

DESIGN AND PERFORMANCE EVALUATION OF PCIe ON DIFFERENT FPGA

*Project report submitted in partial fulfillment of the requirement for the
degree of*

BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING

By

**Himanshu Choudhary (151052)
Lovish Singla (151057)
Gundeep Singh (151066)**

UNDER THE GUIDANCE OF

Dr. Harsh Sohal



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TABLE OF CONTENTS

	Page Number
LIST OF FIGURES	V
LIST OF TABLES	VI
DECLARATION BY THE SCHOLAR	VII
CERTIFICATE	VIII
ACKNOWLEDGEMENT	IX
ABSTRACT	X
CHAPTER-1	
INTRODUCTION	1-4
1.1 INTRODUCTION TO PCI	
1.1.1 BENEFITS OF PCI	1
1.1.1.1 SPEED	1
1.1.1.2 CONFIGURABILITY	2
1.1.1.3 MULTIPLE MASTERS	2
1.1.1.4 RELIABILITY	2
1.2 INTRODUCTION TO PCI EXPRESS	2
1.2.1 BENEFITS OF PCIE	3
1.2.1.1 LAYERED ARCHITECTURE	3
1.2.1.2 HIGH PERFORMANCE	3
1.2.1.3 I/O SIMPLIFICATION	3
1.2.1.4 EASE OF USE	4
CHAPTER-2	
LITERATURE SURVEY	5-25
2.1 OVERVEIW OF PCI PROTOCOL	5
2.2 PCI SIGNAL DESCRIPTION	8

2.2.1 SYSTEM PINS	8
2.2.1.1 CLK	8
2.2.1.2 RST#	9
2.2.2 ADDRESS AND DATA PINS	9
2.2.2.1 AD [31:0]	9
2.2.2.2 C/BE [3:0] #	9
2.2.2.3 PAR	10
2.2.3 INTERFACE CONTROL PINS	10
2.2.3.1 FRAME#	10
2.2.3.2 IRDY#	10
2.2.3.3 TRDY#	10
2.2.3.4 STOP#	11
2.2.3.5 LOCK#	11
2.2.3.6 IDSEL	11
2.2.3.7 DEVSEL#	12
2.2.4 ARBITRATION PINS	12
2.2.4.1 REQ#	12
2.2.4.2 GNT#	12
2.2.5 ERROR REPORTING PINS	13
2.2.5.1 PERR#	13
2.2.5.2 SERR#	13
2.2.6 INTERRUPT PINS	13
2.2.6.1 INTA#, INTB#, INTC#, INTD#	13
2.2.7 COMMUNICATION PINS	14
2.2.7.1 CE	14
2.2.7.2 R_W	14
2.2.7.3 READ_EN	14
2.2.7.4 WRITE_EN	14
2.3 PCI TIMING DIAGRAMS	15
2.3.1 BASIC READ TRANSACTIONS	15
2.3.2 BASIC WRITE TRANSACTIONS	17
2.4 WORKING PRINCIPLE OF PCIe	19
2.4.1 DIFFERENTIAL SIGNALLING	19
2.4.2 LINKS AND LANES	19
2.4.3 PCI EXPRESS VERSIONS	21
2.4.4 DEVICE TYPES	23

2.4.4.1 ROOT COMPLEX	23
2.4.4.2 PCIe TO PCI BRIDGE	23
2.4.4.3 ENDPOINT	23
2.4.4.4 SWITCH	23
2.4.5 PCIe TRANSACTIONS	24
2.4.5.1 TRANSACTION TYPES	24
2.4.6 ARCHITECTURE BUILD LAYERS	25
CHAPTER-3	
XILINX VIVADO TOOL	26
CHAPTER-4	
EXPERIMENTS AND RESULTS	27-36
CHAPTER-5	
CONCLUSION AND FUTURE SCOPE	37
REFERENCES	38

LIST OF FIGURES

Figure1.1: Combination of PCI and PCIe slots	4
Figure2.1: Interface block	8
Figure2.2: Read Transaction	15
Figure2.3: Write Transaction	17
Figure2.4: A Differential signal pair and subtract	19
Figure2.5: X1 Link	20
Figure2.6: The standard PCI Express slot sizes	21
Figure2.7: The Three Architecture Build Layers	25
Figure3.1: Vivado Home Page.	26
Figure4.1: PCIe Design Block	27
Figure4.2: PCIe Simulated Result	27
Figure4.3: Simulated result for X1 lanes	28
Figure4.4: Power Analysis for X1 lanes	28
Figure4.5: Simulated result for X4 lanes	29
Figure4.6: Power Analysis for X4 lanes	29
Figure4.7: Simulated result for X8 lanes	30
Figure4.8: Power Analysis for X8 lanes	30
Figure 4.9: Simulated result of Zynq	31
Figure4.10: Power Analysis for Zynq	32
Figure 4.11: Simulated result for Kintex-7	32
Figure 4.12: Power Analysis for Kintex-7	33
Figure 4.13: Simulated result for Kintex Ultrascale	33
Figure 4.14: Power Analysis for Kintex Ultrascale	34

LIST OF TABLES

Table2.1: PCIe Size Comparison Table	20
Table2.2: PCI Express version Performance Comparison Table	22
Table4.1: Resource Utilization for zynq 7000	35
Table4.2: Resource Utilization for Kintex Ultrascale	35
Table4.3: Delay, Latency and Interval Comparison Table	36

DECLARATION

We hereby declare that the work reported in the B-Tech thesis entitled “**DESIGN AND PERFORMANCE EVALUATION OF PCIe ON DIFFERENT FPGA**” submitted at **Jaypee University of Information Technology, Waknaghat India**, is an authentic record of our work carried out under the supervision of **Dr. Harsh Sohal**. We have not submitted this work elsewhere for any other degree or diploma.

