pcit1 PCI Target MegaCore Function

July 1998, ver. 1.01 Data Sheet

Features

- Parameterized `pcit1` MegaCore™ function implementing a 32-bit, 33-MHz peripheral component interconnect (PCI) target interface
- Fully compliant with the PCI Special Interest Group (SIG) timing and functional requirements; see "Compliance Summary" on page 4
- Optimized for FLEX® 10K and FLEX 6000 architectures
- Extensively hardware tested using the following hardware and software (see "Compliance Summary" on page 4):
  - FLEX 10K PCI prototype board
  - HP E2925A PCI Bus Analyzer and Exerciser
  - Tested with the most popular Intel PCI/host bridges and DEC PCI/PCI bridges
- Includes sample test vectors for user simulation
- No-risk OpenCore™ feature allows designers to instantiate and simulate designs in the MAX+PLUS® II software prior to licensing
- Uses approximately 550 to 790 FLEX logic elements (LEs), depending on the target device and number of base address registers (BARs) used
- Unlimited cycles of zero-wait-state memory read/write burst transactions, achieving a sustained bandwidth of 132 Mbytes per second
- Type zero configuration space
- Parameterized configuration registers: device ID, vendor ID, status, command, class code, revision ID, BAR0 through BAR5, subsystem ID, subsystem vendor ID, interrupt pin, and interrupt line
- Parity error detection
- Supports up to 6 BARs with adjustable memory size to achieve adaptability, efficient silicon implementation, and flexible system memory allocation
- Supports the most PCI bus commands in the industry: configuration read/write, memory read/write, I/O read/write, memory read multiple (MRM), memory read line (MRL), and memory write and invalidate (MWI)
- Local-side interrupt
- Local-initiated termination transactions: target retry, disconnect, and abort

General Description

Traditionally, PCI local bus applications have been targeted for low- to high-end desktop PCs. Today, the PCI interface is a common, fundamental building block for servers, LAN, SCSI, FDDI, and other high-bandwidth I/O applications. The pcit1 function is a hardware-tested, high-performance, flexible implementation of the PCI target interface (ordering code: PLSM-PCIT1).

Because the pcit1 function handles the complex PCI protocol and stringent timing requirements internally, it represents a significant time-saving solution. The easy-to-use local interface can be adapted for a wide range of applications. For example, the separate input and output 32-bit local data path—coupled with the local byte enable signals (l_ben[3..0])—offers the option of a merged or divided connection to peripheral devices.

Optimized for Altera® FLEX 10K and FLEX 6000 devices, the pcit1 function supports configuration, I/O, and memory transactions. With the high density of FLEX devices, designers have ample resources for custom local logic after implementing the PCI interface. The high performance of FLEX devices also enables the pcit1 function to support unlimited cycles of zero-wait-state memory-burst transactions, thus achieving 132 Mbytes per second throughput, which is the theoretical maximum for a 32-bit, 33-MHz PCI bus.

