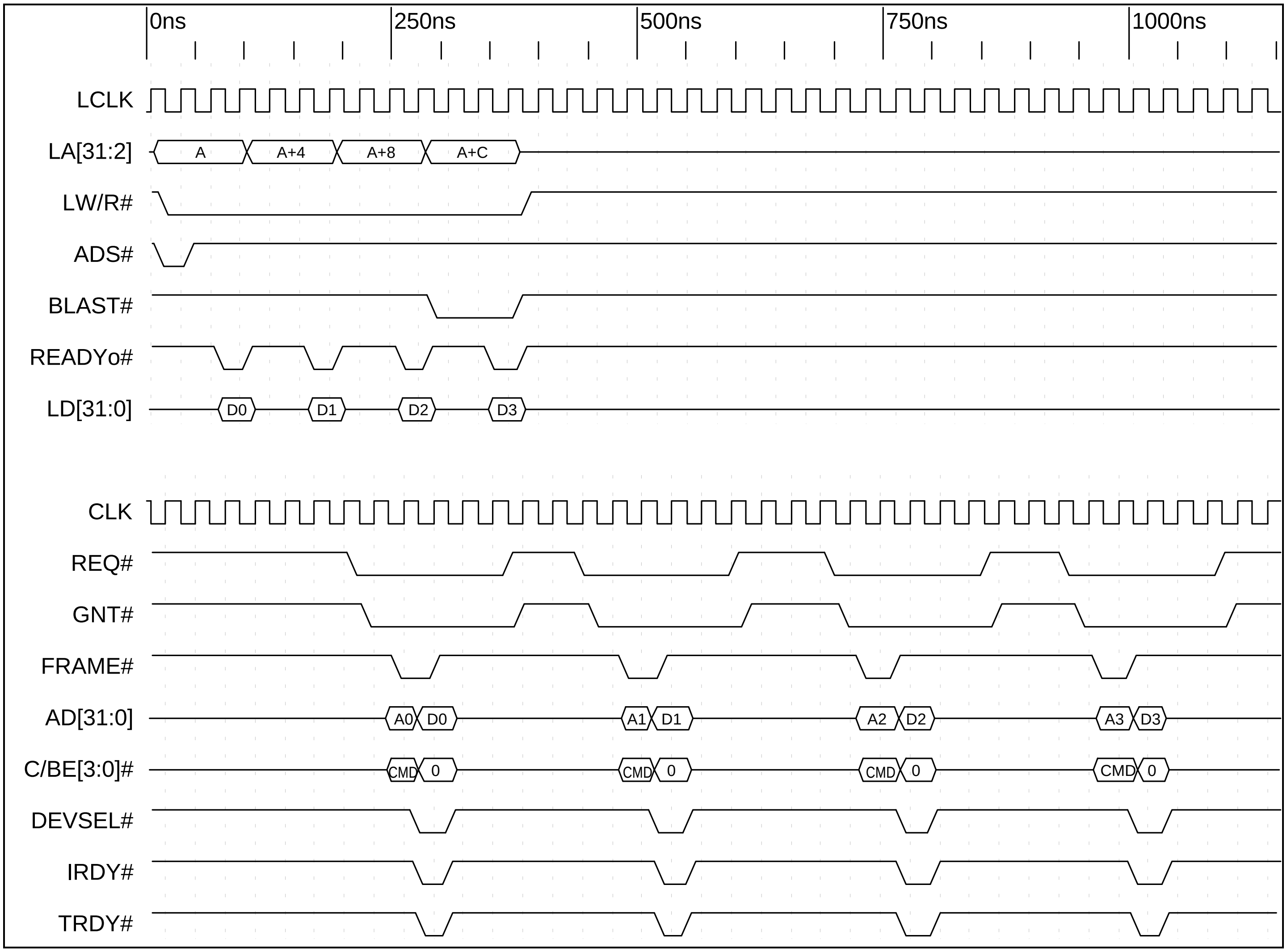
Timing Diagram 8-38. (C Mode) PCI 9080 Direct Master Configuration Read—Type 1 or Type 0



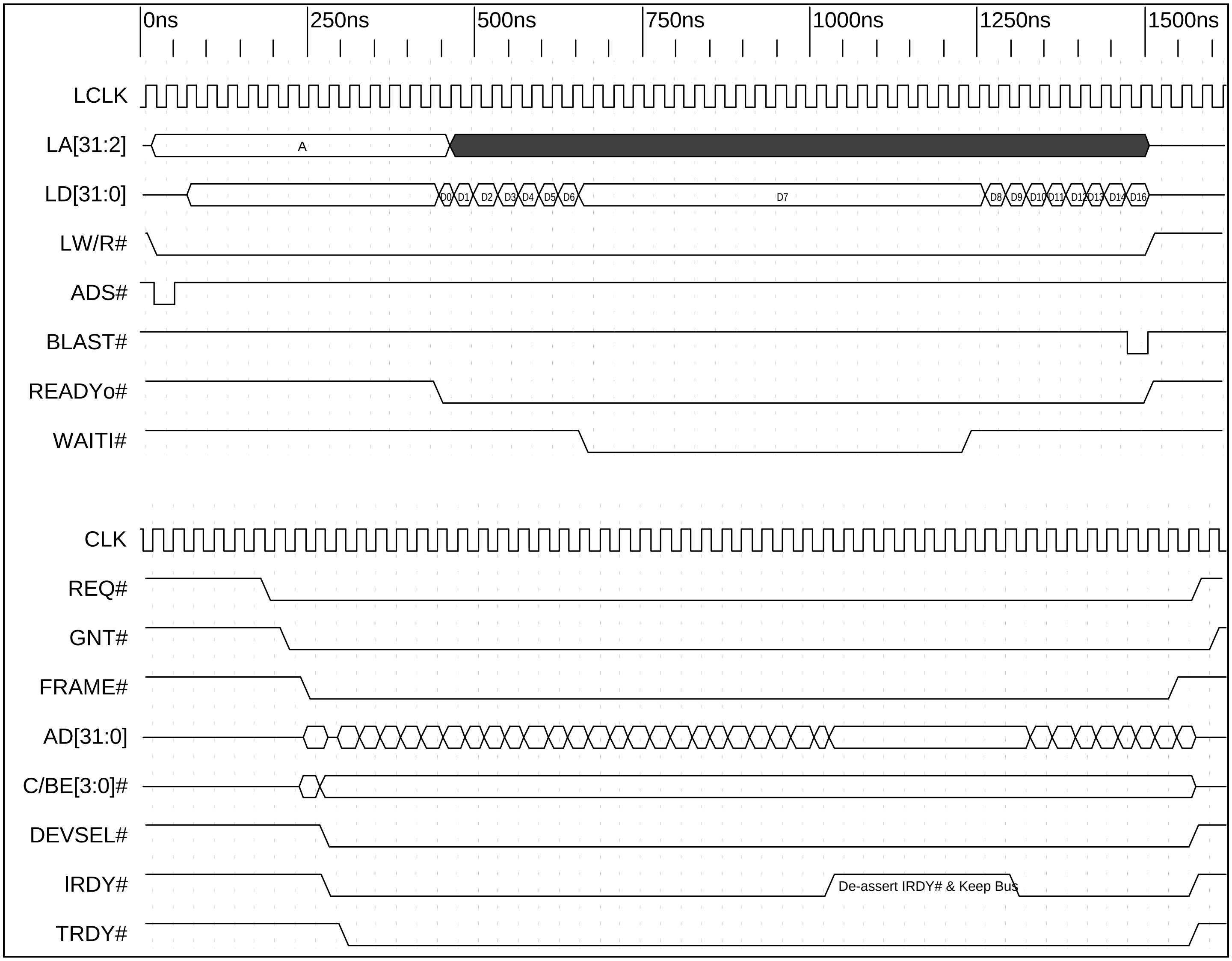
Timing Diagram 8-39. (C Mode) PCI 9080 Direct Master Configuration Write—Type 1 or Type 0



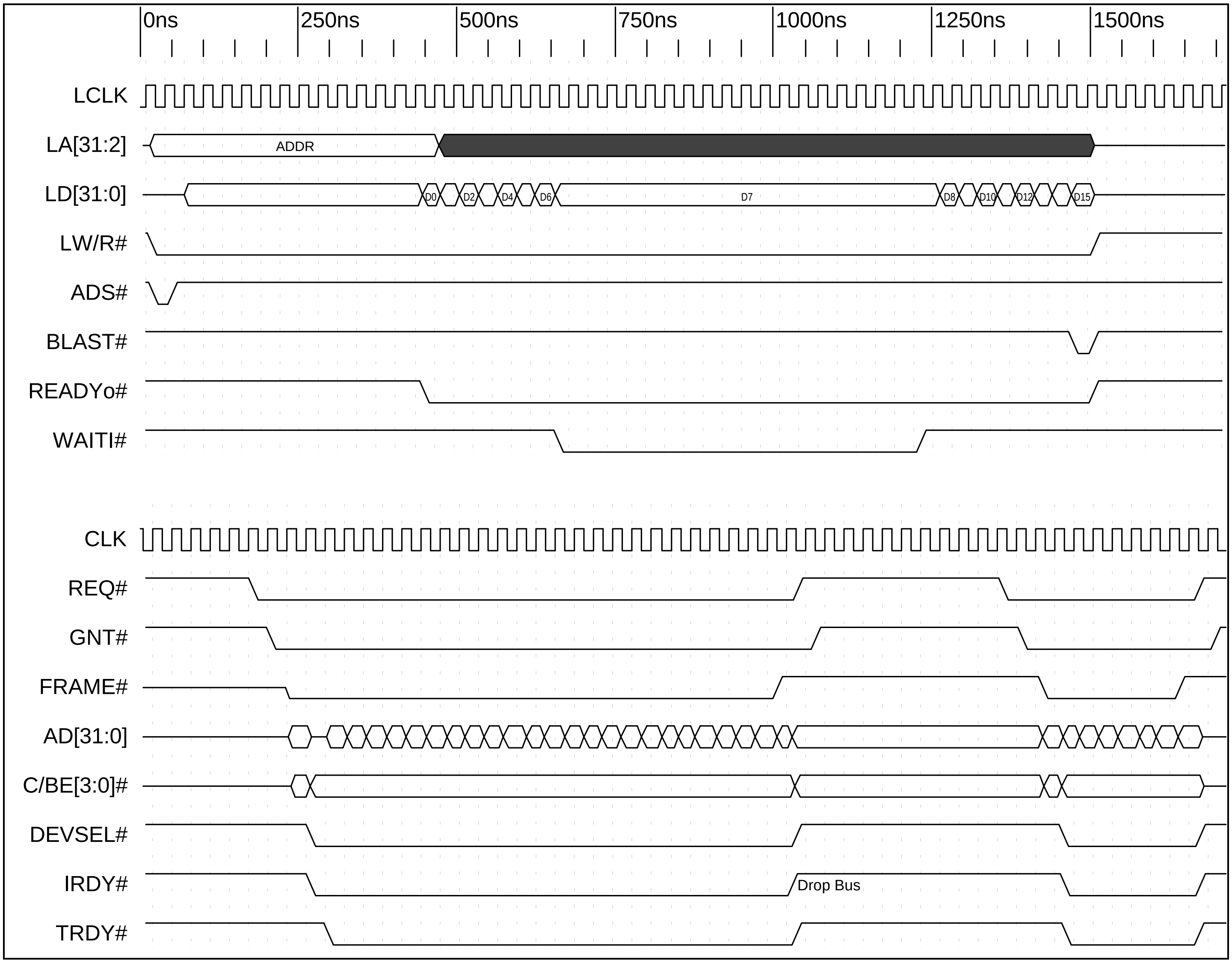
Timing Diagram 8-40. (C Mode) Local Bus Direct Master Read from PCI I/O



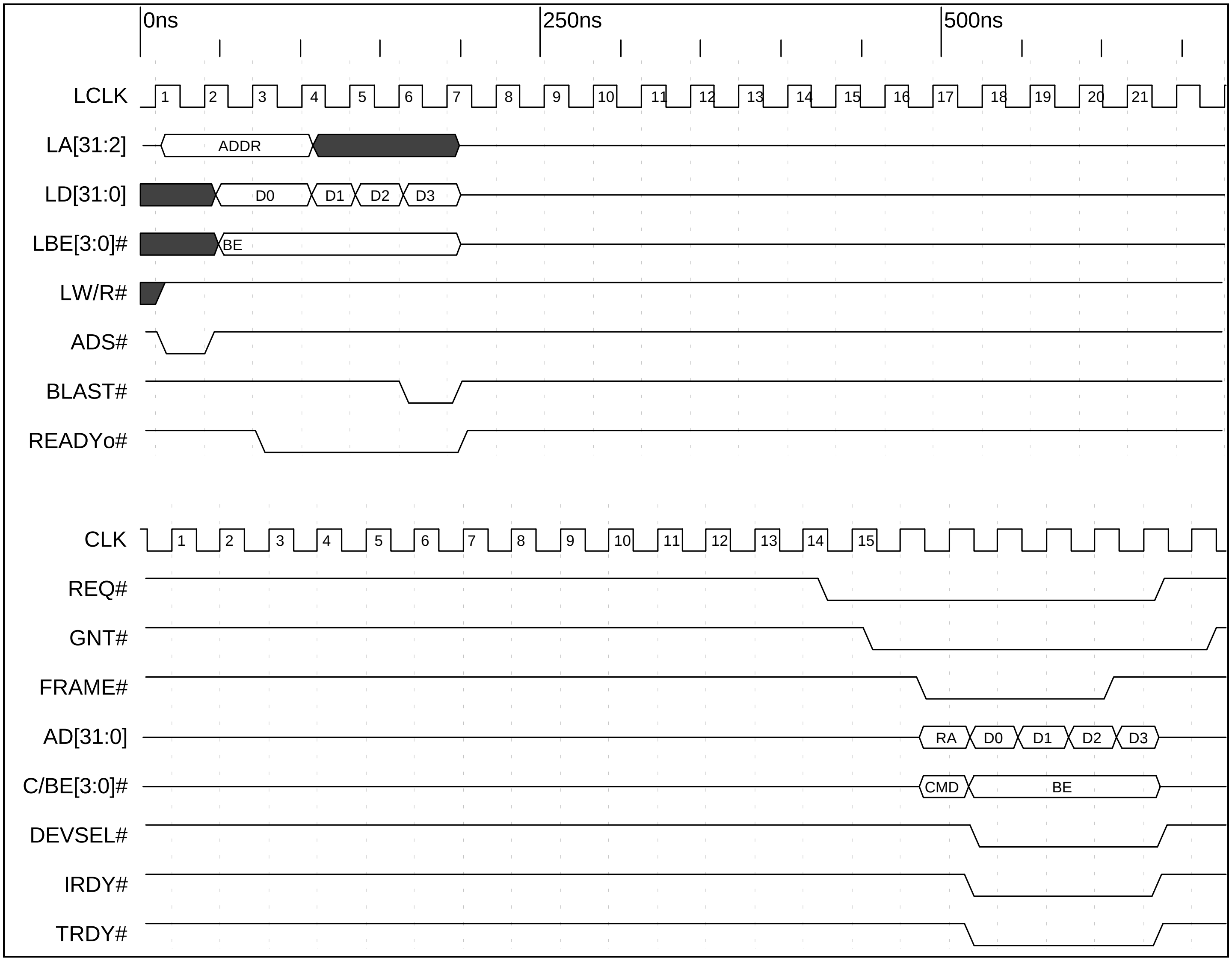
Timing Diagram 8-41. (C Mode) Direct Master Write to PCI I/O



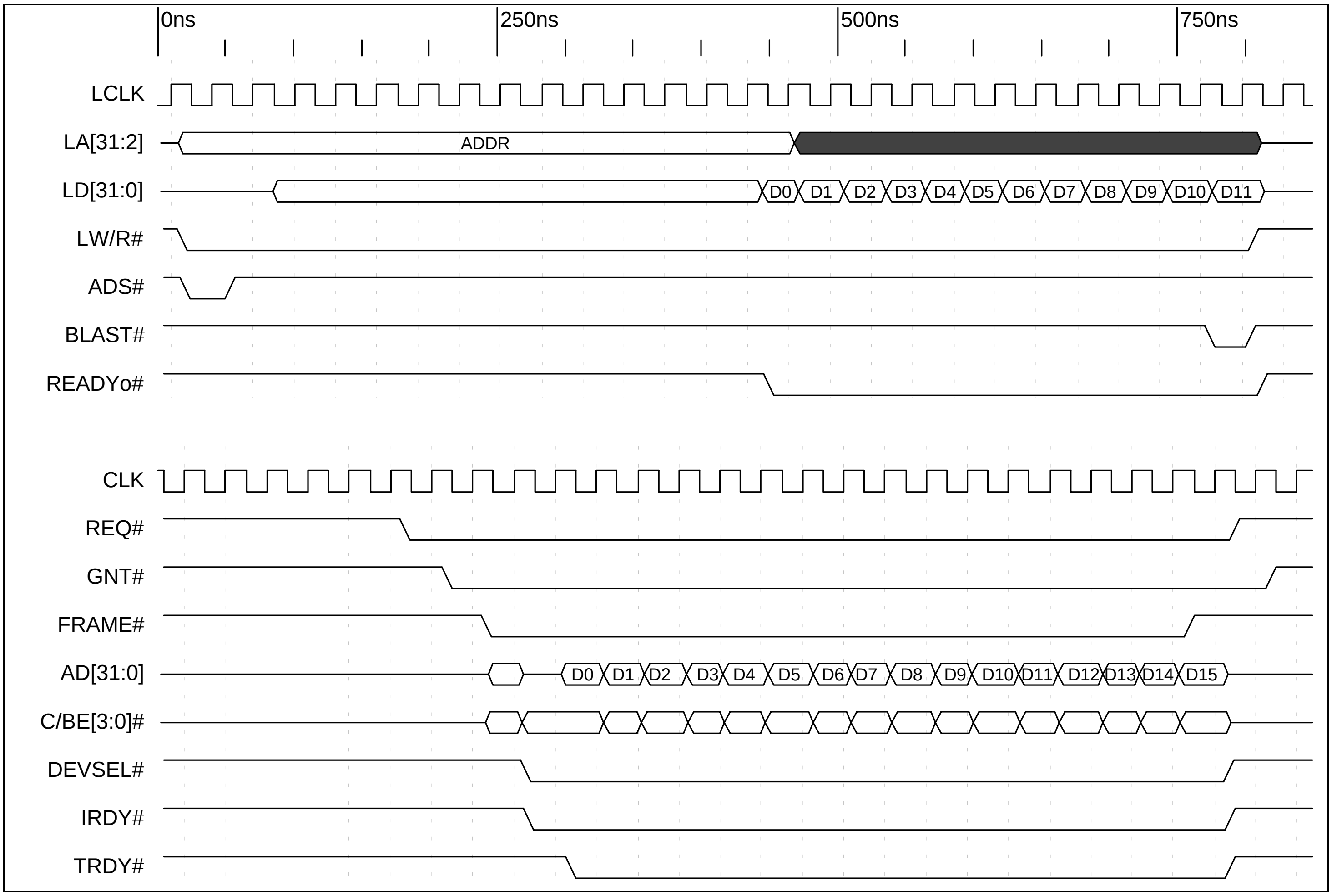
Timing Diagram 8-42. (C Mode) PCI 9080 Direct Master Memory Read—Keep Bus



