Development of a PCI-Express Device Verification Model

Youngwoo Kim*, Sungnam Kim**, Kyoung Park*** and Myungjoon Kim****
* Electronics and Telecommunications Research Institute, Daejeon, 305-350, Republic of Korea
Tel : +82-42-860-1619 Fax : +82-42-860-6699 E-mail: bartmann@etri.re.kr
** Electronics and Telecommunications Research Institute, Daejeon, 305-350, Republic of Korea
Tel : +82-42-860-1020 Fax : +82-42-860-6699 E-mail: ksn@etri.re.kr
*** Electronics and Telecommunications Research Institute, Daejeon, 305-350, Republic of Korea
Tel : +82-42-860-3809 Fax : +82-42-860-6699 E-mail: kyoung@etri.re.kr
**** Electronics and Telecommunications Research Institute, Daejeon, 305-350, Republic of Korea
Tel : +82-42-860-6310 Fax : +82-42-860-6699 E-mail: joonkim@etri.re.kr

Abstract: In this paper, a verification method and model for a PCI-Express device are described. PCI-Express technology is one of new I/O interconnection technologies which is intended to replace conventional PCI based technology, and is introduced by PCI-SIG in 2002. For a fast prototyping, a verification suite which includes a behavioral model and stimuli is needed before actual design is finished. And also it should be simple in structure and accurate enough to verify the design. In this paper, an Early Verification Suite (EVS) which complies with PCI-Express protocol is developed and tested.

PCI-Express, Endpoint device, Verilog, Verification model, SystemC

1. INTRODUCTION

The recent development of semiconductor technologies results the birth of high-density and high-speed micro-processors which operate over multi-Giga hertz speed. In these days, these processors are used in various kinds of fields like - consumer, mobile, desktop and high performance computing and etc. And at this moment, the various kinds of user requirements for more and more performance improvement thrust the evolution of semiconductor technology.

Based on the improvement of semiconductor technologies and the evolution of the Internet, a user can enjoy various kinds of multimedia contents. These flooding of multimedia contents – such as streaming video and sound like MP3 – require more efficient data transfer between processor(s) and various kinds of peripherals. And this leads to the birth of a different breed of I/O subsystem mechanism.

The next generation technologies for I/O subsystem can be divided into two categories. The one is a traditional bus based connection scheme like PCI-X, HyperTranport, and RapidIO. And the other is a high speed interconnect scheme like PCI-Express which adopts point-to-point packet transfer mechanism [1]. PCI-Express is a relatively new technology and is now competing with traditional PCI/PCI-X technology.

As the new technologies appearing and life cycle of new products getting shorter, the need for fast prototyping and shortening development cycle are required more than ever. But, usually a new technology is tends to be more complex and needs more exhaustive work and time for verification and test.

To shorten the development cycle, the design and verification of the product is strongly required before actually launching the hardware design. By developing verification model prior to design, a designer can shorten the design time and also find design bugs before the design is finished.

In general, verification models are developed at the higher level of abstraction by using high level language like C and SystemC. A verification model using high level language, generally, has characteristics of high speed execution, high readability and easy construction. As a result, it is very useful at the very early stage of development like product concept establishing and/or functional specification verification stage. But, high level language model, inherently, lacks timing information, so at the middle of design or prototype verification stage it is required to make another verification model which can verify timing relationship between design entities.

To develop a PCI-Express Endpoint device, we designed and test a PCI-Express behavioral model for verifying functions of the device and behavior of each modules of the device. We use Verilog HDL to develop early stage verification model instead of high level language, because it is a well known design language, can be easily integrated in the actual design code, and is easy to verify timing of the design.

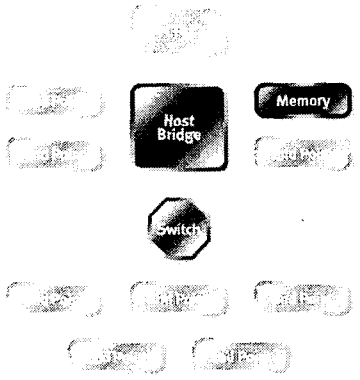
In this paper, we describe the design of verification model and its architecture. In section 2, a brief introduction for PCI-Express is presented. In section 3, the architecture and operation of the verification model is explained and the experimental results is described. In section 4, finally, the conclusion is presented.

2. PCI-EXPRESS TECHNOLOGY

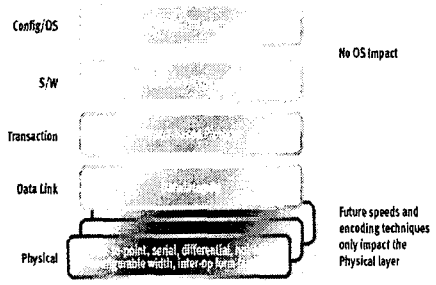
As mentioned before, PCI-Express technology adopts a point-to-point packet switching technology using serial interconnect and PCI-SIG announced PCI-Express Rev. 1.0a in 2003. PCI-Express has software compatibility to conventional PCI software model and has switched fabric topology to overcome the speed bottleneck of conventional bus based I/O subsystem.

