ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2015

An Efficient Architecture for PCI Bus Design

Amit Sanjayrao Mamidwar[1], Vrushali G. Raut[2]

PG Student [Digital System], Dept. of E&TC, Sinhgad College of Engineering, Pune, Maharashtra, India[1]

Assistant Professor, Dept. of E&TC, Sinhgad College of Engineering, Pune, Maharashtra, India [2]

ABSTRACT: In several application of embedded system, the digital components interconnects are needs flexible to power and area of the devices. Also it requires a bus width with several versions like 64 bits or 32 bits with different frequencies (Ex. 33 MHz or 66 MHz). PCI bus is used for connecting these hardware devices in computer. Instead of using different buses in computer, it can use single bus for different frequencies with multiplexing bus width. In order to solve this problem, this paper designs FPGA based PCI bus with these requirements. Through analysis of PCI bus, the top-down method applies on the PCI bus with several functional modules. Also it uses the master and target as coprocessor of the device which are programmed with finite state machine using VHDL language and structured in detail which helps to minimize the power consumption. For selecting bandwidth and frequency, it has used separate multiplexing module. Through the analysis of test bench and synthesis report it will prove that the bus complete requirements of power consumption on the basis of utilization of registers, CLB's and flip flops and helps to design with dual frequency.

KEYWORDS: PCI bus; power consumption of device; multiplexing module.

I.INTRODUCTION

The peripheral component interconnect (PCI) bus standard for I/O has one of the most useful device in embedded system. Though PCI is widely used nowadays, it has some limitations. When it checks for power consumption, it needs used as a low power device in system. Also it has a limitation for high bandwidth demand and varying it as per requirement from low bandwidth to high.

The design of high speed bus with minimum power consumption is very difficult task. Also the PCI has to meet the requirements, which gives minimum distortion. The implementation of the PCI bus includes the master and target device. When the master device is utilized, PCI has to take initiative to transmission of signals and when it is used as target, a host has to take initiative for controlling signals. The data has to transmit in a single byte mode or burst mode. This has to be control by control signals of PCI master or target module. It gives the different signals for single byte or burst mode with different module in a master or target which may cause the violation of the timing parameters. This has been solved to meet the timing parameters by using top-down approach.

The PCI has to meet the requirements of high bandwidth with maximum speed. Considering PCI bus of 33 MHz and 66 MHz for 32-bit as well as 64-bit are multiplexing the speed with bandwidth. The bus has to meet the characteristics of the device; otherwise it will introduce the timing and signal issues. This paper represents the modelling for the PCI bus which uses the top-down approach for power consumption and prototype design with a frequency 33 MHz/ 66 MHz for 32-bit and 64-bit from simulations and matching the timing specifications.

II.LITERATURE SURVEY

PCI is a bus and the latest version of PCI is PCIe 3.0 x 16 which work as a point to point connection. It works for only two devices and no other device can connect. And PCI express works on serial communication whereas PCI bus uses parallel communication. PCI bus is a high performance I/O bus utilizes interconnect peripheral devices in applications such as communication devices. PCI is an I/O device interconnects bus which connects the embedded computing and communication platforms. It is having the bandwidth up to 16GB/s. This existing system interfaces the PCI devices 64 bit data slot.

DOI: 10.15662/ijareeie.2015.0405074

PCI bus having the capability to get the inform operation to function with power management. But each internal device like configuration space, master state machine and address command controller requires informing the register information, state information and enhancement in plug and play capability. It consist device controller, message controller, mode controllers etc which maximizes the power consumption for the device up to several watts (25 watts) for normal PCI bus.

To this existing system, this design aims to replace the PCIe with PCI bus which having dual bandwidth connects to more than two devices as per required application frequency and replacing the current devices in the PCI bus into direct master and target state machine which will minimizes the power consumption.

a. PCI Bus Overview:

PCI bus is widely used in embedded applications for data transmission in burst mode. The architecture of the PCI bus is shown in Figure 1. PCI Bridge connects to the I/O devices as well as to the bus controller. This PCI Bridge controls the communication between I/O devices and buses. Bus controller connects to memory and processor. This bridge provides low latency path through which it can access to any devices. The bridge may include optional function such as hot plugging and arbitration.

