PCI/WISHBONE Bridge

Reference Design

FPGA-RD-02135-1.4

January 2020

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

1. Introduction

PCI Local Bus is an industrial standard developed to seamlessly integrate modern embedded applications into complex systems. Features include a well-documented standard supported by a special interest group and the performance of a 33 MHz, 32-bit version of the specification reaching 132 Mbytes per second at its peak transfer rate. The performance and compatibility offered by PCI Local Bus allow the standard to be quickly adopted in most digital systems.

The WISHBONE bus is intended to be used as a general purpose interface. It fosters design reuse by providing a common bus interface for all IP cores. Its architectural advantages include a flexible integration solution for specific applications and a variety of bus cycles and data path widths. In addition, a variety of suppliers can create designs for the WISHBONE bus interface. This enables lower-cost and higher-performance alternatives to be used.

This reference design provides an interface between the PCI initiator and the WISHBONE slave device to take advantage of both standards. It provides a bridging function between multiple IP cores to a common PCI interface.

2. Features and Limitations

This reference design offers the following features:

- Independent clock domains for the PCI and WISHBONE sides of the bridge
- Supports 33 MHz,32-bit PCI target functions with PCI Local Bus Specification Rev. 2.2
  - Two base address regions (I/O and memory regions)
  - Single cycle and burst modes for read and write cycles
  - Implementation of all required PCI configuration registers
  - Parity generation for all read cycles
- Two FIFOs between PCI target module and WISHBONE master module for clock domain transfers
- WISHBONE bus revision B.3 compliant with master interface

The reference design does not support the following features:

- PCI target interface
  - PERR and SERR
  - Expansion ROM
  - Built-in Self Test (BIST)
  - Burst cycles into the configuration register space
  - Cache line register
  - Interrupt signal from PCI target to PCI initiator
- WISHBONE master interface
  - Respond to signal of RYT_I and ERR_I
  - WISHBONE clock frequency less then 33 MHz

3. Theory of Operation

3.1. PCI Bus

PCI is a highly documented specification. A special group (SIG) called PCI SIG publishes the PCI Local Bus Specification. To understand the basic functionality of the PCI bus, the PCI Local Bus Specification Rev. 2.2 is recommended reading for this reference design.

3.2. WISHBONE Bus

WISHBONE is also a highly documented specification. To understand the basic functionality of the WISHBONE bus, the WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores Specification Rev. B.3 is recommended reading for this reference design.

4. Functional Description

This PCI/WISHBONE bridge reference design is used to interface a PCI initiator, or master, and a WISHBONE slave device. It acts as a target on the PCI side and as a master on the WISHBONE side. It is located between the PCI initiator and the WISHBONE slave device to reduce effort required to deal with the PCI command. The PCI/WISHBONE bridge responds to read/write cycles started by the PCI initiator. The controller functions as a data path controller, transferring data to and from the PCI initiator onto the WISHBONE slave device. Figure 4.1 shows the relationship of the bridge between the PCI initiator and the WISHBONE slave device.

