Circuit Design of PCI Express Retry Mechanisms

BING LI
School of Integrated Circuits
Southeast University
Sipailou No.2, Nanjing, Jiangsu Province, China
CHINA

YAN DING
School of Integrated Circuits
Southeast University
Sipailou No.2, Nanjing, Jiangsu Province, China
CHINA

YONG LIU
School of Integrated Circuits
Southeast University
Sipailou No.2, Nanjing, Jiangsu Province, China
CHINA

tianxindingyan@163.com

Abstract: - TLP retry mechanism is an effective measure of PCI Express to ensure data reliable transmissions. The paper theoretically analyzes reasons for retransmission, operating rules of retransmission and factors affecting the size of retry buffer. On this basis, proposes a circuit structure of implementing TLP retry mechanisms, describing the implementation process of retry management, and conducts functional verification based on VMM verification platform, using VCS simulation tool. Verification results show that this method not only implements the function of the retry mechanisms, but also guarantee data transmission normally and orderly. What's more, it would save the storage space of retry buffer and improve the efficiency of the link transmission.

Key-Words: - PCI Express, TLP retry mechanism, Retry buffer, VMM.

1 Introduction

PCI Express is a high-performance internet communications protocol [1], with point to point serial connection between devices. The data transfer rate of PCI Express has significantly improved, reaching far beyond the transfer rate of the PCI bus. With the improvement of transfer speed, the reliability of data transmission has to satisfy higher requirements.

The structure of PCI Express core consists of three layers, respectively Transaction Layer, Data Link Layer and Physical Layer, shown in Fig.1 [2]. Each of these layers is divided into a transmitting component and a receiving component [3]. TLP retry mechanism is located at Data Link Layer which is responsible for reliably conveying Transaction Layer Packets [4].

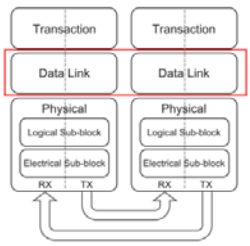


Fig.1 hierarchy structure of PCI Express

As an important technology of PCI Express, TLP retry mechanism is responsible for reliable data transmissions. When data transmission errors occur, retry mechanisms [5] ensures that data with transmission errors could be sent correctly finally. Therefore, the function of retry mechanisms is defined as guaranteeing data transmissions orderly and correctly. The implementation quality of the retry mechanisms has very important impact on the function and performance of PCI Express. In this

paper, an optimized program to implement retry mechanisms has been proposed, not only to ensure the normal and correct transmission of data, but also improve the performance of PCI Express.

2 Retry Mechanisms

2.1 Causes of TLP retransmission

When errors occur or data lost during the process of TLP transmission, it is time to start retry mechanisms and retransmit related TLP data. Direct causes of data retransmission are divided into the two following aspects.

1) Receive NAK (negative acknowledgment) DLLP (Not Acknowledged DLLP) from remote device. NAK DLLP is Data Link Layer packet in response to TLP transmission not correctly.

2) REPLAY TIMER expired. REPLAY TIMER is used to count time started at the last symbol of any TLP transmission or retransmission until corresponding ACK (acknowledgment)/NAK DLLP received. When sending the last symbol of any TLP, REPLAY TIMER starts to count, and reset to 0 when the corresponding ACK/NAK DLLP received. If no response is received from the remote device during set time, there may be a problem with the link, and some TLP data needs to retransmission.

2.2 Retransmission Operations

When receiving NAK DLLP from the remote device, the transmitter blocks new TLP packets from Transaction Layer and stops all of data transmissions and enters retry mode. According to the sequence number field of the received NAK DLLP, we can confirm the sequence number of the first TLP which needs retransmission. The sequence number is defined SEQ. Thus, the copies of TLP in retry buffer of which sequence numbers are before SEQ including the one that sequence number equal to SEQ could be removed and retransmission TLP packets which sequence numbers are after SEQ. Before the receipt of ACK DLLP, if the number of retry is more than three times, the Physical Layer will be notified to link retrain. At the end of the link-retraining, try again

2.3 Retry Buffer

Before sending a TLP, a copy of the TLP needs to be stored to prepare for retransmission. The buffer used to store copies of TLP called retry buffer. Copies of the TLP which have been acknowledged in retry buffer will be removed and storage space released timely; Copies of the TLP which have not been acknowledged will not be removed in retry buffer until they are recognized to transmit correctly.

The size of retry buffer is configurable according to TLP maximum payload size. The retry buffer is required to store all fields of each TLP. Theoretically, it should be large enough to ensure that TLP transmission will not be blocked because retry buffer is full and lead to reducing transmission speed. The following factors should be considered to determine the size of retry buffer.

1) The delay caused by ACK/NAK DLLP transmission from receiver to transmitter.

