RESEARCH ARTICLE OPEN ACCESS

Implementation of PCIe Using Lower-node Technology

Sneha*, Pushpanjali J**,Kumaraswamy K.V***
**(Department of Electronics and Communication Engineering, Bangalore Institute of Technology, Bengaluru)*
*** (Department of Electronics and Communication Engineering, Bangalore Institute of Technology, Bengaluru)*
****(Senior Technical Manager, Tech Labs, Bengaluru)*

ABSTRACT
PCIe (Peripheral Component Interconnect Express), is a point-to-point connection and a serial computer expansion, high-speed, I/O bus standard. The devices integrated on a motherboard such as, graphic cards, Solid-State Devices, Ethernet, and network cards are generally linked to the chipset of the main PCB of the computer using allotted PCI Express connections. Multiple devices communicate with each other simultaneously using this PCIe link. PCIe standards differ greatly from the older standards, including bandwidth, slot size, compatibility with existing operating systems, speed, better performance scaling for bus devices. Hence, PCIe 3.0 soft IP is developed as per ASIC and FPGA design methodology and this has been designed using the Verilog HDL, Incisive Enterprise Simulator and Questa simulator for simulation, Genus Synthesis Solution and Precision RTL for Synthesis, Conformal Equivalence Checker and FormalPro for Equivalence Check are used for ASIC and FPGA design respectively. A clock frequency of 100 MHz and the Library of 65nm standard cell libraries for synthesis are used in this project.

Keywords - *PCIe 3.0, bus standard, Verilog HDL, Incisive Enterprise Simulator, Genus Synthesis, Conformal Equivalence Check, Questa, Precision RTL, Formal Pro.*

Date of Submission: 17-07-2021 Date of Acceptance: 02-08-2021

I. INTRODUCTION

A connection was developed which was earlier known as “3GIO” and formally contracted as PCIe by the PCI-SIG (PCI Special Interest Group), more than 900 companies groups maintain the conventional PCI specifications.A point- to-point, computer expansion PCIe(Peripheral Component Interconnect Express) is a high-speed bus standards. The devices integrated on a motherboard such as, graphic user interface cards, Solid State Devices, video cards,Ethernet, and network cardsare generally linked to the motherboard chipset using allotted PCI Express connections. The older PCI, PCI-X, and AGP bus standards are restored using this PCIe. PCIe standards differ great ly from the older standards, including bandwidth, slot size, compatibility with existing operating systems, speed, better performance scalingfor the bus device.

The way peripheral devices communicate with the processor in an advanced level is developed by PCI Express connection. It shows improvement in various aspects as mentioned above, but the most important one is that the way the information is transferred. The shift in transmission of data from parallel communication to serial communication, is the best example of this PCI Express connection, that is commond set of control, data lines and set of address are used by PCI hot devices and it sshres parallel architecture Whereas, every device connecting to the root complex(RC) with a separate links to the point-to-point topology of PCIe.

The size of lane from one to 32 are used by the PCIe link in-between the two devices. The PCIe connection is found to be on the concept of lane which is a high-speed serial communication, full-duple x, single-bit. To increase bandwidth, lanes can be assembled and allows slots to have separate physical sizes, as that of the number of lanes connected to the slot. It is also advantageous to use PCIe in applications where high throughput is not required and also it is cost-sensitive. considering an example, of two devices which are communicating using four lanes, they will be able to gain quad times more bandwidth than a x1 lane. Any number of lanes from one to 32 lanes can be merged, the most used number of lanes are x16, x4, and x8.

PCI Express 3.0 enhance the process by encoding to 12b/130b from the previous 8b/10b encoding, decreasing the bandwidth to 1.54% from the 20% of PCI Exp ress 2.0. A desirable DC balance of zero’s and one’s bits in the data stream is established by having a binary polynomial XORing with this data streams in a feedback topology know as a "scrambler”. The data can be recuperated by using the XOR a second time because the scrambling polynomial is known. The scrambling and descrambling are administered at the hardware.

II. HISTORY

In 1981, the first Personal Computer was introduced, the additional cards to add extra capabilities can be installed using expansion slots, where these features are not available in the computer. In the present technology, the most common variant of e xpansion slot accessible is called PCI Express, which was initially named as High-Speed Interconnect(HSI), and later changed to 3GIO(for 3rd Generation I/O) before finalizing to PCI-Express. The first solution for the expansion slot that is PCI was developed by Intel, the lead industry to create the definitive expansion slot, in 1992. The PCI bus was found to be degraded in speed when compared to the high-end video cards, hence the AGP slot was developed. This slot was introduced particularly for video cards. Finally, the PCIe was developed and the standard was composed by a technical working group called the AWG(Arapaho Work Group). The PCIe Exp ress Link is as shown in Figure 1.

