

Columbia University
Department of Electrical Engineering
EECS E4340. Laboratory #3B.
PCI Core
Due: April 12, 2004

Please refer to the following documents which can be found on the *Handouts* page on the class website:

- PCI Core Implementation Guide
- PCI Core Design Guide
- PCI Core Release Notes
- PCI Local Bus Specification. These is a copy of the actual bus specification.
- PCI Core Example Design

We are interfacing to a predesigned core for the PCI interface. This core is described in more detail in the first three documents referenced above. In this lab, I would like you to become familiar with the core and make some modifications to the PING64 reference design (which is documented in the last handout shown above). You will find the source code for the PING64 reference design in:

/tools2/courses/ee4340/pci_reference/example

The simulatable VHDL for the core is in:

/tools2/courses/ee4340/pci_reference/src/xpci

As you will notice, the VHDL for the core is written as a netlist of primitives to deliberately make the details of its implementation opaque.

Using the reference design as a guide, please create a testbench that operates as a dumb host, communicating with the PCI core as a 64-bit target. In the reference design, you can “strip out” the *ARB*, *TRG32*, and *TRG64* components. Please build an interface to the backend of the core that has the following signals; please tie the other signals of the core back-end as appropriate:

- *pci_clk(output, std_Logic)*. The PCI bus clock.
- *pci_rst(output, std_Logic)*. The PCI bus reset signal (inverted) which will be used as an asynchronous reset of the DDR interface.

- *pci_wrtn(output, std_Logic)*. Status indicator to indicate whether (as a target) the PCI core is writing or reading data.
- *pci_sdata(output, std_Logic)*. Status indicator to indicate that the PCI core is in data transfer as a target.
- *pci_base_bit(output, std_Logic_vector(2downto0))*. Indicates that the core has decodes and matches an address in the memory spaces defined by the base address registers.
- *pci_addr(output, std_Logic_vector(31downto0))*. Returns the value of the PCI bus address, latched during the address phase.
- *pci_addr_vld(output, std_Logic)*. Indicates the beginning of the address phase.
- *pci_data_vld(output, std_Logic)*. Indicates that a data transaction has occurred and data is available on the bus.
- *pci_ready(input, std_Logic)*. Used to signal to the core that the under interface is ready to transfer data.
- *pci_term(input, std_Logic)*. Used to signal to the core that the user interface wants to terminate the transaction.
- *pci_abort(input, std_Logic)*. Used to signal an error condition to the core to terminate the transaction.
- *pci_data(inout, std_Logic_vector(63downto0))*. The tri-state time-multiplexed address and data bus.

For the PCI core, please use the configuration file contained in:

/tools2/courses/ee4340/pci_reference/cfg.vhd

Please only use the memory space defined by Base Address Register 0, which will have to be configured as part of your PCI start-up from your testbench. Please also do some simple reads and writes to the memory space.