The Realization of PCI Communications Based on the FPGA and PCI9656

LIU Weichen, JI Zhenzhou and TANG Shuofei,

Dept. of Computer Science and Engineering, Harbin Institute of Technology, Harbin, China

Summary
The paper introduced the design of the data channel between the hardware firewall and the host computer based on the advanced Virtex-II-series FPGA from Xilinx, Inc. and the bridge chip-PCI9656 from PLX Technology. The design of FIFO with double channels, double ports and double clocks, three control logics and the queue descriptors makes it a high work frequency and transfer speed, then a good performance in capability and stability.

Key words:
FPGA, PCI9656, FIFO, Descriptor

Introduction
The design of the PCI interface is under the demand of the “Gigabit Hardware Firewall”. The function of the interface is to construct a communication channel between the host computer and the firewall. Accordingly, the host can receive messages from the firewall, send control messages the firewall, expand the functions of the firewall, and improve the flexibility of the firewall. As we know, PCI bus is the most widely used high speed synchronous bus. At present, PCI bus with bus width of 32 bits, clock frequency of 33 MHz and highest transfer speed of 132 Mbps is used in most PCI applications, and it can not satisfy the demand of the progressive data processing [1]. Especially, the demand is more rigorous to transfer data at a high speed between the host and the hardware firewall used in the core of the network. Thereby, it is a good resolvent to develop the new applications based on the PCI bus criterion with bus width of 64 bits, clock frequency of 66 MHz and highest transfer speed of 528 Mbps. PCI9656 is a 64-bit & 66 MHz PCI interface chip presented by PLX Technology, and it has a good performance in flexible connection and I/O acceleration [2]. Using PCI9656 to develop the PCI bus interface can shorten the developing periods, reduce the developing difficulty and get good capability. With the cooperation of the other modules on the firewall, it realizes the whole functions of the firewall to design the interface logic and communication module with PCI9656 on the FPGA.

2. Design Principle
In the interest of expanding the functions of the board adapters, PLX Technology brings out PCI9656 to provide a high-powered object-mode PCI bus interface. The circuit can provide 64-bit mini PCI bus object interface for use of adapters. The target of the paper is to design a channel for data transfer between PCI9656 and the firewall, and it meets the request of speed and stability. Then, as shown in Figure 1, a whole “data channel” between the host computer and the firewall is set up when the driver of PCI9656 is correctly installed at the host, and the appropriative application software is written based on the PLX API DLL.

Based on the advanced Virtex-II series FPGA made by Xilinx Inc., what is to be designed is realized with Verilog HDL. Therefore, first of all, we should know some necessary background information and knowledge of this FPGA and PCI9656.

2.1 Characteristic of Xilinx Virtex-II FPGA
The products of Virtex-II series are high-powered flat-level logic parts which are designed with 0.15-micron and
8-layer coherer technics. Its characteristic is listed below [4]:

a) Clock frequency inside attains to 420 MHz, and in-out frequency to 840 MHz.
b) Interior memories are made up by the high-powered SelectRAM structure, the capacity of every block RAM is 18Kb, and it supports the operation on both ports at the same time. It supports the high-powered exterior memory interface, SDR/DDR SDRAM/SRAM, FCRAM, QDR SRAM and CAM interface included.
c) There are maximum of 93,184 registers/flip-latches drove by clock, and maximum of 93,194 LUTs and shift-registers based on the LUTs. It supports aclinic cascade connection chain, “multiplication accumulating” multiplex selector, and the ample and flexible interior logic resources made up by tristate bus framework.
d) There are 12 DCM modules and 16 clock multiplex buffers at most, and they make up of the abundant and efficient interior clock resource. Accordingly, flexible system clock solutions are provided.

The excellent functions and capabilities of the product and the integrated support software environment provide great advantage for the development.

2.2 PCI9656

PCI 9656 Block Diagram is shown in Figure 2. It adopts the PLX Data Pipe Architecture, has DMA engine, programmable master controller, dependent data transfer mode and PCI info transfer. It can build a PCI interrupt INTA by 2 local bus interrupt LINTi and LINTo. Local clock and PCI clock work asynchronously, and local bus clock is independent of the PCI clock [3]. All these characteristics help to make decision of designing the required module. Choose DMA mode to transfer data, choose interrupt for communications with the host, and design asynchronous high speed local part from the PCI clock.