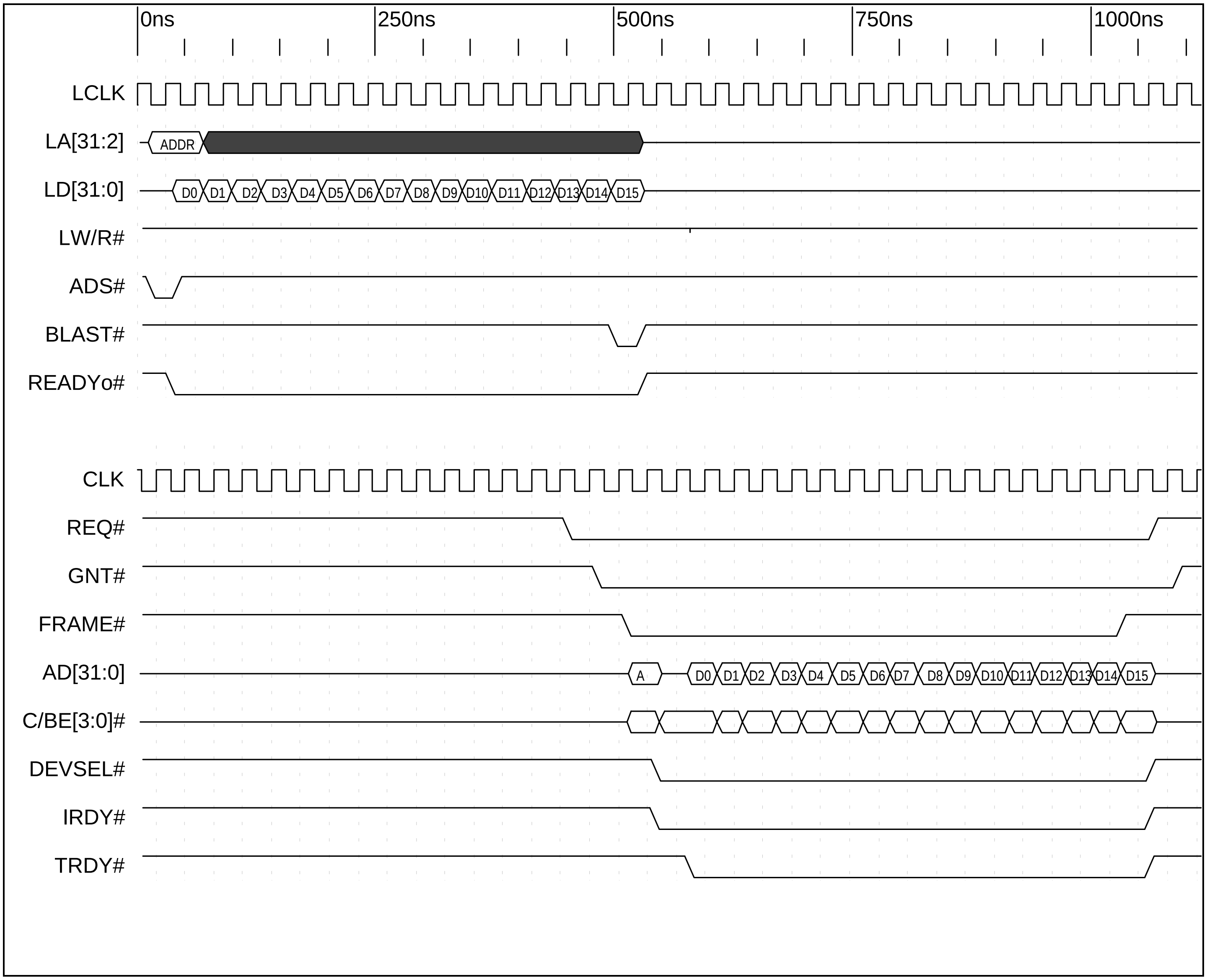
Timing Diagram 8-43. (C Mode) PCI 9080 Direct Master Memory Read—Drop Bus



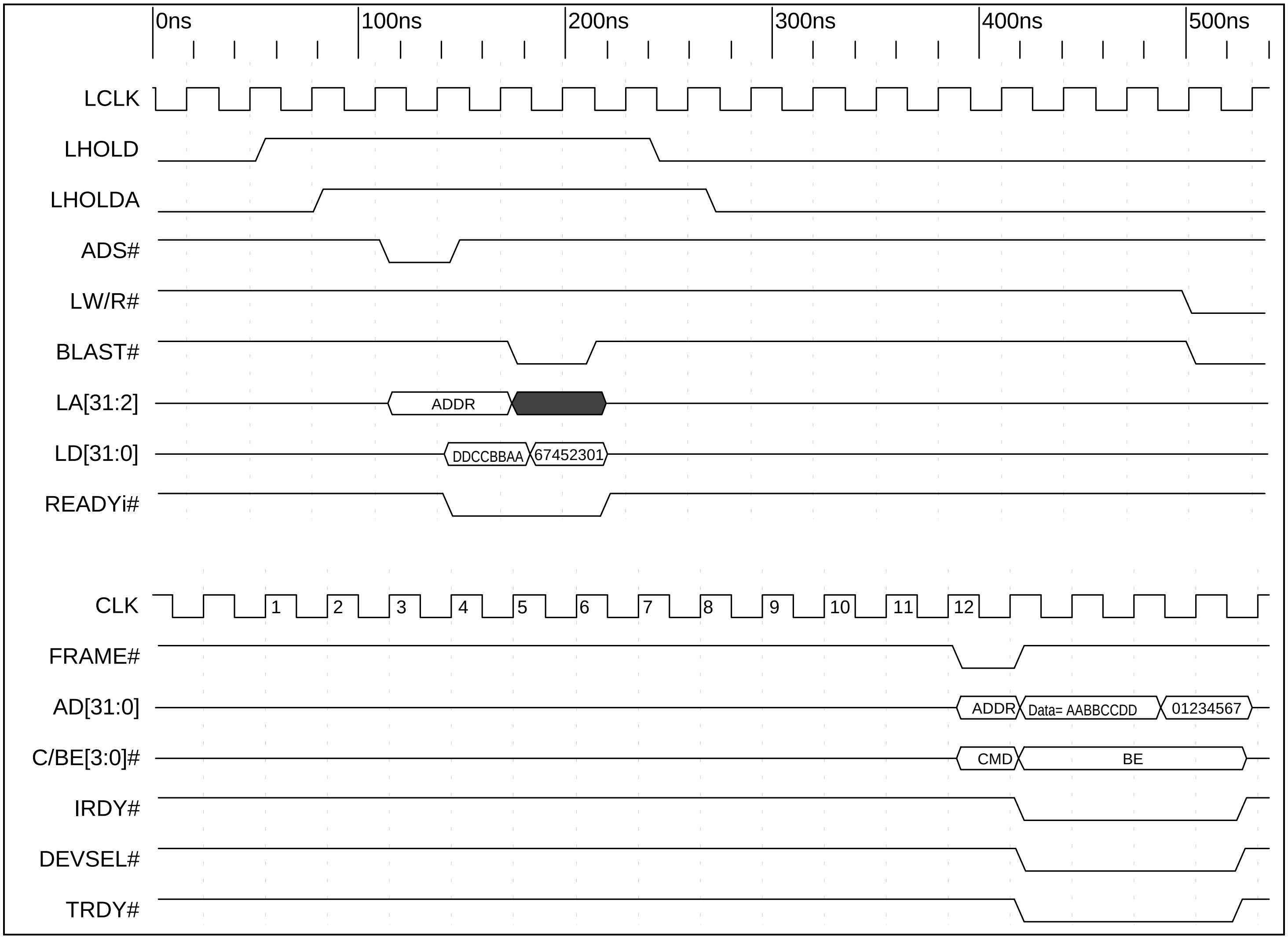
Timing Diagram 8-44. (C Mode) PCI Bus Request (REQ#) Delay During Direct Master Write (8 PCI Clock Delay)



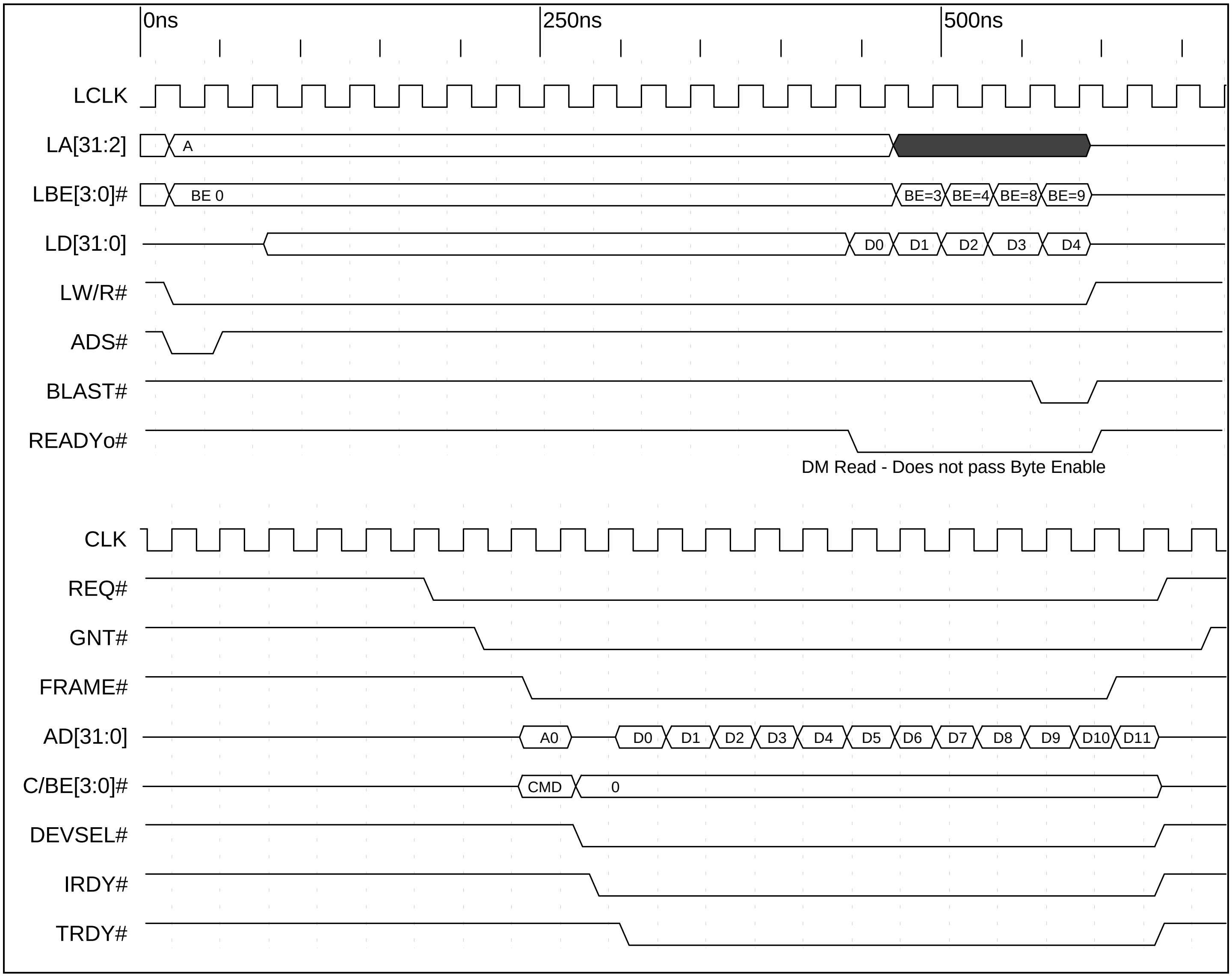
Timing Diagram 8-45. (C Mode) Direct Master Memory Read, Prefetch of 16



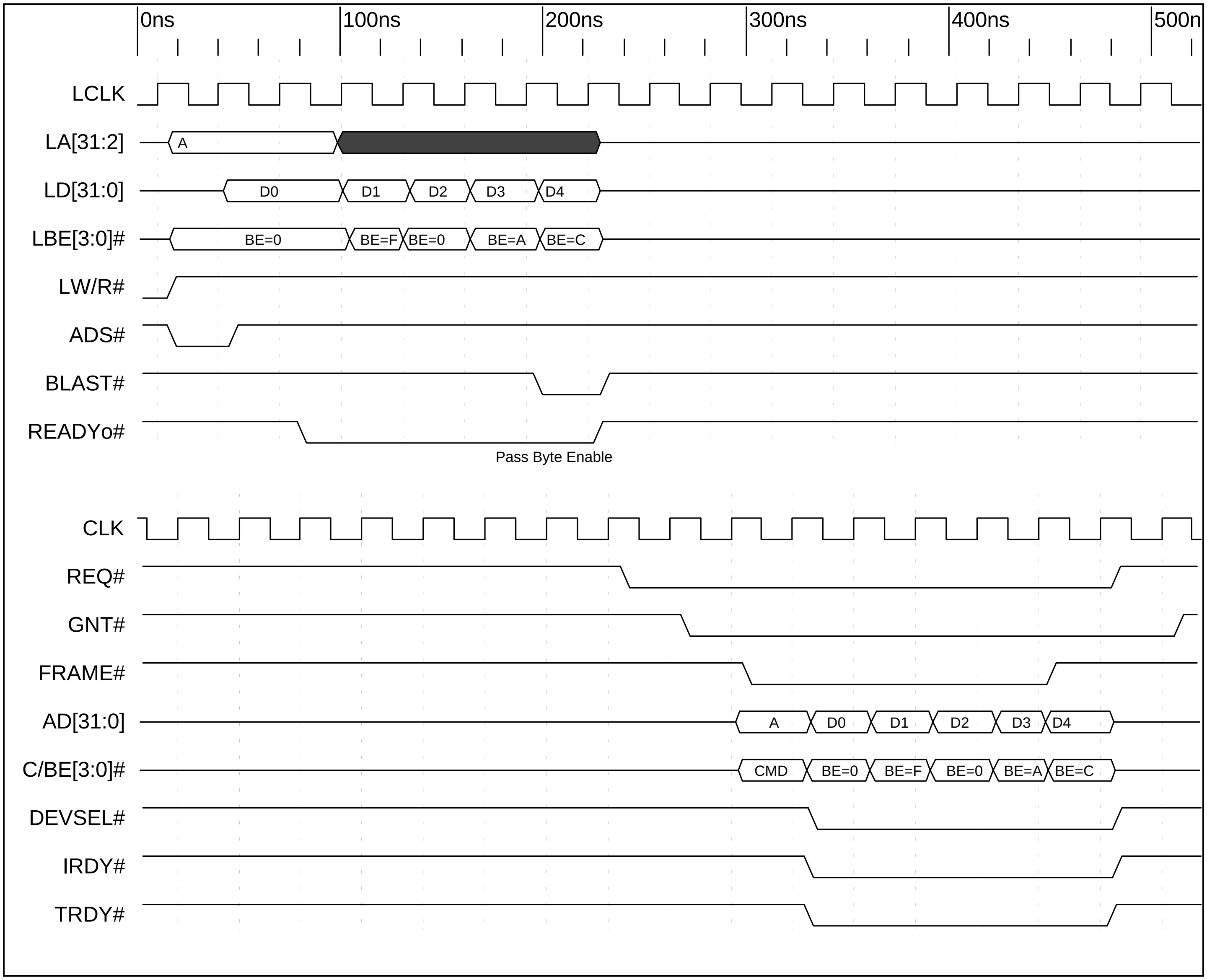
Timing Diagram 8-46. (C Mode) Direct Master Memory Write and Invalidate (MWI)—Cache Line Size of 8



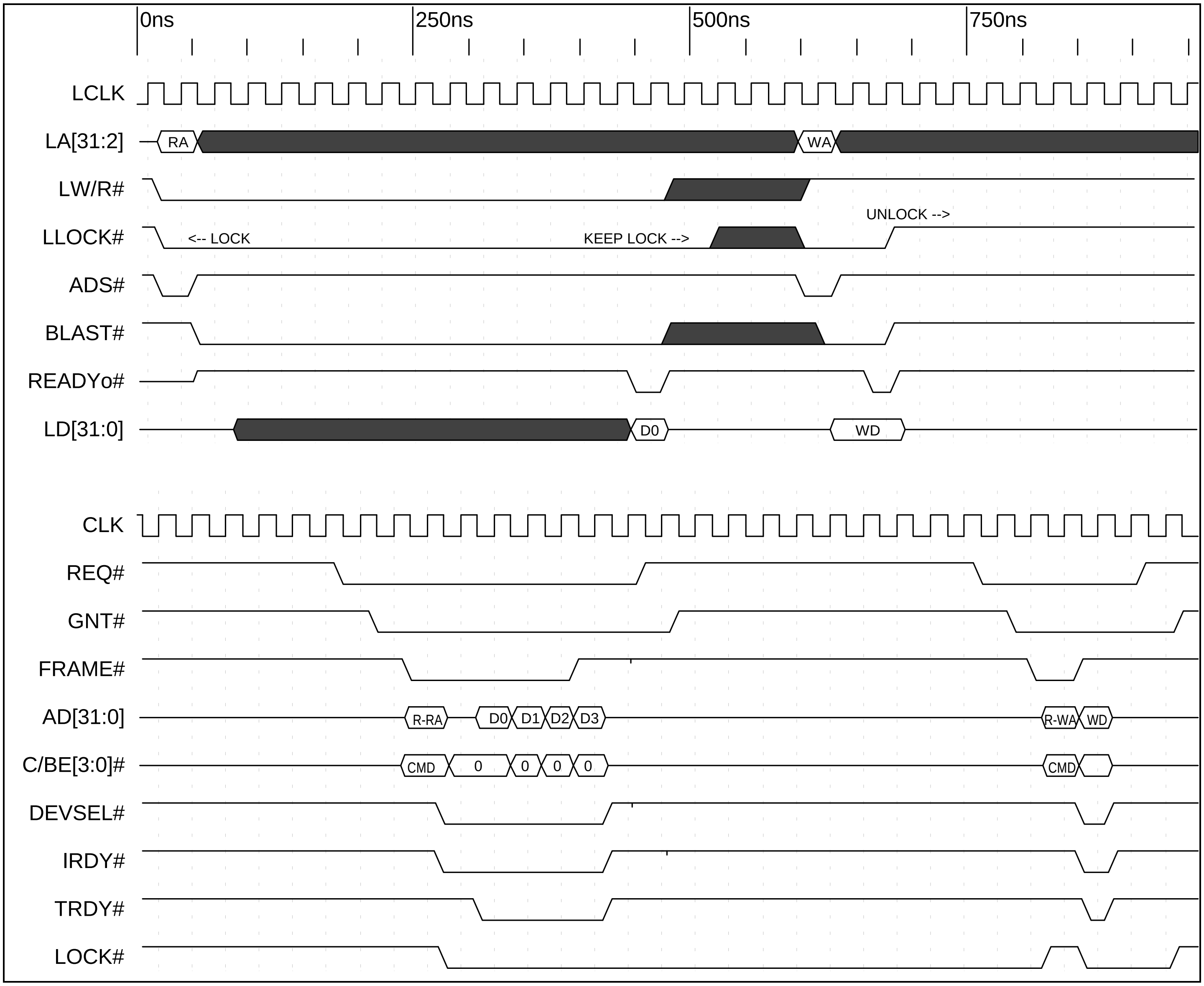
Timing Diagram 8-47. (C Mode) Direct Master in BIGEND Local Bus with BIGEND# Input or Interrupt



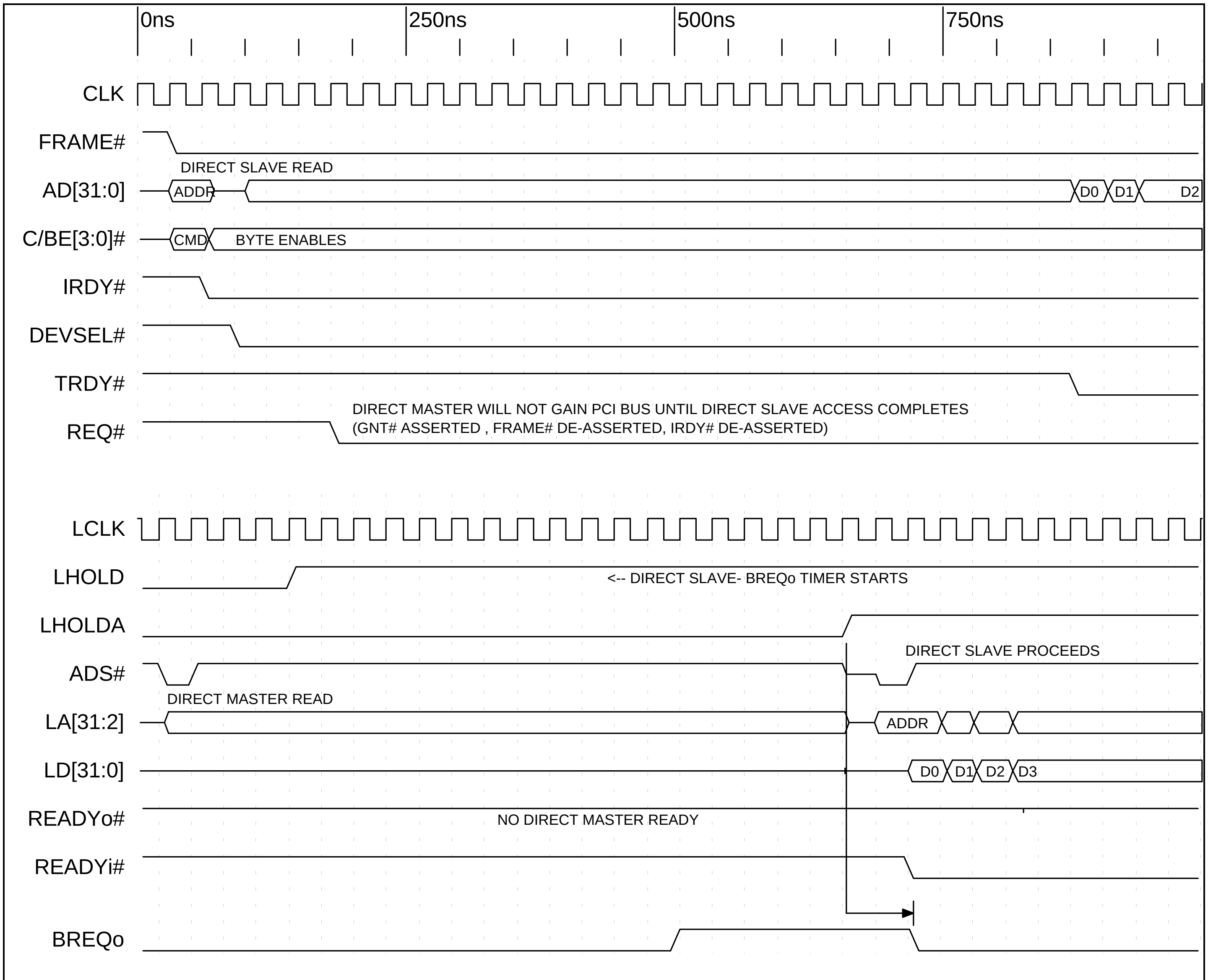
Timing Diagram 8-48. (C Mode) Direct Master Burst, Memory Read Cycles (Changing LBE[3:0]#)



Timing Diagram 8-49. (C Mode) Direct Master Five Lword Burst Write (Changing LBE[3:0]#)



Timing Diagram 8-50. (C Mode) Direct Master Locked Read Followed by Write and Release (LLOCK# and LOCK#)



BREQo timer expires and asserts BREQo to indicate potential deadlock condition.

