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. 3, Issue 4, April 2014

Calculation of ECRC FOR TLP-Packets of PCIe Protocol

Shoeb Mohammed Balabatti[1], Radha R C[2]

M.Tech Student, Department of Electronics & Communication, BMSCE, Bangalore, Karnataka, India [1]

Assistant Professor, Department of Electronics & Communication, BMSCE, Bangalore, Karnataka, India [2]

ABSTRACT: Platform-based post-silicon validation includes platform, memory & other peripherals. The data which is being sent from the peripherals like PCIe to the platform need to be checked for the errors. The errors are generated in the data because of signal integration. The basic goal of this project is to support development of software that can be used for automatically generating ECRC error bits for the data which is being sent from Graphics/Video Cards to CPU through PCIE, this software is used for data integrity. This is the research study project showing how the data can be protected.

KEYWORDS: TLP (Transaction Layer Packet), ECRC(End to End Cyclic redundancy check), PCIe(Peripheral Component Interface Express) , LA(Logic Analyzer).

I.INTRODUCTION

There are many components externally connected to the CPU in the present day systems, some of the components to be noticed are video/graphics card connected to the PCIe protocol of the CPU. There occurs transfer of data between the CPU and the card. Due to the high complexity of the system some errors can be incurred during data transfer. In order to provide the End to End Data Integrity, to avoid data corruption ECRC bits are used with the data. This allows debug-engineers to focus on data analysis and debug issues, thus increasing the overall productivity of post-silicon validation process[1][2].

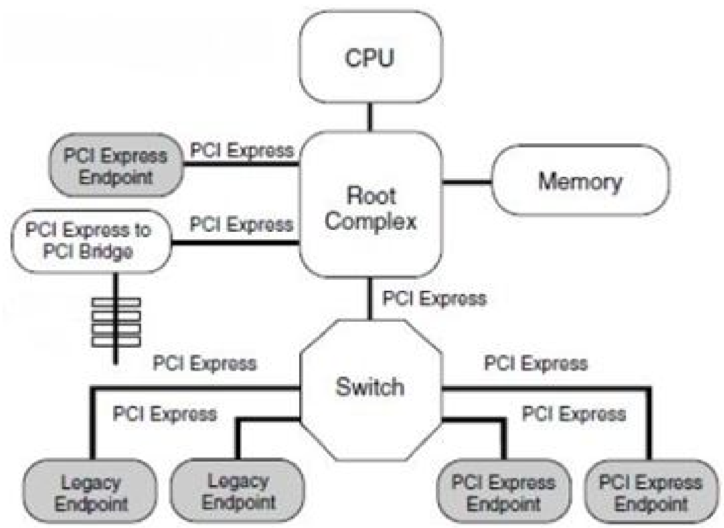

Fig 1.1 Block Diagram of CPU

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. 3, Issue 4, April 2014

II.LITERATURE SURVEY

The paper "Post-Silicon Validation Opportunities, Challenges and Recent Advances" elaborates on the opportunities, challenges and the recent advances in post-silicon validation and also an overview of comparative study between pre-silicon, post-silicon validation and manufacturing testing. Post-silicon validation is used to detect and fix bugs in integrated circuits and systems after manufacture. Due to sheer design complexity, it is nearly impossible to detect and fix all bugs before manufacture. Post-silicon validation is a major challenge for future systems. Today, it is largely viewed as an art with very few systematic solutions. As a result, post-silicon validation is an emerging research topic with several exciting opportunities for major innovations in electronic design automation. Post-silicon validation has significant overlap with pre-silicon design verification and manufacturing (or production) testing [1].
The paper "A Multiple Bits Error Correction Method Based on Cyclic Redundancy Check Codes" and Wikipedia of CRC explains in detail the principle of CRC and error correction method, which became useful during the project[2][5].
The paper "Design and verification for PCI Express controller" gives in detail about PCIe architecture, flow of packets through different layers and PCIe topology[3].
The details of the PCI Express and packets which are used in PCIe at the different levels are understood from PCI-SIG and Wikipedia of PCIe[4][5][7].

III. DETAILS OF PCI PROTOCOL

PCIe architecture is a high performance interconnects for peripherals in computing/communication platforms. This is evolved from PCI and PCI-X™ architectures. PCI Express is a serial point-to-point interconnect between two devices. Implements packet based protocol for information transfer. Scalable performance based on number of signal [3][4][5][7].

PCIe is three layer protocol :

- Each layer split into TX and RX parts.
- Ensures reliable data transmission between devices.