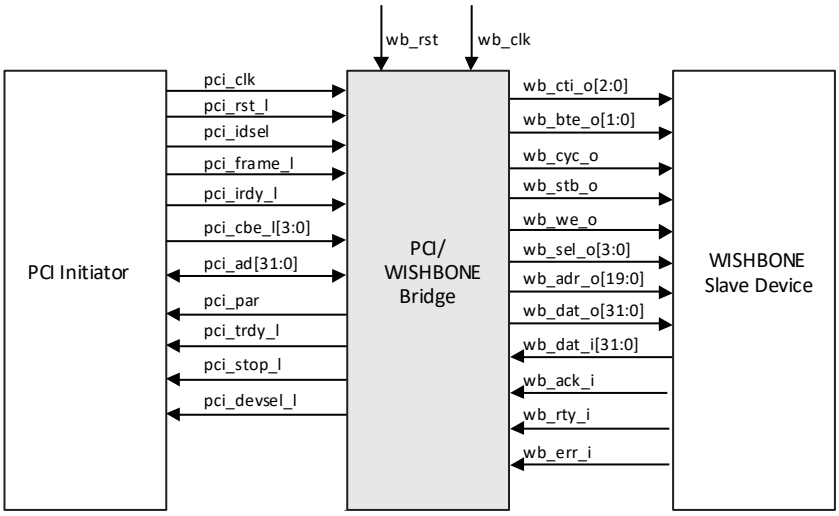

Figure 4.1. PCI/WISHBONE Bridge System

Table 4.1. PCI/WISHBONE Bridge Pin Descriptions

Signal Name	Signal Direction	Active State	Definition
PCI Initiator Interface			
pci_clk	Input	N/A	The clock input to all PCI devices on the PCI bus including PCI targets and PCI initiators.
pci_rst_l	Input	Low	The active low reset for all PCI devices on the PCI bus.
pci_idsel	Input	High	The PCI initiator will drive this signal high at the input of the PCI target that should complete the current configuration cycle on the PCI bus.
pci_frame_l	Input	Low	The PCI initiator drives this signal low at the beginning of a cycle and high at the clock edge before the last data phase on a burst operation.
pci_irdy_l	Input	Low	The PCI initiator drives this signal low prior to the positive edge of a clock when it can complete a data phase.
pci_cbe_l	Input	Low	The multiplexed PCI command/byte enables.
pci_ad	Bi-Directional	N/A	The multiplexed PCI address/data bus.
pci_par	Bi-Directional	Even Parity	The even parity bit. The PCI target drives this signal during read cycles. The PCI initiator drives this signal during the address phase of all transactions and the data phase during writes.
pci_trdy_l	Output	Low	The PCI target drives this signal low prior to the positive edge of a clock when it can complete a data phase.
pci_stop_l	Output	Low	The PCI target drives the stop signal low during a transaction to indicate the termination of a cycle.
pci_devsel_l	Input	Low	The PCI target drives this signal low to indicate the address of the current transaction is in the address space of the base address registers.
WISHBONE Slave Interface			
wb_cti_o	Output	N/A	This signal provides information about the current cycle type identifiers.
wb_bte_o	Output	N/A	This signal provides information about the current burst type.
wb_cyc_o	Output	High	When asserted, indicates that a valid bus cycle is in progress.
wb_stb_o	Output	High	The strobe output signal indicates a valid data transfer cycle.
wb_we_o	Output	1 = Write 0 = Read	This signal is negated during read cycles and is asserted during write cycles.
wb_sel_o	Output	High	The select output signals indicate where valid data is expected on the DAT_I signal during read cycles, and where it is placed on the DAT_O signal during write cycles.
wb_adr_o	Output	N/A	The address output.
wb_dat_o	Output	N/A	The data output during write cycles.
wb_data_i	Input	N/A	The data input during read cycles.
wb_ack_i	Input	High	When asserted, indicates the normal termination of a bus cycle.
wb_rty_i	Input	High	This signal indicates that the interface is not ready to accept or send data, and that the cycle should be retried.
wb_err_i	Input	High	This signal indicates an abnormal cycle termination.
wb_clk	Input	N/A	The WISHBONE system clock.
wb_rst	Input	High	The active high reset for all WISHBONE devices on the WISHBONE bus.

5. Register Transfer Level Implementation

The RTL block diagram of the PCI/WISHBONE bridge is shown in Figure 5.1 It consists of five modules: the top level module, the PCI target module, the WISHBONE master module, the data buffer module and the synchronization module.