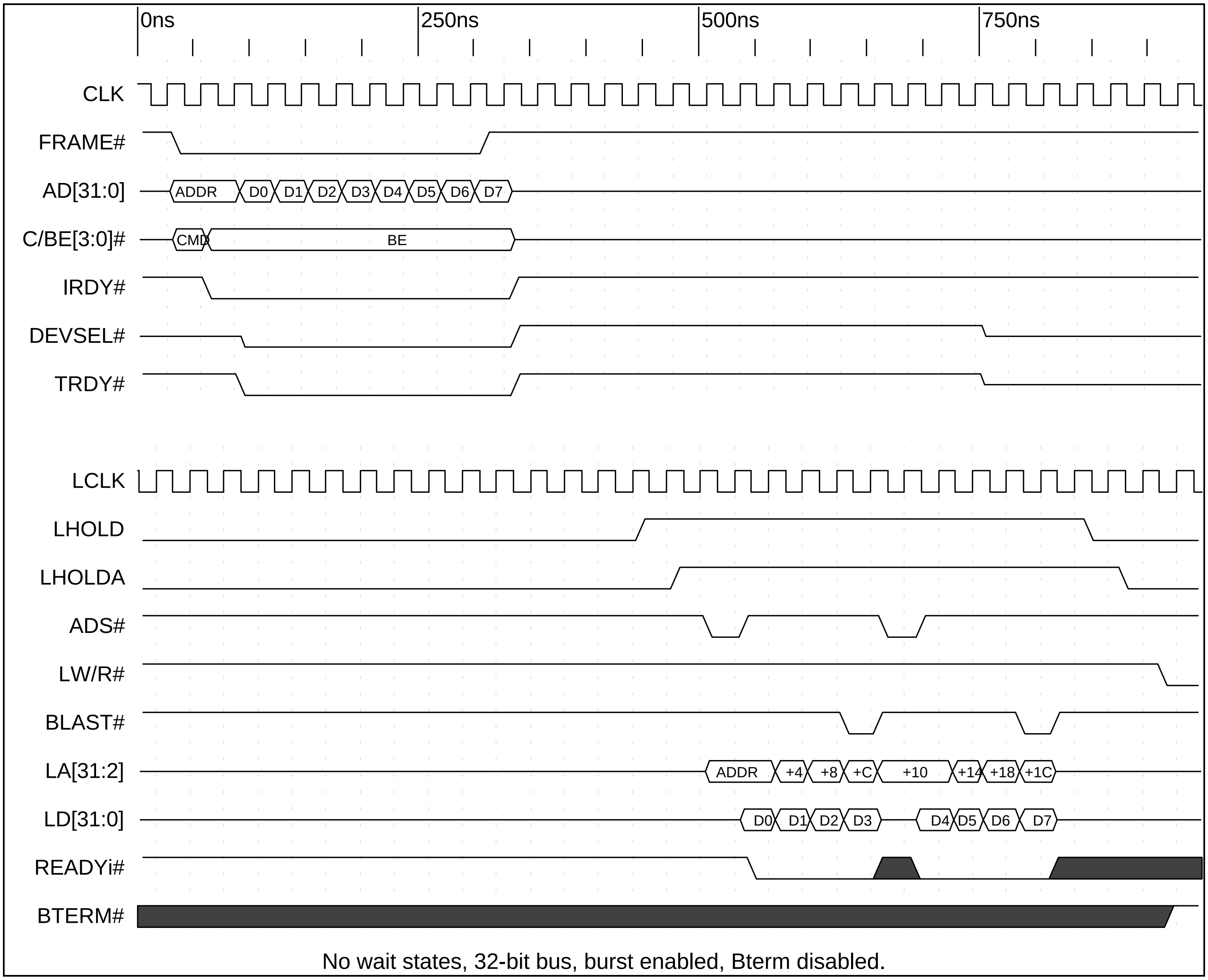
External logic backs off Direct Master transfer and asserts LHOLDA to grant the local bus for a Direct Slave transfer.

Note: *For partial deadlock, PCI retry timer bits [31:28] of the Local Bus Region Descriptor Register can be used to issue RETRYs to the PCI Master attempting the Direct Slave access.*

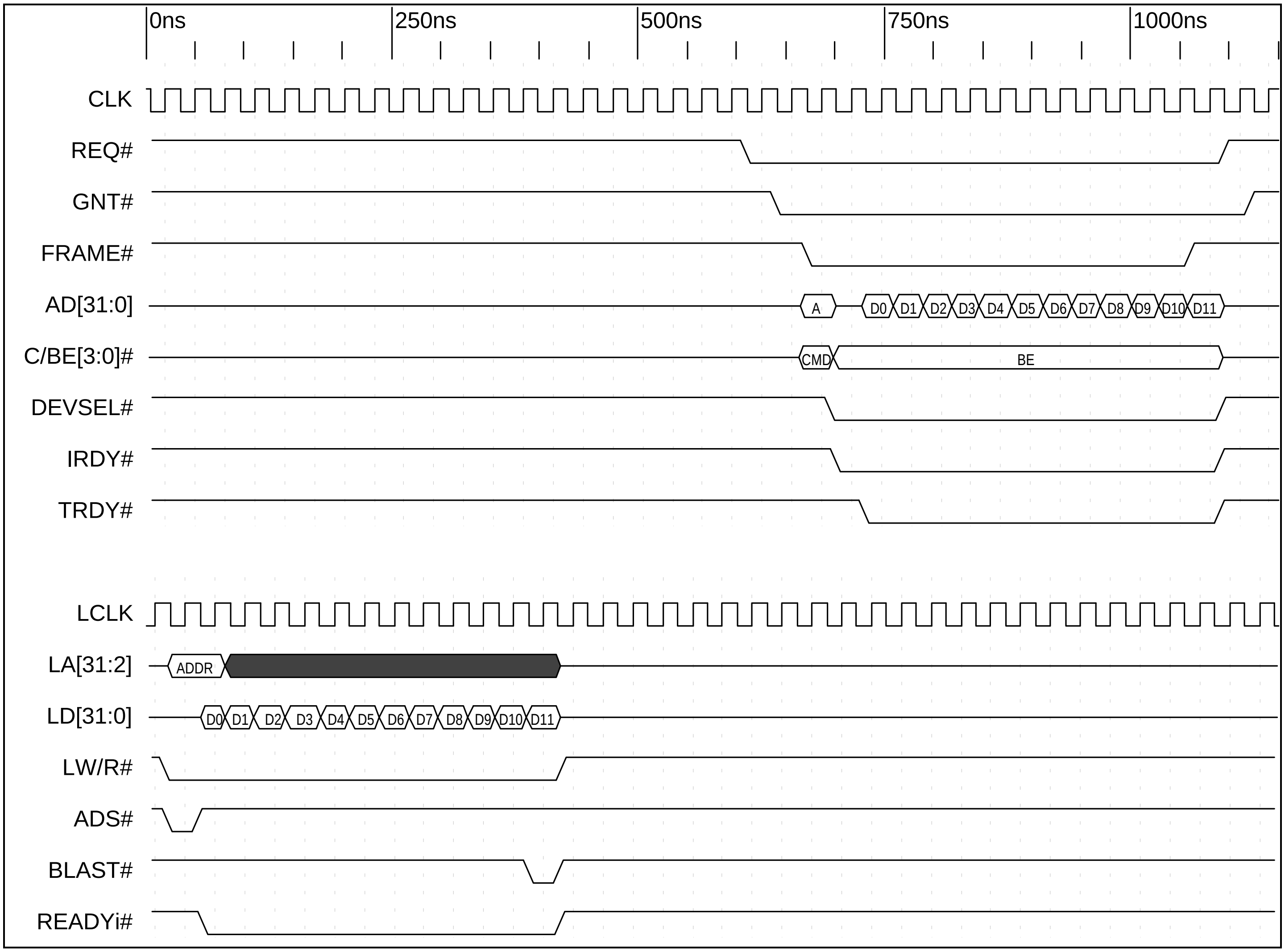
Refer to Section 3, “Functional Description,” for a description of deadlock.

Timing Diagram 8-51. (C Mode) BREQo and Deadlock

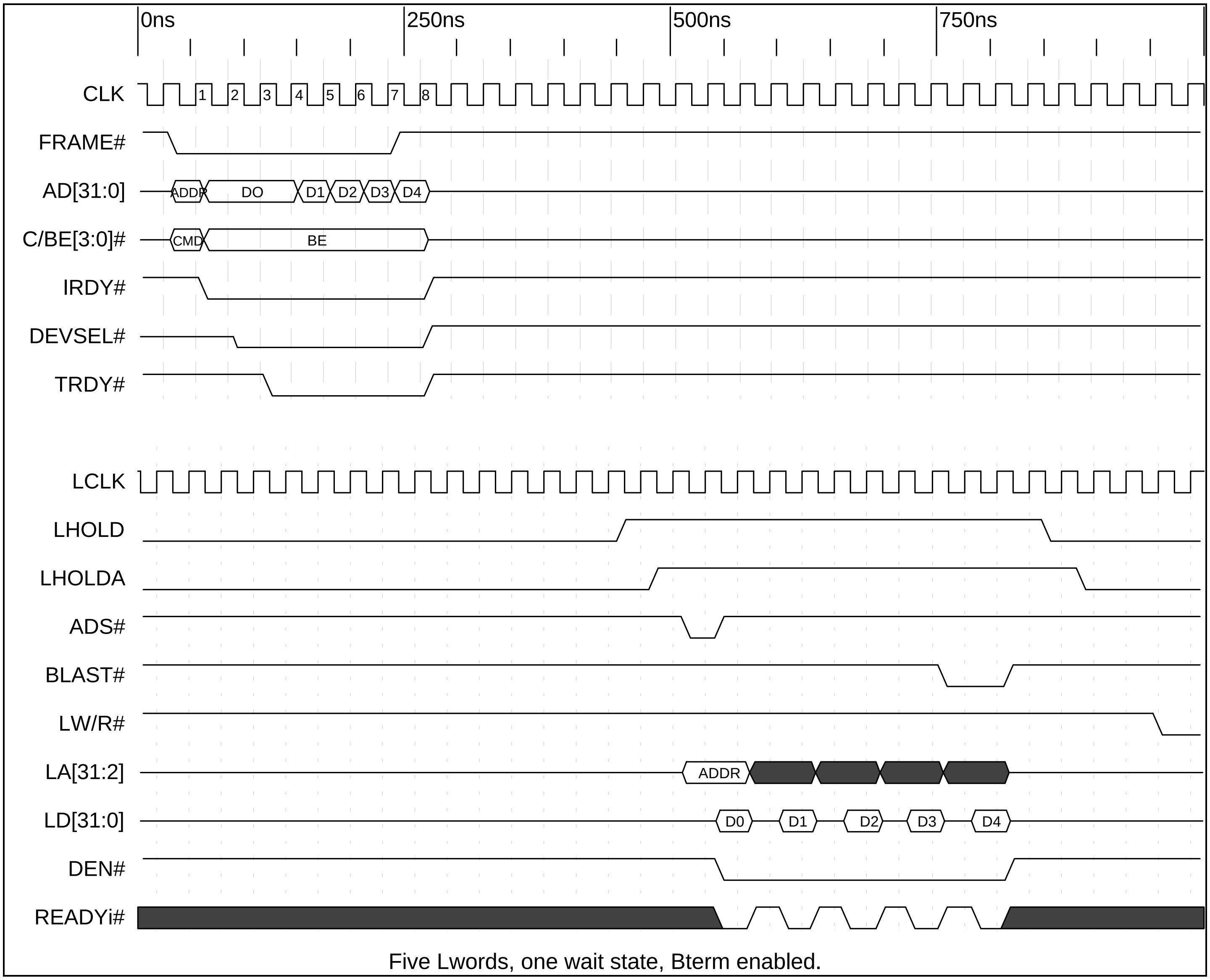
8.3.3 C Mode DMA



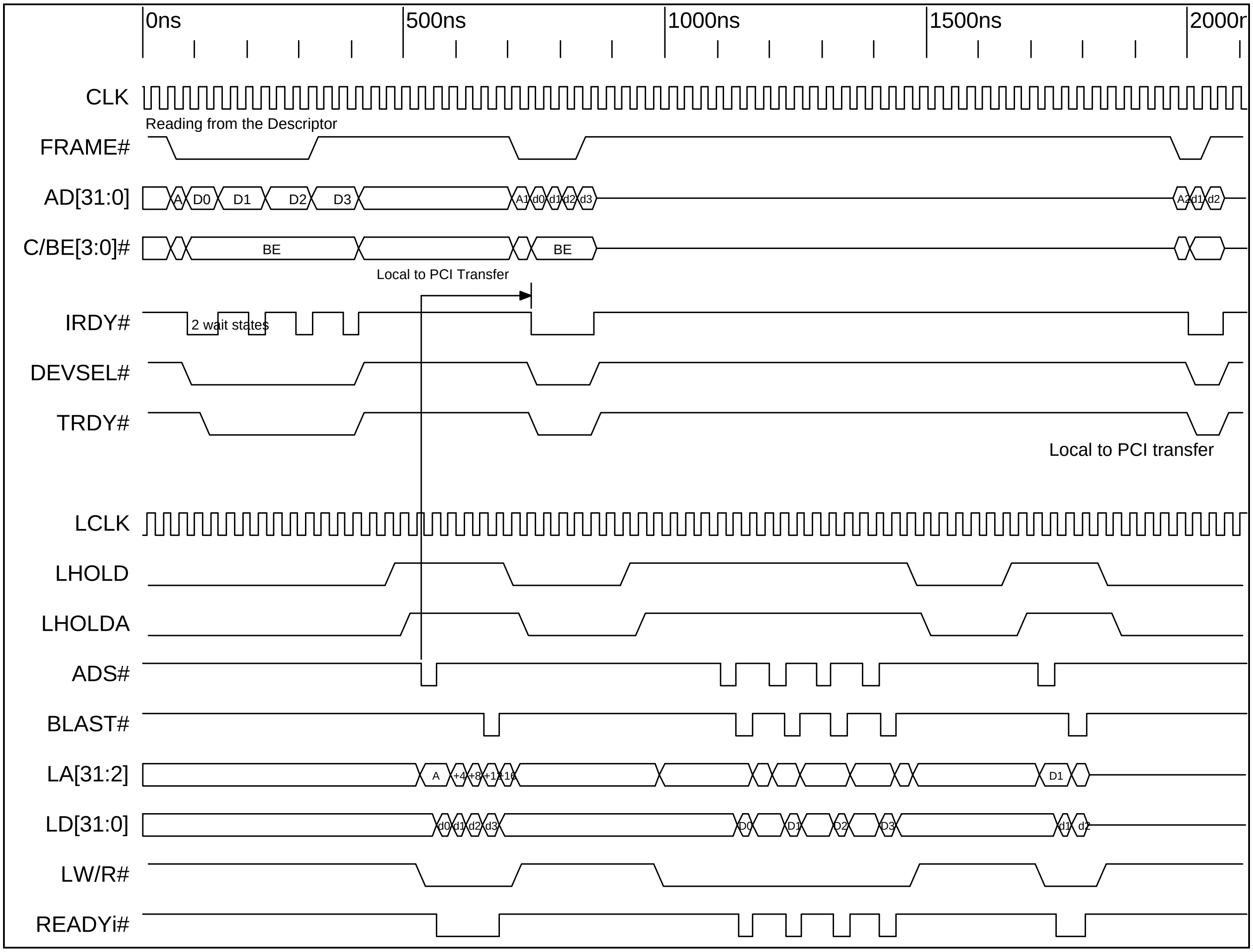
Timing Diagram 8-52. (C Mode) DMA Aligned PCI Address to Aligned Local Address, Bterm Disabled



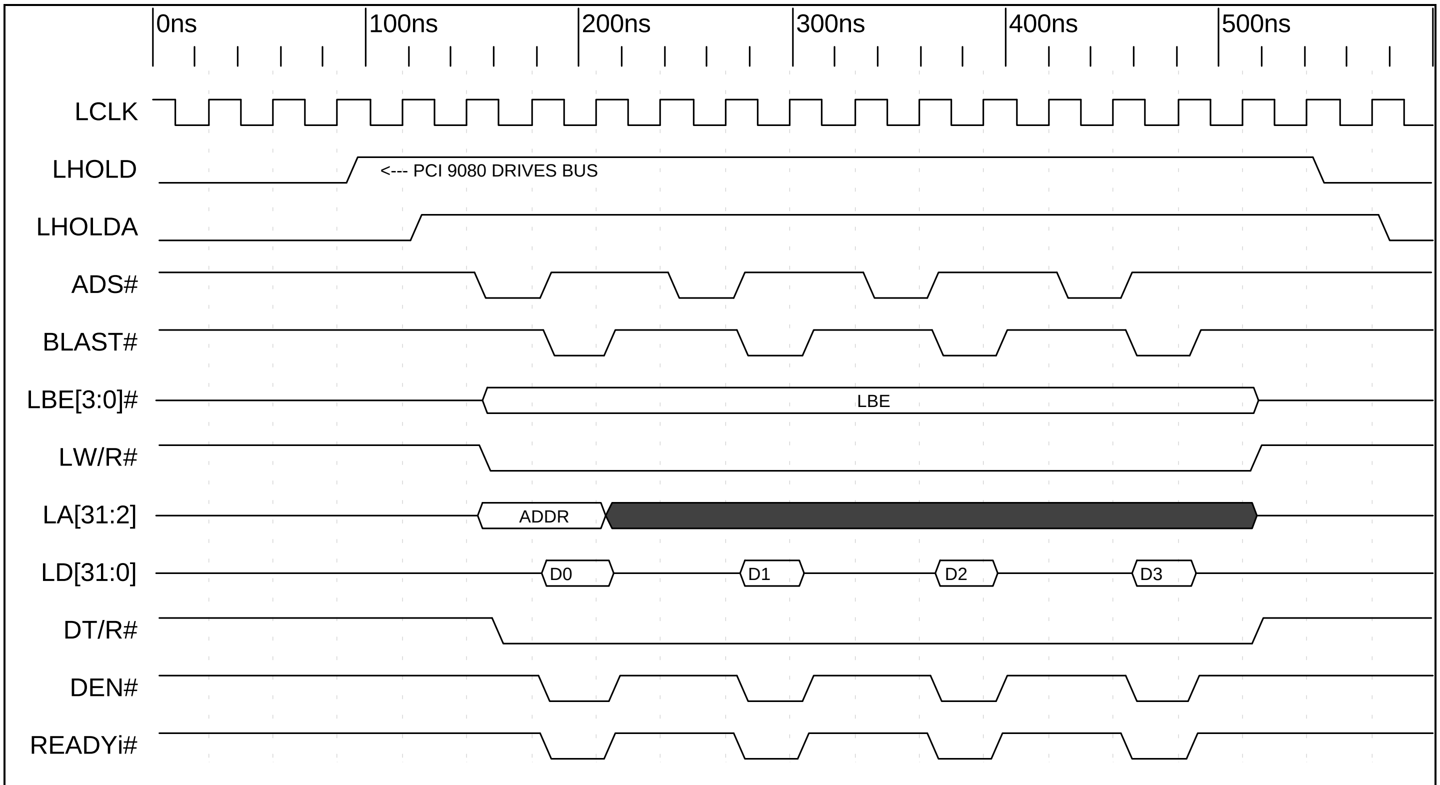
Timing Diagram 8-53. (C Mode) DMA Aligned Local Address to Aligned PCI Address, Bterm Enabled