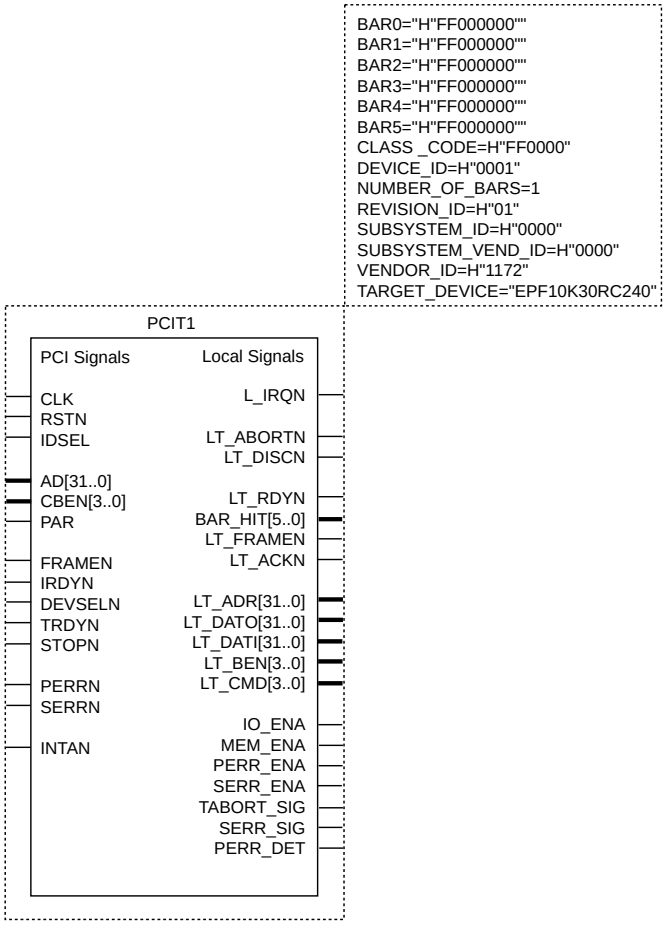
To ensure timing and protocol compliance, the pcit1 function has been vigorously tested. See "Compliance Summary" on page 4 for more information on the hardware test performed. Random PCI commands and data cycles—of various wait states and lengths—were tested using the HP E2925A PCI Bus Analyzer and Exerciser. In fact, more than 65 billion data cycles and one hundred test scenarios were used for testing.

As a parameterized function, the pcit1 function has configuration registers that can be modified upon instantiation. These features provide scalability, adaptability, and efficient silicon implementation. As a result, the same pcit1 function can be used in multiple PCI projects with different requirements.

For example, the pcit1 function offers up to six BARs for multiple local-side devices. However, some applications require only one memory range. PCI designers can choose to instantiate only one BAR, which reduces logic cell consumption. After designers define the parameter values, the MAX+PLUS II software automatically and efficiently modifies the design and implements the logic.

Figure 1 illustrates the `pcit1` symbol as used in a MAX+PLUS II Graphic Design File (**.gdf**).

Figure 1. pcit1 Symbol



Compliance Summary

The pcit1 function is compliant with the requirements specified in the PCI SIG's ***PCI Local Bus Specification, Revision 2.1***, and ***Compliance Checklist, Revision 2. 1***. The pcit1 function is shipped with MAX+PLUS II Simulator Channel Files (**.scf**), which can be used to validate the function in the MAX+PLUS II software. Altera has performed vigorous hardware tests of the pcit1 function against the most popular Intel PCI/host bridges, such as the 430NX, 430VX, 430TX, and 430HX bridges. The function was also tested against several DEC PCI/PCI bridges, such as the DEC 21052-AB bridge. The following tests were performed on each type of hardware:

- Implementing the pcit1 function with a memory interface in an Altera FLEX 10K PCI prototype board
- Using the HP E2925A PCI Bus Exerciser and Analyzer for exercising the pcit1 function, PCI protocol, and checking, as well as for monitoring the pcit1 function in runtime transactions. These tests included:
  - Memory read/write, memory write and invalidate, memory read line, and memory read multiple
  - Configuration read/write
  - I/O read/write
  - Abnormal target terminations (abort, retry, disconnect),
  - Random transaction wait states, burst length, and termination