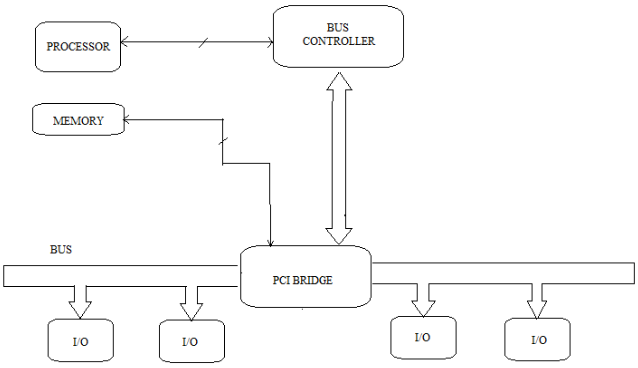

Fig.1. PCI Bus architecture

b. PCI Bus Operation

PCI bus processes the data, address, control signals and system functions which need to be accessed through modules, master or slave. The master device controls the bus and it selects the target device which doesn't have control on bus authority. In transmission of data, it includes an address pulses as well as one or more data pulses which depends on data transmission mode. It consist address/data (AD) multiplexed bus.

A read operation starts with address which occurs when frame get asserts initial clock and then occurs at next clock. During transmission of address AD [63:0] and C/BE [7:0] contains valid information. During the data phase, C/BE# indicate which byte lane are occupied in the current data phase. A data phase may consist either wait cycles or a data transfer. The C/BE# output must stay enabled from the first clock of the data phase through the end of the transaction. The C/BE# controls valid byte enable information through the entire data phase.

The data phase follows the turnaround condition which is used at the time of target select. In this case, the address is valid on the second clock and then the master stops driving AD. The target can provide valid data is the fourth clock. The target must drive the AD lines following the turnaround cycle when DEVSEL# is asserted.

The data phase completes when TRDY# are asserted on the same rising clock edge. When TRDY# is declaiming, a wait cycle is inserted and no data is transferred. When IRDY# is asserted then FRAME# goes active high or declaims.
In the write transaction mode, the process starts when FRAME# goes active low or assert. Since master provides both address and data for write transaction, there is no turnaround cycle require. Data phase works similar at write and read transaction.

III.EFFICIENT PCI MODEL DESIGN

The PCI bus implementation designed using several modules, it consists master as well as target module. PCI Bus divides the module in master/target state machine, address/command controller, parity checker, configuration space, address decoder and device clock manager.

Figure 2 shows the functional block diagram of PCI Bus with several modules. These modules connect with modular method for design operation. All of the read and write operations must be allocated by the CPU to control the bus for use. When used as the master device, it can apply to take the bus initiatively, and then the data is transmitted in bursts or single byte to the destination address. A burst transfer consist segment of address and several segments of data. It requires that the target device and the master device must understand the implicit addressing. The state machine module is separated into a master device and the target device. Each module is designed with VHDL by XILINX.

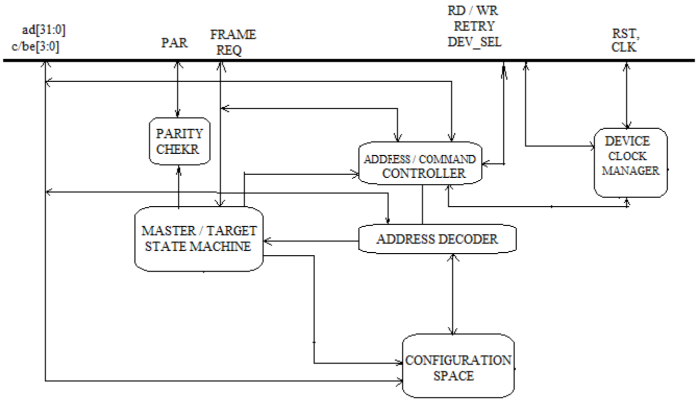

Fig.2. Functional Block Diagram of PCI Bus

A. Configuration Space

This block is used to configure the related register of PCI bus in the configuration space. Each device of PCI bus must get a configuration space for the function of bus. The core interface, by default, implements a complete type zero configuration space header.