Himanshu Choudhary Lovish Singla Gundeep Singh

Department of Electronics and Communication Engineering

Jaypee University of Information Technology, Waknaghat , India

13/05/2019

CERTIFICATE

This is to certify that the work titled “**DESIGN AND PERFORMANCE EVALUATION OF PCIe ON DIFFERENT FPGA**” submitted by Himanshu Choudhary, Lovish Singla, Gundeep Singh in partial fulfillment for the award of degree of Bachelors of Technology in Electronics And Communication Engineering of Jaypee University of Information Technology, Solan has been carried out under my supervision.

Signature of Supervisor

Name of Supervisor - **Dr. Harsh Sohal**

Designation - **Assistant Professor (S.G)**

ACKNOWLEDGEMENT

May 13, 2019

We would like to express our sincere gratitude to our guide Dr. Harsh Sohal, for his Constant support, encouragement and direction without which the realization of This project would not have been possible. His friendly disposition and invaluable Support and encouragement has proved to be very rewarding.

ABSTRACT

This B.Tech project is based on DESIGN AND PERFORMANCE EVALUATION OF PCIe ON DIFFERENT FPGA. Today's computer systems demand high performing interconnects that can help in providing high definition graphics, full momentum videos, high bw networking etc. PCIe (**peripheral component interconnect Express**) came out as the dominant model for interconnecting all the elements of present day, high performing computer systems.

In our work we have designed PCIe model, worked on the idea of different lanes and how they affect the energy consumption. Also energy efficient PCIe will increase the lifetime of a computer system and PCIe with low delay & latency would raise the efficiency of the system. Three distinct FPGA are considered and the design is realized on the three ICs & we find out the most energy saving architecture & also find the design that will provide high performance among the three designs taken under discussion. There is 46.75% depletion in latency when we shift our PCIe design from 28 nanometer technology based on seven series architecture to 20 nanometer technology based on ultra scale architecture.

CHAPTER 1

INTRODUCTION

1.1 Introduction to PCI

The present PC frameworks, with the emphasis on high definition graphics, full movement videos, high data transfer capacity organizing, etc, go a long ways past the abilities of the engineering that introduced the age of the PC in 1982. Current PC frameworks request elite interconnects that likewise enable devices to be changed or overhauled with least exertion by the end client.

Because of this requirement, PCI (peripheral component interconnect) developed as prevailing mechanism for interlinking the components of present day, elite PC frameworks. It is a well considered standard with various positive highlights that will keep it applicable well into the following century. Initially, considered as an instrument for interlinking peripheral devices on the motherboard, PCI advanced into an event about six diverse physical executions coordinated at explicit market portions yet all utilizing a similar fundamental transport convention.

1.1.1 BENEFITS OF PCI

PCI gives benefits over four fundamental vectors: Speed, Configurability, Multiple Masters and Reliability.

1.1.1.1 SPEED

The essential PCI protocol can exchange up to 132 Megabytes every second, well over a request of greatness quicker than ISA. All things being equal, the interest for data transfer capacity is voracious. Augmentations to the essential convention yield data transfer capacities as high as 512 Mbytes every second and improvement right now under way will push it to a GB.

1.1.1.2 CONFIGURABILITY

PCI offer the capacity to arrange a framework consequently, assuaging the user undertaking of the framework setup. It could be contended that PCI's prosperity owes a lot to the very reality that user need not know about it.

1.1.1.3 MULTIPLE MASTERS

Preceding PCI, most busses uphold one and only "master," the processor. High transfer speed devices could have direct access to memory through a component called DMA (direct memory access) yet devices, when all is said in done, couldn't converse with one another. In PCI, any device can possibly assume responsibility for the transport and start exchanges with some other device.

1.1.1.4 RELIABILITY

"Hot Plug" & "Hot Swap," characterized individually for PCI, offers the capacity to supplant modules while not disturbing the framework's activity. This significantly decreases MTTR (mean time to repair) to yield the vital level of up-time expected of mission-basic frameworks, for example, the phone network.

WHY PCIe?

PCI has a few deficiencies. As the processors, video cards, sound cards and systems have become quicker & all the more dominant, PCI has remained the equivalent. PCIe has a fix width of 32 bits and deals with just 5 gadgets at once. Another convention called PCI Express (PCIe) disposes of a great deal of these weaknesses furnishes more data transfer capacity and is good with existing working frameworks.