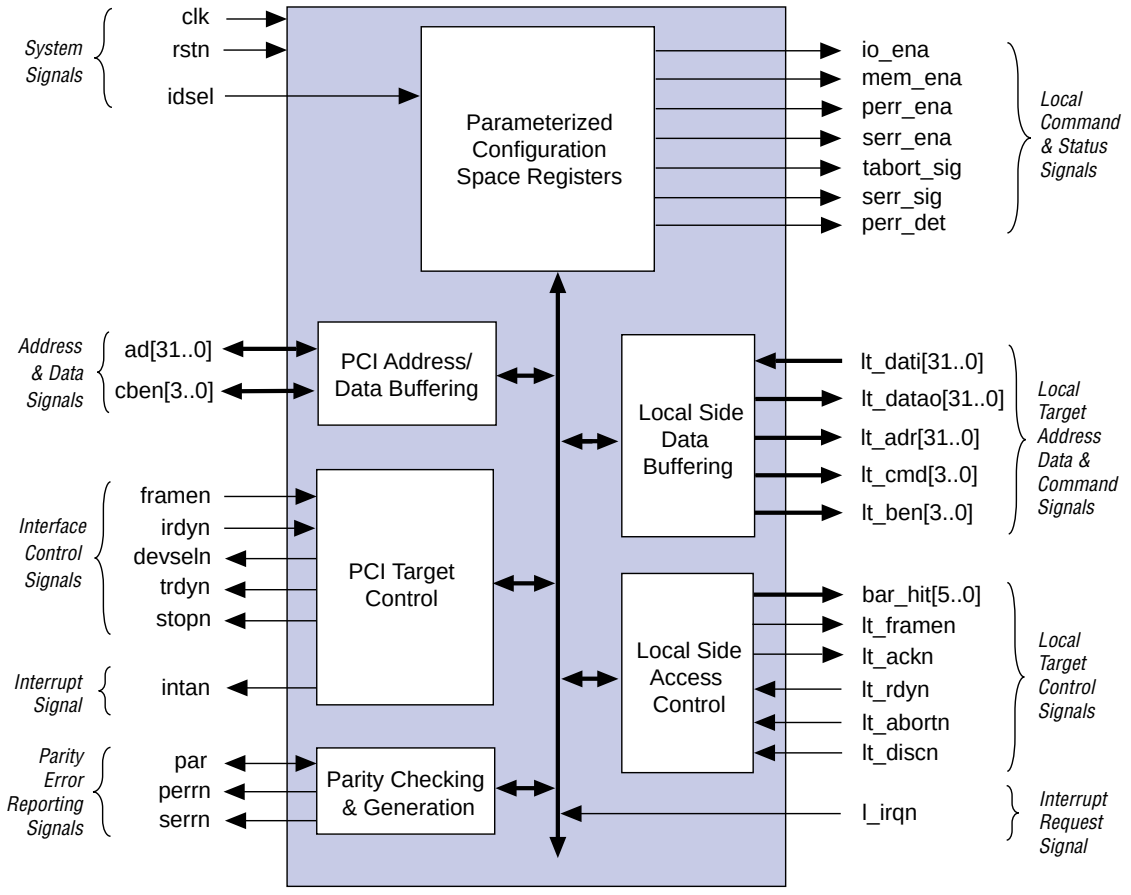
Functional Description

The pcit1 function consists of three main components:

- Parameterized PCI bus configuration register space
- Target interface control logic
- Parity checker and generator

Figure 2 shows the pcit1 block diagram. The signals connecting the pcit1 function to the PCI bus are shown on the left side of the figure. The signals are grouped by functionality, and signal directions are illustrated from the perspective of the pcit1 function as it connects to the PCI bus.

Figure 2. pcit1 Functional Block Diagram



Bus Commands

Table 1 shows the PCI bus commands supported by the pcit1 function. In accordance with the PCI SIG's ***PCI Local Bus Specification, Revision 2.1***, all unsupported commands are ignored by the pcit1 function.

Table 1. PCI Bus Commands Supported by the pcit1 Function *Note (1)*

cben[3..0] Value	Bus Command Cycle	Supported
0000	Interrupt acknowledge	Ignored
0001	Special cycle	Ignored
0010	I/O read	Yes
0011	I/O write	Yes
0100	Reserved	Ignored
0101	Reserved	Ignored
0110	Memory read	Yes
0111	Memory write	Yes
1000	Reserved	Ignored
1001	Reserved	Ignored
1010	Configuration read	Yes
1011	Configuration write	Yes
1100	Memory read multiple (MRM), *Note (2)*	Yes
1101	Dual address cycle	Ignored
1110	Memory read line (MRL), *Note (2)*	Yes
1111	Memory write and invalidate (MWI), *Note (2)*	Yes

Notes:

(1) During the address phase of a transaction, the cben[3..0] bus is used to indicate the type of command.