Major features of PCI-Express like scalability through point-to-point packet switching, layered architecture, integrity mechanism for each layer, QoS mechanism via Virtual Channel, and etc. are differentiate PCI-Express from conventional PCI or PCI-X based system.

PCI-Express adopts lane based serial link connection mechanism and each link consists of multiple lanes. Each lane communicates between components at a rate of 2.5Gbps. PCI-Express link can be expanded up to 32 lanes (32x) with speed of 80Gbps. The layered architecture of PCI-Express minimizes data rate changes and functionality changes of layers as depicted in figure 1 [2]~[4].



(a) PCI Express Topology



(b) Layered architecture of PCI Express

Fig. 1. PCI Express Technology [3].

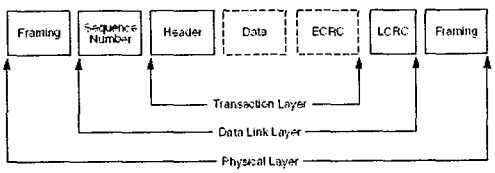


Fig. 2. PCI Express Packet.

PCI-Express had three layers – Transaction, Data Link, and Physical layer. Data flows down through each layer and encapsulated according to the layer (see figure 2).

The Transaction layer provides upper layer interface, performs data segmentation and reassembly, and conforms Transaction Layer Packet (TLP) by adding transaction header. To ensure link level data integrity, the Data Link layer generates header and CRC and also processes Data Link Layer Packet (DLLP) which is used to manage the link initialization, flow control, etc. Finally, the Physical layer performs parallel-serial conversion via SerDes, and provides physical medium interface and control.

As depicted in figure 1. (a) PCI-Express topology consists of three major components – Root Complex (RC), Switch, and Endpoint device. Root Complex is a device to initiate and terminate all PCI-Express communication and it has responsibilities to configure the entire PCI-Express fabric and manages the topology. The Root Complex plays similar roles of a North Bridge as in conventional PCI based computer systems. Endpoint is a device which connects a peripheral device to PCI-Express fabric and terminates PCI-Express packets. Switch devices connect between Root Complex and Endpoint device, form PCI-Express fabric and bypass transactions between RC and Endpoint devices. Using these components, PCI-Express constructs switched fabric via Link. After Link establishing procedure – Link between two components is established automatically – the Root Complex configures overall topology of the conformed fabric [5].

3. PCI-EXPRESS VERIFICATION SUITE

In this paper, a verification model for a PCI-Express Endpoint device is developed and verified. In contrast with the conventional bus based PCI system, a PCI-Express based device communicates each other with packets to transfer data and needs more complex model compared to bus functional model (BFM) in the conventional PCI based system for verifying an endpoint device. The following sections describe the developed PCI-Express verification model and its operations.

3.1. PCI-Express verification model

As described before, PCI-Express endpoint device consumes and responds most of request packets from Root Complex device. Thus the verification model for a PCI-Express Endpoint device should implement Root Complex functionalities – topology management, link control, physical port management, etc. and especially generation of configuration packets.

The developed verification model, named Early Verification Suite (EVS), has functionalities of three layers according to the PCI-Express architecture – downstream port model including Transaction Layer, Data Link Layer and Physical Layer, and an additional layer, host model, for performing host system functionalities and some Root Complex functionalities. The figure 3 depicts the overall structure of the EVS.

The downstream port model is to perform actual PCI-Express packet generation and reception. The

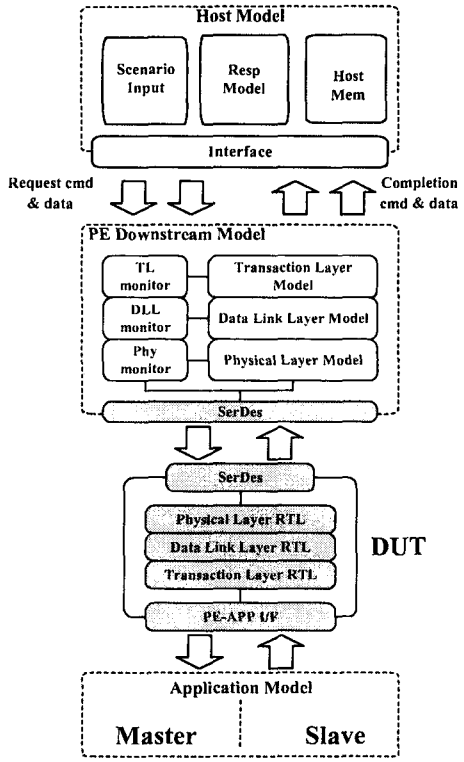


Fig. 3. Conceptual diagram of verification model.

