Philippine Engineering Journal (2000) XXI (1): 38-41

PCI PROTOTYPING PLATFORM

Richard T. Garcia, Ronoel M. Sarte, and Louis P. Alarcon
Intel Microprocessors Laboratory
Department of Electrical and Electronics Engineering
University of the Philippines

ABSTRACT

Peripheral Component Interconnect or PCI, a high-speed bus technology is being considered as superior to other bus technologies when it comes to design and performance. Different adapters, which need fast accesses with one another or with system memory, can be connected to the PCI bus with the clock frequency approaching the processor's full speed.

The stringent electrical, functional, and timing specifications of PCI make it a formidable design challenge. A flexible PCI solution is needed to meet the current and the future requirements. Using a Very High Speed Integrated Circuit Hardware Description Language (VHDL) and Xilinx Field Programmable Gate Array (FPGA) Technology, a PCI prototyping platform card can be developed.

The PCI prototyping platform card will be implemented using PCI 32-bit wide bus implementation with a clock frequency of 33MHz. This card is capable of performing read and write transactions. Three registers will be provided for prototyping purposes. It is here where data can be read and written.

The PCI prototyping platform card will be designed using VHDL and will be implemented using Xilinx FPGA technology.

It is the objective of this project to develop and implement a PCI prototyping platform card capable of performing read and write transactions which will serve as a benchmark for future PCI-based projects.

I. Introduction

Slow execution time related to memory and I/O functions adversely affects the performance of computer systems. Such scenario is very evident in peripheral devices such as LAN, SCSI and motion video.

The PCI bus, jointly developed by Intel and other industry leaders including IBM, is the latest bus architecture that addresses performance issues related to bandwidth problems. It provides a data path capable of accessing up to 264Mbytes per second [1]. Both the present and future processors of different architecture can be fully supported by PCI bus component and add-in card interface since they are processor-independent and it is also the first bus to support both 3.3V- and 5V- cards.

Mandatory configuration registers containing device-specific information enable the system BIOS to automatically configure PCI bus components and add-in cards. Through automatic device configuration, the need for hardware jumpers and software utilities can be eliminated and assignment of same system resource to two or more devices can be avoided.

Adding new components, such as add-in cards, to the computer system increases the existing capabilities of the machine. However, there is a need to configure these new components. An installer is usually needed to run software set-up utilities in configuring these devices, adding more trouble to the user.

With the advent of Plug and Play architecture, which is fully supported by the PCI bus, nothing will be required of the user except to "plug" the device into the computer and it will automatically "play".

Plug and Play requires identity of the device's function, what services the device provides, and the system resources the device needs in order to function, which is implemented in the PCI Configuration Space header region [2] .

II. Project Description

The objective of the project is to design and fabricate a 5V PCI Prototyping Platform Card that has the following features:

- PCI compliant 32-bit, 33 MHz PCI Interface (Target only)
- Supported Target Functions
  - Type Zero Configuration Space Header
  - Parity Generation and Parity Error Detection
  - I/O Read and I/O Write commands
  - Configuration Read and Write Commands
- Supports Plug and Play Architecture

The main components comprising the FPGA implementation of the hardware are shown in Figure 1 and these are the following:

- Registers 1, 2 and 3 – These three 32-bit I/O registers handle the I/O read and I/O write transactions.
- Configuration Space – This component provides the mandatory configuration registers and the I/O Base Address Registers needed in the implementation of Plug and Play architecture of the PCI card.
- Parity Generation/Checking – This block aids in the error detection and handling of the different transactions occurring in the PCI bus. It utilizes XOR gates to generate and check the parity during the address and data phases [3].