1.2 INTRODUCTION TO PCIe

The General reason IO interconnect customary is named, Peripheral component Interconnect express (PCIe), that is upgraded highlight adaptation of PCI bus customary and more cost-effective than PCI-X. Peripheral part Interconnect express - because the name infers this can be a fringe contraption interconnect bus customary. PCIe replaces parallel bus design of additional seasoned rendition, for instance, PCI and

PCI-X, with new versatile sequent purpose to point interface with packet base transmissions.

A fast sequent association, which may work additional sort of a system as hostile a bus, is named PCI specific. PCIe incorporates a switch, that controls some point-to-point sequent associations, that are essentially yield from a amendment, guiding straight toward the devices purpose, wherever info has to go. each device has its very own submitted association. PCIe has no transfer speed sharing as would be expected bus.

1.2.1 BENIFITS OF PCIe

PCI Express gives benefits over five fundamental spaces: Layered architecture, High Performance, I/O Specifications and Ease of use.

1.2.1.1 Layered Architecture

The PCI Express layered engineering improves functionality and versatility. The 3 layers which structure the center of PCIe are Transaction Layer, Data Link Layer and Physical Layer.

1.2.1.2 High Performance

The underlying usage of PCIe use a 2.5 gigabits for every second per course data transmission however the capacity of the bus can possibly grow up to 10 gigabits for each second per heading. PCIe gives the transfer speed adaptability of 250 megabytes for each second per course for beginning single path to 32000 megabytes for each second per heading for motioning crosswise over 32 lanes.

1.2.1.3 I/O Simplification

PCIe gives a one of a kind interface innovation serving numerous market sections. For instance a PC chipset architect may actualize a x16 PCIe design for graphics card, a x1 arrangement for universally useful I/O and a x4 setup as a rapid chip to chip interconnect.

1.2.1.4 Ease of Use

PCIe locally bolsters hot swap and hot plug. Hot Swap implies that you can expel the drive and supplant it with another drive without noteworthy interference to the

framework. If there should arise an occurrence of a reflected circle condition, the framework should re-sync with the new drive to restore the reflected pair. On account of a RAID arrangement, the framework execution might be debased until the drive is supplanted and the checksum information is spread over the new drive, yet once more, there is no noteworthy intrusion to support.

Hot Plug ordinarily implies that you can include another FRU (a disk drive in our eg.), yet you can't remove the FRU without taking a type of blackout. I state that Hot Plug is harder as in you should be cautious and not decipher Hot Plug as having Hot Swap ability.



FIGURE1.1: Combination of PCI and PCIe slots

CHAPTER 2

LITERATURE SURVEY

2.1 Overview of PCI Protocol

PCI bus is a sync bus layout where the exchange of data is performed in regard to the system clk. Our PCI layout works on a clock cycle of 33 megahertz which allows one bus exchange carried out every thirty nanoseconds. Afterward, PCI layout expanded the bus architecture to assist tasks at 66 megahertz; nonetheless most of the current PCs still work on 33 megahertz clock speed.

PCI implemented a multiplexed bus for address and data i.e. AD[31:0], a 64 bit data bus was also implemented but majority of a current PCs bolster a 32bit data exchanges via the base 32-bit PCI connectors. At 33 Megahertz, an information rate of 132 mb/sec is supported by the 32bit bus and an information rate of 264 mb/sec is supported by the 64bit bus.

Due to the multiplexing of Address & information bus, PCI permits a lower pin count of the PCI connectors which empowers low price and smaller size of the package for PCI segments. Traditional thirty two-bit PCI add on boards utilize simply around fifty signal pins on PCI connectors out of that 32 of them are multiplexed AD bus. The cycles of the PCI bus are started by passing a address on AD[31:0] lane amid 1st clock edge referred to as address part. The address part is motioned by initiation of a signal called FRAME#. The subsequent clock edge starts the primary of a minimum of 1 information phases during which data is changed over the AD [31:0] signals.

In PCI choice of words, data values are changed linking associate degree instigators that is the bus master, associate degrees an objective that is called bus slave. Instigator derived the C/BE[3:0]# signal amid the address part for flagging the exchange (memory scan, memory write, Input output read, input output write, then forth.). Amid information part the C/BE[3:0]# signals fill in as computer memory unit empower to demonstrate that data bytes are legitimate. Each of the instigator and target could plant postponement state into information transferred by dissenting the IRDY# and TRDY# signals. Legitimate data exchanges happen on every clk edge IRDY# and TRDY# are declared.

The PCI bus exchange comprises of a single address phase but arbitrary number of data phases. Input output activities which get to register inside PCI ordinarily have only one data's phases. Memory exchanges which shifts block of data's comprise off different data's phase which write and read numerous back to back memory areas. Both initiator as well as the target might end the bus transfer succession whenever. The initiator signal finish off the bus exchange via dissenting the FRAME signals amid least data phases. An objective might end bus's transfer via stating the STOP signal. At point when our initiators identify a functioning STOP signal, it should end the present transfer and re-arbitrate for bus and then proceed. In the event that STOP# is attested with no data phases finishing, the objective has issued a retry. On the off chance that STOP# is attested after at least one data phase has effectively finished, the objective has issued distinction.