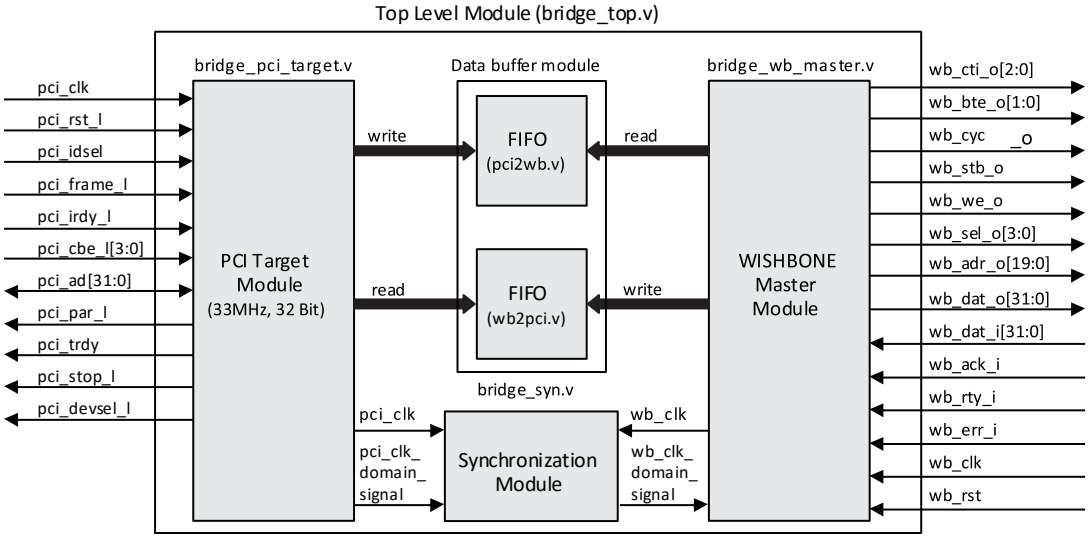

Figure 5.1. Block Diagram

5.1. Top Level Module

This is the top level HDL block of the reference design. This module is created to instantiate the lower level modules of the design.

5.2. PCI Target Module

This module is based on Designing a 33 MHz, 32-Bit PCI Target Using Lattice Devices (FPGA-RD-02116). It contains the functional blocks listed below. A detailed description of each functional block in this module can be found in FPGA-RD-02116.

- Miscellaneous glue logic block
- Configuration multiplexer block
- Base address check block
- State machine
- Parity generation block
- Retry counter block

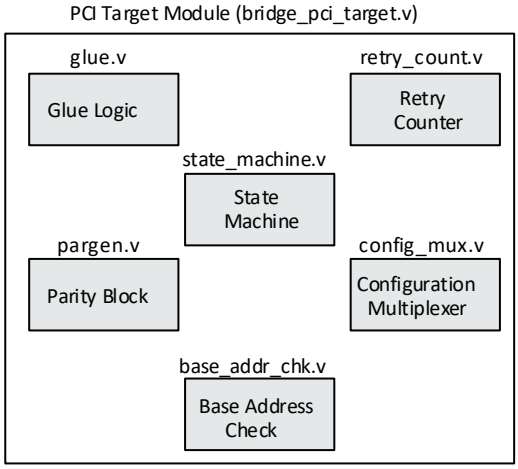

Figure 5.2. PCI Target Module

5.3. Data Buffer Module

The data buffer module is implemented by two FIFOs which are generated by IPexpress™ in the ispLEVER® design tool. When the PCI initiator initiates a WRITE command, the PCI target module stores the data in a FIFO and the WISHBONE master module checks the same FIFO to see if it is empty. If the FIFO is not empty, the WISHBONE master module reads data from the FIFO and writes data to the WISHBONE slave device. On the other hand, when the PCI initiator initiates a READ command, the PCI target module transfers a read signal to the WISHBONE master module and waits for the ready signal from the WISHBONE master module. Once the WISHBONE master module detects a read signal from the PCI target module, it sends a READ command to the WISHBONE slave device and writes data to the FIFO.

