

August 1, 1996

Summary

The LogiCore PCI Interface, Version 1.1, represents a significant advancement in features and capability over Version 1.0. This document describes the differences between the two versions.

Table of Contents

Symbol Name Changes	1
Interface Signal Changes	1
Speed Grade Recommendation.....	3
Base Address Register Differences	3
Command/Status Register Differences	3
Parity Functional Differences	3
Output Enables and Turn-Around Cycles.....	3
STOP- Logic Functional Changes	4
Initiator Functional Differences	4
General Functional Differences	4

Symbol Name Changes

The LogiCore PCI interface symbol is now called PCI_LC_1.1 in Version 1.1. It was called PCI_LC.1 in Version 1.0. The name was changed to reflect that it supports Initiator functionality and to avoid potential problems in existing designs with two similarly named symbols containing different functionality.

The old and new symbols are shown in Figure 1 and Figure 2, respectively.

Interface Signal Changes

There are a few changes to the interface signals between Version 1.1 and Version 1.0. The differences are summarized in this section. Please refer to the "LogiCore PCI Interface User's Guide" for more detailed information.

New Signals on Version 1.1 Symbol

The following signals were not on the Version 1.0 symbol but have been added to the Version 1.1 symbol. These include:

- **IRDY-**: The PCI IRDY# signal captured in an input flip-flop. This signal replaces the IRDY_I- and IRDYQ- signals available on the older Version 1.0 symbol. The IRDY_I- signal came directly from the PCI bus and had difficulty meeting PCI timing in the user application. The IRDYQ- signal was not functional.
- **TRDY-**: The PCI TRDY# signal captured in an input flip-flop. This signal replaces the TRDY_I- and

TRDYQ- signal available on the older Version 1.0 symbol. The TRDY_I- signal came directly from the PCI bus and had difficulty meeting PCI timing in the user application. The TRDYQ- signal was not functional.

- **STOP-**: The PCI STOP# signal captured in an input flip-flop.
- **DEVSEL-**: The PCI DEVSEL# signal captured in an input flip-flop.
- **SRC_EN**: A signal used in burst applications to advance the pointer for the source data in the user application. Used only during Target Read or Initiator Write operations, where the LogiCore PCI interface is the source of data. SRC_EN also indicates when the output flip-flops are enabled to capture the source data available on the ADIO[31:0] bus.
- **KEEPOUT**: Used in conjunction with TERM asserted High and READY deasserted Low to provide exclusive access to the internal ADIO[31:0] bus. Prevents any Target operation from using the internal bus lines.
- **TIME_OUT**: Indicates when the Initiator's Latency Timer has expired, if enabled in the macro. Replaces the LAT_TIME[7:0] bus which provided the contents of the Latency Timer Register to the back-end user application. The Latency Timer is now integrated in the Initiator state machine.
- **CSR[39:0]**: Contains the extended Command/Status Register. CSR[15:0] matches the contents of the LogiCore macro's Command Register. CSR[31:16] matches the contents of the LogiCore macro's Status Register. CSR[39:32] defines the current status of a bus transaction.

Old Signals with New Names

Some of the signals on the symbol have new names to clarify their function. These signals, listed by their new name, include:

- **S_WRDN**: Was called WR_RD-IN. S_WRDN is the write/read direction control for slave (Target) accesses.
- **S_CBE[3:0]**: Was called CBE_IN[3:0]; S_CBE[3:0] is the command/byte enables for slave (Target) accesses.

LogiCore PCI Version 1.1 Compared to Version 1.0

- **M_WRDN:** Was called WR_RD_OUT. M_WRDN is the write/read direction control for master (Initiator) transactions.
- **M_CBE[3:0]:** Was called CBE_OUT[3:0]: S_CBE[3:0] is the command/byte enables for master (Initiator) accesses.
- **M_ADDR:** Was called ADDR, active-High. **IMPORTANT:** M_ADDR- is the active-Low address state from the Initiator state machine. This is the opposite polarity from the previous version. Indicates when the user application should present address and command values. This signal was inverted in order to reduce a level of logic when enabling address values onto the bus through three-state buffers.