Timing Diagram 8-54. (C Mode) DMA Aligned PCI Address to Aligned Local Address (External Generation of Wait States)



Timing Diagram 8-55. (C Mode) Read of DMA Chaining Parameters from PCI and Local Buses

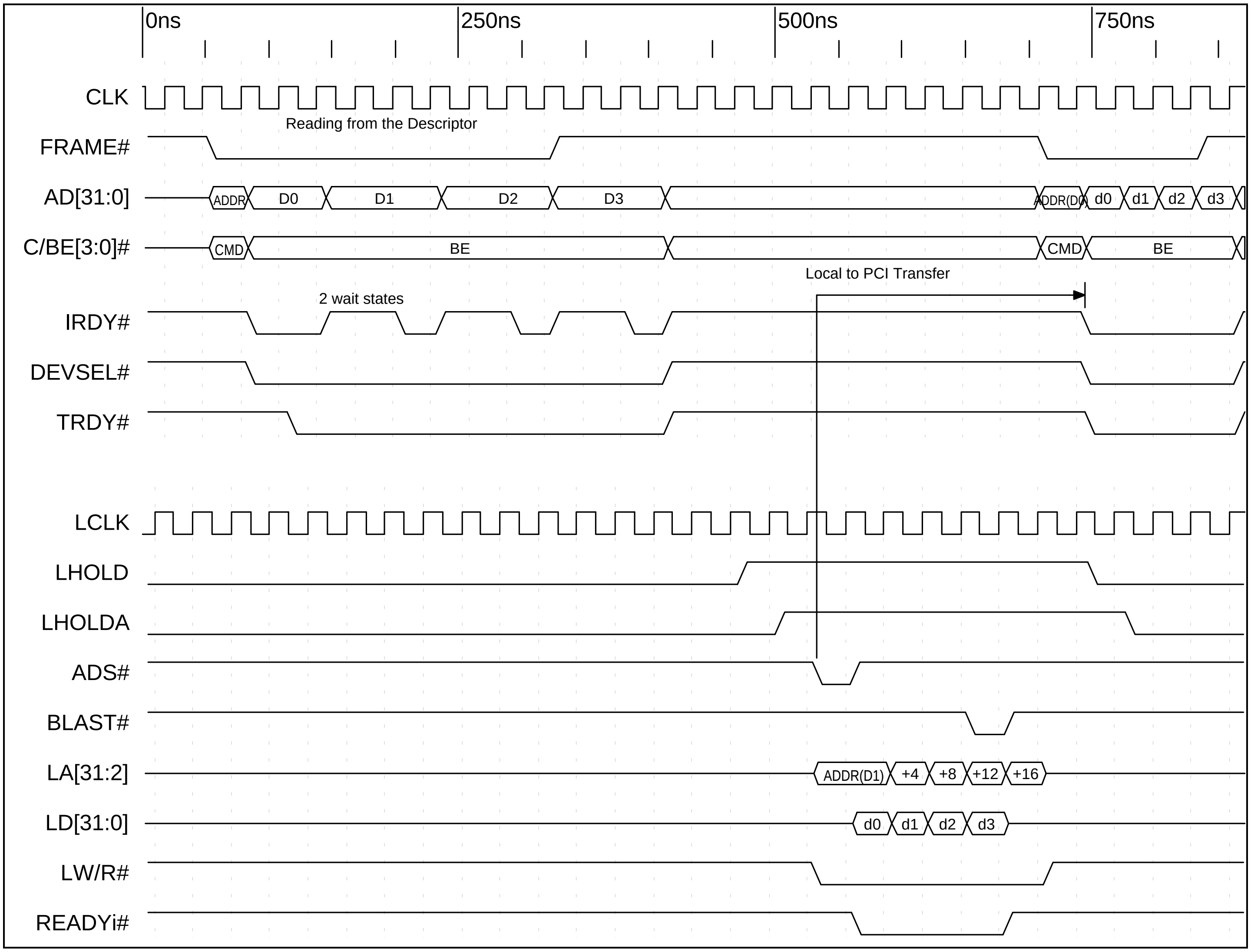


First Address A are bits [31:4] of next Descriptor Pointer Register.

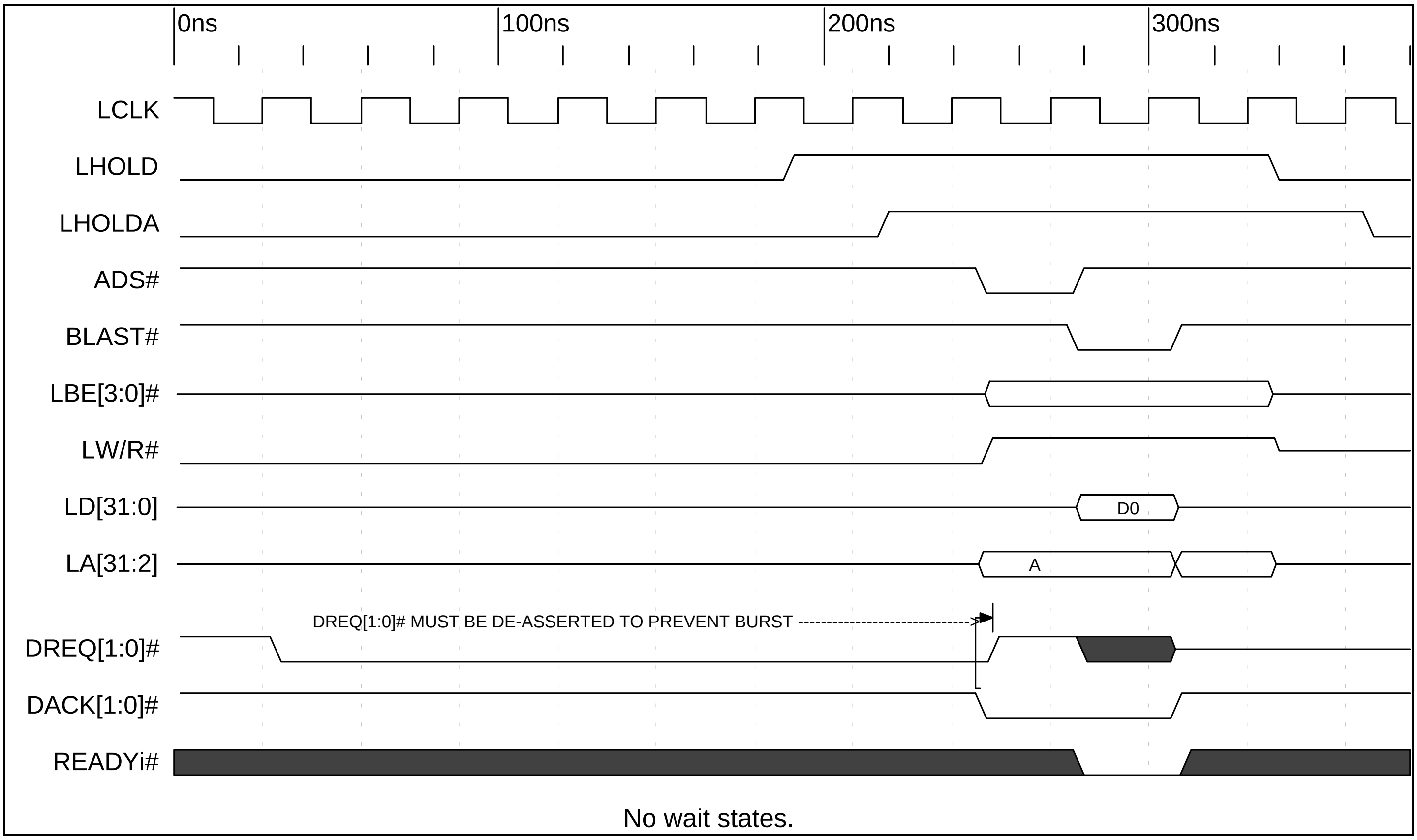
D0: PCI start address
D1: Local start address
D2: Transfer count (bytes)
D3: Next descriptor pointer

No wait states.

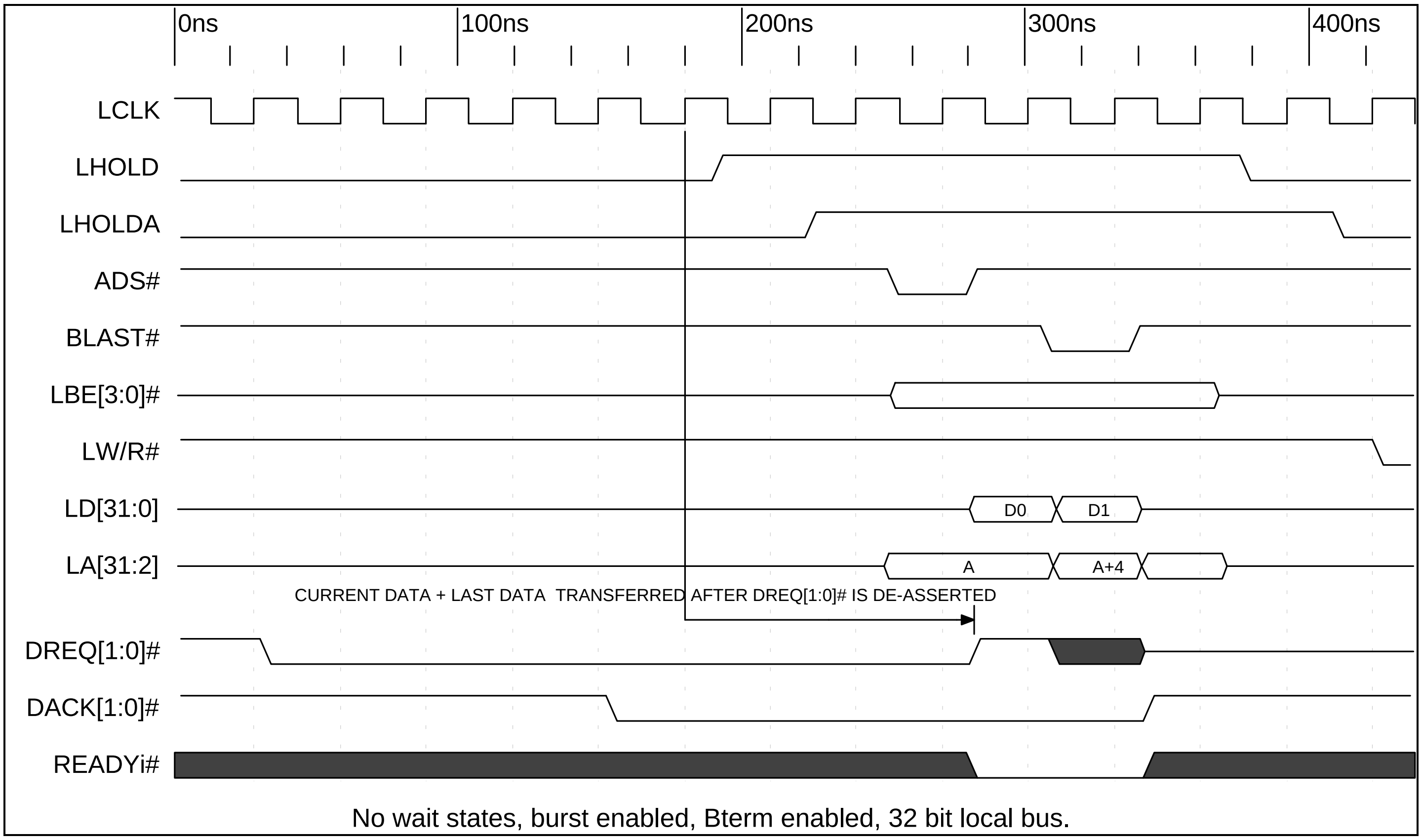
Timing Diagram 8-56. (C Mode) PCI 9080 DMA Read of Chaining Parameters from Local Bus



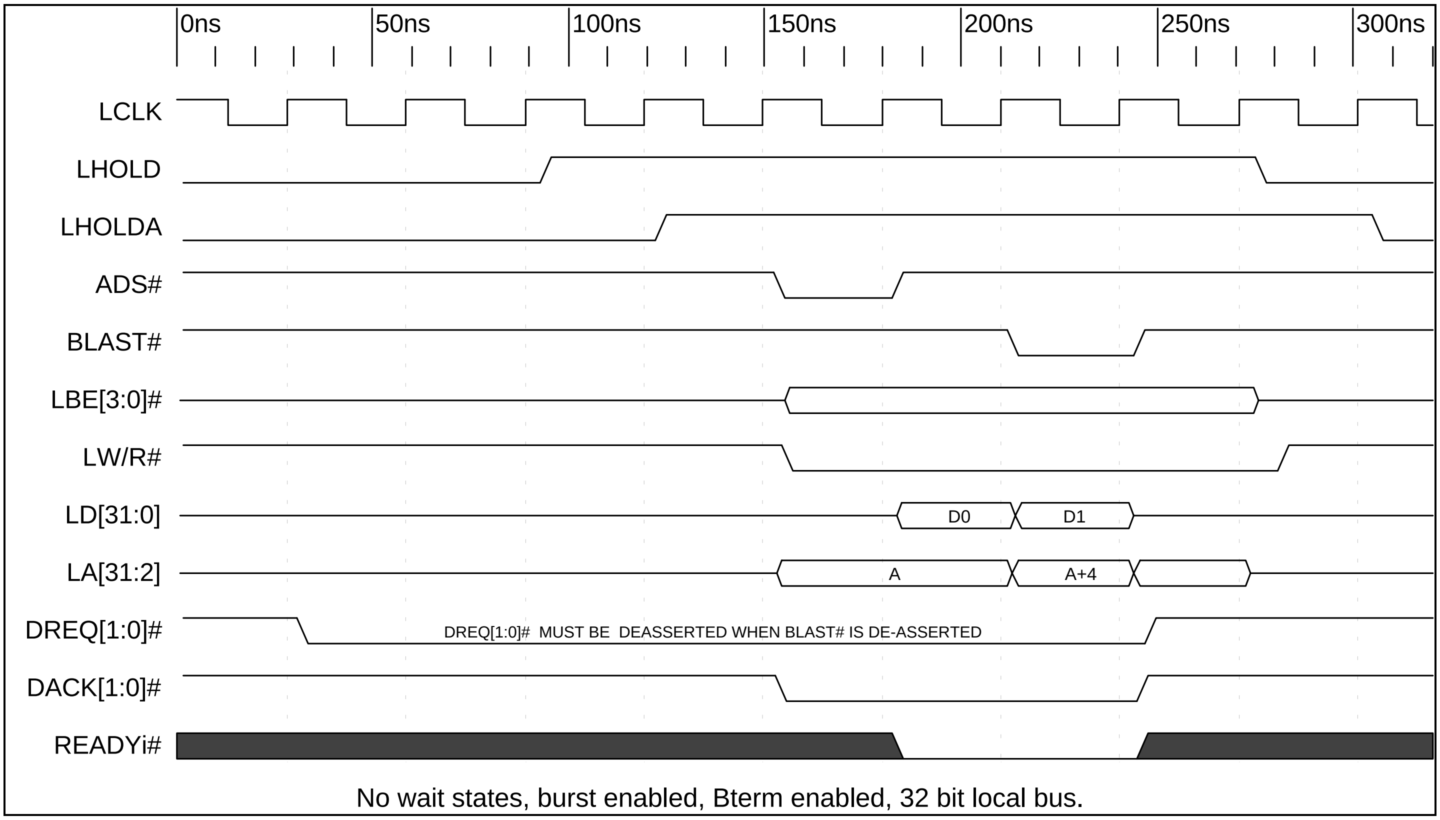
Timing Diagram 8-57. (C Mode) Read of DMA Chaining Parameters from PCI Bus (Local to PCI Transfer)



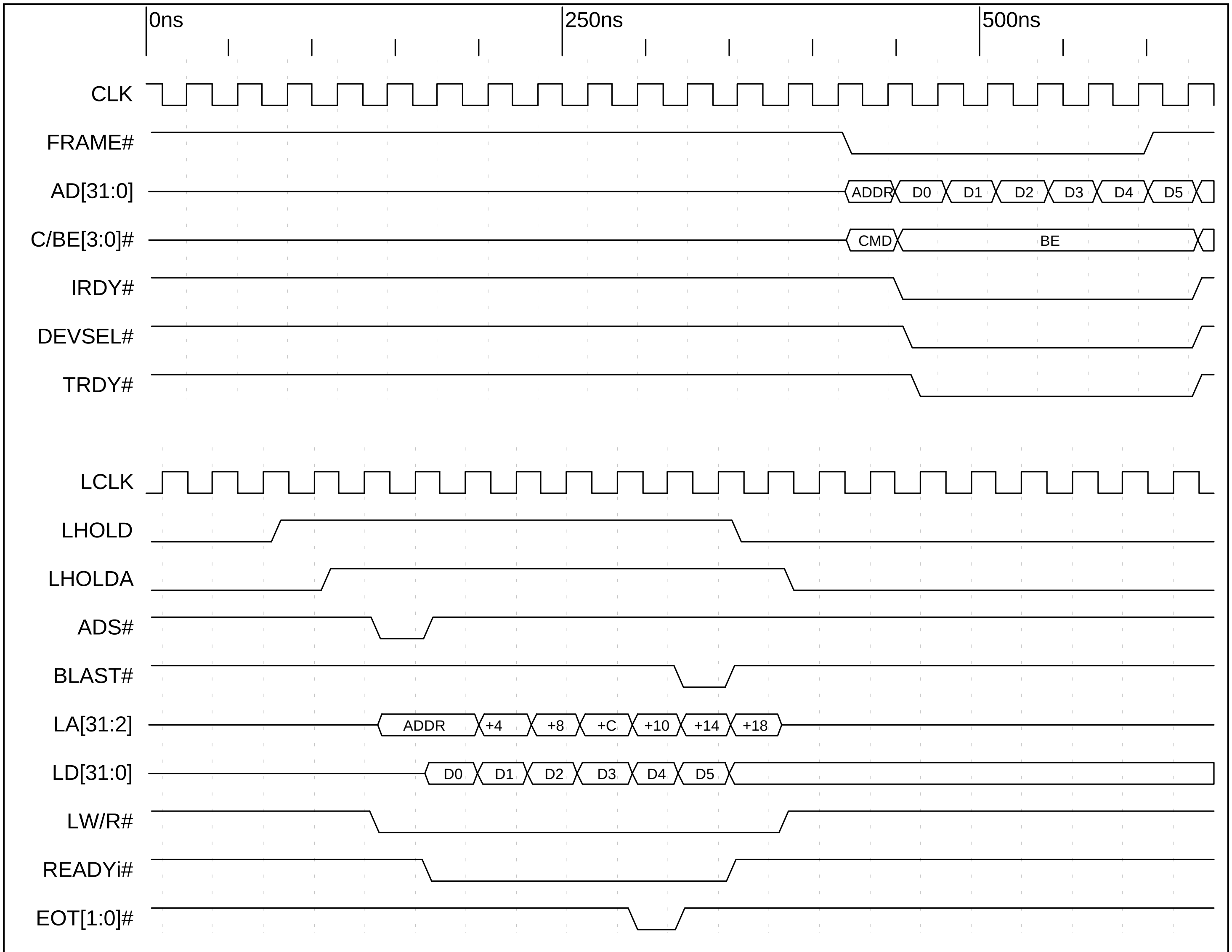
Timing Diagram 8-58. (C Mode) Single Cycle DMA Demand Mode PCI to Local


0ns
100ns
200ns
300ns
400ns
LCLK
LHOLD
LHOLDA
ADS#
BLAST#
LBE[3:0]#
LW/R#
LD[31:0]
D0
D1
LA[31:2]
A
A+4
CURRENT DATA + LAST DATA TRANSFERRED AFTER DREQ[1:0]# IS DE-ASSERTED
DREQ[1:0]#
DACK[1:0]#
READYi#
No wait states, burst enabled, Bterm enabled, 32 bit local bus.