Initiator mediates for responsibility of the bus by affirming REQ# sign to focal referee. The referee awards responsibility for bus via stating the GNT# signals. The REQ# signal and GNT# signal are one of a kind on a for every space premise enabling the mediator to actualize a bus reasonableness calculation. Intervention in PCI is "hidden" as in it doesn't expend clock cycle. The present initiator's bus transfer are covered with the intervention procedure which decides the following proprietor of the bus.

PCI supports thorough automatic design instrument. Every PCI device incorporates a lot of setup registers which permit recognizable proof of the kind of device like SCSI, videos, Lan etc. & an organization who delivered it. Different registers permit design of the device's Input Output address, memory address and so on.

In spite of the fact that it isn't generally actualized, PCI bolsters 64-bit tending to. Not at all like the 64 bit information bus alternative which needs a more extended connector that has extra 32 bit of information signals, 64 bit tending to be bolstered via base 32 bit connectors. Double Address Cycles has issued where the lower request 32 bits of the location are being driven onto the AD [31:0] signal amid the first addresses phase, & the higher request 32 bits of the location are driven into the AD [31:0] signal amid a 2nd address phase. A rest of its exchange proceeds with like an ordinary bus transfer.

PCI characterizes supports for every 5 Volts & 3.3 Volts flagging dimensions. The PCI connector characterizes pins location for every 5 Volts and 3.3 Volts level. In any case, earlier PCI frameworks were 5 Volts just, & didn't give dynamic power on the 3.3 Volts connector pins. After some time more utilization of the 3.3 Volts interface has normal, however include loads up which must work in more established heritage frameworks

are confined to utilizing just 5 Volts provide. The “keying” conspire an actualized inside PCI connector to counteract embeddings a include board in to a framework within incongruent supplied voltage.

Albeit utilized most broadly in computer good frameworks, PCI bus engineering as processor autonomous. PCI signal definitions are nonexclusive enabling bus for be utilized in the frameworks dependent on the further processor family. PCI incorporates severe details for guarantee a signals standard required for activity at the 33Mhz and 66 MHz. Segments and include sheets must incorporate one of a kind bus drivers which are explicitly intended for the usage in a PCI bus condition. Regular T T L gadgets utilized in the past bus usage, for example, ISA are not agreeable with necessities of PCI. This confinement alongside the fast bus speed directs that nearly all PCI devices are actualized for customised ASICs.

The fast speed of PCI restricts the quantity of extension spaces on solitary bus to close to three or four, when contrasted with six or seven prior bus models. To allow extension buses that have more than three or 4 spaces, the PCI-SIG characterized a PCI to PCI Bridge component. PCI to PCI Bridges are ASICs which electrically confine 2 PCI bus while permitting bus exchanges to be sent starting with a single bus then onto the next. Each extension device has an "essential" PCI bus & an "auxiliary" PCI bus. Different extension devices might be used for making a framework with numerous PCI bus.

2.2 PCI Signal Description

2.2.1 System Pins

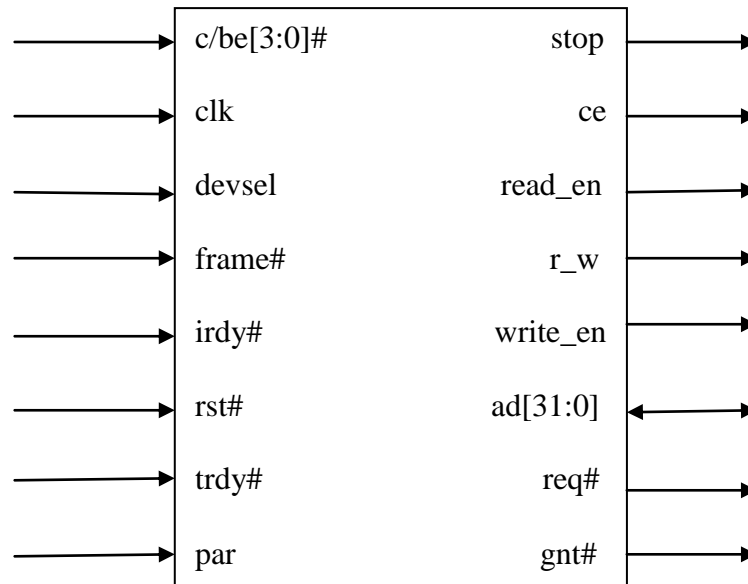


Figure2.1: Interface block

2.2.1.1 CLK

Clock's gives timing references to each and every exchange on our PCI bus. All of the PCI signals with the exception of reset & interrupts are mapped on the rising edge of CLK signals. Each bus timings particulars are characterized with respect to rising edge of our clock. For majority PCI frameworks the CLK signals works with a frequency level of 33 Megahertz. To work at 66 Megahertz frequency level, both the PCI framework and the PCI board should be explicitly intended to work with the higher CLK frequency. The 66 Megahertz framework supplies a 66 Megahertz CLK only if the included board underpins, & supplies a default value of 33 Megahertz CLK if the included board does not bolster the higher frequency. In like manner, in the event that a framework is fit for giving just 33 Megahertz clock then at that point a 66 Megahertz included device must most likely work utilizing the lower frequency value. The base recurrence of the Clock signal is determined at 0 Hz allowing Clock to be "suspended" for the energy sparing reasons.