5.4. Synchronization Module

The PCI target module and the WISHBONE master module of this design communicate with each other to complete READ or WRITE requests initiated by the two interfaces. This module provides a synchronization mechanism between the PCI target module and the WISHBONE master module because they are operated in different clock domains. In this module, the signals generated in the PCI clock domain are sampled by the WISHBONE clock before they come into the WISHBONE clock domain for purposes of synchronization.

This design presumes the WISHBONE clock frequency to be faster than the PCI clock frequency. Thus, the signals from the PCI clock domain are sampled by the WISHBONE clock. In situations where the WISHBONE clock frequency is less than 33 MHz, the design should be modified to meet the requirements of the sampling theory.

5.5. WISHBONE Master Module

The WISHBONE master module is a 32-bit WISHBONE master interface and implements a WISHBONE master function as defined in the WISHBONE specification. Requests are sent to the WISHBONE slave device through this module.

The WISHBONE master module performs burst transfers as block cycles, as described in the WISHBONE specification. The cycle type defined by the signal CTI_O is either “Incrementing burst cycle” or “End-of-Burst” based on the response of the WISHBONE slave device. The burst type defined by the signal BTE_O is always “Linear burst type”.

When the PCI target module decodes an address to fall within a range of one of its enabled images, it sends signals to the WISHBONE master module and these signals trigger the state machine in the WISHBONE master module to generate the corresponding operations. Figure 5.3 shows the state diagram of the state machine in the WISHBONE master module.