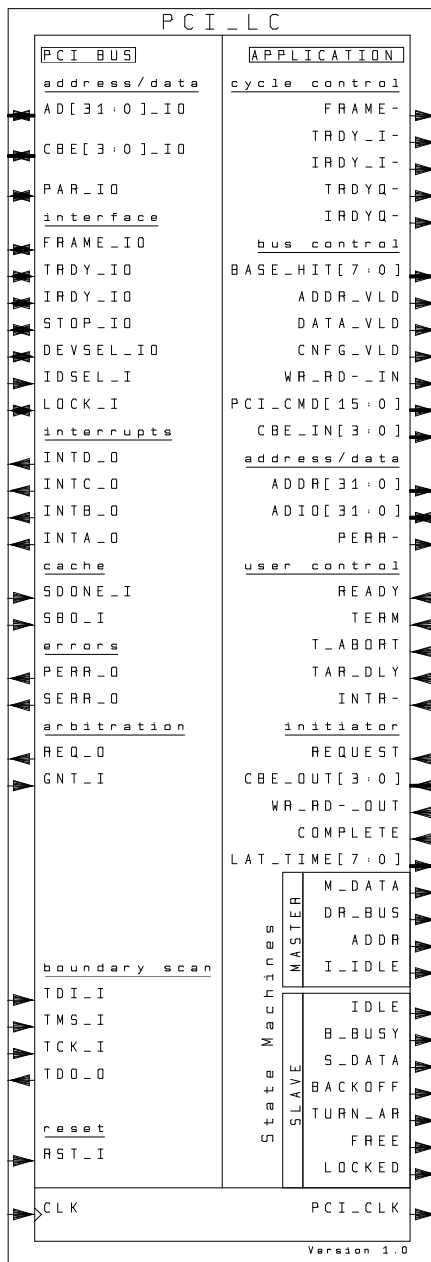


Figure 1. Version 1.0 Symbol (PCI_LC).

Old Signals Deleted in Version 1.1

A few of the signals were deleted between revisions. These were deleted because they became obsolete with new functionality or were no longer needed.

- **TRDY_I-**: Replaced by TRDY- signal.
- **IRDY_I-**: Replaced by IRDY- signal.
- **TRDYQ-**: Replaced by TRDY- signal.
- **IRDYQ-**: Replaced by IRDY- signal.
- **TAR_DLY**: Only required for Targets that support fast decode. Not applicable to this design.
- **LAT_TIME[7:0]**: Replaced by TIME_OUT signal. The

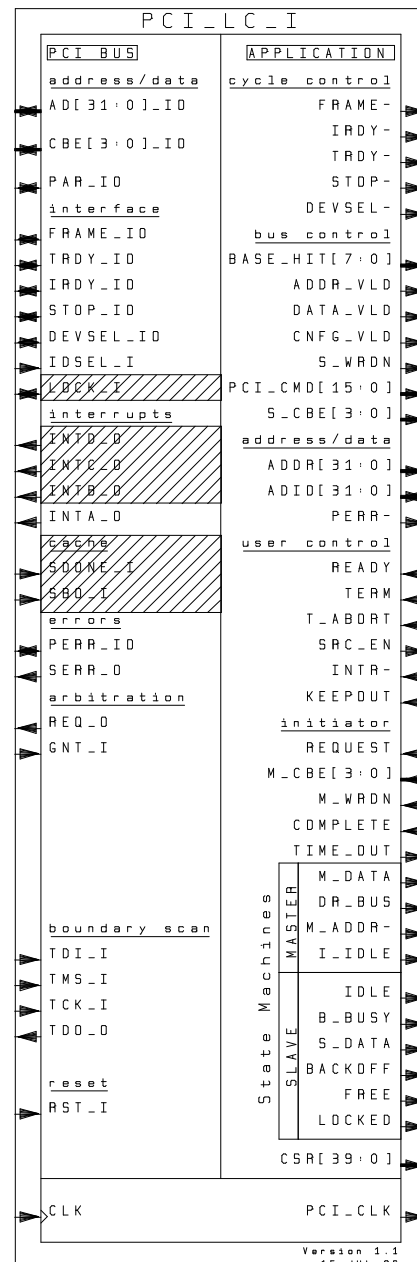


Figure 2. Version 1.1 Symbol (PCI_LC_I).

Latency Timer is now integrated in the Initiator state machine.

- **TURN_AR**: The Target turn-around state is included as part of the S_DATA state because of data pipelining. TURN_AR was tied Low in Version 1.0.

Speed Grade Recommendation

The XC4000E-2 speed grade is required for all 33 MHz Initiator designs and is recommended for all 33 MHz LogiCore PCI designs in general. Some Target-Only applications may be able to use the XC4000E-3 speed grade.

Base Address Register Differences

Base Address Register Size and Type

In Version 1.0, the Base Address Registers (BARs) were customized by substituting various schematic pages to decode different address ranges. This process is replaced by a single symbol (BASE_REG) that can be customized for any decode size and memory type.

Memory Base Register Command Decoding

In Version 1.0, memory base registers only responded to memory read and write commands. In Version 1.1, memory base registers also respond to Memory Read Multiple (MRM), Memory Real Line (MRL), and Memory Write and Invalidate (MWI) commands.

Command/Status Register Differences

In Version 1.0, the bits of Status Register were provided, but were left to the user to connect through the user application. In Version 1.1, the full Command/Status Register is supported and pre-implemented in the macro.

Command Register

The following changes were made to the Command Register:

- **Bus Master Enable** – selectively enables the Initiator state machine. This bit must be set before the Initiator state machine can request the bus.
- **Memory Write and Invalidate Supported** – made read-only and defined as '0'.
- **VGA Palette Snoop Enable** – made read-only and defined as '0'.
- **Report Parity Error** – selectively enables parity error response.
- **Address Stepping Supported** – made read-only and defined as '0'.
- **SERR- Enable** – selectively enables SERR- system error response.