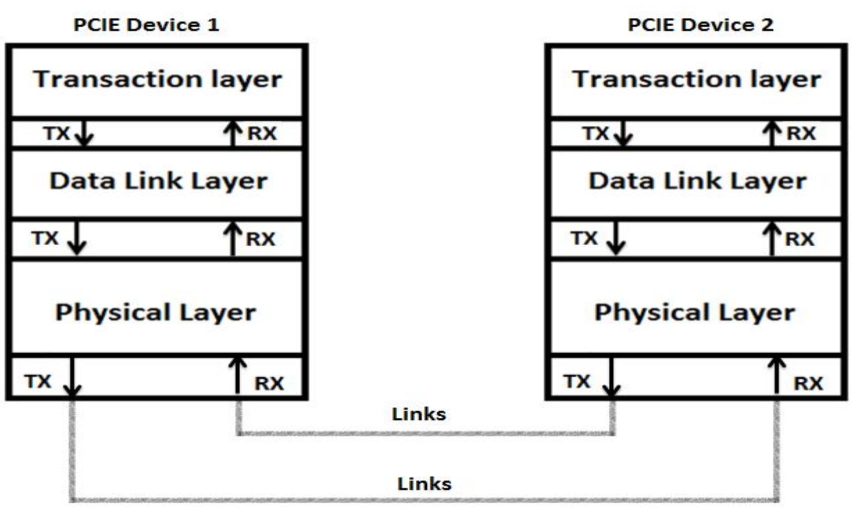

Fig 1.2: Different layers of PCIe Protocol

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. 3, Issue 4, April 2014

PCIe employs packets to accomplish data transfer between devices. A root complex can communicate with endpoint and vice versa. An endpoint can communicate with another endpoint. Communication involves transmission and reception of packets called Transaction Layer packets (TLPs) through lanes.

PCIe transactions can be classified as below

1. Memory Read or Memory Write: These packets are used for data transfer between memory and CPU.
2. I/O Transactions: These packets are used for data transfer between I/O and CPU. These transactions are restricted to supporting endpoint devices.
3. Configuration Read or Configuration Write transactions: These transactions are used for program features, and check status in the PCI Express configuration space.
4. Message Transactions: These transactions are meant for event signaling and general purpose messaging

Transactions are defined as series of one or more packet transmissions required to complete an information transfer between requester and completer. These transactions can be categorized into Posted and Non-Posted transactions [4][5][7].

Table 1.3: Different types of PCIe Transactions

Transaction Type	Non Posted or Posted
Memory Read	Non Posted
Memory Write	Posted
Memory Read Lock	Non Posted
IO Read	Non Posted
IO Write	Non Posted
Configuration Read	Non Posted
Configuration Write	Non Posted
Message	Posted

In the table 1.3, it shows different types of PCIe transactions of TLP

For Non-Posted Transactions, a requester transmits a TLP packet to a completer. At a later time, the completer returns a TLP completion packet back to the requester. The purpose of completion TLP is to confirm to the requester that the completer has received the request TLP [4][7].

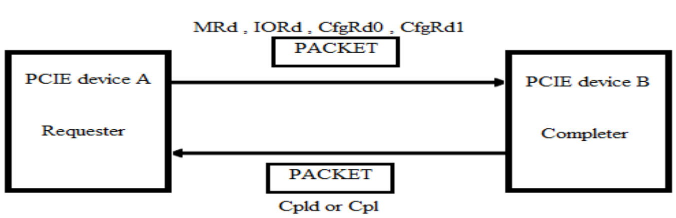

Fig 1.4: Non Posted Transactions

MRd = Memory Read Request

IORd = IO Read Request

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. 3, Issue 4, April 2014

CfgRd0 = Type 0 configuration Read Request

CfgRd1 = Type 1 configuration Read Request

CplD = Completion with Data for normal operation

Cpl = Completion without Data for normal operation

For Posted Transactions, a requester transmits a TLP packet to a completer. The completer doesn’t return a completion TLP packet back to the requester. Posted transactions are optimized for best performance in completing the transaction at the expense of the requester not having knowledge of successful reception of the request by the completer[4][7].

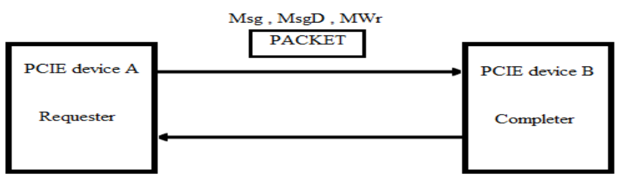

Fig 1.5: Posted Transactions

MWr = Memory Write Request

Msg = Message transaction

A. Building Transactions

Transmitter Side:

Fig 1.6: Packet Formation at Transmitter Side

- At the Transaction layer of PCIe protocol a Transaction layer packet is formed which contains (**Header** and **Data**) along with this packet **ECRC** field is appended for the purpose of Error checking.
- At the Data Link Layer of PCIe protocol **Sequence number** field and **LCRC** field is appended to the packet coming from Transaction Layer.
- At the Physical Layer of PCIe protocol **Start** and **End** symbols are appended to the packet coming from the Data Link Layer.