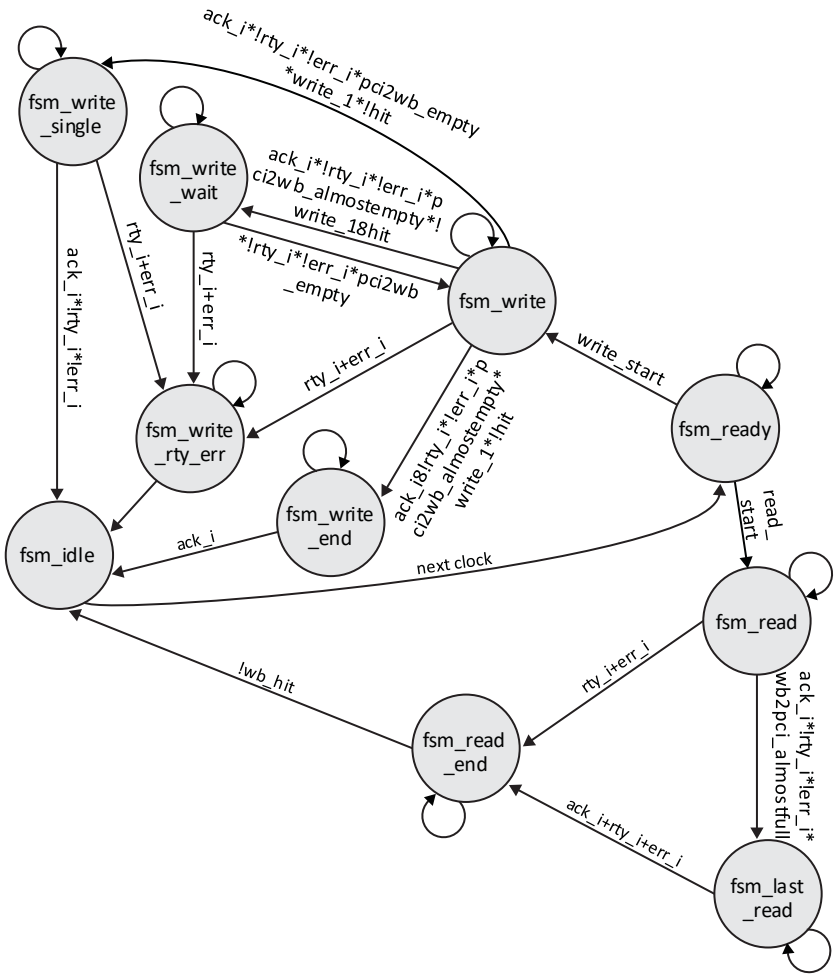

Figure 5.3. WISHBONE Master State Machine

PCI to WISHBONE Write Cycles

During reset, the state machine is forced to the FSM_IDLE state. After reset, the state switches from FSM_IDLE to FSM_READY state at the positive edge of the signal WB_CLK. The state machine stays in FSM_READY as long as signal WRITE_START or READ_START is inactive which indicates no operation on the PCI bus. In this state, the module generates the signal READY which is used in the PCI target module. Once the signal WRITE_START is active, the PCI bus initiates write cycles, the state machine switches to FSM_WRITE state, and the WISHBONE master module performs write cycles. The state machine fetches the address into the address counter. This register counter is connected directly to address output bus. Then it fetches each data byte with byte enables from the pci2wb_fifo and puts it on the WISHBONE bus until there is no more data in pci2wb_fifo and the PCI bus completes the write cycles. In the mean time, address counters are incremented for each data phase. The data is terminated with signal WB_ACK_I. The state machine samples the signal EMPTY, ALMOSTEMPTY of the pci2wb_fifo and PCI target module synchronous signal DATA_WRITE_L and WB_HIT to estimate whether the PCI bus performs single write cycles or burst writes cycles. If there is only one data in the pci2wb_fifo and the PCI bus completes the write cycles, the state machine switches to the FSM_WRITE_SINGLE state. If no data is left in the pci2wb_fifo and the PCI bus has not completed the write cycles, the state machine switches to the FSM_WRITE_WAIT state until the pci2wb_fifo is not empty. Whenever the WRITE is terminated with signal WB_RTY_I or WB_ERR_I, the state machine switches to the FSM_RTY_ERR state to stop the WISHBONE bus WRITE cycles and meanwhile sends the signal WB_STOP to the PCI target module.

PCI to WISHBONE Read Cycles

When the PCI bus performs READ cycles, the signal READ_START must be active and the state machine switches to the FSM_READ state. In READ cycles, the state machine does not know the burst read length requested by the PCI bus. It reads 16 data bytes (the length of wb2pci_fifo) with all bytes enabled from the WISHBONE slave device when data is terminated with signal WB_ACK_I. If the signal ALMOSTFULL of the wb2pci_fifo is active, the state machine switches to the FSM_LAST_READ state to perform the last READ cycle. If the READ cycle is terminated with signal WB_RTY_I or WB_ERR_I, the state machine switches to the FSM_READ_END state to stop WISHBONE bus READ cycles and meanwhile sends the signal READY to the PCI target module, which informs the PCI target module to read data from the wb2pci_fifo.

When the burst read length requested by the PCI bus is more than 16 bytes, the signal EMPTY of wb2pci_fifo will stop the PCI bus transfer. In addition, the signal WB2PCI_FIFO_RST is generated in the FSM_READY state to clear the wb2pci_fifo before every READ cycle starts.

6. HDL Verification

6.1. Functional Blocks of the HDL Test Suite

The HDL code for the PCI/WISHBONE Bridge reference design test suite contains the following blocks:

- Test bench top level
- PCI initiator generator
- Clock and reset block
- WISHBONE slave device generator
- PCI/WISHBONE bridge under test

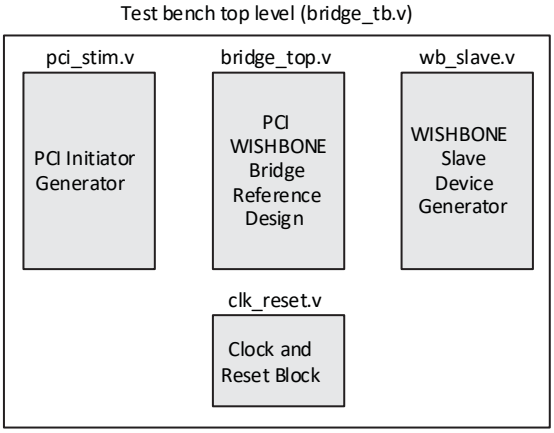

Figure 6.1. Test Bench Block Diagram

6.2. Test Bench Top Level

The PCI/WISHBONE bridge test bench is used to verify the timing and functionality of this reference design. It simulates of the PCI initiator read or write transfers and the interface with the WISHBONE slave device.