Transaction Layer generates and consumes PCI-Express defined transactions – configuration, memory read and write, I/O read and write and additional transactions like various kinds of messages. The Data Link Layer is to guarantee the link integrity while sending and receiving packets. To do this functionality, the DLLPs are defined and implemented. The role of the Physical Layer is to establish physical link and train it between the model and the DUT.

3.2. Implementation of EVS

The downstream port model and host model are implemented in the forms of Verilog tasks and some control state machines. In each model, state machines control the flow of the transactions according to the verification stimuli and check the verification results.

The host model generates and verifies various kinds of verification stimuli by controlling the downstream port model. It consists of host system memory model, configuration space model, and stimuli for verifying the system. Because verification procedures are involved in generating and receiving various kinds of transactions, the host model and the Transaction layer are tightly coupled in the EVS. Each send and receive transaction in model communicates each other to verify the correctness of the request and response relations between them. The figure 4 depicts more detailed block diagram of the model.

In the figure 4, the Physical layer for the model is omitted because the functionalities of the Physical layer for downstream port are more complex and requires more precise operation than those of the other layers. So the Physical layer is implemented in Verilog RTL for more accurate operations.

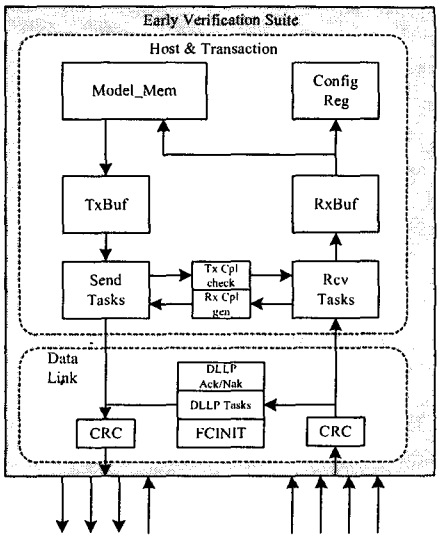


Fig. 4. Detailed block diagram of host and downstream port model.

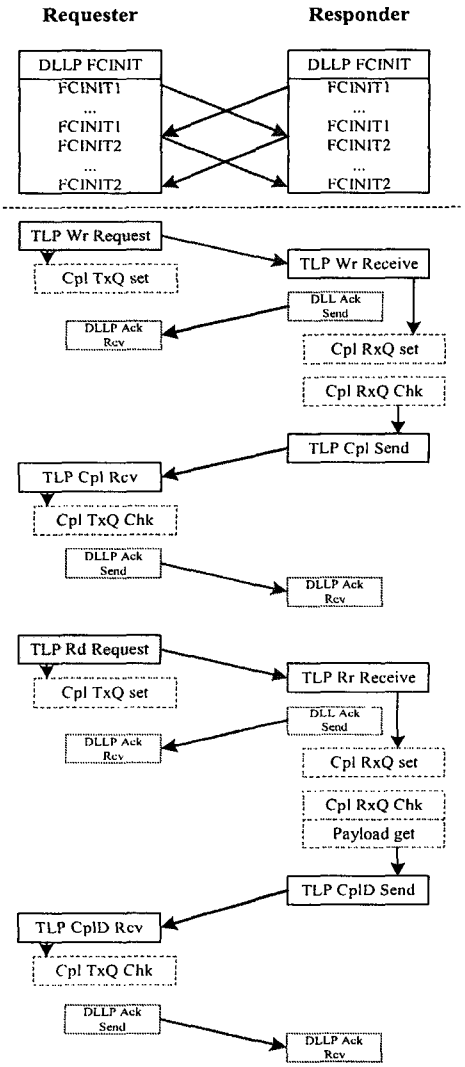


Fig. 5. Initialization and Transaction processing

3.3. Operation of EVS

The verification process is done according the following procedures:

1) Physical Link setup (this is done by automatically by the Physical layer)
2) Data Link layer flow control setup
3) PCI-Express Configuration space setup and verification
4) PCI-Express transaction verification