(2) The pcit1 function interprets the MRM and MRL commands as memory read transactions, and the MWI command as a memory write transaction. The local side detects the exact command on the lt_cmd[3..0] bus. Any special handling required by these commands is the responsibility of the local side.

Target Response

For memory transactions, the pcit1 function supports an unlimited cycle of zero-wait-state burst transfers. Therefore, both memory read and write transfers achieve a sustained throughput of 132 Mbytes per second, which is the maximum bandwidth attainable in a 32-bit, 33-MHz PCI bus system. However, if the local logic cannot handle burst data during data cycles, it may insert a wait state to stall data exchange on the PCI bus via the lt_rdyn signal. For slower applications, it is usually desirable to implement a data buffer at the local side of the interface to store or pre-fetch data. The pcit1 function provides separate input and output local data as well as an address bus, thus allowing data buffers of various widths and depths to be added to the local logic.

In addition to high-performance data operation, the pcit1 function supports local-initiated abnormal termination. Under most conditions, data is successfully exchanged between master and target devices. However, when the local logic is unable to complete the request, it may use the lt_abortn or lt_discn signal to terminate the current PCI cycle. Depending on the inputs from the local logic and the PCI bus, the pcit1 function drives a combination of the control signals devseln, trdyn, and stopn to terminate the transaction by issuing a retry, stop, or abort request.

Configuration Registers

Each logical PCI bus device includes a block of 16 configuration DWORDS (32-bit blocks of data) reserved for implementing its configuration registers. The format of the first 16 DWORDS is defined by the PCI SIG's ***PCI Compliance Checklist, Revision 2.1***, which defines two header formats, type one and type zero. Header type one is used for PCI-to-PCI bridges; header type zero is used for all other devices.

Table 2 displays the 64-byte configuration space. The registers within this range are used to identify the device, control PCI bus functions, and provide PCI bus status. The shaded areas indicate registers that are supported by the `pcit1` function.

Table 2. PCI Bus Configuration Registers

Address	Byte			
	3	2	1	0
00H	Device ID		Vendor ID	
04H	Status Register		Command Register	
08H	Class Code			Revision ID
0CH	BIST	Header Type	Latency Timer	Cache Line Size
10H	Base Address Register 0			
14H	Base Address Register 1			
18H	Base Address Register 2			
1CH	Base Address Register 3			
20H	Base Address Register 4			
24H	Base Address Register 5			
28H	Card Bus CIS Pointer			
2CH	Subsystem ID		Subsystem Vendor ID	
30H	Expansion ROM Base Address Register			
34H	Reserved			
38H	Reserved			
3CH	Maximum Latency	Minimum Grant	Interrupt Pin	Interrupt Line

Parameters

Table 3 describes the pcit1 parameters, which set PCI bus configuration registers. All parameters—except TARGET_DEVICE, NUMBER_OF_BARS, and BAR0 through BAR5—set read-only PCI configuration registers, which are device identification registers.

Table 3. pcit1 Parameters

Name	Format	Default Value	Description
NUMBER_OF_BARS	Decimal	1	Number of BARs used. This parameter controls the number of BARs instantiated in the pcit1 function at compile time. BARs are instantiated in sequential order, starting with BAR 0.
INT_ARBITER_ENA	Binary	Off	Internal arbiter enabled. When set to “on”, the reqn and gntn become internal signals, reducing the number of I/Os used by the pci_b function; when set to “off”, the reqn and gntn signals become tri-stated signals.
HOST_BRIDGE_ENA	Binary	Off	Host bridge enabled. When set to “on”, the master bit is tied to high.
BAR0	Hexadecimal	H"FF000000"	BAR 0
BAR1	Hexadecimal	H"FF000000"	BAR 1
BAR2	Hexadecimal	H"FF000000"	BAR 2
BAR3	Hexadecimal	H"FF000000"	BAR 3
BAR4	Hexadecimal	H"FF000000"	BAR 4
BAR5	Hexadecimal	H"FF000000"	BAR 5
CLASS_CODE	Hexadecimal	H"FF0000"	Class code register
DEVICE_ID	Hexadecimal	H"0001"	Device ID register
VENDOR_ID	Hexadecimal	H"1172"	Device vendor ID register
REVISION_ID	Hexadecimal	H"01"	Revision ID register
SUBSYSTEM_ID	Hexadecimal	H"0000"	Subsystem ID register
SUBSYSTEM_VEND_ID	Hexadecimal	H"0000"	Subsystem vendor ID register
TARGET_DEVICE	String	"EPF10K30RC240"	Device in which the pcit1 is implemented