Timing Diagram 8-59. (C Mode) Multiple Cycle (Burst) DMA Demand Mode PCI to Local with No Wait States

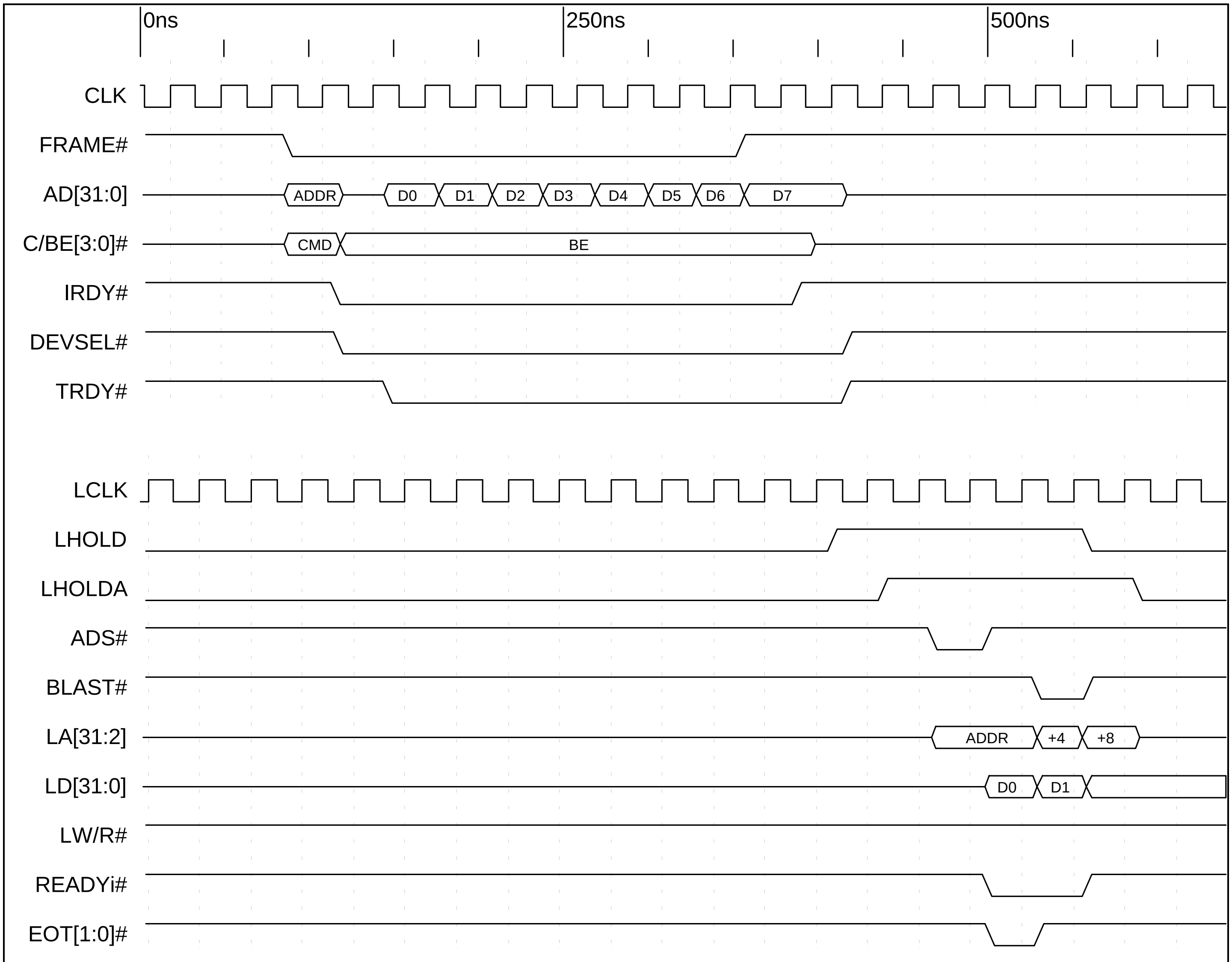


Timing Diagram 8-60. (C Mode) DMA Demand Mode Terminated with BLAST# (Local to PCI)



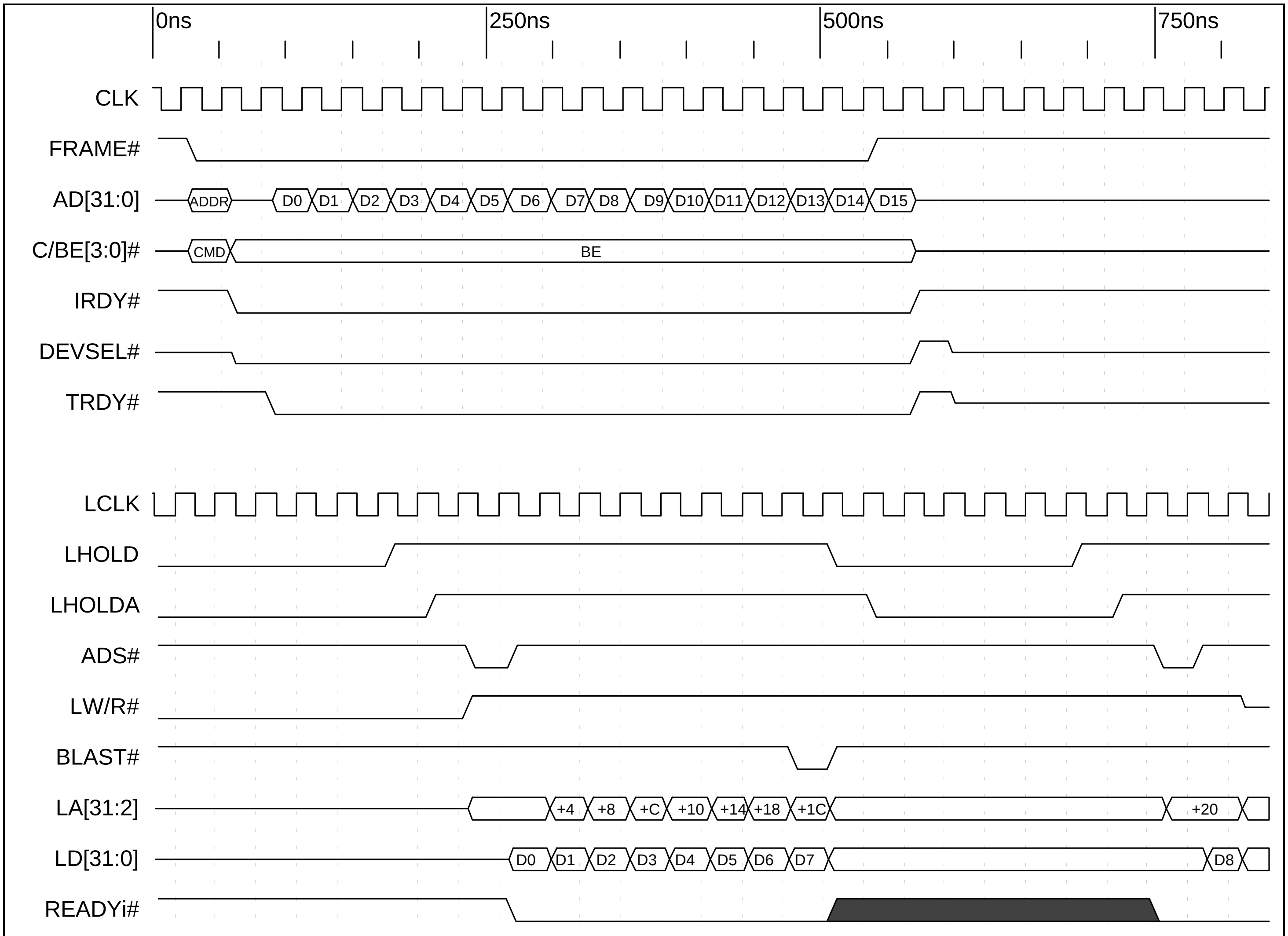
Stop Data Transfer Mode bit is not set. If this bit is set, there is no BLAST# and D5 is not transferred. See Table 4-62[15] or Table 4-67[15].

Timing Diagram 8-61. (C Mode) DMA Local to PCI, Terminated with EOT[1:0]#



Stop Data Transfer Mode bit is not set. If this bit is set, there is no BLAST# and D5 is not transferred. See Table 4-62[15] or Table 4-67[15].

Timing Diagram 8-62. (C Mode) DMA PCI to Local, Terminated with EOT[1:0]#

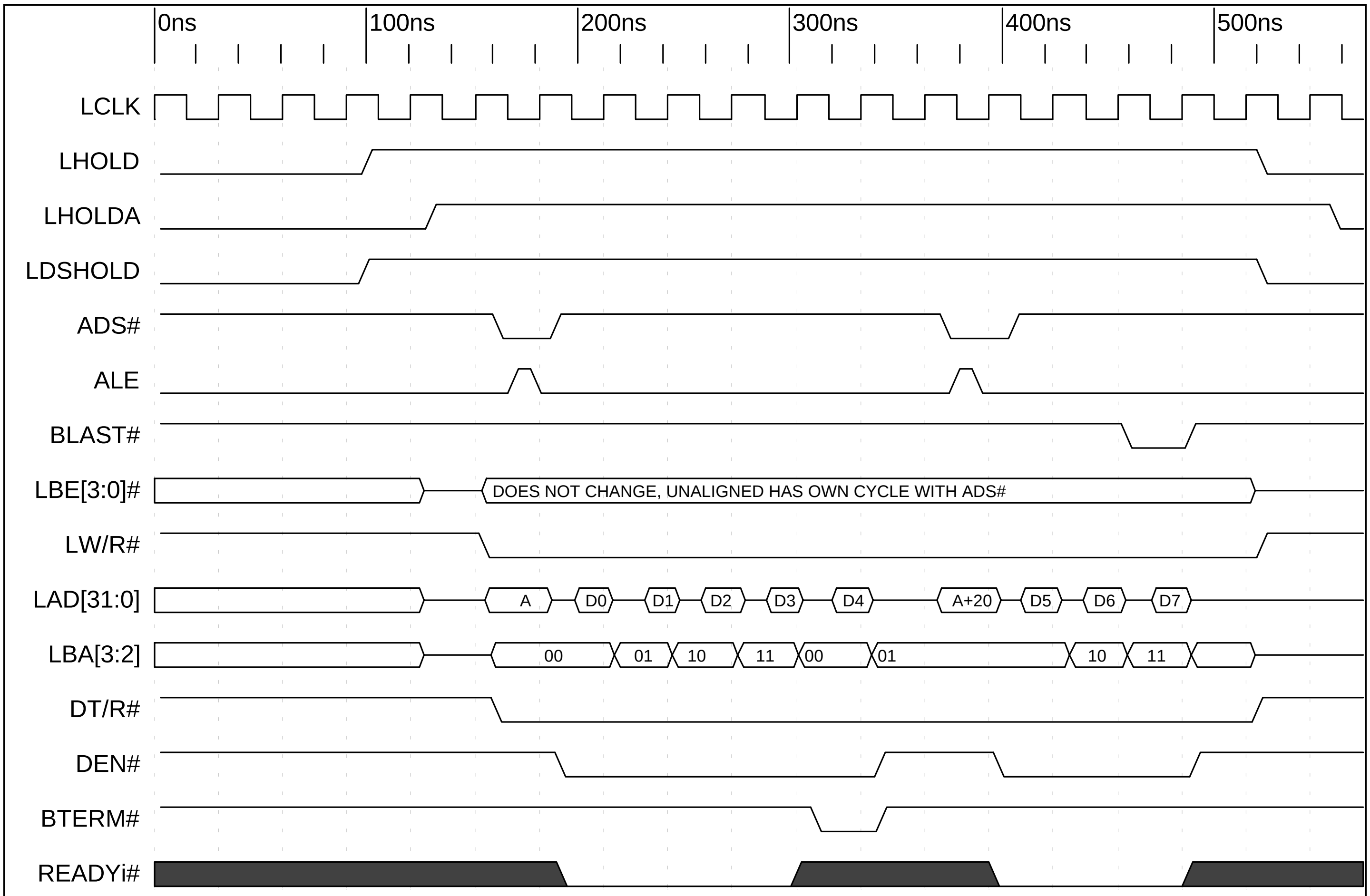


Latency Timer = 7 CLK, Pause Timer = 4 CLK.
PCI 9080 has internally added another clock to the pause timer.
Therefore, the time when LHOLD is de-asserted to when it is re-asserted is 5 CLKS for a pause timer of 4 CLK.

Timing Diagram 8-63. (C Mode) DMA PCI to Local with Local Pause Timer and Local Latency Timer

8.4 J MODE

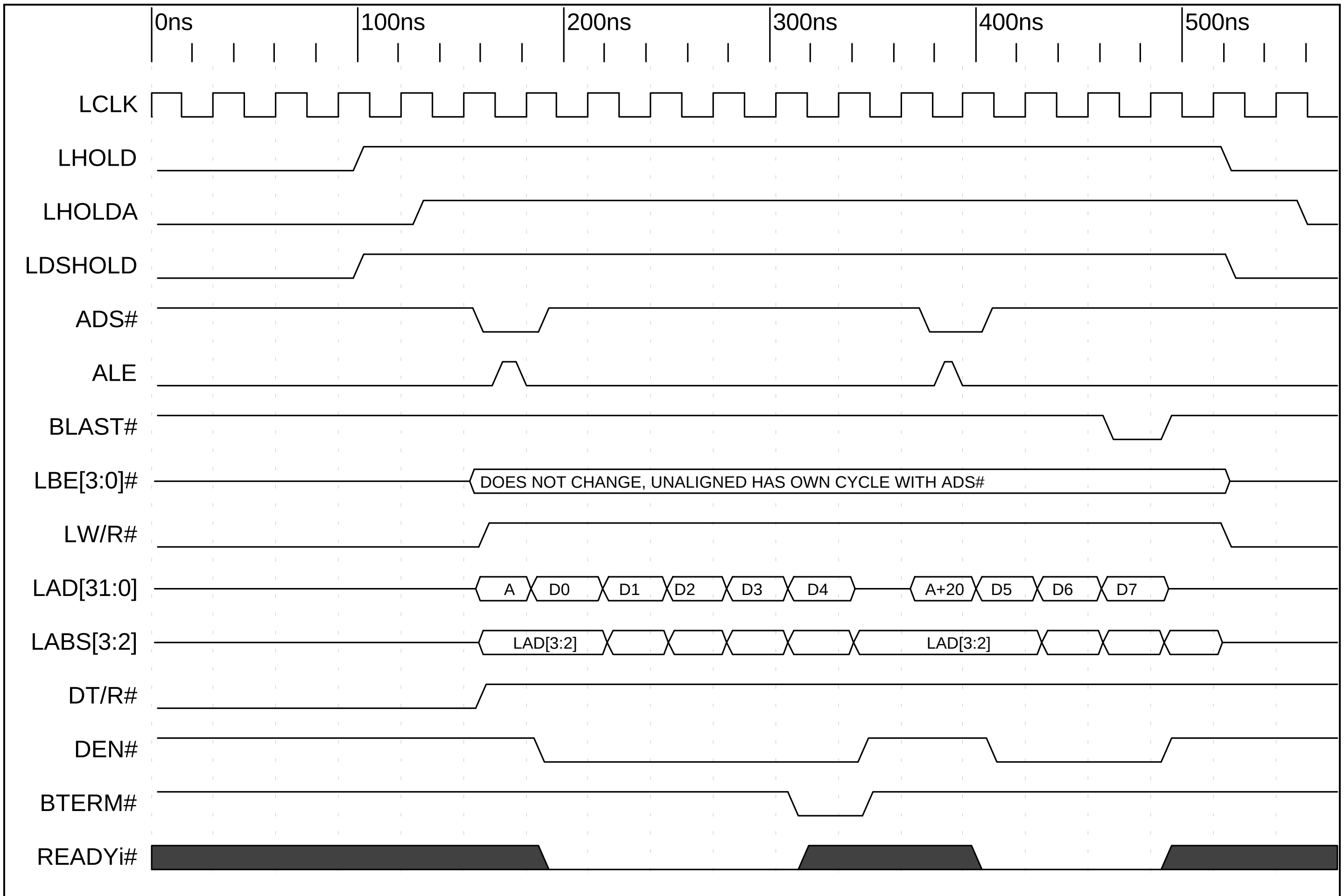
8.4.1 J Mode Direct Slave



Eight Lword burst, no wait states, burst enabled, Bterm enabled, 32 bit local bus.

Note: *If Bterm is disabled, a new ADS# cycle starts every quad Lword boundary.*

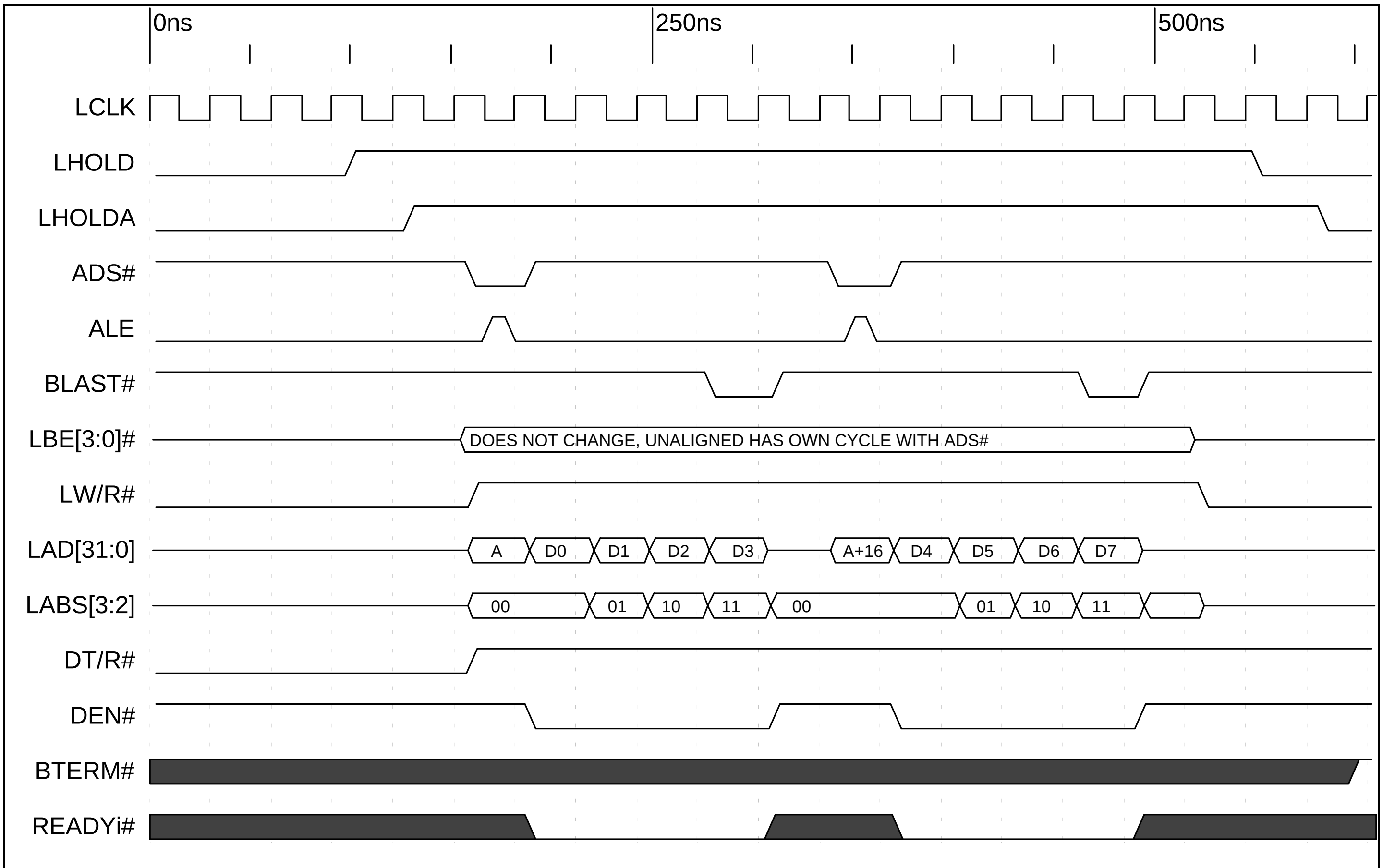
Timing Diagram 8-64. (J Mode) PCI 9080 Direct Slave Burst Read from Local Bus



Eight Lword burst, no wait states, burst enabled, Bterm enabled, 32 bit local bus.