These packets are transmitted to the Receiver on the PCIe Link [3][5][7].

Receiver Side:

Fig 1.7: Packet Formation at Receiver Side

- At the Physical Layer of PCIe protocol **Start** and **End** symbols are stripped from the packet coming from the Transmitter.

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. 3, Issue 4, April 2014

- At the Data Link Layer of PCIe protocol **Sequence number** field and **LCRC** field is striped from the packet coming from Physical Layer.
- At the Transaction layer of PCIe protocol a Transaction layer packet is taken out which contains (**Header** and **Data**) and transmitted to the particular destination specified in the header field. The **ECRC** field is striped for the purpose of Error checking [3][5][7].

B. Details of TLP Packet

The TLP packets contain the Header and Data Field.

- Data Field contains the data to be transmitted.
- Header Field contains type of TLP packet, address and other information.

The detailed description of header field given below

- Format of the packet.
- Type of the packet.
- Length for any associated data.
- Transaction Descriptor, including:
  - Transaction ID
  - Attributes
  - Traffic Class
- Address/routing information.
- Byte Enables
- Message encoding
- Completion status
- TLP Digest field contains the calculated ECRC value of the packet

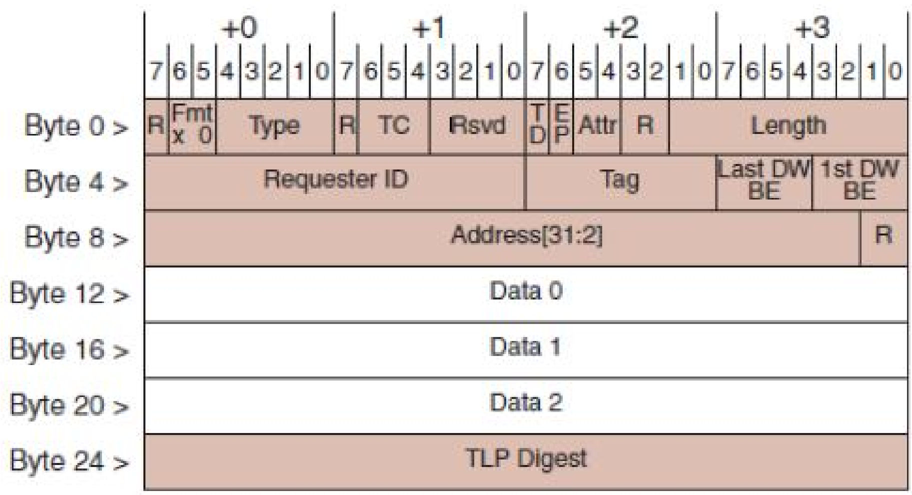

Fig 1.8: Generalized TLP packet.

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. 3, Issue 4, April 2014

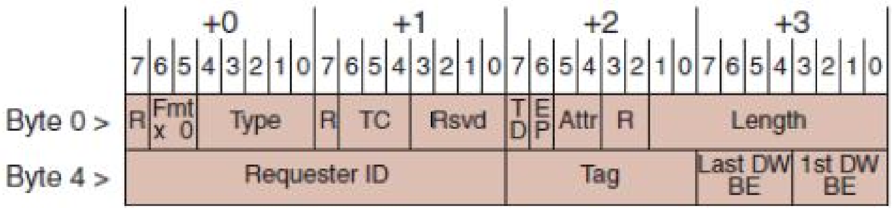

Fig 1.9: Header of TLP packet

- Fmt – size of the header, is there payload present in the packet is depend on this field.
- Length – defines the payload size in DW.
- EP – Poisoned bit.
- TC – Traffic class.
- TD – TLP digest – ECRC field.
- Attr – status (success, aborted).

IV. GENERATION OF ECRC

To ensure end-to-end data integrity detection in systems that require high data reliability, a Transaction Layer 32-bit (ECRC) can be placed in the Digest field at the end of a TLP. The ECRC covers all fields that do not change as the TLP traverses the path. The ECRC is generated by the Transaction Layer in the source component, and checked by the ultimate PCI Express Receiver and optionally by intermediate Receivers. A Switch that supports ECRC checking must check ECRC on TLPs targeting the Switch itself. Such a Switch can optionally check ECRC on TLPs that it forwards. On TLPs that the Switch forwards, the Switch must preserve the ECRC as an integral part of the TLP, regardless of whether the Switch checks the ECRC or if the ECRC check fails [4].