The NUMBER_OF_BARS parameter defines the number of BARs to be instantiated in the pcit1 function during compilation. Depending on the value of the NUMBER_OF_BARS parameter, some of the BAR0 through BAR5 parameters are ignored by the MAX+PLUS II software. For example, setting NUMBER_OF_BARS to 1 will cause the BAR1 through BAR5 parameter values to be ignored because the corresponding BARs will not be used during compilation.

The BAR0 through BAR5 parameters control the following pcit1 properties:

- Type of address space reserved (i.e., memory or I/O)
- Amount of memory or I/O space that is reserved
- Whether the memory space is pre-fetchable
- Whether the memory space can be located anywhere in the 32-bit address space, or if it must be mapped below 1 Mbyte

For example, if BAR0 = H"FF000008", the pcit1 function will contain a pre-fetchable, $2^{(32-8)}$ bytes (i.e., 16 Mbytes) of memory space that can be located anywhere in the 32-bit address memory space.

Signals

The pcit1 function uses the following PCI bus signals:

- *Input*—Standard input-only signal.
- *Output*—Standard output-only signal.
- *Bidirectional*—Tri-state input/output signal.
- *Sustained tri-state (STS)*—Signal that is driven by one agent at a time (e.g., a device or host operating on the PCI bus). An agent that drives a sustained tri-state pin low must actively drive it high for one clock cycle before tri-stating it. Another agent cannot drive a sustained tri-state signal any sooner than one clock cycle after it is released by the previous agent.
- *Open-drain*—Signal that is wire-ORed with other agents. The signaling agent asserts the open-drain signal, and a weak pull-up resistor deasserts the open-drain signal. The pull-up resistor may take two or three PCI bus clock cycles to restore the open-drain signal to its inactive state.

PCI Signals

Table 4 summarizes the PCI bus signals that connect the pcit1 function to the PCI bus. See "Local-Side Signals" on page 12 for information on local-side signals.

Table 4. PCI Bus Signals Connecting the pcit1 Function to the PCI Bus (Part 1 of 3)

Name	Type	Polarity	Description
clk	Input	–	Clock. The clk input provides the reference signal for all other PCI interface signals except rstn and intan.
rstn	Input	Low	Reset. The rstn input initializes the pcit1 interface circuitry, and can be asserted asynchronously to the PCI bus clock edge. When active, the PCI output signals are tri-stated and the open-drain signals (e.g., serrn) float.
idsel	Input	High	Initialization device select. The idsel input is a chip select for configuration transactions.

Table 4. PCI Bus Signals Connecting the pcit1 Function to the PCI Bus (Part 2 of 3)