6.3. Clock and Reset Block

The clock and reset block is used to generate the 33 MHz PCI clock and WISHBONE clock for the test suite.

6.4. PCI initiator Generator

The PCI initiator generator block is the heart of the PCI test suite. The stimulus generator creates all the PCI cycles as a PCI initiator would on a PCI bus. It also checks the data returned from the PCI target and the condition of its control signals. All the simulation tasks and functions are defined and executed in this block. It controls all the automation and self-check verification features of the test suite. Each major task is derived from minor functions used to generate the transactions. The following is a list of tasks contained in this module:

- **pci_reset** – Sets the stimulus block up for a reset.
- **read_config** – Performs a PCI configuration read.
- **write_config** – Performs a PCI configuration write.
- **read_cycle** – Performs a single, burst, memory or I/O PCI read cycle.
- **write_cycle** – Performs a single, burst, memory or I/O PCI write cycle.
- **pci_sniff** – Calls the entire configuration read and write cycle set needed to initialize the PCI target.
- **write_read_test** – Calls memory and I/O read write cycles to test burst and single cycle transactions.
- **write_special_cycle** – Calls special write cycles to test WISHBONE slave device stop cycles.

- **check_cycle** – A task developed to test the termination of any PCI cycle and determine if the cycle was a data transfer, a data transfer with a stop, a data retry, an abort or a violation of the PCI protocol. The result of this check is passed onto the other tasks for self-checking.
- **kill_time** – Used to run the simulation and space out PCI transactions by five PCI clock cycles.
- **check_data** – Checks data returned by the PCI target during configuration, memory or I/O reads and compares it with expected values.
- **check_parity** – Checks the PCI parity value returned during a configuration, memory or I/O read and compares it with the expected parity values.

6.5. WISHBONE Slave Device Generator

This generator is a WISHBONE slave device behavioral model used to model the functionality of a WISHBONE slave device. If a write is made to address 00050h, the generator responds with signal WB_ERR_O instead of signal WB_ACK_O, causing a data error termination.

7. Timing Specifications

The following timing diagrams show the major timing milestones in the simulation.