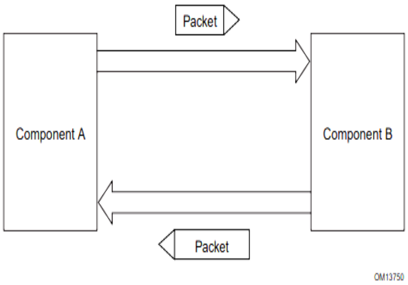


Fig. 1. PCIe Express Link

III. PCI EXPRESS LAYERING ARCHITECTURE

PCI Express protocol mechanism consists of three layers.

- Transaction-Layer
- Data-Link Layer
- Physical-Layer

The communication linking the receiver(Rx) and transmitter(Tx) happens as per the below layers, shown in Figure 1.3. Packets are used to communicate data between components in PCI Express. The Transaction layer creates the packets and carrying of the information between the sender component and the acceptor component is performed by Datalink Layer.

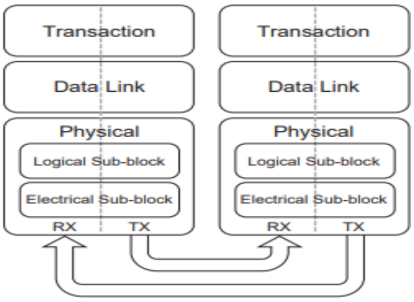


Fig. 2. PCIe Express Architecture

Additional information that are required to handle packets at different layers are added, as these packets flow through layers. The reverse process occurs at the receiver side when compared to the transmitter side that is, as the packets flow to Data Link Layer from their Physical Layer and (for Transaction Layer Packets) to the Transaction Layer the packet get into a form that can be treated at the receiving device. The Figure. 1.4 indicates how an input data packet flows through different layers.

3.1. Transaction Layer

In the architecture of the PCIe the top layer is the Transaction Layer. Assembly and disassembly of Transaction Layer Packets (TLPs) are the primary principle of this layer. Events such as write, read and certain types of event are transferred with these TLPs in transaction. There are two types of transcations:

- Posted Transaction : In this type of transactions are one where the requester does not anticipate to and will not receive a completion Transaction Layer Packet (TLP). That is requests for memory write, without completion.
- Non-Posted Transaction: This type of transactions are one where the requester anticipate to accept a completion Transaction Layer Packet (TLP) from the device. That is it requests for memory read,with completion.

3.2. Datalink Layer

An in-between stage of the Physical-Layer and Transaction-Layer the Data-Link Layer is present. Link acknowledgment, flow initialization protocol, error correction protocol, reliable mechanism for transferring Transaction Layer Packets (TLPs) connecting the two components on a Link and power management are some of the basic responsibilities of the data-link layer.Given the Transaction Layer Packets (TLPs) from the Transaction Layer, this Layer is in-charge of reliably transfer the packets through a PCI Express Link to the receiving component’s Transaction Layer. TLP boundary details to the Data Link Layer is providedd by the Transaction Layer. TLP Sequence Number and a Link CRC(LCRC) for error detection to the TLP are the components of this layer.

3.3. Physical Layer

In Physical Layer, both TLP(transaction Layer Packets) and DLLP(Data-link Layer Packet) type packets are transferred from the Data Link Layer to the Physical Layer for transmission over the channel. Also, packets are accepted by the Physical Layer from the Link and sent to the Data Link Layer of the receiver side. This Layer includes all circuitry for interface operation, including driver and input buffers, parallel-to-serial and serial-to-parallel conversion, PLL(s), and impedance matching circuitry. This Layer is in charge for converting information accepted from the Data Link Layer into an exact format and sending it across the PCI Express Link with compatible device link to the other side of the Link.

IV. DESIGN OBJECTIVE AND METHODOLOGY

The main objective of this project is to design a behavioral model of the Peripheral Component Interconnect Express 3.0 using Verilog language and verify the functionality of the behavioral model to an acceptable level of coverage. ASIC design methodology is used.