ad[31..0]	Tri-state bidirectional	–	Address/data bus. The ad[31..0] bus is a time-multiplexed address/data bus; each bus transaction consists of an address phase followed by one or more data phases. The data phases occur when irdyn and trdyn are both asserted.
cben[3..0]	Tri-state input	Low	Command/byte enable. The cben[3..0] bus is a time-multiplexed command/byte enable bus. During the address phase, the bus indicates the command; during the data phase, the bus indicates byte enables.
Name	**Type**	**Polarity**	**Description**
par	Tri-state bidirectional	–	Parity. The par signal is even parity across the ad[31..0] and cben[3..0] buses. In other words, the number of 1s on ad[31..0], cben[3..0], and par equal an even number. The parity of a data phase is presented on the PCI bus during the clock cycle of the following data phase.
framen	Input	Low	Frame. The framen signal is an input from the current bus master that indicates the beginning and duration of a bus transaction. When framen is initially asserted, the address and command signals are present on the ad[31..0] and cben[3..0] buses. The framen signal remains asserted during the data operation and is deasserted to identify the end of a transaction.
irdyn	Input	Low	Initiator ready. The irdyn signal is an input from a bus master to the pcit1 function and indicates that the bus master can complete a data transaction. In a write transaction, an active irdyn indicates that valid data is on the ad[31..0] bus. In a read transaction, irdyn indicates that the master is ready to accept the data on the ad[31..0] bus.
devseln	Sustained tri-state output	Low	Device select. Target asserts devseln to indicate that the target has decoded its own address and accepted the transaction.
trdyn	Sustained tri-state output	Low	Target ready. The trdyn target output indicates that the target can complete the current data phase. In a read operation, an active trdyn indicates that the target is providing valid data on the ad[31..0] bus. In a write operation, an active trdyn indicates that the target is ready to accept data on the ad[31..0] bus.
stopn	Sustained tri-state output	Low	Stop. The stopn signal is a target device output that signals the bus master to terminate the current transaction. The stopn signal is used in conjunction with trdyn and devseln to indicate the type of termination initiated by the target.
perrn	Sustained tri-state output	Low	Parity error. The perrn signal indicates a data parity error, and is asserted one clock following the par signal or two clocks following a data phase with a parity error.
serrn	Open-drain output	Low	System error. The serrn signal indicates system error on address parity error. The pcit1 function asserts serrn if a parity error is detected during an address phase and the serr_ena and perr_ena bits in the PCI command register are set.

Table 4. PCI Bus Signals Connecting the pcit1 Function to the PCI Bus (Part 3 of 3)			
intan	Open-drain output	Low	Interrupt A. The intan signal is an active-low interrupt to the host, and must be used for any single-function device requiring an interrupt capability.

The PCI bus, FLEX 10K, and FLEX 6000 devices support IEEE Std. 1149.1-1990 Joint Test Action Group (JTAG) boundary-scan testing (BST). To use IEEE Std. 1149.1-1990 BST, designers should connect the PCI bus JTAG pins to the FLEX 10K or FLEX 6000 device JTAG pins. See Table 5.

Table 5. Optional IEEE Std. 1149.1-1990 JTAG Signals			
Name	**Type**	**Polarity**	**Description**
TCK	Input	High	Test clock. The TCK input is used to clock the test mode and test data in and out of the device.
TMS	Input	High	Test mode select. The TMS input is used to control the state of the Test Access Port (TAP) control in the device.
TDI	Input	High	Test data. The TDI input is used to shift the test data and instructions into the device.
TDO	Output	High	Test data. The TDO output is used to shift the test data and instructions out of the device.

Local-Side Signals

Table 6 summarizes the pcit1 target signals that connect the pcit1 function to the local-side peripheral device(s).

Table 6. pcit1 Signals Connecting the pcit1 Function to the Local Side (Part 1 of 2)

Name	Type	Polarity	Description
l_irqn	Input	Low	Local interrupt request. The local-side peripheral device asserts l_irqn to signal a PCI bus interrupt. Asserting l_irqn forces pcit1 to assert the intan signal for as long as l_irqn is asserted.
lt_abortn	Input	Low	Local target abort request. The lt_abortn signal is asserted when the local-side peripheral device encounters a fatal error and cannot complete the current transaction. Therefore, lt_abortn is asserted to request that the pcit1 function issue a target abort.
lt_discn	Input	Low	Local target disconnect request. It is used to request the pcit1 function to issue a retry or diconnect. The pcit1 function will issue a retry or a disconnect depending on when lt_discn is asserted. The PCI bus requires the target device to issue a disconnect whenever the transaction exceeds its memory space. When the transaction exceeds the target device's memory space, the local side is responsible for asserting lt_discn.
lt_rdyn	Input	Low	Local target ready. The local side asserts lt_rdyn to indicate either a valid data input during a target read, or that the local side is ready to accept data during a target write. During a target read, the deassertion of lt_rdyn suspends the current transfer, i.e., a wait state is inserted by the local side. During a target write, an inactive lt_rdyn directs the pcit1 function to insert wait states on the PCI bus. The pcit1 function inserts wait states during a burst transfer only when lt_rdyn inserts wait states on the local side.
lt_dati[31..0]	Input	–	Local target data bus input. The lt_dati[31..0] bus is driven active by the local-side peripheral device during PCI bus-initiated read transactions.
bar_hit[5..0]	Output	High	BAR hit. When the bar_hit[5..0]bus is asserted, the PCI address matches a BAR address, which prompts the pcit1 function to claim the transaction. Each bit in the bar_hit[5..0] bus is used for the corresponding BAR, i.e., bar_hit[0] is used for BAR 0. The bar_hit[5..0] bus has the same timing as the lt_framen signal.
lt_framen	Output	Low	Local target frame request. The lt_framen signal is asserted while pcit1 is engaged in a PCI transaction. The lt_framen signal is asserted one clock before pcit1 asserts devseln; it is released after the last data phase of the current PCI transaction is complete.