A. Rules for Generating ECRC

A 32-bit ECRC is calculated for the entire TLP (header and data payload) using the following algorithm and appended to the end of the TLP:

The ECRC value is calculated using the following algorithm.

- The polynomial used has coefficients expressed as x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 + x^10 + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1
- The packet inputs are XORed with the coefficients of polynomial until 32bit value is obtained.
- The invariant fields of the TLP’s, header, and the entire data are used in ECRC calculation
- Payloads (if present) are included in the ECRC calculation, all bits in variant fields must be set to 1(binary) for ECRC calculations.
  - Bit 0 of the Type field is variant
  - The EP field is variant
  - all other fields are invariant
- ECRC calculation starts with bit 0 of byte 0 and proceeds from bit 0 to bit 7 of each byte of the TLP
- The result of the ECRC calculation is complemented, and the complemented result bits are mapped into the 32-bit TLP Digest field.
- The 32-bit ECRC value is placed in the TLP Digest field at the end of the TLP.

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. 3, Issue 4, April 2014

- For TLPs including a TLP Digest field used for an ECRC value, Receivers which support end to end data integrity checking, check the ECRC value in the TLP Digest field by applying the same algorithm used for ECRC calculation to the received TLP, not including the 32-bit TLP Digest field of the received TLP
- Comparing the calculated result with the value in the TLP Digest field of the received TLP

Receivers which support end-to-end data integrity checks report violations as an ECRC Error.

V. IMPLEMENTATION OF WORK

For Implementation of generating ECRC for TLP packets the Tools used are:
A. Python
Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. Python code can be packaged into standalone executable programs. Python interpreters are available for many operating systems [6][8].
B. Logic Analyzer
Designed for developers and valuators, the Logic Analyzer allows semiconductor, motherboard and add-in card manufacturers to capture, analyze and view PCI Express traffic
Faster interpretation and debug of PCI Express traffic with color-coded, clearly labeled protocol elements in a graphical display [9].
The scripting of algorithm is done in the Python language and output is seen in LA by capturing the traffic. This script is run on the different transactions of TLP and correct results are observed in LA.

VI. RESULT AND DISCUSSION

Packet	2.5	Packet Error	TLP	Mem	MRd (64)	Length	Requester ID	Tag	Address	1st BE	Last BE	ECRC	LCRC
0	x4	ECRC Err	2000		001:00000	8	000:00:0	0	00000001:25912020	1111	1111	0x21354789	0x6088B245

Fig 1.9: Mem Rd TLP Packet captured in LA with Wrong ECRC

In the fig 1.9, it shows how the captured TLP packet is seen in LA. When the calculated ECRC is wrong, we see the ECRC field in red color in the packet.

Packet	2.5	Packet Error	TLP	Mem	MRd (64)	Length	Requester ID	Tag	Address	1st BE	Last BE	ECRC	LCRC
0	x4	ECRC Err	2000		001:00000	8	000:00:0	0	00000001:25912020	1111	1111	0x45BEB9B1	0x0B810DED

Fig 1.10: Mem Rd TLP Packet captured in LA with correct ECRC

In fig 1.10, it shows how the captured TLP packet is seen in LA. When the calculated ECRC value is correct, we see the ECRC field color changes to white in the packet.

VII.CONCLUSION

The objective of the present work is to understand how the traffic moves from PCIe card to CPU and how it can be used in generating of ECRC for the packets. The same script is run on different packets of PCIe to calculate the correct ECRC value for them. The method used here is efficient and used in many of the industrial environment. By using this ECRC value we can come to know in which path to the CPU the error has been occurred, and this can help to understand in detail about the errors occurred.

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. 3, Issue 4, April 2014

REFERENCES

[1] Subhasish Mitra, Sanjit A. Seshia, Nicola Nicolici, "Post-Silicon Validation Opportunities, Challenges and Recent Advances", Design Automation Conference (DAC), 2010 47th ACM/IEEE
[2] Yanbin Zhang, Qi Yuan, "A Multiple Bits Error Correction Method Based on Cyclic Redundancy Check Codes", 2008 IEEE
[3] Eugin Hyun, Kwang-Su Seong, "Design and verification for PCI Express controller", 2005 IEEE
[4] PCI Specification : http://www.pcisig.com
[5] PCI Express Details : http://en.wikipedia.org/wiki/PCI_Express
[6] Python tool : http://www.tutorialspoint.com/python/
[7] Pci-express-tlp-pcie-primer-tutorial-guide-1: http://xillybus.com/tutorials/pci-express-tlp-pcie-primer-tutorial-guide-1
[8] Python Programming Language : http://en.wikipedia.org/wiki/Python_(programming_language)
[9] Logic Analyzer : http://en.wikipedia.org/wiki/Logic_analyzer