The first 64 bytes are used for the standard header, and then the next 64 bytes are reserved for future capability items to be added to the PCI interface. The remaining 128 bytes are under the control of the user application. Typically, the user application should return zero on all configuration accesses. However, the user application can implement additional capability items in this area, or implement registers that are outside the scope of the specification.

B. Address/Command Controller

This consists of PCI bus complex control logic design. Generally the logical sequences are designed on the terminal of the AD and C/BE. A different stage of address and data, commands and byte is used for allow multiplexed output. When the module address signal is asserted, the output is addresses to the transaction of address.

C. Address Decoder

This module is proposed to verify the device's IO/MEM base address allotted by the system during the base register configured write cycle, and then in the address cycle of read and write operations, This module compares the address from PCI bus data sampled with the previous base address, so that it could determine whether the current operation of the device is for itself.

D. Parity Checker

This module is used to determine whether the master device addressing to the correct target device or the data transmission is correct. Specific verification process: in the device through latching AD# and CIBE# the XOR or parity is generated. At the next clock edge the device will generate the XOR between its internal calculation and even parity

PAR. If the two values are same, even parity is correct. Otherwise, at the next clock pulse the device will report valid data errors via PERR# or SERR#.

E. Master/Target State Machine

The module of state machine is the most necessary part of PCI bus interface. The state machine decides the current status of bus operation by determining the signal on the PCI bus. Thereby it can deal with the process of other modules and it receives signals from other modules. By analysing the PCI bus and the signal from other modules, it changes the control signals. The state machine outputs related PCI bus signals in accordance with PCI, combined with the input from the PCI bus. The PCI bus transaction completes the state machine design which is divided into the master device and the target device.

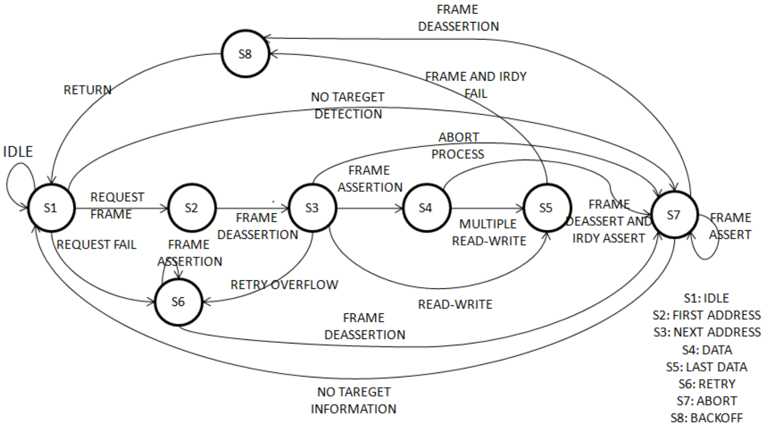

Fig.3 Sate Machine for Master

Figure 3 showing the state machine of master. It has 8 states which takes one clock for each transition. The state machine has 5 kinds of situations as unsuccessful occupancy, no answer from target, single data read operation, information retry operation and multiple data read-write operation.