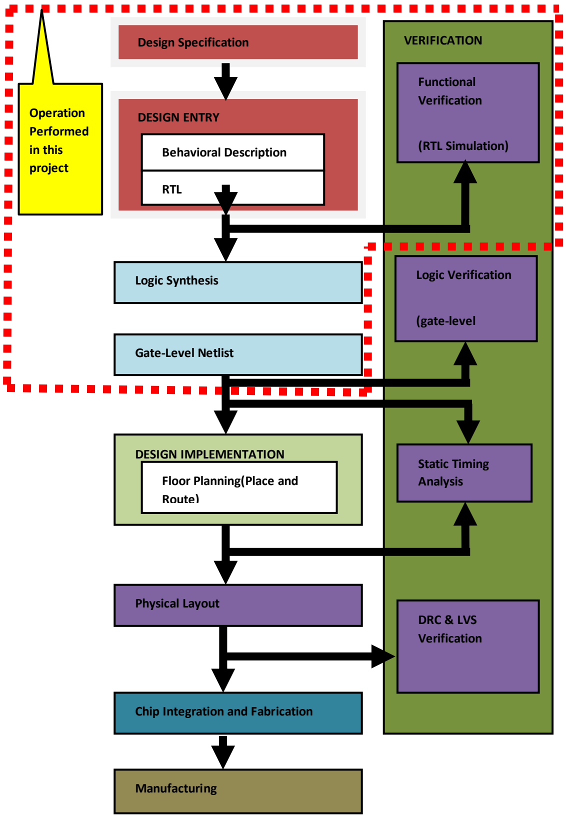


Fig. 3. ASIC Design Flow

3.4. Design Specification

The first phase in the ASIC design flow is distinguishing the specifications. This specification stage is an extremely important part of the design and development process. A document with a list of how a device must function and perform in various operational situations is included in this stage. As technology becomes more advanced, it is important to update new features and design improvements from the older devices, which include high-speed processing and low power consumption. A top-down design approach is employed to manage the complexities of the ASIC design process, and as an initial step, a proper detailed specification is developed. A detailed documented working specification helps in understanding the design process, with the advantage of design being less prone to errors and estimation of project schedule and cost. For the proposed design in this project, the specification is as listed below in Table 1.

3.5. Verification

Functional verification ensures the functionality and logical behavior of the circuit by simulating on a design entry level. The logical design is verified for equating the original design and implementation at several stages in the entire design process to make sure there is a precise ASIC outcome. The verification process, build-in test cases to the design description, and confirming that the expected behavior is obtained. Verification is also performed at other stages of the design, using advanced Electronic Design Automation tools to compare gate-level netlists to the design description and actual layout implementation to the synthesized netlist. If any verification test fails during the process, the design is transmitted back for correction. Verification based on Simulation(also called dynamic verification) is conventionally utilized to simulate the design. Input is given to exercise each line in the HDL code. Providing significant scenarios to check that provided certain input, the design performs to specification a test-bench is built. That is to functionally verify the design.

TABLE I. Design Specification

Design Implementation Methodology	ASIC/FPGA Design Flow
Frequency of Operation	100MHz
Data Input	32-bit
Reset Signal	Active High

Technology Specific Library	65nm
Header	EB(base-16) (Resource of Request Type)
	BE(base-16) (Resource of Completion Type)
Tail	77(base-16)(End of Data)

*[Examples of headers and tail are used and are for illustration purpose only]

4.3 Code Coverage

Coverage computes the number of all the feasible configurations of a design that have been simulated, that is, validates and estimates verification of tested and untested section of the design. It is a tool used for performing design verification. Coverage is defined as the percent of verification attributes that have been considered.

4.4 Logic Synthesis

Synthesis transform HDL description or RTL to gate-level implementation using standard gates, cell library, specified constraints and optimization settings. It consists of three steps: translation, optimization and mapping. The translation is the transformation of the HDL code to gate level boolean equation. Optimization stage involves in bringing down the Boolean equation, then these technology in-dependent boolean logic equations are plotted to technology dependant library logic gates based on design constraints, available library of technology gates. The obtained optimized gate level representation defined as gate level netlist is generally represented in Verilog.

4.5 Equivalence Check

LEC (Logic Equivalence Check) is also called as formal verification. As the design goes through multiple steps like synthesis, place and route and numerous optimizations before it the final tape-out is made of a digital chip. Because of any of the automated or manual changes the logical functionality at every stage does not volatile, we need to make sure that the is exact. The chip becomes useless, if the functionality difference is found at any point during the entire process. This is the reason that, in the entire chip design process LEC is one of the most important checks. With shrinking technology nodes and increasing complexity, logical equivalence check takes a major role in ensuring that two representations of a circuit design exhibit exactly the same behavior.

V. PROPOSED DESIGN

The proposed design will take 32-bit data as input from the user. Then the input data flows through all the three layers of PCIe which are transaction, data-link, and physical layers and is processed at each of the three layers present at transmitter and receiver respectively, according to their functionalities. At the receiver, received data will be verified at each of the layers to ensure data reliability, data integrity and error-free transmission through the physical links. A proper handshake mechanism(ACK/NACK) will be used by the receiver to notify the transmitter about the status of the data. This design functions on a reset high signal. The Block diagram showing the input and output ports of the PCIe 3.0 design is as shown in Figure 4.