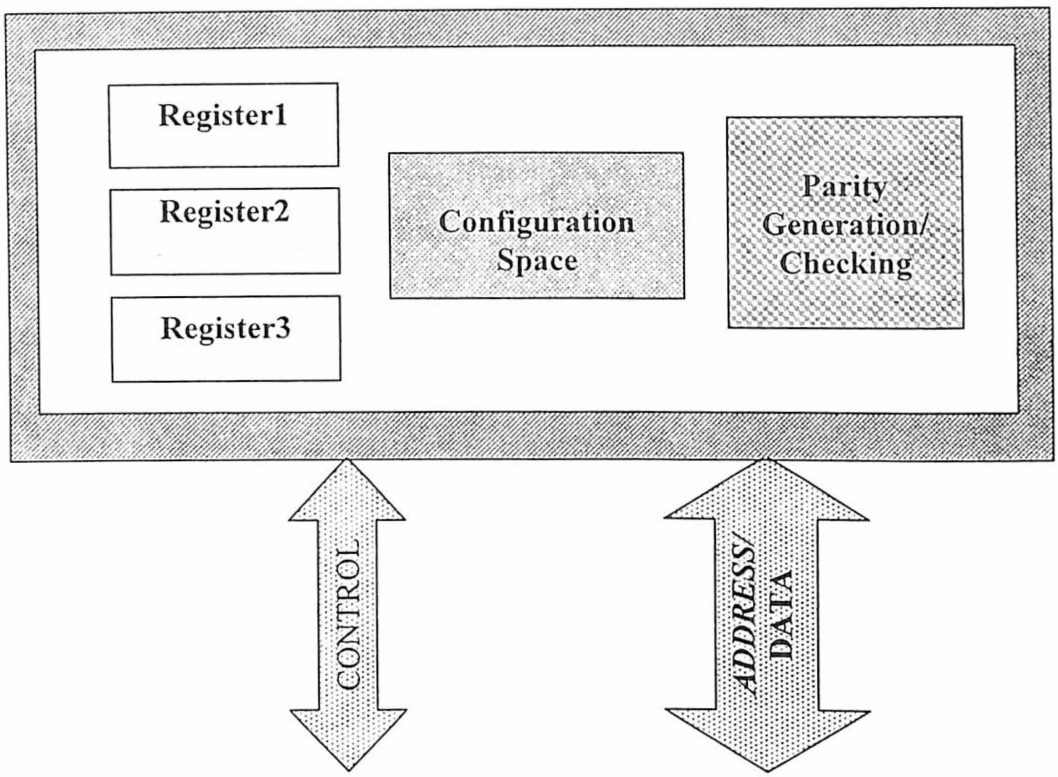

Figure 1. Main Components of FPGA Hardware Implementation

III. Design Methodology and Testing

The project has undergone several stages of design and testing. The first one involves the use of Leapfrog™ VHDL Simulator where the design's functionality is verified with the help of a test bench or stimulus, modeling the PCI bus master or initiator. Different types of transaction such as Configuration Write, Configuration Read, I/O Write and I/O Read were tested. Transactions involving parity errors during the address phase and data phase were also simulated.

To check the design implemented in gates, the Verilog XL Simulations using the Leapfrog™ VHDL Simulator was used. These simulations verify the functional integrity of the design, but this time, gate delays were taken into consideration.

The real target FPGA for the design is the Xilinx XC4013E HQ240, which has 576 Configurable Logic Blocks (CLBs) [4]. For debugging purposes, the Xilinx XC4010E PG191 FPGA was used instead since the optimized design consumes only less than 400 CLBs, which is the maximum capacity of the XC4010E. The design was downloaded to this FPGA and was tested using the Logic Analysis System to check for the actual hardware implementation of the design. The same stimulus was encoded in the pattern generator of the Logic Analysis System. Two additional signals, namely the TST_AD and TST_PAR, were added in order to tristate the AD lines and the PAR line respectively. When TST_AD is set to '1', it means that the AD lines have been tristated. On the other hand, when TST_PAR is set to '1', the PAR line takes the value of 'Z' or high-impedance.

The final test of the PCI Card involves the use of the C codes provided by Ivor Bowden from the PCI Special Interest Group (PCI SIG). Upon the installation of the 5V card in the Pentium motherboard's PCI slots, the configuration registers will be read, specifically the Base Address Register 0, to know the start address of the I/O space allotted by the configuration software for the PCI card. Using those allotted addresses, I/O write transactions and I/O read transactions would then be implemented

IV. Conclusions and Recommendations

The use of Field Programmable Gate Array Technology proved to be an effective tool in implementing the hardware design of the Peripheral Component Interconnect (PCI) Prototyping Platform. The module for the PCI Prototyping Platform uses only 400 out of the 576 Configurable Logic Blocks or CLBs available in the Xilinx XC4013E HQ240, thus enabling the initial tests to be done with the use of Xilinx XC4010E PG191, which has 400 CLBs.

Leapfrog™ Simulations of VHDL and Verilog XL verified that the behavioral description of the PCI Prototyping Platform meets the timing specification for the following PCI commands or transactions: I/O Read, I/O Write, Configuration Read and Configuration Write. This was affirmed by testing the hardware design of the PCI Prototyping Platform using the Logic Analysis System where the hardware delays were taken into consideration.

In light of the researchers' thorough research in Peripheral Component Interconnect, it is recommended that other PCI's capabilities and features be explored and implemented in hardware. This includes burst transfer, data stepping, fast back-to-back transactions, etc. Also, the PCI 64-bit implementation appears to be a very interesting endeavor.

V. References

[1] Solari, Edward and Willse, George. PCI Hardware and Software, architecture and Design, 2nd ed., San Diego, USA: Annabooks, 1995.

[2] Shanley, Tom and Anderson, Don. PCI System Architecture, 3rd ed., Massachusetts: Addison-Wesley Publishing Company, 1995.

[3] LogiCORE PCI Master and Slave Interfaces Version 1.2.0.Xilinx Inc., 1997.

[4] The Programmable Logic Data Book. Xilinx Inc., 1996.