Note: *If Bterm is disabled, a new ADS# cycle starts every quad Lword boundary.*

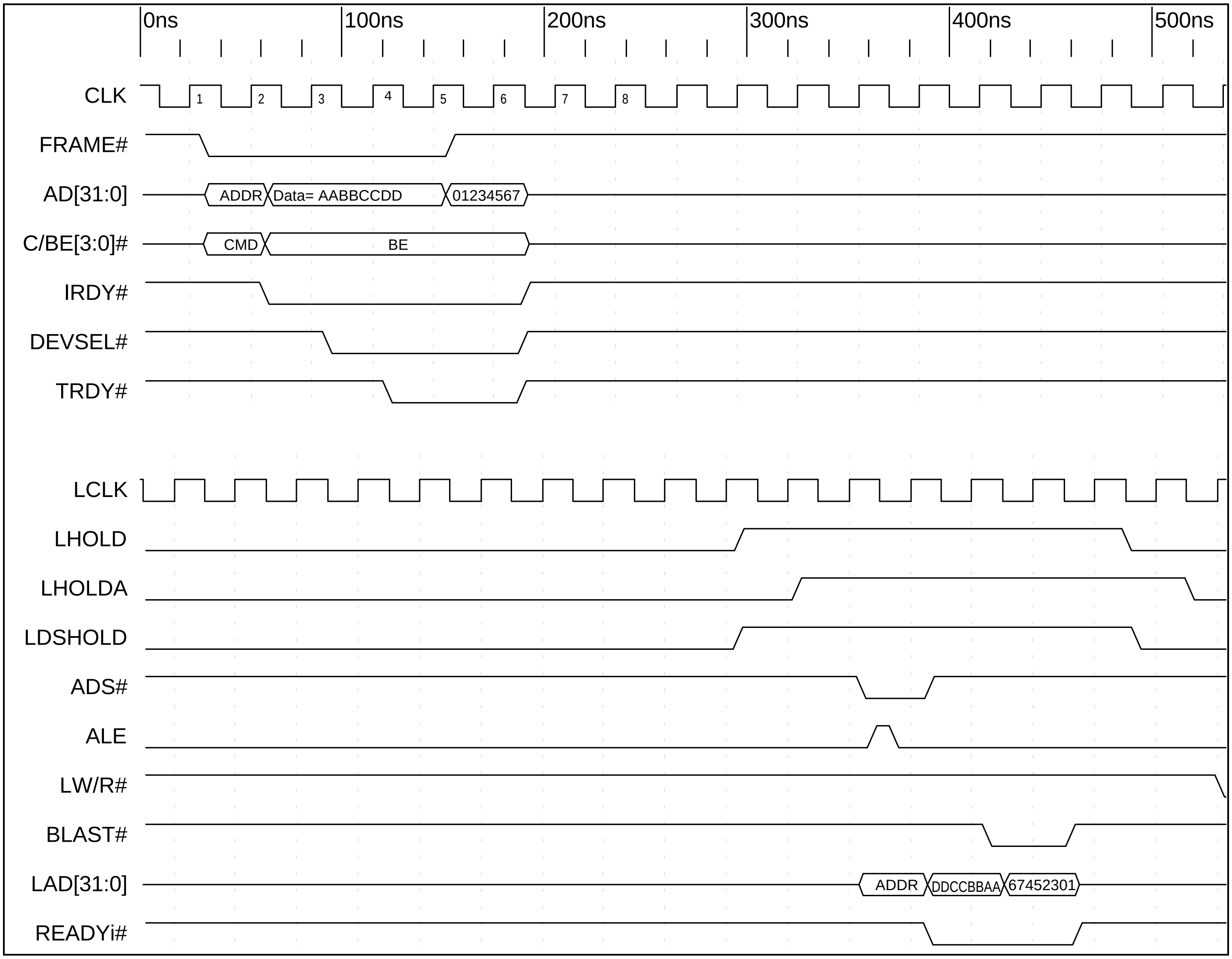
Timing Diagram 8-65. (J Mode) PCI 9080 Direct Slave Burst Write to Local Bus, Bterm Enabled



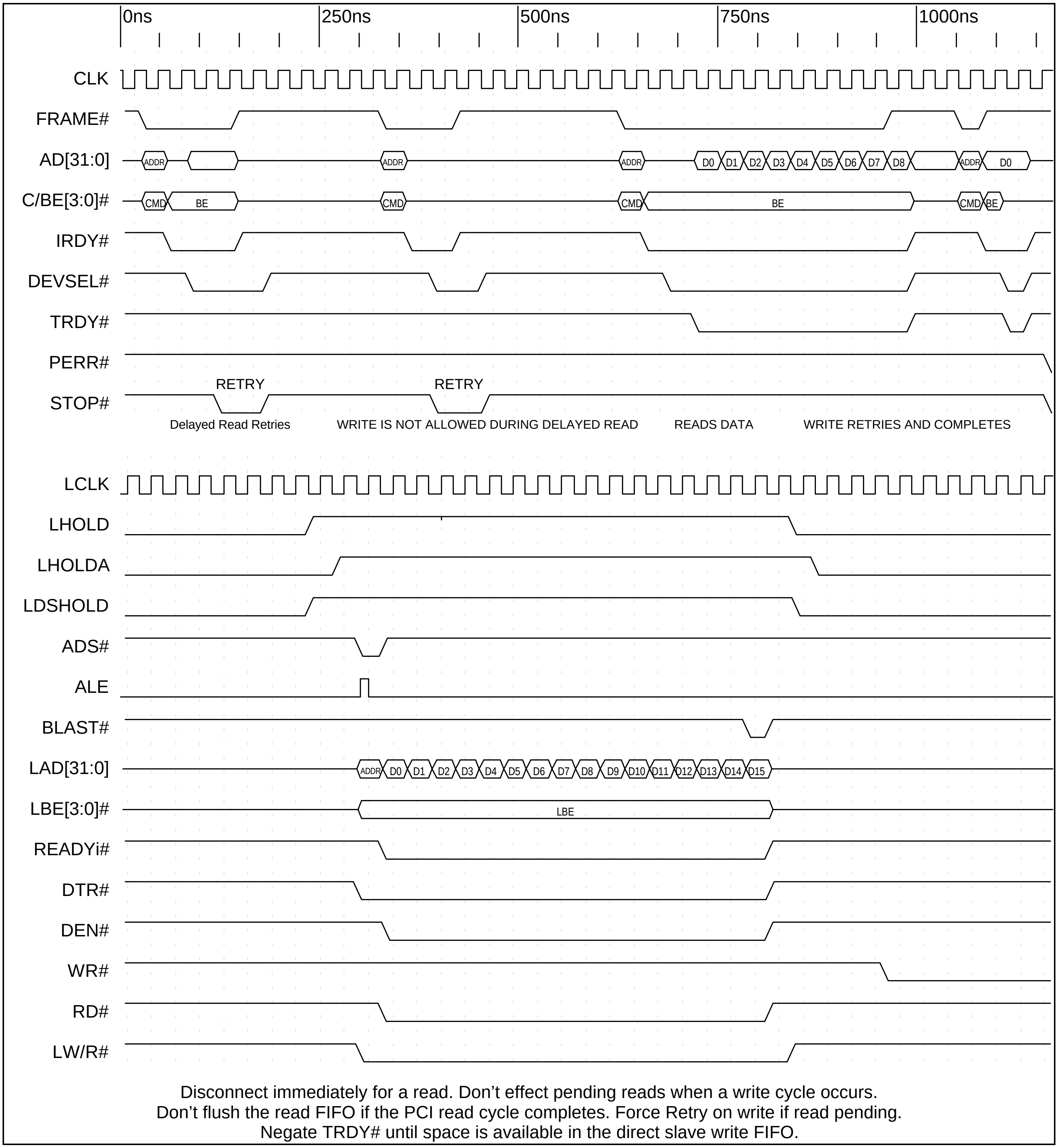
Eight Lword burst, no wait states, burst enabled, Bterm disabled, 32 bit local bus.

Note: *If Bterm is disabled, a new ADS# cycle starts every quad Lword boundary.*

Timing Diagram 8-66. (J Mode) PCI 9080 DMA or Direct Slave Burst Write to Local Bus, Bterm Disabled

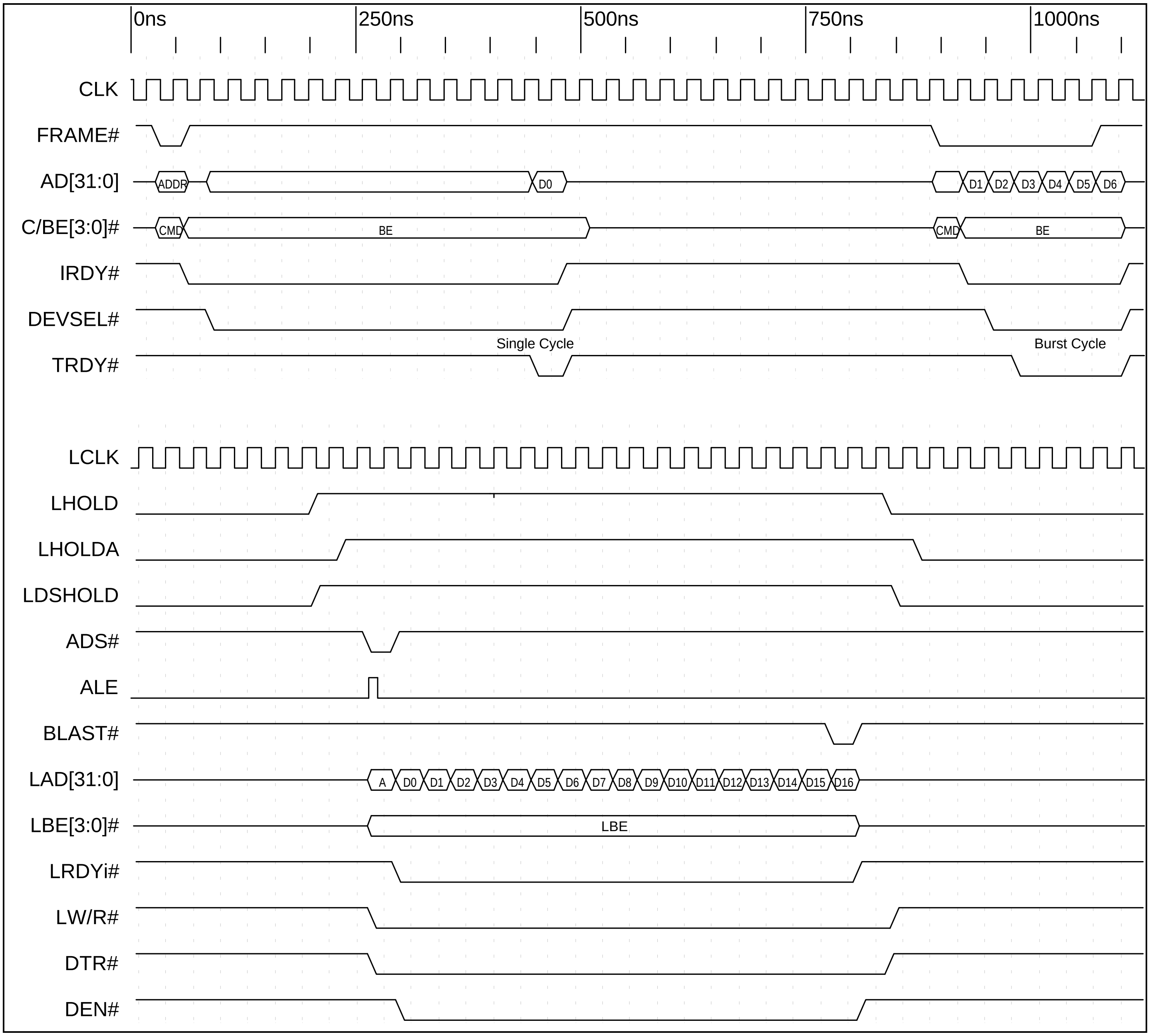


Timing Diagram 8-67. (J Mode) Direct Slave in BIGEND Local Bus with BIGEND# Input or Internal Register Setting

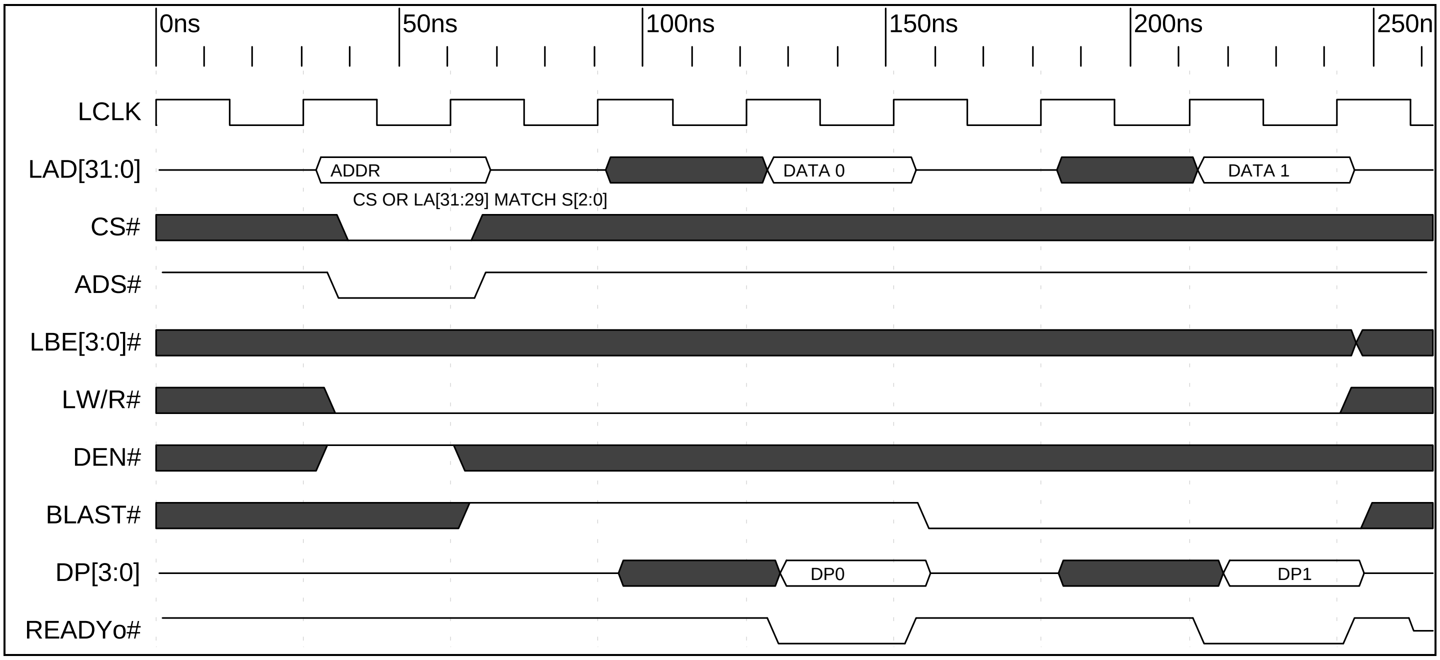


Disconnect immediately for a read. Don’t effect pending reads when a write cycle occurs.
Don’t flush the read FIFO if the PCI read cycle completes. Force Retry on write if read pending.
Negate TRDY# until space is available in the direct slave write FIFO.

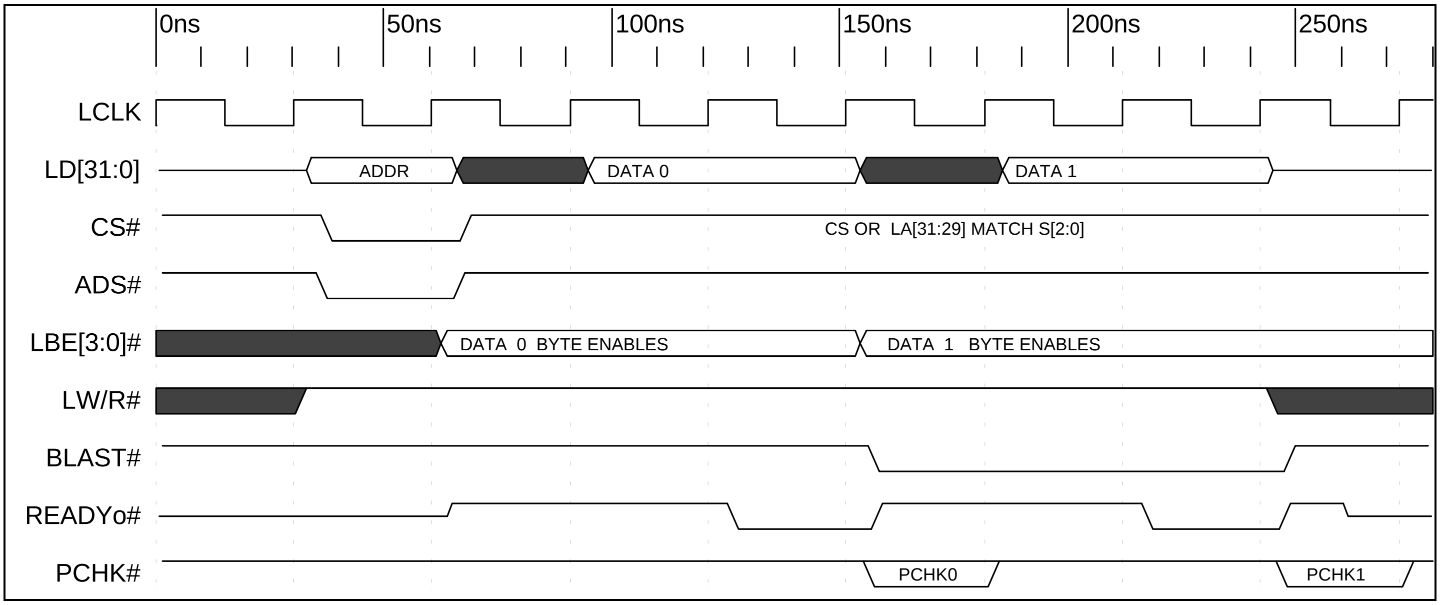
Timing Diagram 8-68. (J Mode) Direct Slave Read v2.1 Spec



Timing Diagram 8-69. (J Mode) Direct Slave Read No Flush Mode (Read Ahead Mode)