2.2.1.2 RST#

Reset is operated dynamic low to make a hardware reset off a PCI devices. Reset will make the PCI device's design registers, state machines, & yield sign to put in their underlying state. The RST# is stated & disserted nonconcurrently to CLK signal. This will stay dynamic for any rate 100 ms after the CLK ends up steady.

2.2.2 ADDRESSES & DATA PINS

2.2.2.1 AD [31:0]

Addresses and Data are multi-plexed into these pins. AD [31:0] exchanges a 32 - bit physical location amid "address phases", & exchanges 32 - bits of information data's throughout "data phases". A address phases happens amid a clock follows a higher to lower change in a FRAME# signal. An data phases happens when these IRDY# & TRDY# attested lower. Amid compose exchanges a initiator driven legitimate information on AD [31:0] amid every of the cycles it drive's IRDY# lower. Target drive's TRDY# lower when it can acknowledge the compose information. At the point when these IRDY# and TRDY# are lower, the objective catches the compose data & the exchange's finished. Read to exchanges a inverse happens. The objective drive's TRDY# lower when substantial data is driven on AD [31:0], and the initiator drives IRDY# low when it can acknowledge the data. At the point when these IRDY# & TRDY# are lower, initiator catches the data & exchange is finished. Bit 31 is the hugest Addresses and Data bit. Bit 0 is the lesser noteworthy Addresses and Data bit.

2.2.2.2 C / BE [3:0] #

Bus Command & Byte Enable are multi-plexed into the pins. At the time of address phase an exchange of the sign convey bus order that characterizes a kind of exchange to be performing. At the time of data phase an exchange of the sign convey bytes empower data. C / BE [3] # is byte empower of the maximum critical byte (AD [31:24]) & C / BE [0] # is bytes empower of the rent huge bytes (AD [7:0]) & C/ BE [3:0] # signals are being driven just by a initiator & are effectively determined through all the addresses & data phase of the transaction.

2.2.2.3 PAR

Equality is uniform equality over a AD [31:0] & C / BE [3:0] # signal. Indeed, uniform equality suggests there are even numbered '1's in the AD [31:0], C / BE [3:0] #, & PAR signal. PAR signals has an indistinguishable timings from AD [31:0] signal, yet it delay by the one cycles which enable max opportunity for figure legitimate equality.

2.2.3 INTERFACE CONTROL PINS

2.2.3.1 FRAME#

Cycles Frame is driven lower to the initiator for flag in the beginning of the new bus transactions. Addresses phase happens amid a main clock cycles after the higher to lower progress in FRAME# signals. In event that the initiators expect for play out an exchange with just a solitary data phase, at that point it will give FRAME# back higher after just single cycles. In the event that different data phase is to be played out, the founder will hold the FRAME# to low in everything except end data phase. Initiator flag it plans for play out an ace started end by driven FRAME# higher amid last data phase an exchange. At the time an objective started end the initiator will keep on driving FRAME# low through the finish of the exchange.

2.2.3.2 IRDY#

Initiator Ready is driven lower by the founder, a sign is prepared for finish present information phase to exchange. At the time composes shows the founder has set legitimate information for AD [31:0]. At the time understands it shows that initiator has prepared to acknowledge information in AD [31:0]. When stated, initiator hold IRDY# lower till TRDY# has driven lower to finish a exchange, or objective uses a STOP# signals for terminating without performs the information transfer. IRDY# licenses the founder to embed hold up state as expected to moderate the informaton transfers.

2.2.3.3 TRDY#

Target Ready is taken lower to the target as a sign it is prepared to finish the current data phase of the exchange. At the time writes it shows the founder has set legitimate information to AD [31:0]. At the time read to shows the founder is prepared for valid

data to AD [31:0]. When stated, target hold TRDY# lower until IRDY# is taken low to finish the exchange, or objective use to STOP# signals to end without perform the information transfer. TRDY# licenses a founder to embed hold up state as expected for moderate the information transfer.

2.2.3.4 STOP #

Stop has driven lower for the objective to demand a founder end present transactions. If an objective requires a significant lot of the time for react to the transactions, it might utilize a STOP # sign to end the transactions so that the transport could be utilized for performing different moves then. At the point when the objective ends a transaction without playing out any information stages it is known as a retry. On the off chance that at least one information stages are finished before the objective ends the transactions, as it's known as a distinction. Retry or separate flag a founder which should return sometime in the future to endeavor playing out the transaction once more. In case of a deadly mistake, for example, an equipment issue the objective may utilize STOP # and DEVSEL # to flag the anomalous end of a transport exchange known as objective prematurely end. The initiator can utilize the objective prematurely end to flag framework programming that a lethal blunder has been recognized.