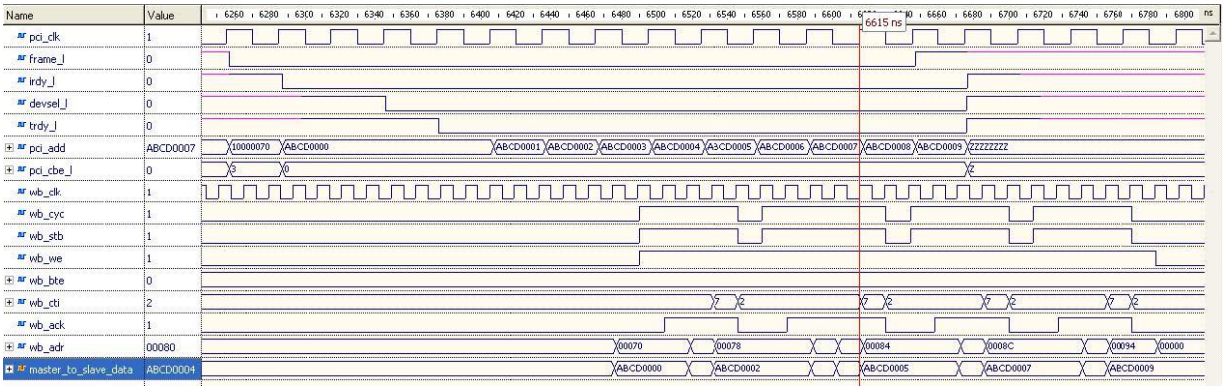

Figure 7.1. PCI to WISHBONE Write Cycles

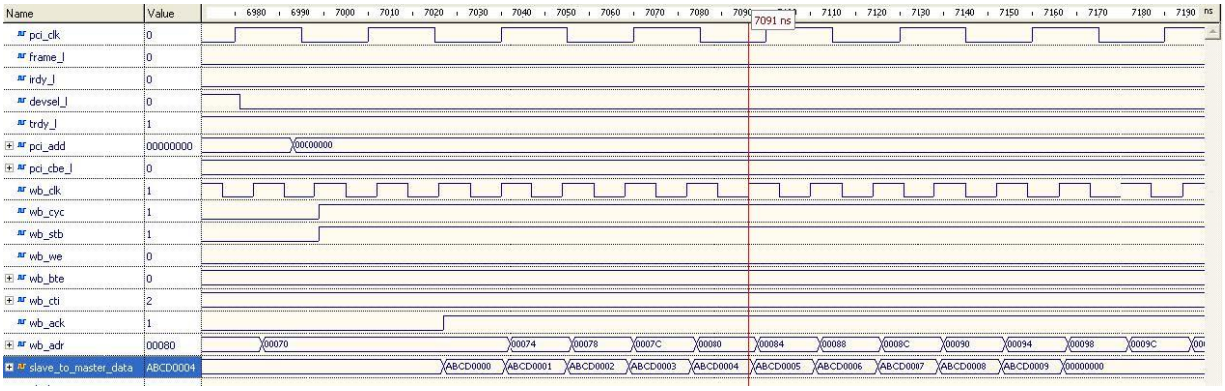

Figure 7.2. PCI to WISHBONE Read Cycles

8. Implementation

This design is implemented in Verilog and VHDL. When using this design in a different device, density, speed, or grade, performance and utilization may vary. Default settings are used during the fitting of the design.

Table 8.1. Performance and Resource Utilization

Device Family	Language	Speed Grade	Utilization (LUTs)	f_{MAX} (MHz)	I/Os	Architecture Resources
LatticeECP3™ [1]	Verilog	-8	627	33	146	2 EBRs
	VHDL	-8	602	33	146	2 EBRs
LatticeXP2™ [2]	Verilog	-7	628	33	146	2 EBRs
	VHDL	-7	604	33	146	2 EBRs
MachXO™ [3]	Verilog	-5	422	33	146	2 EBRs
	VHDL	-5	414	33	146	2 EBRs

Notes:

1. Performance and utilization characteristics are generated using LFE3-70E-8FN484C with Lattice Diamond™ 1.2 design software.
2. Performance and utilization characteristics are generated using LFXP2-5E-7FT256C with Lattice Diamond 1.2 design software.
3. Performance and utilization characteristics are generated using LCMXO-2280C-5FT256C with Lattice Diamond 1.2 design software.

References

- PCI Local Bus Specification Rev. 2.2
- WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores Specification Rev. B.3
- Designing a 33MHz, 32-Bit PCI Target Using Lattice Devices (FPGA-RD-02116)

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.4, January 2020

Section	Change Summary
All	• Changed document number from RD1045 to FPGA-RD-02135. • Updated document template.
Disclaimers	Added this section.

Revision 1.3, April 2011

Section	Change Summary
Implementation	Added support for the LatticeECP3 device family and Lattice Diamond 1.2 design software.

Revision 1.2, January 2010

Section	Change Summary
Implementation	Added support for the LatticeXP2 device family.

Revision 1.1, December 2009

Section	Change Summary
All	• Added VHDL support. • Resolved a transfer issue between the PCI clock and WISHBONE clock in the bridge_syn.v module.

Revision 1.0, February 2009

Section	Change Summary
All	Initial release.

www.latticesemi.com