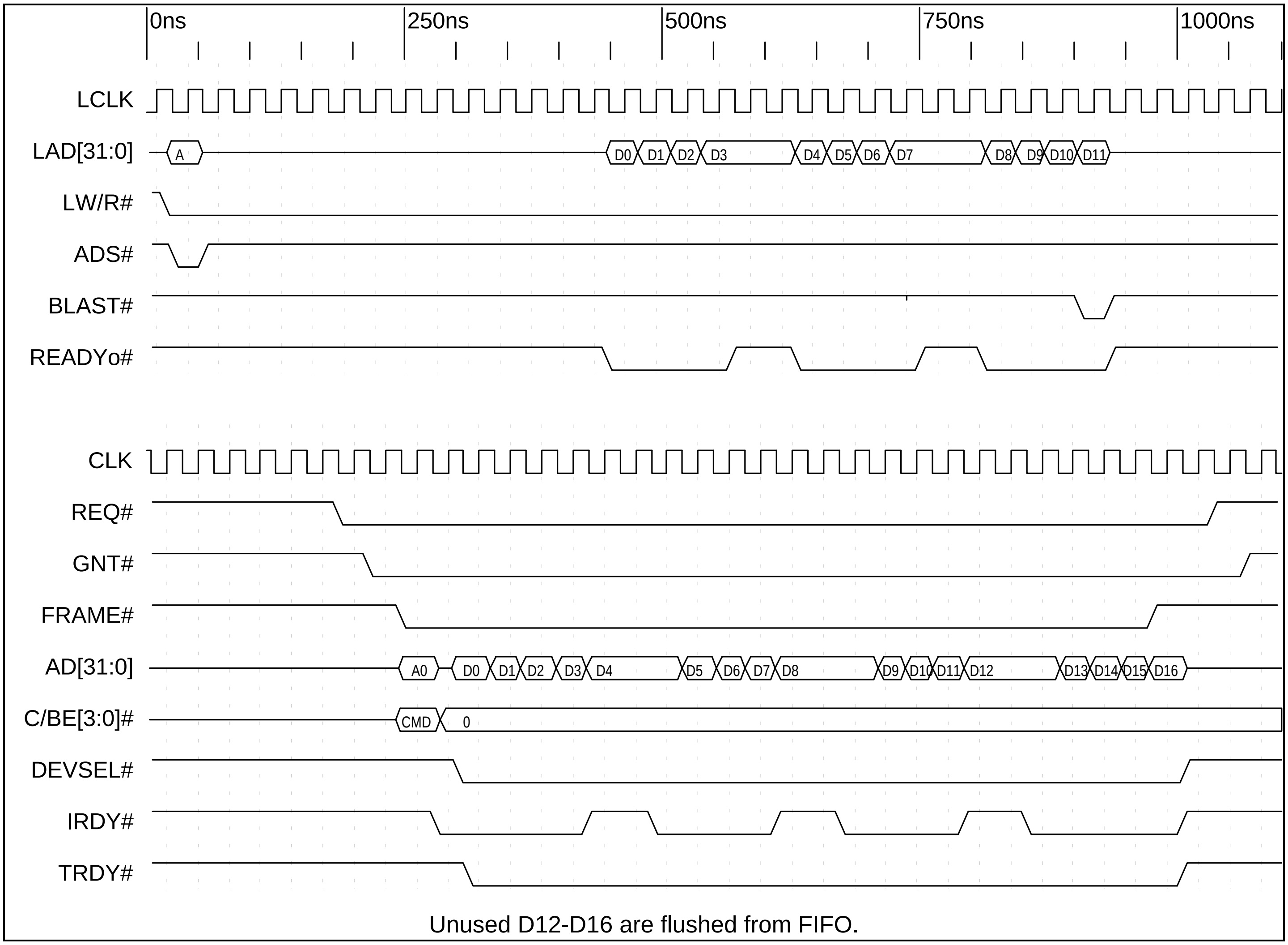


Timing Diagram 8-70. (J Mode) Local Bus Read from PCI 9080 CFG Registers



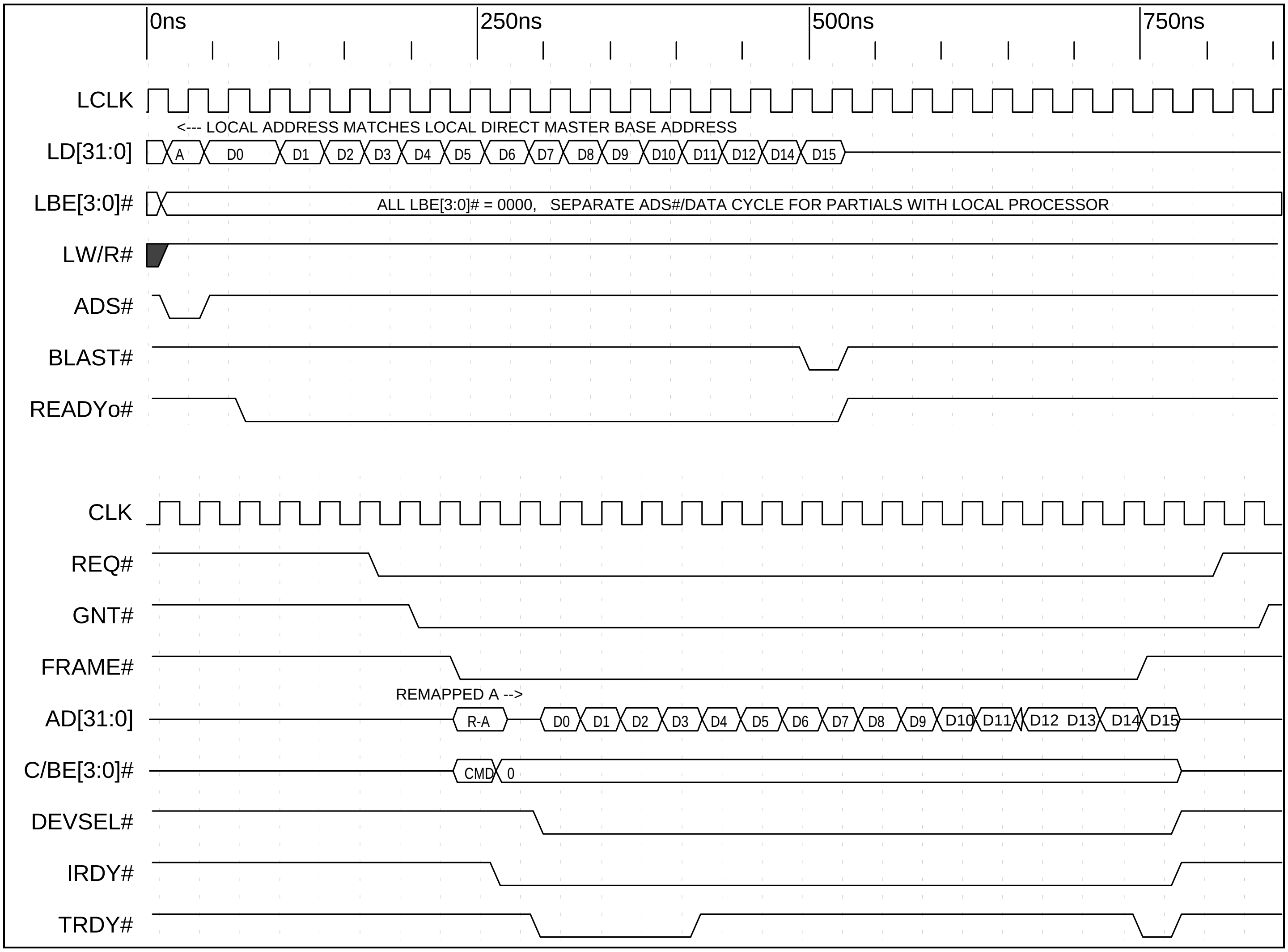
Timing Diagram 8-71. (J Mode) Local Bus Write to PCI 9080 CFG Registers

8.4.2 J Mode Direct Master

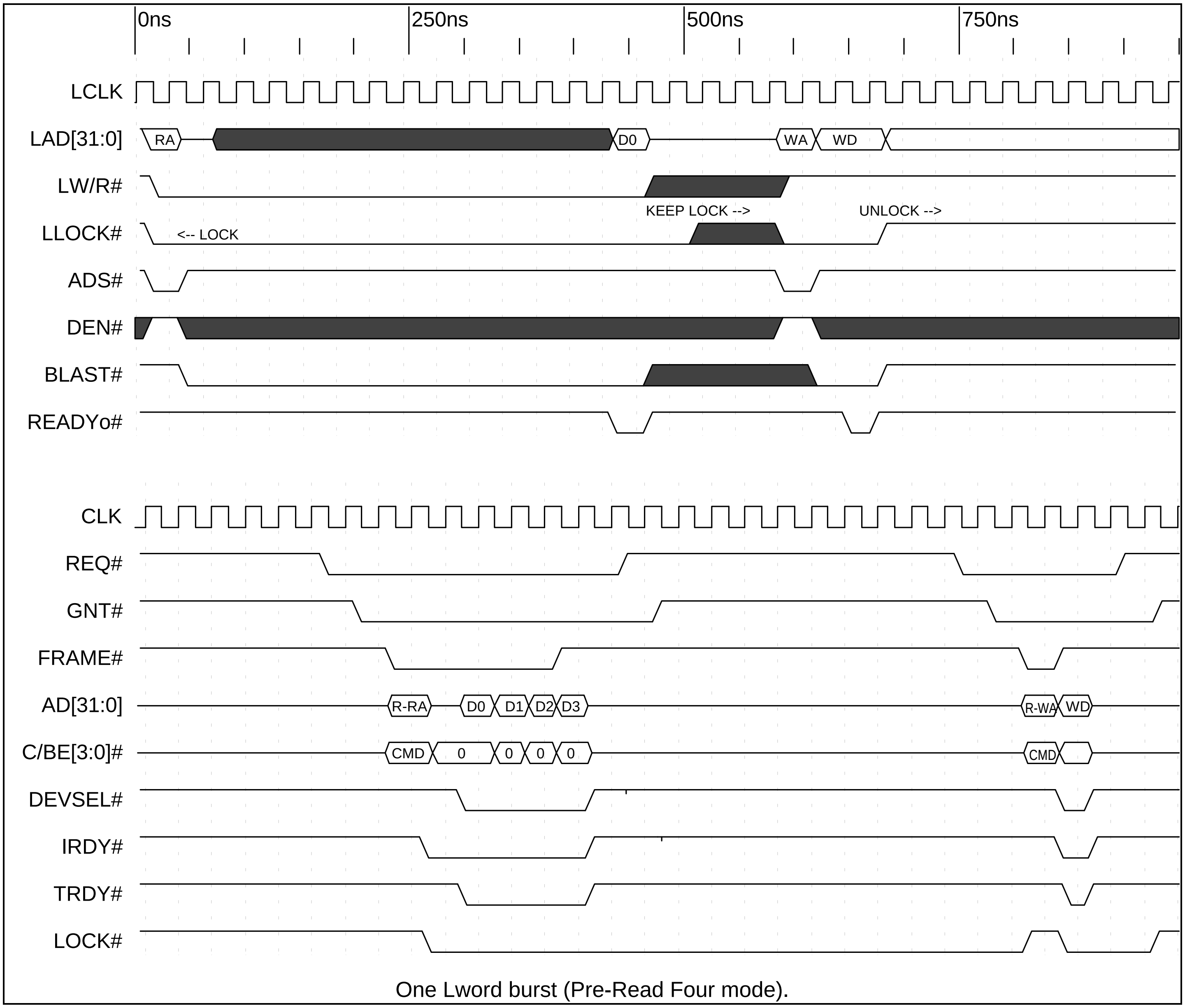


Unused D12-D16 are flushed from FIFO.

Timing Diagram 8-72. (J Mode) Direct Master Read Access from PCI Bus (Keep PCI Bus If Read FIFO Full Mode), No PCI Disconnects


0ns
250ns
500ns
750ns
LCLK
<--- LOCAL ADDRESS MATCHES LOCAL DIRECT MASTER BASE ADDRESS
LD[31:0]
A D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D14 D15
LBE[3:0]#
ALL LBE[3:0]# = 0000, SEPARATE ADS#/DATA CYCLE FOR PARTIALS WITH LOCAL PROCESSOR
LW/R#
ADS#
BLAST#
READYo#
CLK
REQ#
GNT#
FRAME#
REMAPPED A -->
AD[31:0]
R-A D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
C/BE[3:0]#
CMD 0
DEVSEL#
IRDY#
TRDY#

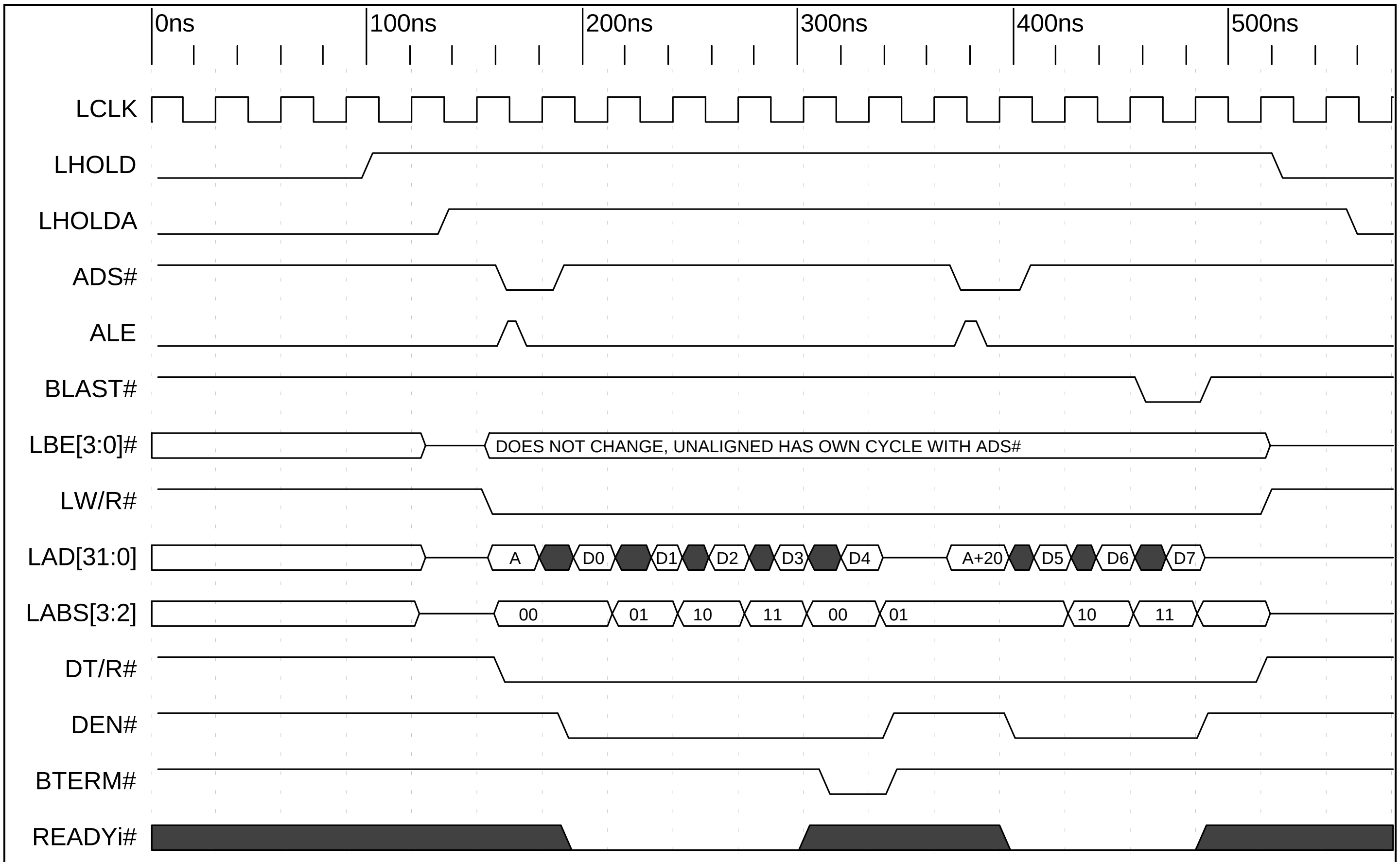
Timing Diagram 8-73. (J Mode) Local Bus Direct Master Burst Write Access to PCI Bus, Continuous If Same Clock Rate and No PCI Disconnects



One Lword burst (Pre-Read Four mode).

Timing Diagram 8-74. (J Mode) Local Bus Direct Master Lock Memory Read Access from PCI Bus Followed by Write and Release

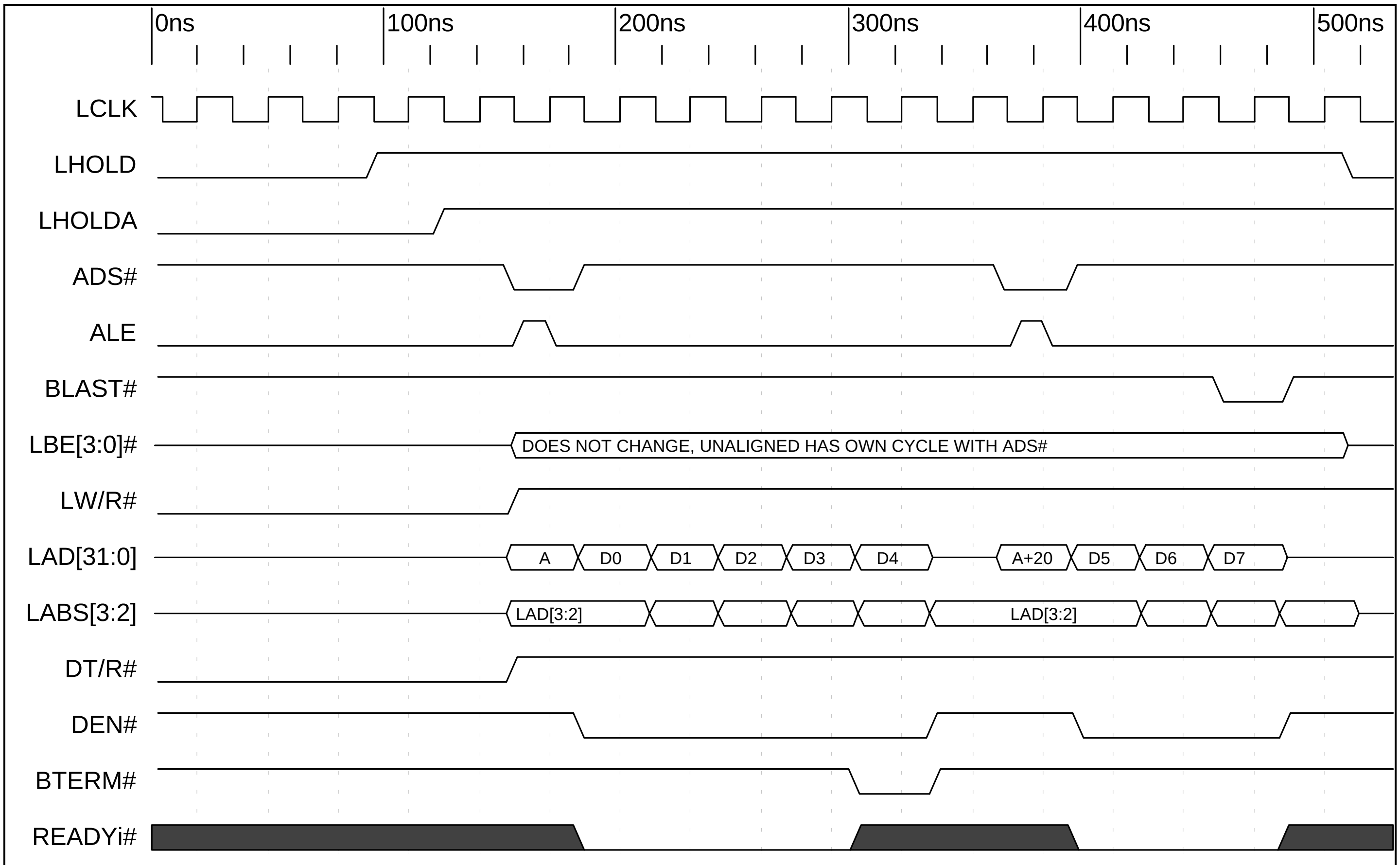
8.4.3 J Mode DMA



Eight Lword burst, no wait states, burst enabled, Bterm enabled, 32 bit local bus.