2.2.3.5 LOCK#

Lock might affirmed by the initiator for demand selective accesses as playing out numerous transaction for an objective. Keeps different initiator for changing locked address until a specialist starting a lock could finish it's transaction. Just explicit locales (at least 16 byte) of objectives locations is lock for select access. This LOCK # is affirmed, others non-selective transaction will continue with the addresses which are not presently locked. Be that as it may, any non-select gets for the objective's lock location space may be denied by means of retry task. LOCK # is proposed for used by bridge device for prevent dead lock.

2.2.3.6 IDSEL

Introduction Device Select for utilized a chip select while PCI arrangement write and read transaction. IDSEL is taken by PCI framework and one of its kind for each

opening premise. This permits the PCI arrangement of mechanism for independently addresses every PCI devices in a framework. The PCI devices are selected via configuration cycles just if IDSEL# is higher, AD [1:0] is "0" (indicates the sort 0 configuration cycles), & command place on a C / BE [3:0] # signal amid addresses stage is "configuration read" & "configuration write". AD [10:8] might utilize to choose 1 of 8 "functions" inside PCI devices. AD [7:2] selects one by one configurations enlists inside the devices & functions.

2.2.3.7 DEVSEL #

Devices Selected is taken dynamic lower by the PCI targets then it distinguishes it's location in a PCI transport. DEVSEL # might be taken 1, 2, or 3 clocks follows a location stage. DEVSEL # always asserts with or preceding a check edges in it a TRDY # signals are assert. DEVSEL # had been assert, it's can't be disserted till a last information stage had finished, & a objective issues an objective aborted. On an off chance that the initiator never gets a functioning DEVSEL # its ends transactions which are named as master aborted.

2.2.4 ARBITRATION PINS

2.2.4.1 REQ #

Requests are utilized by the PCI gadget to requests utilization of a transport. Every PCI gadget had interesting REQ # signals. A referee in a PCI framework gets a REQ # signal by every gadget. It's significant that these sign be tri stated since RST # is assert avert the framework hung. The sign that's executed just by gadgets equipped for be n starter.

2.2.4.2 GNT #

Grants demonstrate that the PCI devices solicitation is utilize a bus had been grant. Every PCI devices had its very remarkable GNT # signals by a PCI framework referee. On the off chance that devices GNT # signals are dynamic amid 1 clock's cycle, at that point devices might start the transactions on a accompanying clock's cycle by attesting a FRAME # signals. That signs are actualized just by device equipped for be a starter.

2.2.5 ERROR REPORTING PINS

2.2.5.1 PERR#

Parity Error is utilized for detailing information parity mistakes amid all PCI transactions with the exception of an "Special Cycle". PERR # has taken lower 2 clock's period after a information stage with awful parities. It has taken lower for at least 1 clock's periods. PERR # has share with every PCI device & has taken with the tri state drivers. The draw up resistors guarantees a signs are continued in the inert states then none of devices are driven into it. In the wake of being affirmed lower, PERR # always be taken higher 1 clock's before be tri stated to reestablish a sign to it's latent states. That guarantees a sign do not stay lower in accompanying cycles in light of the moderate ascent because of the draw up.

2.2.5.2 SERR#

Framework Error is for announcing address parity blunders, information parity mistakes amid the Special Cycles, & some another lethal framework mistake. SERR # shares with every PCI device & are driving just as the channel signals (its driven lower or tri stated PCI device, however none determined higher). It's actuated recurring to CLK, however then discharged would skim higher non-synchronously from a draw up resistors.

2.2.6 INTERRUPT PINS

2.2.6.1 INTA#, INTB#, INTC#, INTD#

Interrupt is taken lower with the PCI device to demand consideration through there devices drivers programming. This are characterized "level sensitive" & is driven lower as the open channel signals. When affirmed, the INT x # sign would keep on being stated as PCI devices till the devices drivers programming clear a pending solicitation. The PCI devices which contain just the solitary capacity will utilize just INTA#. Multi-work devices, (for example, a mix LAN/modem include board) may utilize different INT x # line. The solitary capacity devices utilize INT A #. The 2 capacity devices utilize INT A # & INT B #, and so forth. Every PCI devices driver always be equipped

for transferring the intrude on level anchoring by different device utilizing an interfere with vector.

2.2.7 COMMUNICATION PINS

2.2.7.1 CE

When it goes low, RAM is active to make the transaction.

2.2.7.2 R_W

When it is high, read operation occurs to the ram and when it is low write operation occurs to the ram.

2.2.7.3 READ_EN

When read_en is high data is read from the register.

2.2.7.4 WRITE_EN

When write_en is high data is written to the register.