- **Fast Back-to-Back Enable** – made read-only and defined as '0'. Cannot be written.

Status Register

The following changes were made to the Status Register:

- **66 MHz Capable** – made read-only and defined as '0'.
- **Fast Back-to-Back Capable** – made read-only and defined as '0'.
- **Data Parity Error Detected** – set by Initiator when it detects a data parity error during a transaction that it initiated.
- **DEVSEL- Timing** – made read-only and defined as '10', which defines a slow decode.
- **Signaled Target Abort** – set by the user application when Target Abort (T_ABORT) is asserted during a Target access.
- **Received Target Abort** – set by the Initiator state machine when it receives a Target Abort condition during a transaction that it initiated.
- **Received Master Abort** – set by the Initiator state machine when it detects that no Target has responded to a transaction that it initiated.
- **Signaled System Error** – set whenever the LogiCore PCI interface detects an address parity error. Parity errors are not checked during Special Cycles.
- **Detected Parity Error** – set whenever the LogiCore PCI interface detects a parity error.

Parity Functional Differences

There were functional errors in the parity logic in Version 1.0. Version 1.0 reported parity errors on PERR- in the wrong clock cycle (one cycle too late). The Version 1.0 Initiator also generated incorrect parity on PAR. These have been corrected and verified in Version 1.1.

Version 1.1 also includes system error, SERR-, support plus parity errors are reported in the Status Register.

Output Enables and Turn-Around Cycles

Post-release testing indicated that Version 1.0 performed some of its turn-around cycles on the wrong clock edge (one clock too late). These problems have been corrected in Version 1.1.

There were also problems in Version 1.0 related to contention on the internal bus, ADIO[31:0], especially during Configuration Read and Write operations. Extra controls were added to the ADIO bus output enables to prevent contention.

Also, a new feature called KEEPOUT provides exclusive access to the internal ADIO bus without interference from Target accesses. See the "LogiCore PCI Interface User's Guide" for more information.

STOP- Logic Functional Changes

The STOP- logic was changed to add Target Abort support (T_ABORT signal) and to make Target Terminations more robust.

Initiator Functional Differences

Version 1.0 contained a pre-release of the Initiator functionality. This version was not fully compliance tested for release. Version 1.1 contains a fully-verified Initiator function. The functional differences are listed below.

Master Enable bit

The Initiator state machine cannot request the bus or initiate transfers until the Bus Master Enable is set in the Command Register. The macro will still support Bus Parking, even if the Master Enable bit is not set.

Latency Timer

In Version 1.1, the Latency Timer is integrated in the Initiator state machine. In Version 1.0, it was left to the user to incorporate into the user application logic.

Bus Parking

Bus Parking was not supported in Version 1.0. It is supported in Version 1.1. When parked (GNT- asserted but no active REQUEST), the macro will drive its AD[31:0] and CBE[3:0] lines, followed one cycle later by PAR. The value driven on the bus is provided by the user application. During Bus Parking, the actual values on the address and CBE pins are not important—the lines must be prevented from floating.

GNT-/FRAME- Latency

In Version 1.0, the Initiator required three clock cycles between recognizing a valid GNT- signal and the time that it first asserted FRAME-. This proved too long for some applications.

In Version 1.1, this GNT-/FRAME- latency time is reduced to a single clock cycle. Once the macro receives GNT-, and the bus is idle, it asserts FRAME- on the next rising clock edge.

COMPLETE logic during proper slot

The COMPLETE signal is used to end an Initiator transaction. In Version 1.0, COMPLETE ended the transaction

by always de-asserting FRAME- two cycles later, regardless of any wait states inserted by either the Target or Initiator.

In Version 1.1, COMPLETE de-asserts FRAME- only after the second-to-last data transfer successfully completes.

In order to guarantee proper operation and 33 MHz performance, the Initiator state machine inserts an IRDY- wait state immediately before de-asserting FRAME-. This extra wait-state is hidden during Initiator Write transactions. On Initiator Read transactions, the extra wait-state is hidden if the Target inserts a TRDY- wait state.

General Functional Differences

Tri-State During Reset

When RST- is asserted, all output pins should be high-impedance. This is guaranteed in Version 1.1 by using the Global Three-State feature available in the XC4000E.

Design Documentation

Documentation on the design hierarchy is imbedded in the schematic. See the schematic page named TREE.1 for more information.

Documentation on how to use the new Base Address Register functions is included in the schematic page named BAR_INFO.1.

Pipelined Data Sourcing

Version 1.1 uses the clock enable feature on XC4000E I/O flip-flops to hold data during Initiator Write and Target Read operations. The data provided on ADIO[31:0] is automatically captured and held in the output flip-flops. This frees the ADIO bus and provides maximum data throughput. The SRC_EN signal can be used to advance the pointer providing source data. Therefore, new data can be presented on the ADIO[31:0] bus while the previous data is held in the output flip-flops awaiting transfer.

LC_PCI Library

The non-modifiable portions of the LogiCore PCI Interface have been grouped into a new library called LC_PCI. This requires a new library entry in the `viewdraw.ini` file.