2) The delay caused by the physical link connection and physical implement.

3) L0s to L0 exit latency of receiving device.

3 Circuit Design of Retry mechanisms

The circuit design of retry mechanisms includes retry management module and TLP package module, shown in Fig.2.

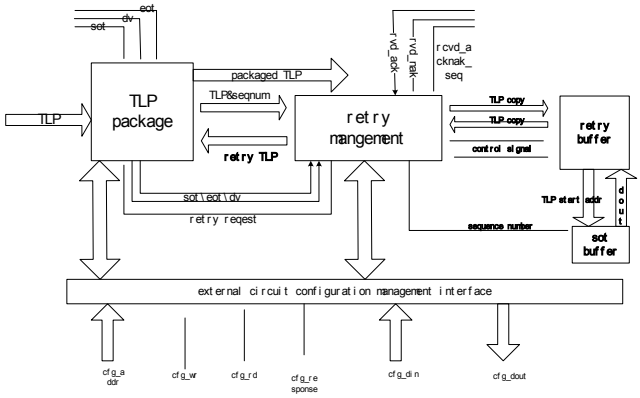


Fig.2 block diagram of retry mechanisms

ACK/NAK protocol [6] plays an important role in the interaction of the two modules. Retry management module is mainly responsible for controls for retransmission process and management of the retry buffer. TLP package module is mainly responsible for package handle of TLP packets, and sent the packaged data to the Physical Layer. In this paper, TLP package module distributes sequence number for every TLP unsent, and store TLP copies to retry buffer. When retransmission, read out packets and complete the packaged process in TLP package module. Compared to the case that copies of packaged TLP are stored into retry buffer after added sequence number and LCRC, the proposed scheme lowers request of retry buffer size and contributes to storage space seducement.

3.1 Specific design for TLP package module

TLP package module is mainly responsible for package handle of packets from the Transaction Layer, and sent packaged packets to the Physical Layer. The block diagram is as shown in Fig.3.

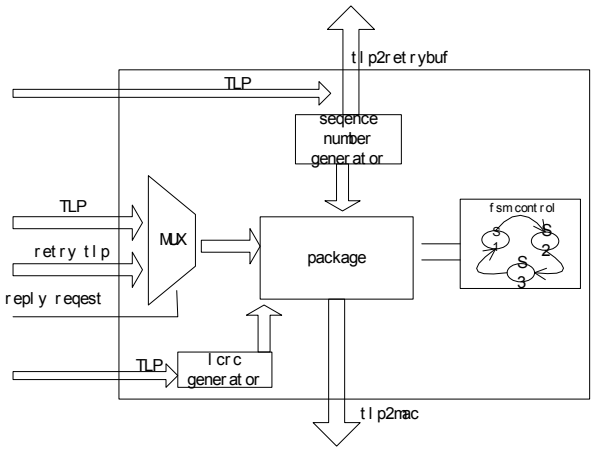


Fig.3 block diagram of the TLP package module

TLP package module is mainly responsible for package handle of packets from the Transaction Layer, and sent packaged packets to the Physical Layer. The module includes sequence number generator, LCRC generator, package arbiter, package control state machine and package process. The sequence number generator is mainly responsible for distributing 12 bits sequence numbers for every TLP unsent. The sequence number successively increases and re-starts counting from 0 after reaching 4095. LCRC generator is used to generate 32 bits LCRC. Based on the content of TLP and sequence number, using CRC algorithm, LCRC generator calculates 32 bits LCRC, and adds to the end of TLP. Package mode arbiter selects and determines the package mode according to reply request. There are two kinds of package mode including normal mode and reply mode. In normal mode, TLP package module packages TLP data from the Transaction Layer; In reply mode, the module packages TLP data from retry buffer. If the reply request is valid, the module selects reply mode. Package control state machine controls the rules of data package in order to avoid package process in disorder. The state machine includes two parts, one is to control the package process of new TLP from the Transaction Layer; the other is to control the package process of replay TLP data. The two parts switches to each other according to package states. Package process is responsible for adding sequence number and LCRC to the start field of and the end field of TLP packets, and sending all of packaged packets to next module.

When the reset signal is asserted, the package control state machine enters the IDLE state. Next clock cycle, if the reply request is valid, the state machine enters reply mode and starts to package replay data and export output results next clock. The reply mode consists of three states, namely reply start state, in reply state and CRC adding state. The CRC adding state is divided into three sub-states according to the number of effectively double words. The package process starts packing operations according to the length of effectively double words. In the reply start state, add header information to TLP packets including header identifier, packet length and sequence number. In addition, for Gen3, it is also to add packet length checksum to TLP header information. Then, the state machine jumps to in reply state and sequentially adds all of data fields. Lastly, when a flag marked packet end is detected, state machine jumps to the corresponding CRC adding state and add CRC checksum.