2.3 PCI Timing Diagrams

2.3.1 BASIC READ TRANSACTION

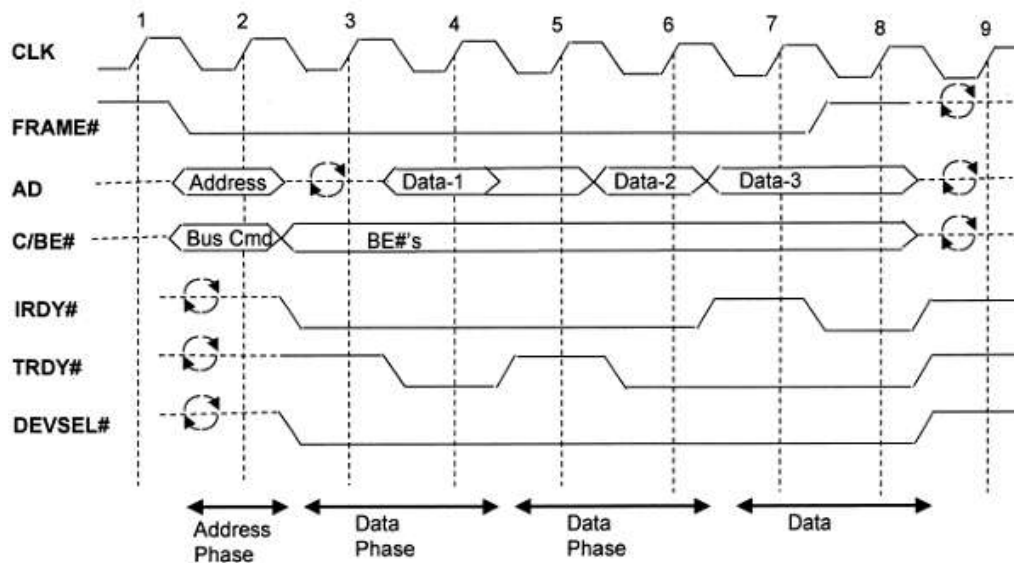


FIGURE 2.2: Read Transaction

Clock 1: The bus is latent and most banner & tri-expressed. The ace for the up and coming exchange has gotten its GNT # and recognized, the bus is inactive so this drive FRAME # higher at first.

Clock 2: AP: The ace drive FRAME # low & place an objective location in Address data bus & bus direction in C/BE # bus. Every objectives lock location & order in rising edge for clk 2.

Clock 3: The master declares the suitable lines of the C / BE # byte enable bus & furthermore declares IRDY # to show that it is prepared to acknowledge read information from target. Our target that perceives its location on the AD bus attests DEVSEL # to recognize its AD. This is additionally a turnaround cycle: In a read exchange transaction, the ace drives the AD lines amid the location stage and the objective drives it amid the information stages. At whatever point beyond what one gadget can drive a PCI bus line, the detail requires a one-clock-cycle turnaround, amid

which neither one of the devices is driving the line, to dodge conceivable dispute that could result in clamor spikes and superfluous power utilization. Turnaround cycles are recognized in the planning graphs by the two round bolts pursuing each other. The ace expresses the appropriate lines of the C/BE # byte enable bus and furthermore affirms IRDY # to demonstrate that it's prepared for acknowledge read information from objective. The objective which perceives particular location on A D bus affirms DEVSEL # to recognize a choice. It's likewise some turnaround cycle: At the read transaction, ace drives Address Data lines amid location stage & the objective takes it amid information stages. At whatever point beyond what one device can drive a PCI bus line, the determination requires a one-clock-cycle turnaround, amid which neither one of the devices is driving the line, to keep away from conceivable dispute that could result in commotion spikes and superfluous power utilization. Turnaround cycles are recognized in the planning outlines by the two roundabout bolts pursuing one another..

Clock 4: Our target posts information on Address Data bus and states TRDY #. Master locks information on rising edge point of clk 4. Information exchange happens on a clk cycle amid that both IRDY # & TRDY # are affirmed.

Clock 5: Our target disserts TRDY # showing that coming data isn't prepared for transferring. Still, our target requires continuing driving Address Data bus so that it is prevented from getting floated. This is a wait cycle.

Clock 6: Our target is set upcoming data object on Address Data bus and asserte TRDY #. Both of them IRDY # & TRDY # is assert so master latches the information bus.

Clock 7: Master is disserted IRDY # indicated that is not ready for upcoming data. This is another wait cycle.

Clock 8: Master is reassert IRDY # & disserted FRAME # for show, it's last information exchange. Accordingly the objective disserts Promotion, TRDY# and DEVSEL#. The master disserts C/BE# and IRDY#. This is an master started end.