3. Function Design

As shown in Figure 3, the PCI module is made up of 5 parts: PCI controller, receiver queue, receiver controller, sender queue, sender controller. The two FIFOs save up the data, the PCI controller communicates with PCI9656 to connect to the host part, and the receiver controller and the sender controller communicate with the scheduler of the FPGA to connect to the Local part.

(1) Data management
Because the block mode of PCI9656’s DMA is chosen for data transfer, how to express the length and the available length of the data package in transfer must be considered. If only the original data is packed, the length information can not be acquired. The solution of the paper is to add a piece of 32-bit head information to the original data package for the transfer at one time. The head information can identify the available length of the data package and the available width of the last data in the package.

(2) FIFO design
In consideration of the request of the system, two FIFO channels are created to store the up data and the down data separately for the credible data transfer. Read and write operations on one FIFO can be done at the same time by using the RAM with a pair of ports and it definitely improves the efficiency of the data transfer. For the best utilization of the FIFO resources, the read and write operations are controlled under different clocks. At the
PCI part, the clock of the PCI bus – 66 MHz is used. At the Local part, the uniform clock of the board – 125 MHz is used. Therefore, the data storage has the characteristic of double channels, double ports and double control clocks, and the superiorities of parallelism and asynchronism are utilized here.

The Block SelectRAM Memory RAMB16_S36_S36 which is provided inside the FPGA is adopted as the basis memory of the two FIFOs. As shown in Figure 4, RAMB16_S36_S36 is indeed the two-port RAM. It contained a storage area of 18kb and two unattached visiting ports A and B. The two ports have independent functions and symmetrical structure, and the different visit to the storage can go along at the same time. Data can be written and read from either port. Every port is synchronous, and it owns clock of itself. These characteristics nicely satisfy the above requirement.

Many factors must be considered when choosing the capacities of the two FIFO, such as the attribute of the data for transfer, the attribute of the data flux, the gross of the rams in the FPGA and its allotment. In our case, the length of the data package is changeless, and the flux of the package is stable correspondingly, so the access to the FIFOs is reversely easy to control for the burst condition is infrequent. Because the word length of the data for storage is commonly big, the mode of word expanding is used for the connection of the rams.

The management method of the FIFO data storage is ordinal cycle, and the general information of the FIFOs is in the descriptors.

(3) Descriptor design
The descriptor is designed to describe the state of the FIFO. The two descriptors of the two FIFOs provide the whole useful information for visiting. The descriptor is a 32-bit binary. It contains two pieces of information: the size and the first address of the free space in the queue. Then, via the total size of the queue, we can know the size, and then the first address of the data space. Thereby, all the information of the FIFO is collected.

A descriptor composed of the size and the first address of the free space of the FIFO is chosen to meet the need of the specific visit to the FIFO in the design of the firewall. Other designers can discretionarily choose the combination of the four states of the FIFO to describe it for your personal purpose of efficiency and optimization.

The principle of updating the descriptor is to keep the integrality of the data. The time of updating the descriptor is immediately after the whole package transfer finished, and the next transfer must start after the update is over. During the maintenance of the descriptor, there are some problems in “read-write” conflict and asynchronism. The design must follow the principle of keeping the integrality of the data, the write of the descriptor must execute after the read of the descriptor finished, and the asynchronous conflict must be managed with great caution.

The descriptor is a big bright spot of the design. Two simple 16-bit binaries and two simple shift operations provide all the needed size and address information for the read and write of the FIFO. The existence of the descriptors greatly reduces the complexity of the queue operations, affords many facilities for the design of the controllers, and consequently improves the efficiency.

(4) Receiver/sender controller design
When the receiver/sender controller is in the process of transfer, the part which sends the sending signal must send the length of the waiting data package at the same time. The receiver part judges the time to respond. After the sender part receives the answer signal, starts the transfer and finishes it, the receiver/sender controller should update the descriptor and turn into the state for the next transfer, besides canceling the transfer control signal to inform the accomplishment.