After the end of packages, packets are sent to next module, and packages control state machine is reset to IDLE state, waiting for the next start of package.

3.2 Design for retry management module

Design diagram of retry management module is as shown in Figure 3. Retry management includes control logic to retry buffer and physical implementation of buffers. The implementation of retry mechanisms requires two buffers; respectively retry buffer and sot buffer. All copies of unacknowledged TLP are stored in retry buffer, and sot buffer is used to store the first addresses of TLP packets stored in retry buffer. The write address of sot buffer updates according to the sequence number of TLP packets. Two buffers are all implemented with single port RAM. For the purpose of the actual situation and save resources, we design the size of retry buffer is capable of

storing two largest maximum TLP, which is set 256 bytes in the design.

The specific implement process of retry management refers to Fig.4.

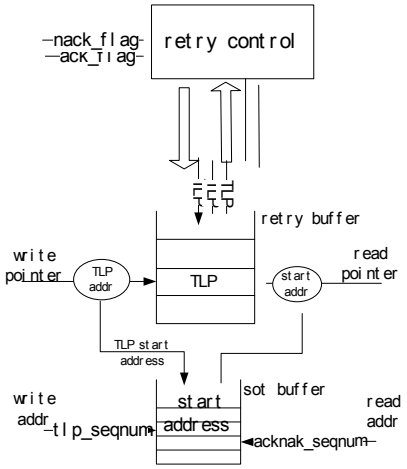


Fig.4 the design diagram of retry management module

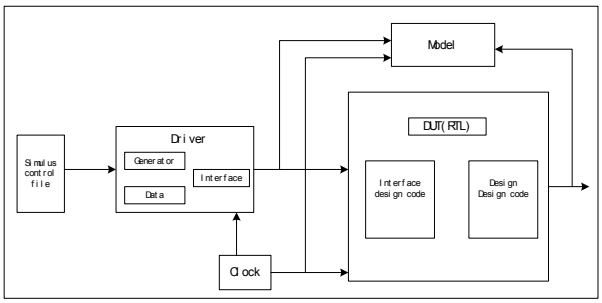
The control logic of retransmission is responsible for updating read and write addresses of retry buffer and sot buffer; According to the response from remote device is ACK DLLP or NAK DLLP, control logic determine whether to launch reply request; Be responsible for removing copies of TLP has acknowledged in the retry buffer, and updating space available of retry buffer; Control the steps of retransmission, and generate reply complete signal when replay has done.

When receiving NAK DLLP, the retry management module launches replay request and sends to the TLP package module. The TLP package module enters retry model and notifies the retry management module that is ready after ending the current task, when receiving the retry request. After above of all, the retry management module starts to control the retransmission process. First of all, we can gain the sequence number of the TLP to be retransmission through the sequence number field of NAK DLLP. Using the sequence number to addressing sot buffer, the start address of TLP will be access. Then, though the start address, addressing the retry buffer, it is easy to gain the address of TLP, and the corresponding TLP will be read out, starting the process of retransmission. During data retransmission, new TLP packets are blocked to store into retry buffer.

In the process of implementation, the operation of removing TLP acknowledged is designed like that new TLP packets updating copies of TLP have been acknowledged, which save additional operations to remove acknowledged TLP packets.

4 Verification and Simulation

The verification of the design in this paper is conducted with VMM verification platform. VMM verification platform designed consist of DUT (Design Under Test), Driver, Model and Clock, as shown in Fig.5.



.

Fig.5 VMM verification platforms

Stimulus Control File is responsible for having Driver enabled and controlled of random data. Driver is the source of all signals in verification, driving different excitation to DUT and Model through interface. Clock offers clock excitation for Driver, Model and DUT, and synchronous clock is used in the paper. Model is the reference model written with SystemVerilog language according to design specifications. Assertion and monitor are used in Model to check the correctness of DUT behaviors. Outputs of DUT is also driven to Model and checked in Model and the simulation will be stopped if the check result is wrong. DUT is the design codes written with Verilog hardware description language. The verification result of retry mechanisms is as shown in Fig.5.

Fig.6 is the work timing of retry mechanisms interfaces and internal circuit nodes, including operations①-⑧. The cooperation of TLP package module and retry management module, contributes to completion of retry mechanisms. Operation① represents receiving NAK DLLP. Operation ② represents retry request generating because of operation①. Operation⑤ represents TLP package module receiving the retry request. Operation ⑥ represents that package control state machine grants to retransmission packets. Operation③indicates the receive of grant. Operation ④ indicates that retry management module starts reading out data from retry buffer. Operation⑦represents TLP from retry buffer has been packaged in TLP package module.