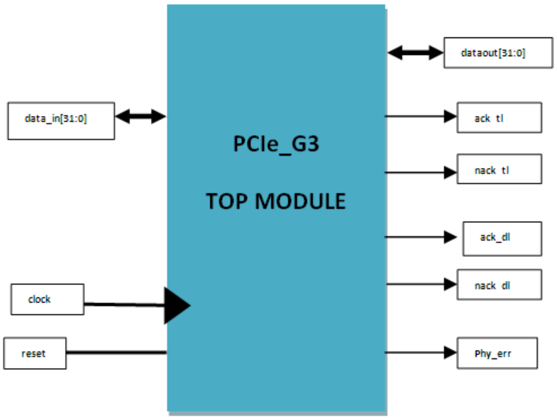


Fig. 4. PCIe 3.0 Top Module Block diagram

The flow chart representing 32-bit input data flowing through all the layers and components has been shown in Figure 5.

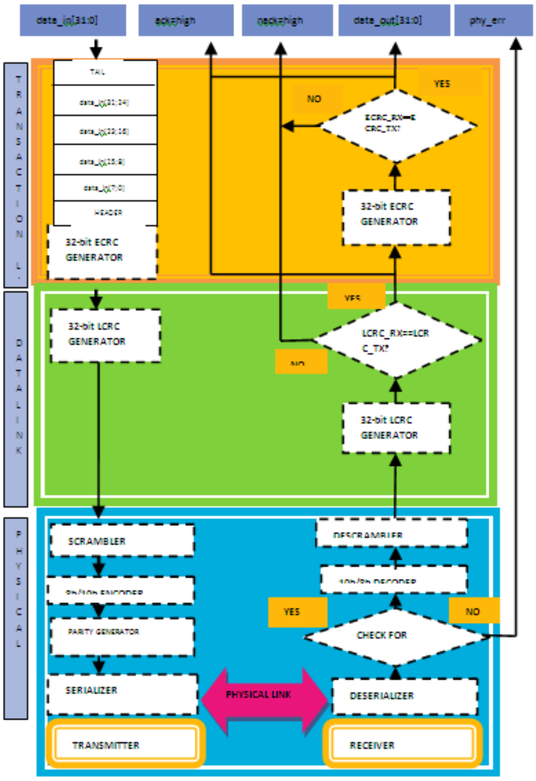


Fig. 5. Data processing through all three layers of PCIe 3.0

5.1 Transaction Layer

At transmitter side transaction layer has the following components:

5.1.2 Header and Tail Append Block

A FIFO buffer with 4stacks containing each with 8bit width will store the incoming 32-bit data into the transaction layer. Now suitable header and tail are appended on either side of the data as shown in Figure 5. In this project, the following header and tail with suitable messages have been used only for illustrations purpose.

5.1.3 32-bit ECRC Generator

The 32-bit ECRC is used to check on the data sent from a transmitter PCIe port to the receiver PCIe port, to make sure that there is an data integrity from end-to end detection in systems that is required for high data reliability. In the transaction layer the transmitter adds the ECRC to the end of the Transaction-Layer-Packets(TLP). The data link layer does not change the ECRC. At the receiver it checks the ECRC against the transmitted TLP. If an error occurs, the receiver passes a message back to the transmitter(NACK). The Advanced Error Reporting (AER) feature of PCIe is used to handle the error.

5.2 Datalink Layer

TLP data integrity is ensured during transfer between Data Link Layers using a LCRC of 32-bit.

5.3 Physical Layer

The following component are present in this Physical Layer:

5.3.1 Scrambler and Descrambler

A technique called data scrambling is implemented to reduce the possibility of electrical resonances on the link. This algorithm is designed using a linear feedback shift register on a various-Lane Link. The functionality of scrambling or descrambling is performed using a serial XOR operation to the data with the initial resuts of a Linear Feedback Shift Register(LFSR) that is synchronized between PCI Express devices.

5.3.2 8b/10bEncoder and Decoder

The 8b/10b coding scheme is implemented for high-speed serial data transmission. On the sender, the encoder plots the 8-bit parallel data input to 10-bit results. The decoder will then pot the 10-bit data to the original 8-bit information. When the 8b/10b coding scheme is worked, the serial data stream is DC-balanced and has a maximum run-length.

5.3.3 Parity Generator and Checker

A digital logic circuit that employees a parity bit in the sender side is called the parity generator. The combinational circuit that estimates the parity in the acceptor is called the Parity Checker. Parity checking is a technique in which an extra bit called parity bit is added, usually at the MSB, of the data stream which needs to be transmitted. Adding one or zero depends upon whether we are encountering even parity or an odd parity

5.3.4 Serializer and Deserializer

In high-speed communications for changing serial data and parallel interfaces in bidirections, a serializer/deserializer(SerDes) is a device employed. It turns the parallel data into serial data so that they can be transmitted over media that does not reliably transfers parallel data or is utilized to lower the bandwidth. The basic SerDes function has two blocks: the Parallel In Serial Out(PISO) block or parallel-to-serial converter, and the Serial In Parallel Out(SIPO) block or serial-to-parallel converter.