(5) PCI controller design
There are two visit modes from PCI9656 to the PCI controller: visiting the descriptors in the mode of Direct Slave, and visiting the data areas of the FIFOs in the mode of DMA. Since PCI9656 is also the sponsor in the DMA mode, the PCI controller is always working as the secondary device and the control signals used in the two modes are absolutely the same.

Because there is only one device on the local bus, all storages (two FIFOs and descriptors included) can be
addressed uniformly. Thereby, the two different kinds of visit can be differentiated by the address signals, also can the visit to the different FIFOs. Because PCI9656 visits the up queue and the descriptors only for read and the down queue only for write (the descriptors can only be updated by FIFO controllers), the address method makes the address signals take powerful information – they can differentiate between read and write, and they can differentiate the object for read and write.

4. Performance Analysis

Capability and stabilization are the biggest request to the design of the data channel with no interest in the data content. All the designs in the paper are debugged and tested in the Xilinx integrated develop environment – ISE, emulated in Modelsim, synthesized in Synplify, and online debugged in ChipScope. The interface to the OS is the SDK software PLXMon powered by PLX Technology. Snifer is used to capture packages in the network for testing. Particularly, PLXMon can get real-time register information of PCI9656, as well as the important parameters for the DMA transfer. It supplies huge convenience to the “visual” online debug. Being familiar with PLXMon and making good use of it can greatly reduce the debug difficulty.

The analysis is processing in two parts: testing the up queue and the down queue. To test the up queue, data packages are brought by network device, and then they pass the firewall and reach the scheduler. The up FIFO incepts them under the control of the receiver controller, the PCI controller sends them to PCI9656, PCI9656 sends the data packages rightly to the memory buffer under the command of PLXMon, and then the validity of the transfer can be checked. Whereas, to test the down queue, PLXMon first sends data packages from the memory buffer to PCI9656, then they are written into the down FIFO under the control of the PCI controller, the sender controller sends them to the scheduler, they pass the firewall and are captured by Snifer, and the validity is checked at last.

(1) Stabilization analysis
Because the design serves for the hardware firewall, very little data transfer with high data density is occurred in actual applications, and then the stabilization is the most important capability aim. In our design, the synthesizing frequency of the local part is high to 154.9 MHz. It exceeds the actual work clock 125 MHz by a great deal. Moreover, the synthesizing frequency of the PCI part exceeds the actual work clock 66 MHz more and is high to 112.1 MHz. As a result, the whole module works fairly stably. It performs good conditions in the test, too. In the test to the firewall that lasts for seven continuous days, the module runs stably and the ratio that the false code appears when sending and receiving data is zero.

(2) Capability analysis
Under the precondition of the stabilization, the design gives attention to the improving of the capability. For the unconventional FIFO design with double channels, double ports and double clocks and the descriptor system, the throughput rate of the interface is improved greatly. In the
test, the visual speed is high to 287.04 Mbps (restricted by the experiment and the actual speed terminal maybe higher), and it attains a high standard of the capability of the 64-bit, 66 MHz PCI bus.

(3) Other analysis
The condition that the module occupies the percentage of the FPGA resources is as follows:
- Total LUTs: 469 (1%),
- Block Rams: 6 of 96 (6%),
- Register Bits: 442 (1%).

It is obvious that the design consumes only a few resources and takes tiny burden to the system. Due to the apt choice of the number matching of the Block Rams, the data throughput is abundant to a high standard. The design accords tightly with the conception of high capability and low cost.

5. Conclusions
The paper introduces the way of setting up the data channel between the local and host using PCI9656. It provides the communication interface to the host computer, and realizes the basis of the function configuration, extensibility control and information feedback to the user for the hardware firewall. It provides the resolvent of high speed PCI transfer based on PCI9656. Its practical frequency 154.9MHz can absolutely satisfy the request of high speed and stabilization. Since the design principle and realization method of the module, based on the powerful functions and simple user interface of PCI9656, fast local bus interface development with low resource usage and good performance can be widely generalized into the applications based on the high speed PCI bus.

References

LIU Weichen received the B.S. degree in Computer Science and Engineering from Harbin Institute of Technology in 2004. Now, he is a graduate student in Computer Science and Engineering in Harbin Institute of Technology. His major research interest is computer system architecture. He has published four papers in the field.