Table 6. pcit1 Signals Connecting the pcit1 Function to the Local Side (Part 2 of 2)

Name	Type	Polarity	Description
lt_ackn	Output	Low	Local target acknowledge. The pcit1 function asserts lt_ackn to indicate that either a valid data output occurred during a target write transaction, or that the pcit1 function is ready to accept data during a target read transaction. During a target read, lt_ackn is deasserted, which indicates that the pcit1 function is not ready to accept data and that the local logic should stop the burst transaction. During a target write, lt_ackn is deasserted, which suspends the current transfer, i.e., a wait state is inserted by the PCI master device. The lt_ackn signal is only deasserted during a burst transaction when the PCI bus master inserts a wait state.
lt_adr[31..0]	Output	–	Local target address output. The lt_adr[31..0] output represents the target memory address for the current local-side data phase. The pcit1 function increments the lt_adr[31..0] bus after a successful data transfer is complete on the local side, i.e., lt_rdyn and lt_ackn are active during the same clock cycle.
lt_dato[31..0]	Output	–	Local target data bus output. The lt_dato[31..0] output is driven active by the local-side peripheral device during PCI-initiated write transactions.
lt_ben[3..0]	Output	Low	Local target byte enable. The lt_ben[3..0] output represents the byte enable requests from the PCI master during data phases.
lt_cmd[3..0]	Output	–	Local target command. The lt_cmd[3..0] output represents the PCI command for the current claimed transaction. The lt_cmd[3..0] bus uses the same encoding scheme as the cben[3..0] bus.

Local-Side Command/Status Signals

Table 7 summarizes the pcit1 control signals that are driven to the local logic from the command/status register of the pcit1 function. For a detailed description of the registers, refer to the ***PCI Local Bus Specification, Revision 2.1***.

Table 7. pcit1 Local-Side Command/Status Signals

Name	Type	Polarity	Description
io_ena	Output	High	I/O space enable. PCI command register bit 0.
mem_ena	Output	High	Memory space enable. PCI command register bit 1.
perr_ena	Output	High	Parity error response enable. PCI command register bit 6.
serr_ena	Output	High	System error. PCI command register bit 8.
tabort_sig	Output	High	Target abort signaled by the pcit1 function. PCI status register bit 11.
serr_sig	Output	High	Signaled system error. PCI status register bit 14.
perr_det	Output	High	Data or address parity error detected by the pcit1 function. PCI status register bit 15.

References

Reference documents for the pcit1 function include:

- ***PCI Local Bus Specification. Revision 2.1***. Portland, Oregon: PCI Special Interest Group, June 1995.
- ***PCI Compliance Checklist. Revision 2.1.*** Portland, Oregon.
- ***1998 Data Book***. San Jose, California: Altera Corporation, January 1998.

Revision History

The information contained in the *pcit1 PCI Target MegaCore Function Data Sheet* version 1.01 supersedes information published in previous versions. In version 1.01, the name of the HP E2925A PCI Bus Analyzer and Exerciser was corrected.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
(888) 3-ALTERA
lit_req@altera.com

I.S. EN ISO 9001