VI. ASIC DESIGN FLOW RESULTS AND INTERFERENCE

RTL Simulation results of Integrated Design of PCIe3.0 using Verilog source code was simulated using NCSim tool of Incisive Enterprise Simulator suite, A coverage analysis tool named IMC

(Integrated Metrics Center) was used to yield coverage reports. The testbench was able to cover **94.88** percent of the entire Verilog source code as shown in Fig 16. The schematic and output waveform is as shown in Figure 14 and Figure 15 respectively. The overall delay (from input to output) was observed to be 165 ns. The first input data $(\text{B17C\_AE56})_{16}$ has a header value equal to $(\text{EB})_{16}$. Hence, this data is of request type. In the same way, the second input data $(\text{EF45\_A8BC})_{16}$ has a header value $(\text{BE})_{16}$, hence it is of completion type (as per Table I).

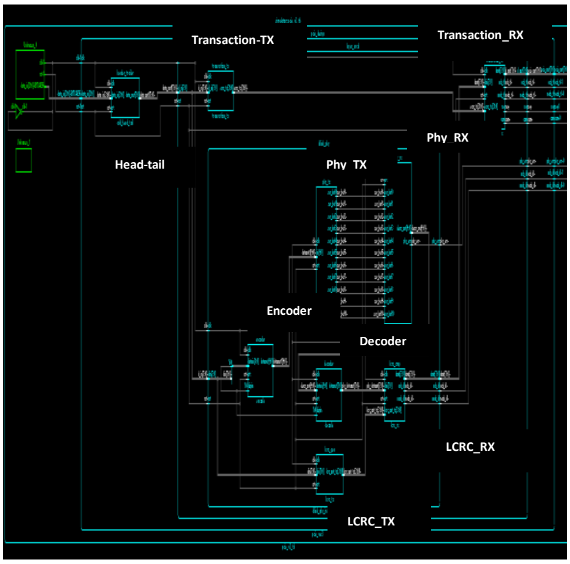


Fig. 14. Schematic of PCIe 3.0

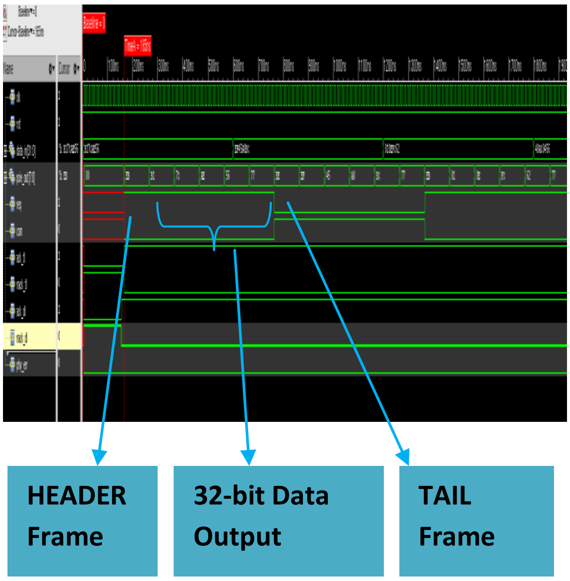
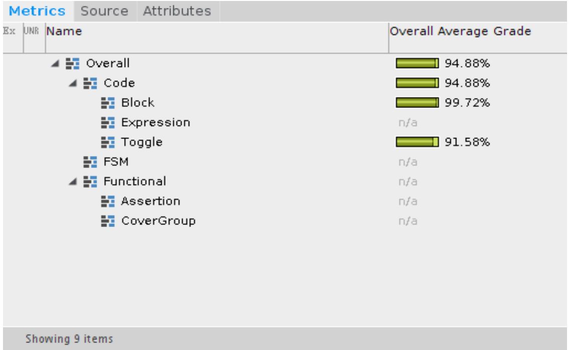


Fig. 15. Simulation Waveform of PCIe 3.0



Metrics Source Attributes

Name	Overall Average Grade
Overall	94.88%
Code	94.88%
Block	99.72%
Expression	n/a
Toggle	91.58%
FSM	n/a
Functional	n/a
Assertion	n/a
CoverGroup	n/a

Showing 9 items

Fig. 16. Detail Code Coverage report of Testbench

6.1 Logic Synthesis results of Integrated Design of PCIe3.0 Using Genus

Genus Synthesis Solutions was use to perform the Logic Synthesis of the design PCIe 3.0. s per the specification given in the tabe I. The design was synthesized with a clock frequency of 100 MHz and 65 nm technology library. The Technology Schematic obtained after synthesis is shown in the Figure 17 below. Summary of synthesis reports, that is timing, area and power reports has been tabulated in the Table II below.