Finally, packaged data would be transmitted after all the above operations.

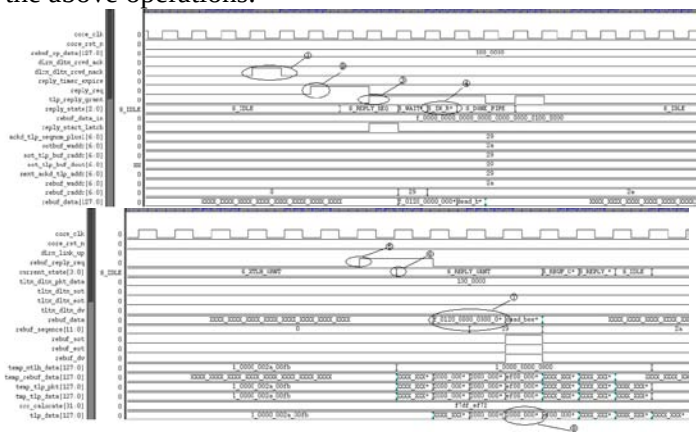


Fig.6 simulation results of retry mechanisms

The above series of operations are namely the data transfer process of TLP retry mechanisms.

5 Performance Analysis

After extensively search for related papers and network resources of PCI Express for a long time, we found that there is no paper that has proposed the implement method of TLP retry mechanisms.

Only few papers mention the design of retry buffer and almost could be summed up in the same method that store copies of packaged TLP into retey buffer after adding the 1B packet header, 2B sequence number (including 4 bits reserved field), 4B crc and 1B end of Packet fields, typical such as paper of 'Proposal of Implementation of the "Data Link Layer" of PCI-Express'.

Assume that the min size of retry buffer in the method above is also two TLP with maximum packet length. Theoretically, taking 128 bytes of TLP max payload for example, the minimum size of retry buffer would be 272B in PCI Express1.0/2.0 protocol and 273B in PCI Express3.0 protocol which is calculated through the summation of all fields of packaged TLP multiplied by 2 in the method above, while it only requires 256B in PCI Express1.0/2.0/3.0 protocol which calculated through 128 multiplied by 2 in the method proposed in the paper. In PCI Express1.0/2.0 protocol, the storage savings rate of retry buffer is 6%, and the storage savings rate of retry buffer is 6.6% in PCI Express3.0 protocol, shown in Fig.7.

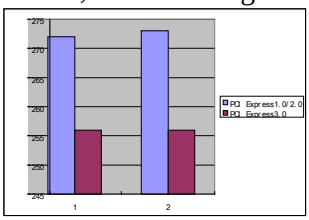


Fig.7 the comparison chart of performance of two methods

6 Conclusion

All in all, the proposed implement method in the paper has two advantages. Firstly, only storing TLP content into retry buffer, instead of storing packaged TLP including sequence number field and LCRC field, reduces requirements for the retry buffer size, and saves storage space. Secondly, the behavior of updating copies of TLP acknowledged using new TLP packets is convenient and avoids additional operations. Everything has two sides. The proposed circuit structure of implementing retry mechanisms also has some limitations. Such as, each TLP packet would be packaged twice. Each TLP has been packaged when it was transported at the first time and was packaged again if it needs retransmission.

Given the importance of retry mechanisms, the implementation method of retry mechanisms is particularly important. According to the simulation and experiment, it has proved that the program could implement all the functions of retry mechanisms, and the performance optimized. Though the specific elaboration, the advantages of the proposed method are so obvious that it is worthy of adoption and promotion. The implement method of retry mechanisms in the paper would be recommended in the design of PCI Express controller and Cloud Network.

References:

[1] [1] Peter B¨ohm, Incremental and Verified Modeling of the PCI Express Protocol, *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, 2010, 1495-1508.

[2] PCI Express Base Specification Revision 3.0, 2008.

[3] Eugin Hyun, Kwang-Su Seong, Design and Verification for PCI Express Controller, *Third International Conference on Information Technology and Applications*, 2005, 581–586.

[4] M. Aguilar, A. Veloz, and M. Gumin, Proposal of Implementation of the "Data Link Layer" of PCI-Express, *2004 1st International Conference on Electrical and Electronics Engineering*, 2004, 64-69.

[5] Xu Jiamou, Li Xuwen, Jia Kebin, The Research and Implementation of Interfacing Based on PCI Express. *The Ninth International Conference on Electronic Measurement & Instruments*, 2009, 116-121.

[6] Chungwon Park, Hee Yong Youn, Youngmin Kwon, Efficient Buffer Management for Retry Mechanism in InfiniBand. *IEEE Asia Pacific Conference on Circuits and Systems*, 2006, 302 – 304.