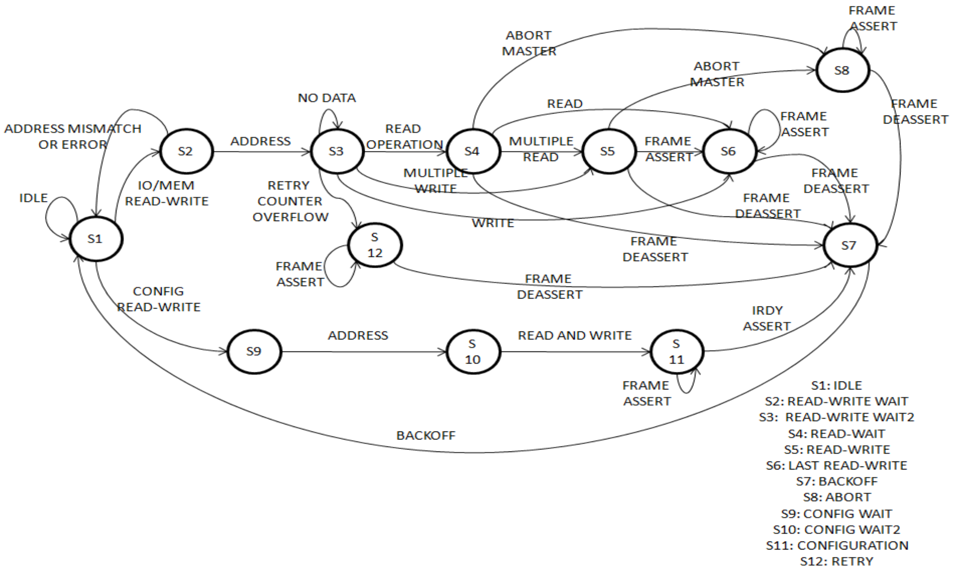

Fig.4 Sate Machine for Target

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2015

Figure 4 showing the state machine for target device. It has 12 states. Similar to master state machine, it requires single clock cycle for each transition. State machine works for unsuccessful occupancy, operation read-write for configuration, single data read, multiple data read, single data write, multiple data write, retry operation and backoff operation.

IV.RESULT AND DISCUSSION

By designing the FSM machine by using both state machines for master and target, it will improve its work which will minimize the power consumption. Figure 5 showing the controlling FSM state machine for the target that shows the combine operation for single cycle read-writes, multiple cycle read-writes and retry operation. This minimizes the slices due to combination all operation in a single FSM.

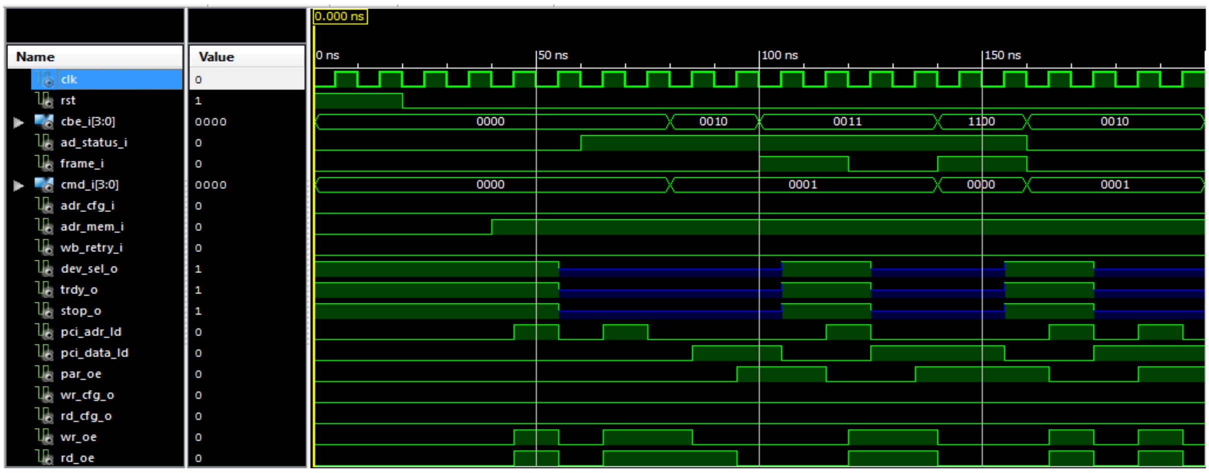

Fig.5 Simulation of Target State Machine for PCI

Initially the cbe_i[3:0] takes “0000” command, it gets interrupt command, i.e. no operation command. For “0000” command frame is getting assert for few cycle and later it get deassert. If frame asserted and address status gets low, the device select dev_sel_o, target ready trdy_o and stop_o goes to ‘X’ condition which means target has not got the address. Similarly it checks parity condition on par_oe for operation of pci. Also it checks the operation for read-write on wr_oe, rd_oe, wr_cfg_o, rd_cfg_o for memory, I/O devices or configuration space. If the value of cbe_i[3:0] is “0010” means it takes I/O read command, similarly if it is “1100”, so it will memory read with burst data.