The figure 5 illustrates the initialization and transaction processing flow between requester and responder – in this model requester is the verification model and responder is DUT. In each procedure, the EVS generates stimuli, checks protocols, and reports the results for the steps according to the PCI-Express specifications. After the Physical layer setting up the link, Data Link layer performs flow control initialization. In EVS, Data Link layer model generates FCINIT1 and FCINIT2 DLLP sequence. During flow control initialization, EVS and DUT exchange DLLPs according to the defined state, then finally wake up the Data Link layer. Host model detects the Data Link layer state and starts Endpoint configuration. During configuration phase each configuration write and read request requires appropriate completion. Host model sends the requests and checks the responses according to the requested transaction.

```
Send_TLP_CfrWr (CfgType, TD, EP, FstBE, BDF, RegNo, ExtRegNo, CfgData)
Send_TLP_CfgRd (CfgType, TD, ED, FstBE, BDF, RegNo, ExtRegNo)
Send_TLP_MWr (Addr, AdFmt, TC, TD, EP, Attr, Length, FstBE, LstBE, S_Addr)
Send_TLP_MRd (Addr, AdFmt, Lock, TC, TD, EP, Attr, Length, FstBE, LstBE)
Send_TLP_IOWr (Addr, TD, EP, FstBE, IOData)
Send_TLP_IORd (Addr, TD, EP, FstBE)
```

Fig. 6. Some Verilog tasks.

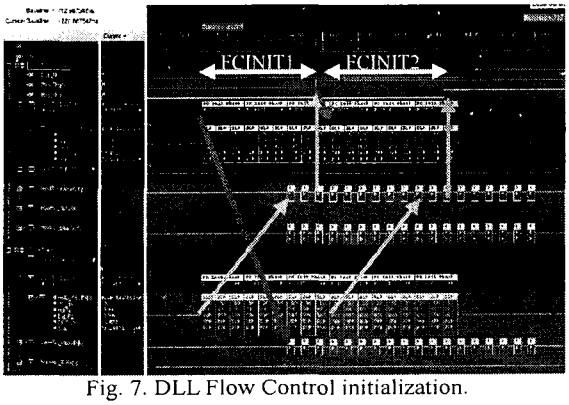


Fig. 7. DLL Flow Control initialization.

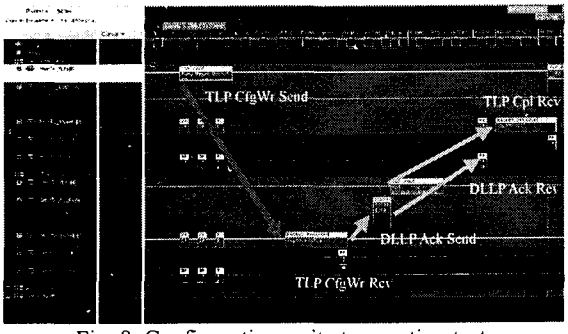


Fig. 8. Configuration write transaction test.

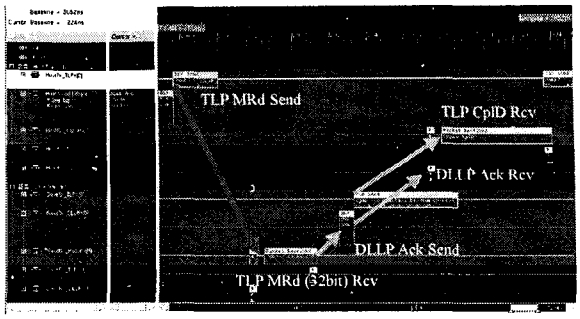


Fig. 9. Memory read transaction test

Because these procedures are also designed using Verilog tasks, a user can easily modify the test stimuli by calling the tasks. The figure 6 shows some Verilog tasks which generates and checks transaction.

The figure 7 through 9 shows the verification process simulation results. In figure 7, the Flow Control initialization process is simulated and verified. The figure 8 and 9 shows the process of verifying configuration write transaction and memory read transactions.

4. CONCLUSION

In this paper, an Early Verification Suite which complies with PCI-Express protocol is developed and tested. The proposed verification model enables fast prototyping by using well known Verilog HDL language and simple layered model. The core EVS is designed using Verilog HDL and offers a basic framework for PCI-Express transaction verification and compliance check. The designed EVS covers various kinds of PCI-Express compliance test features such as initialization check, transaction information check, protocol sequence check, configuration check, etc. The layered architecture of the EVS enables easy expansion to higher level verification and modeling language adaptation such as SystemC.

By using the EVS, we verified a PCI-Express Endpoint device design at the early stage of the design process. And also we developed more accurate and complex SystemC model. The EVS model is also used in the SystemC model in the form of transaction monitor.

The designed PCI-Express Endpoint device is now under test.

References

[1] Linda Dailey Paulson, "The Ins and Outs of New Local I/O Trends," IEEE Computer, vol. 36, issue 7, July, 2003, pp.13~16.

[2] @IT特集, PCI-X 2.0 と PCI-Express のインパクト 新しい・張インタ・フェイスの・体とその影響を探る; http://www.atmarkit.co.jp

[3] Intel whitepaper, "Advanced Switching for the PCI-Express Architecture," http://www.intel.com/design/network/papers/251737.htm

[4] Ajay Bhatt, "PCI-Express Architectural Overview," PCI-SIG Developer's Conference, 2002

[5] PCI-Express Base Specification 1.0a, PCI-SIG