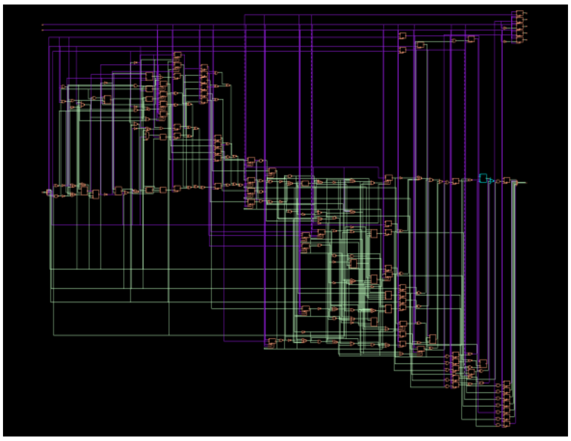


Fig. 17. Technology Schematic of PCIe3.0

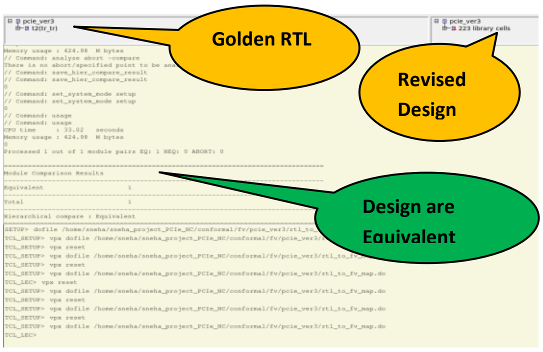
TABLE II. Synthesis Results Report

TYPE OF COVERAGE	PERCENTAGE OF COVERAGE
Area	**700µm²**
Timing	**9.4ns**
Power	**64.41µW**
Number of Cells	**223**

6.2 Equivalence Check Using Conformal LEC

To perform Logical Equivalence check between RTL and Gate-Level Netlist Conformal

LEC was used. Here RTL is considered as Golden design and gate-level netlist is considered as Revised design. After performing LEC the designs were found to be Equivalent. The result of equivalence check is as shown in Figure 18.



```
Memory usage : 424.98  M bytes
// Command: analyze abort -compare
There is no abort/specified point to be ana
// Command: save_hier_compare_result
// Command: save_hier_compare_result
0
// Command: set_system_mode setup
// Command: set_system_mode setup
0
// Command: usage
// Command: usage
CPU time      : 33.02    seconds
Memory usage : 424.98  M bytes
0
Processed 1 out of 1 module pairs EQ: 1 NEQ: 0 ABORT: 0

================================================================================
Module Comparison Results
--------------------------------------------------------------------------------
Equivalent                    1
--------------------------------------------------------------------------------
Total                         1
--------------------------------------------------------------------------------
Hierarchical compare : Equivalent
================================================================================
SETUP> dofile /home/sneha/sneha_project_PCIe_NC/conformal/fv/pcie_ver3/rtl_to_
TCL_SETUP> vpx dofile /home/sneha/sneha_project_PCIe_NC/conformal/fv/pcie_ver3/r
TCL_SETUP> vpx reset
TCL_SETUP> vpx dofile /home/sneha/sneha_project_PCIe_NC/conformal/fv/pcie_ver3/rtl_to_fv_
TCL_SETUP> vpx reset
TCL_SETUP> vpx dofile /home/sneha/sneha_project_PCIe_NC/conformal/fv/pcie_ver3/rtl_to_fv_map.do
TCL_LEC> vpx reset
TCL_SETUP> vpx dofile /home/sneha/sneha_project_PCIe_NC/conformal/fv/pcie_ver3/rtl_to_fv_map.do
TCL_SETUP> vpx reset
TCL_SETUP> vpx dofile /home/sneha/sneha_project_PCIe_NC/conformal/fv/pcie_ver3/rtl_to_fv_map.do
TCL_SETUP> vpx reset
TCL_SETUP> vpx dofile /home/sneha/sneha_project_PCIe_NC/conformal/fv/pcie_ver3/rtl_to_fv_map.do
TCL_LEC>
```

Fig.18. Conformal Equivalence Check

VII. FPGA DESIGN FLOW RESULTS AND INTERFERENCE

RTL Simulation results of Integrated Design of PCIe3.0 using Questa Simuator is as shown below.