Similarly figure 6 showing the result for parity detecting for target. It uses 64 bit checker which has divided into two 32 bit checkers. As per user requirement it will use either one or both checker which will improves its operation faster for the device. Both devices work faster which minimize the power consumption and area.

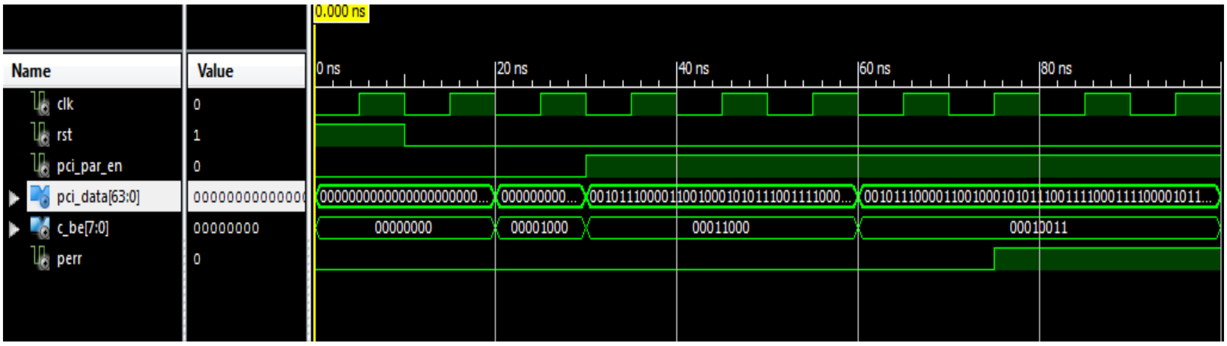

Fig.6 Simulation of Parity Checker for PCI

If pci_par_en is ‘1’, it gets enable to check the parity condition. Perr gives parity checking error for both pci_data[63: 0] and cbe[7:0] with single bit output. Now initially pci_data[63:0] and cbe[7:0] sets to all ‘0’. If pci_data and cbe gets even then perr gives output ‘0’ and if data and cbe is odd then it shos the parity error as ‘1’. It works as successful even parity checker for getting the parity error information.

VI.CONCLUSION

This paper proposed FPGA-based PCI bus of low-power devices. The interface of PCI bus is representing program with the top-down approach. The modules of top layer, state-machine, configuration register, base address check and even-parity are designed. And the simulation of read operation, write operation and configuration operation will prove that the design conform PCI bus specification as compare to PCI 64_ug262. This paper will get the low-power embedded device increase to applications of PCI bus. It will meet requirements of the design. For manufacturers, PCI bus interface would be reserved. It will not effect on resources and its waste.

REFERENCES

[1] Mingji Yang, Lei Wu, Haokun Shi, and Harbin, “Design of PCI-104 Bus Interface of Low-Power Embedded CPU Based on FPGA”, IEEE International Conference on Measurement, Information and control, vol. 01, pp. 275-279, August 2013.

[2] Hossein Kavianipour, Christian Bohm, “A High-Reliability PCIe Communication System for Small FPGAs” , 2013 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), pp.1-4, Oct.2013-Nov.2013.

[3] Mohammad S. Sharawi, “Signal Integrity Characterization and Modeling of a PCI/PCI-x 66/133 MHz Bus”, Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on , pp.490-493, August 2008.

[4] Ms. Awani N. Gaidhane, Mrs. Manish Pankaj Khorgade, “FPGA Implementation of Serial peripheral Interface of Flex-Ray Controller”, Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on , pp.128-132, March 2011.

[5] PCIBook, “PCI Local Bus Specification”, PCI Group, revision 2.3, Portland; 2002.

[6] XILINX User Guide, “LogiCORE IP Initiator/Target v4.18 for PCI”, pci_64_UG262, User Guide XILINX, October 2012.