Note: *If Bterm is disabled, a new ADS# cycle starts every quad Lword boundary.*

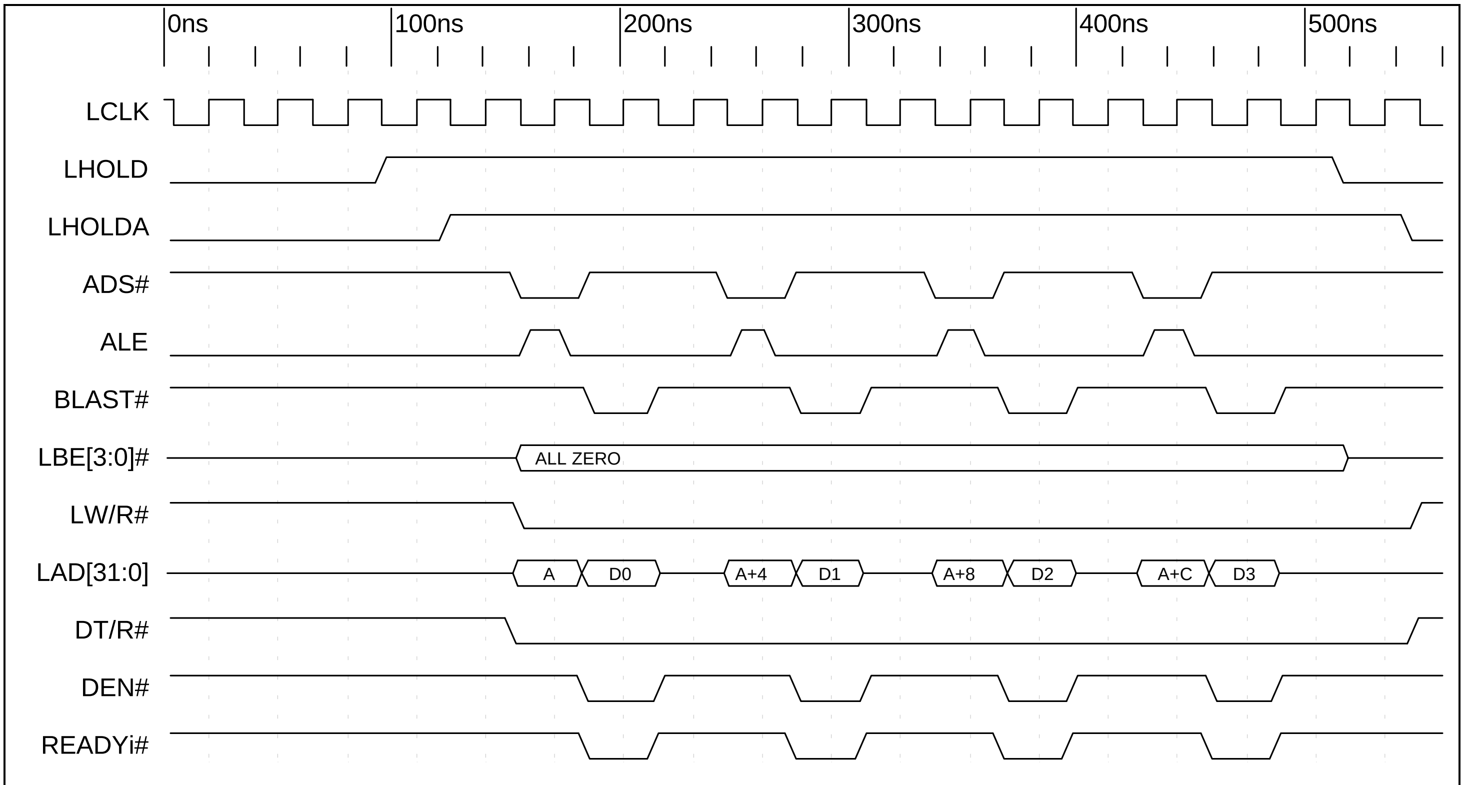
Timing Diagram 8-75. (J Mode) PCI 9080 DMA Local to PCI



Eight Lword burst, no wait states, burst enabled, Bterm enabled, 32 bit local bus.

Note: *If Bterm is disabled, a new ADS# cycle starts every quad Lword boundary.*

Timing Diagram 8-76. (J Mode) PCI 9080 DMA PCI to Local Bus

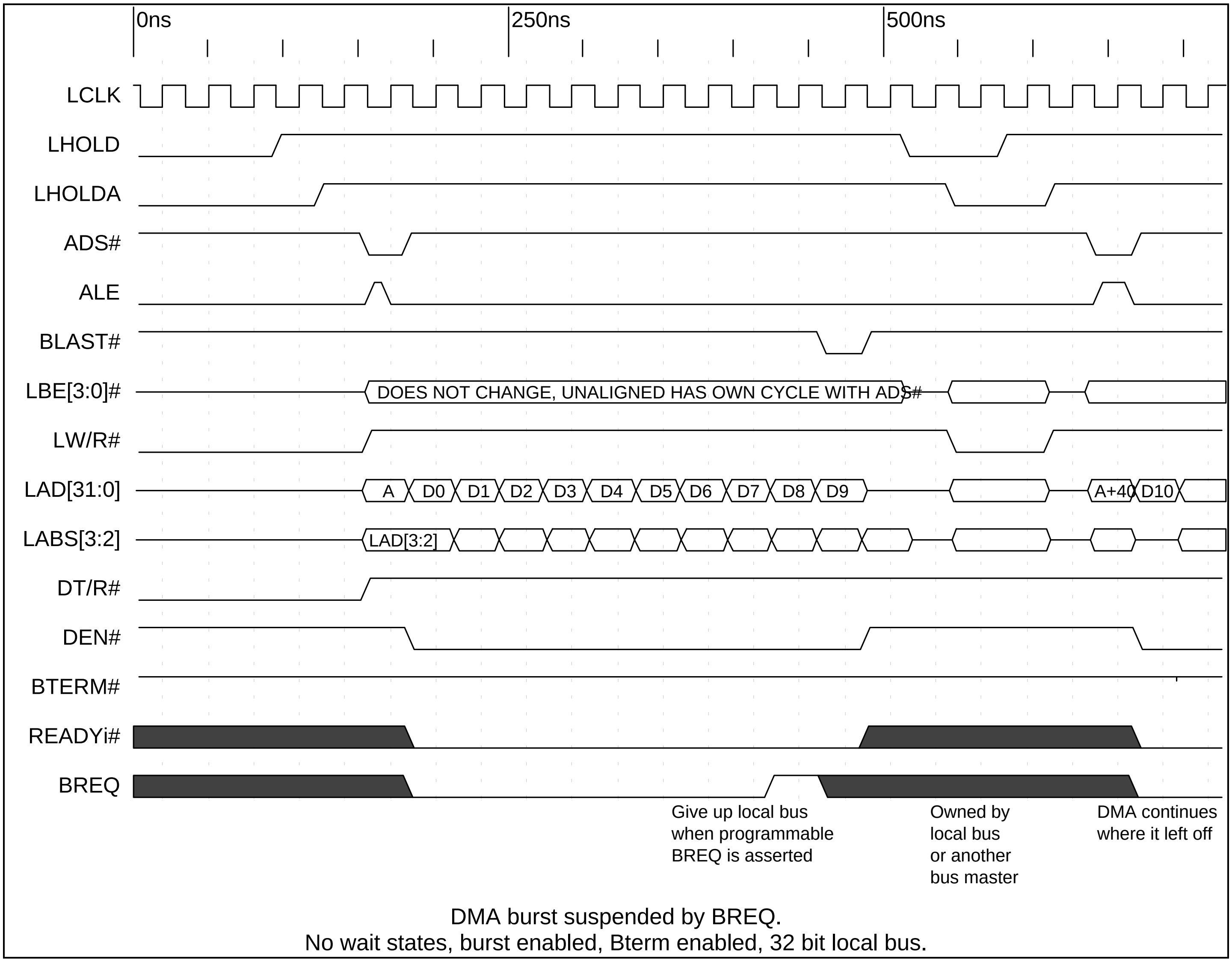


First Address A are bits [31:4] of next Descriptor Pointer Register.

D0: PCI start address
D1: Local start address
D2: Transfer count (bytes)
D3: Next descriptor pointer

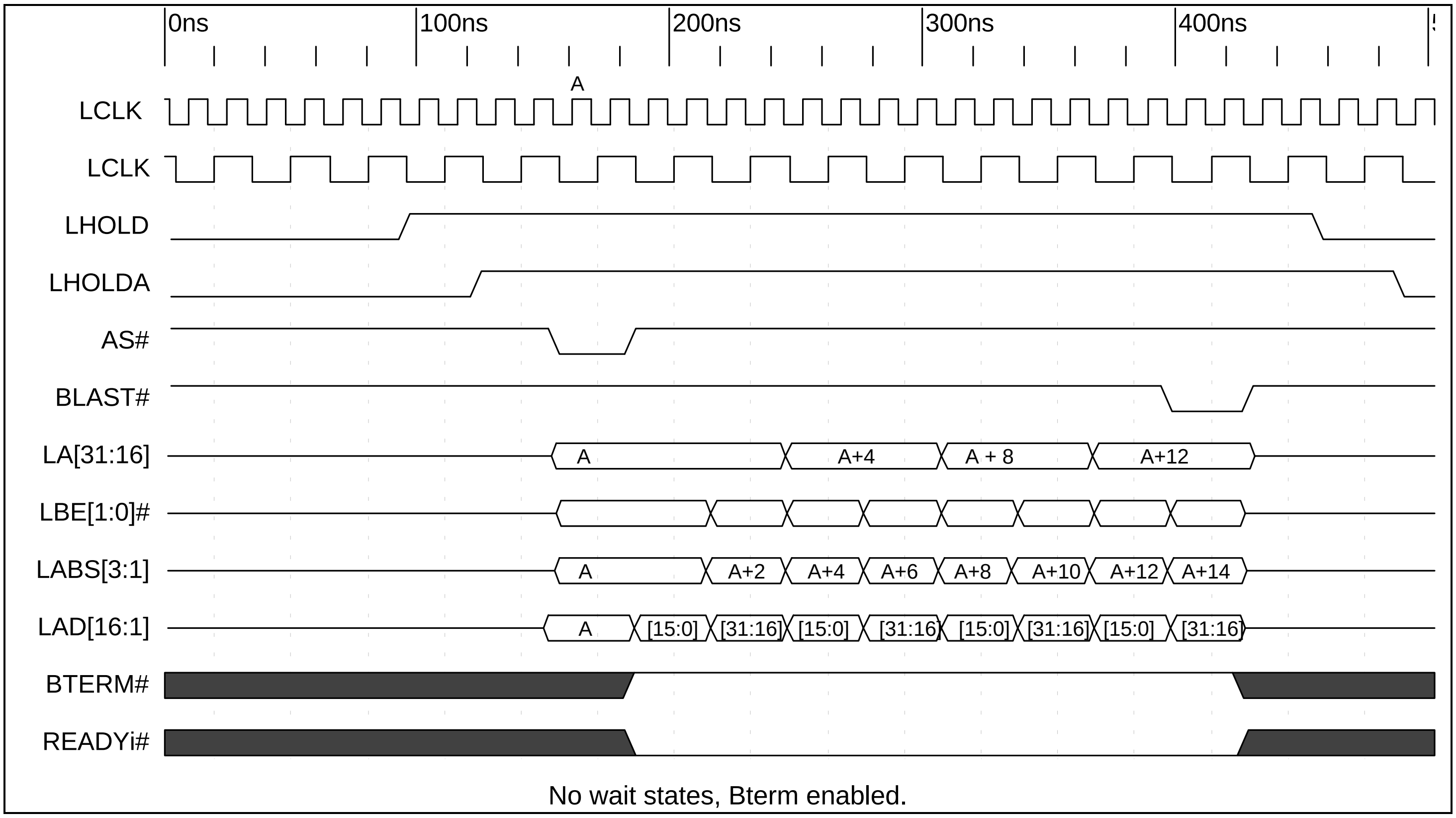
No wait states.

Timing Diagram 8-77. (J Mode) DMA Read of Chaining Parameters


0ns
250ns
500ns
LCLK
LHOLD
LHOLDA
ADS#
ALE
BLAST#
LBE[3:0]#
DOES NOT CHANGE, UNALIGNED HAS OWN CYCLE WITH ADS#
LW/R#
LAD[31:0]
A
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
A+40
D10
LABS[3:2]
LAD[3:2]
DT/R#
DEN#
BTERM#
READYi#
BREQ
Give up local bus when programmable BREQ is asserted
Owned by local bus or another bus master
DMA continues where it left off
DMA burst suspended by BREQ.
No wait states, burst enabled, Bterm enabled, 32 bit local bus.

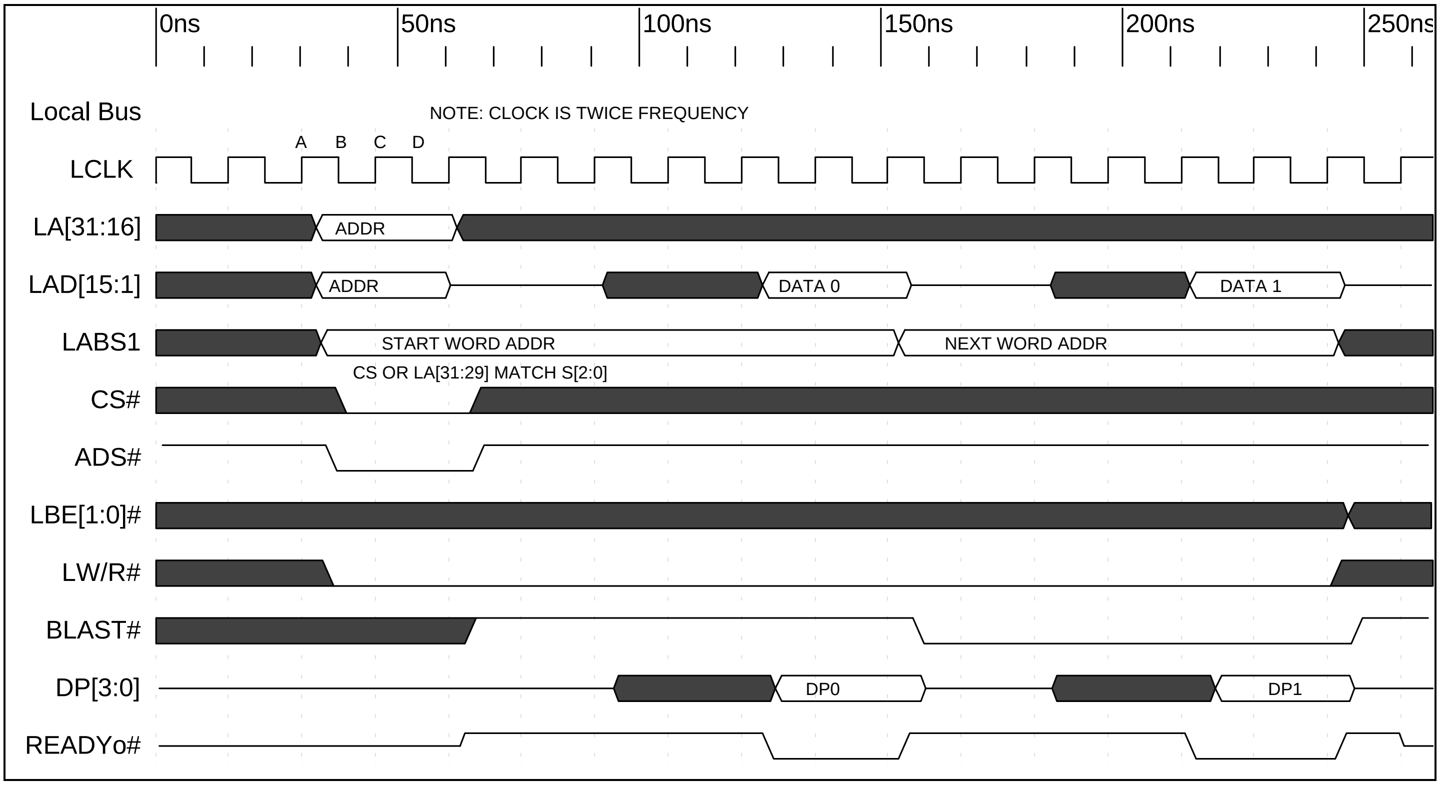
Timing Diagram 8-78. (J Mode) PCI 9080 Write to Local Bus BREQ Asserted

8.5 S MODE

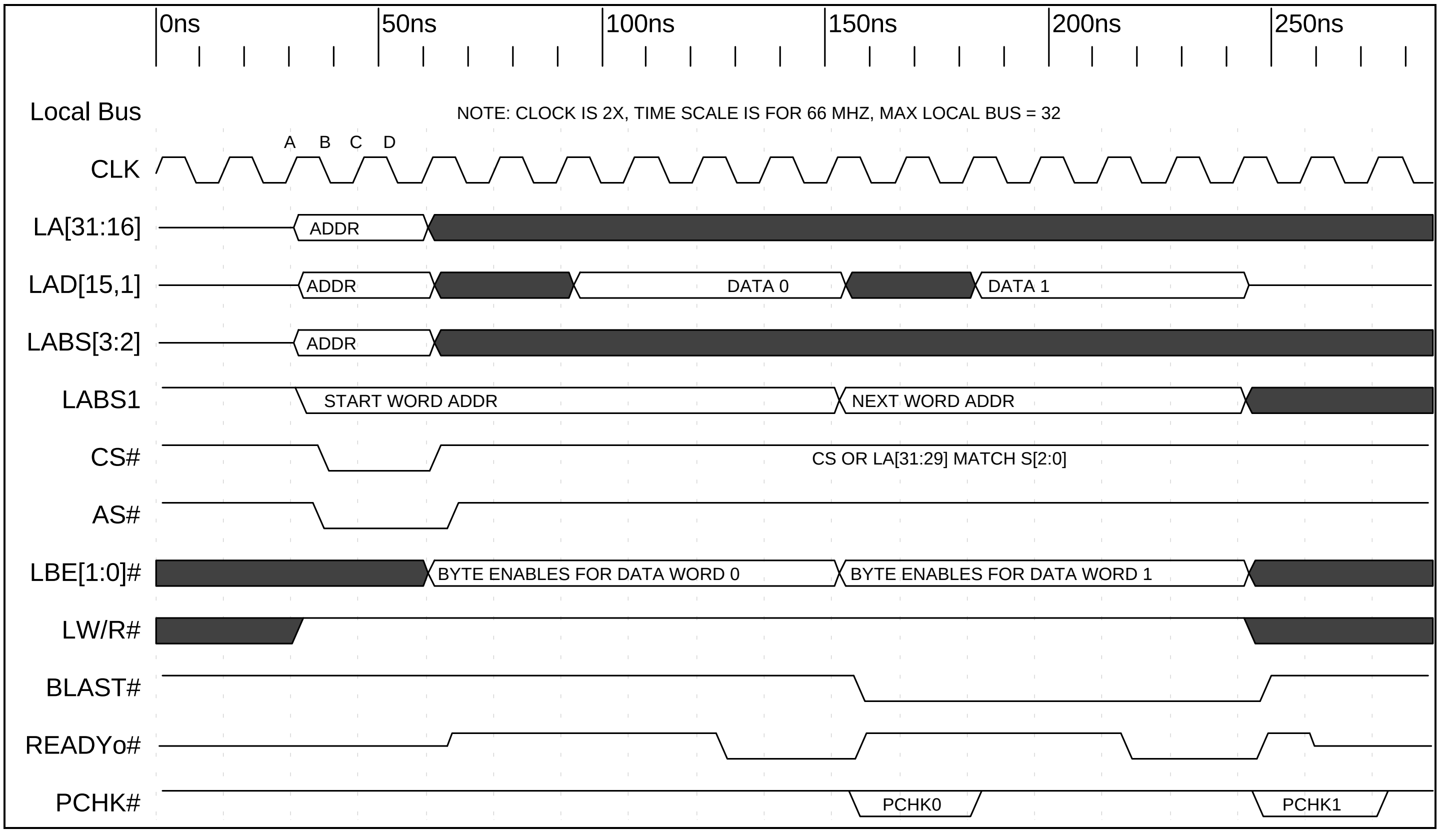


No wait states, Bterm enabled.

Timing Diagram 8-79. (S Mode) PCI 9080 DMA or Direct Slave Two Lword Burst Write to 16 Bit Local Bus



Timing Diagram 8-80. (S Mode) Local Bus Read from PCI 9080 CFG Registers



Timing Diagram 8-81. (S Mode) Local Bus Write to PCI 9080 CFG Registers