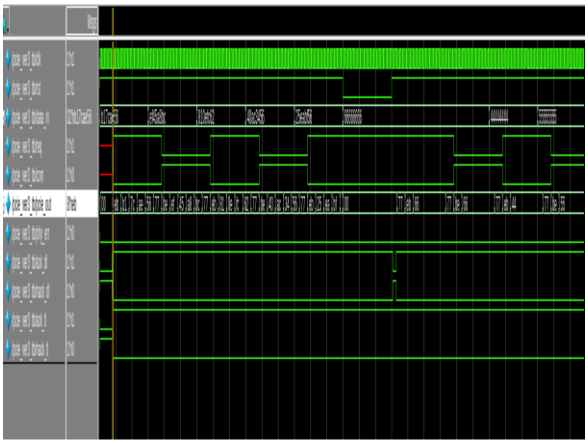


Fig. 19. Simulation waveform of PCIe3.0

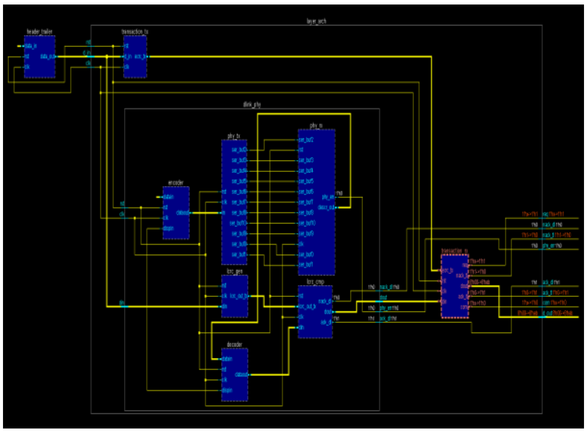


Fig. 20. Schematic of PCIe3.0

7.1. Coverage Report

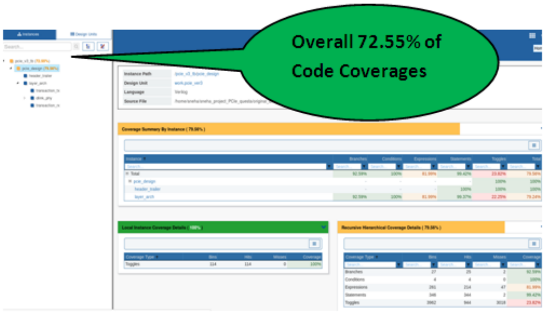


Fig. 21. Code Coverage report of Testbench

7.2 Logic Synthesis Using Precision RTL

For synthesizing PCIe 3.0 design Precision RTL tool was used. Synthesis was carried out with a device clok frequency of 100 MHz (As per the specifications listed in Table I). The FPGA device 7S6CSGA225 belonging to Xilinx vendor, Spartan-7 family with speed grade -1 was selected for the PCIe 3.0 design implementation. Technology Schematic of the PCIe 3.0 is represented in Figure 22 as shown. Table III shows the synthesis results.

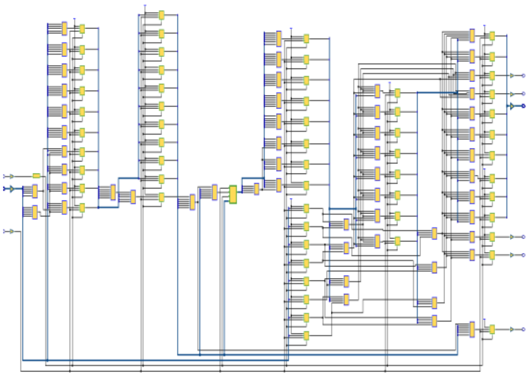


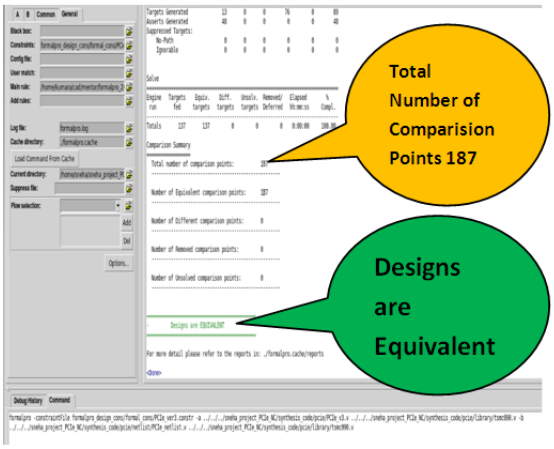
Fig.22. Technology Schematic of PCIe3.0

TABLE III. Synthesis Results Report

TYPE OF COVERAGE	PERCENTAGE OF COVERAGE	
Area	Cell Delay	**38.24%**
	Net Delay	**61.76%**
	Slack	**0.904ns**
Timing	Number of DFFs	**68**
	Number of LUTs	**55**

	Total number of accumulated instances	**147**

7.3 Equivalence Checker Results using FormalPro



Black box:
Constraints: formalpro_design_cons/formal_cons/PCIe
Config file:
User match:
Main rule: /home/kumara/cad/mentor/formalpro_2
Add rules:
Log file: formalpro.log
Cache directory: ./formalpro.cache
Current directory: /home/sneha/sneha_project_P
Suppress file:
Flow selection:

```
Targets Generated        13    0    0    76    0    89
Asserts Generated        48    0    0     0    0    48
Suppressed Targets:
   No-Path                0    0    0     0    0     0
   Ignorable              0    0    0     0    0     0

Solve
==============================================================
Engine  Targets  Equiv.   Diff.    Unsolv.  Removed/  Elapsed    %
 run     fed     targets  targets  targets  Deferred  hh:mm:ss  Compl.
--------------------------------------------------------------
Totals   137     137      0        0        0         0:00:00   100.00

Comparison Summary
==================
  Total number of comparison points:         137
  ------------------------------------------------
  Number of Equivalent comparison points:    137
  ------------------------------------------------
  Number of Different comparison points:     0
  ------------------------------------------------
  Number of Removed comparison points:       0
  ------------------------------------------------
  Number of Unsolved comparison points:      0
  ------------------------------------------------

==============================================
-        Designs are EQUIVALENT
==============================================

For more detail please refer to the reports in: ./formalpro.cache/reports

<Done>
```

```
formalpro -constraintFile formalpro_design_cons/formal_cons/PCIe_ver3.constr -a ../../../sneha_project_PCIe_NC/synthesis_code/pcie/PCIe_v3.v ../../../sneha_project_PCIe_NC/synthesis_code/pcie/library/tsmc090.v -b ../../../sneha_project_PCIe_NC/synthesis_code/pcie/netlist/PCIe_netlist.v ../../../sneha_project_PCIe_NC/synthesis_code/pcie/library/tsmc090.v
```

Fig.23. Formal Equivalence Check after constraining the design

VIII. CONCLUSION

PCIe 3.0 was designed using Verilog, simulated and verified using NCSim and Questa, synthesized successfully with technology library of 65nm using Genus and Precision RTL, performed equivalence check using Conformal LEC and Formal Pro for ASIC and FPGA as per the standard protocol and able to achieve area of 700µm² and power of 64.41µW. The developed soft IP of PCIe 3.0 can be taken for physical design.

IX. FUTURE SCOPE

The developed PCIe 3.0 Soft IP can be taken for Physical Design and Physical Verification. In many application hard IP can be integrated.

REFERENCE

[1]. *PCI Express® Base Specification Revision 3.0*, by PCI-SIG, November 2010.

[2]. Hiroki Nakamura, Hirotaka Takayama, Yoshiki Yamaguchi and Taisuke Boku,2017, *Thorough analysis of PCIe Gen3 Communication*, (IEEE).

[3]. Mallikarjun Vasa, Chun-Lin Liao, Sanjay Kumar, Ching-Huei Chen, Bhyrav Mutnury,2020, *PCIe Gen-5 Design Challenges of High-Speed Servers*, 978-1-7281-6161-7/20/$31.00 ©2020 IEEE

[4]. Sun Zhaopeng, Zhou Kuanjiu, Cui Kai , Hu Shaoqi,2020, *PCIE-based High-Performance FPGA -GPU-CPU Heterogeneous Communication Method* (IEEE).

[5]. Richard A Prasad, Madhusudan Kulkarni, 2017, *Design and Verification of PHY interface for PCIe gen 3.0 and USB gen 3.1 using UVM methodology*, (IRJET).

[6]. Naveen G, Sheetal Kulkarni, Naresh Madhusudhana, K N Raja Rao,2016, *Communication between Host machine and the Host Bus Adapter over PCIE bus using Diagnostic window*,(IJCER)

[7]. Yanhui Han, Yue Feng; Zejie yu, Wenzhong lou, 2016, *Link Initialization and Training in MAC Layer of PCIe 3.0*, (IJCSIT).

[8]. Gokulakrishnan, Radhakrishnan, December 2016, *Design and Verification of ACK/ NAK Protocol of PCI Express Data Link Layer in System Verilog*, (IJRASET).

[9]. Shoeb Mohammed Balabatti, Radha R, April 2014, *Calculation of ECRC FOR TLP-Packets of PCIe Protocol*, (IJAREEIE).

[10]. Rutva Pathak, 2013**,** *Introduction to VIP with PCI Express Technology*,(IJSRD).