2.3.2 BASIC WRITE TRANSECTIONS

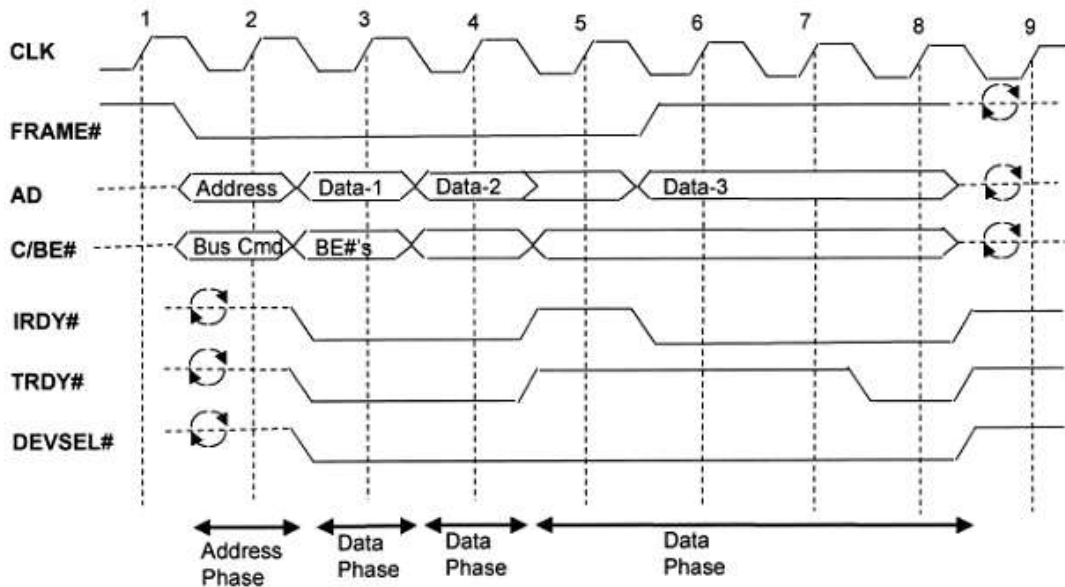


Figure2.3: Write Transaction

Clock 1: Bus is free/idle.

Clock 2: Initiator states substantial address and places a write command on the C / BE # signals. This is the address phase.

Clock 3: Initiator drives legitimate compose data and byte enable signals. The initiator declares IRDY # low showing legitimate compose data is accessible. Our target states DEVSEL # low as an affirmation it has decidedly decoded the location (Our target may not declare TRDY # before DEVSEL #). Our target drives TRDY # lower showing it is prepared to catch data. The primary data stage happens as both IRDY # & TRDY # are low. Our target catches the compose data.

Clock 4: Initiator gives new data & byte empowers. The 2nd data stage happens has both of them IRDY # & TRDY # are low. The objective catches the compose data.

Clock 5: Initiator disserts IRDY # demonstrating it isn't prepared to give the following data. The objective disserts TRDY # demonstrating it isn't prepared to catch the following data.

Clock 6: Initiator gives the following legitimate data and states IRDY # low. Initiator drives FRAME # higher showing it is the last data stage (ace end). The objective is as yet not prepared and keeps TRDY # high.

Clock 7: The target is still not prepared and keeps TRDY # higher.

Clock 8: The target is prepared and states TRDY # low. The 3rd data phase starts as both IRDY # & TRDY # are low. The target catches the write data.

Clock 9: FRAME #, AD, and C/BE# are tri-stated, as IRDY #, TRDY #, and DEVSEL # are driven inactive high for one cycle prior to being tri-stated.

2.4 WORKING PRINCIPLE OF PCIe

At the point when the PC starts, PCIe figure out which all devices are associated with the mother board, sets up connections between them. It directs the progression of the traffic and arranges the width of every connection. The ID of devices and associations is completed by drivers for the PCIe device.

2.4.1 Differential Signalling

The PCIe utilizes differential flagging method, which utilizes two transmission lines for sending one sign. These two sign have positive and negative voltage levels individually. The data signal is transmitted in positive and negative sign and at the collector side they are subtracted to get unique sign. This method is profoundly successful for commotion undoing.

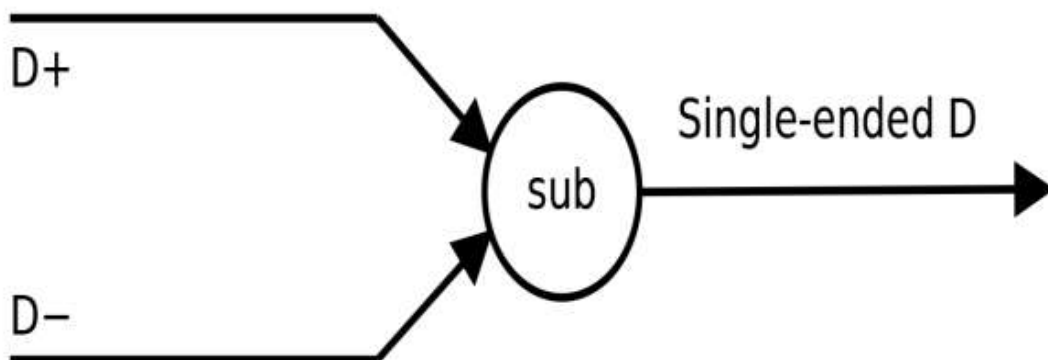


FIGURE2.4: A Differential signal pair and subtract

2.4.2 Links and Lanes

The association between two PCI Express devices is called Connection. A connection comprises of various Paths. A path is the term utilized for a solitary arrangement of differential transmit and get sets. The PCI Express base determination characterizes the accompanying design of sequential connections: X1, X2, X4, X8, X12, X16, and X32.

A X1 setup demonstrates that the connection between two PCI Express devices comprise of a solitary path. A X4 design demonstrates that the connection between two PCI express devices comprise of a solitary path.

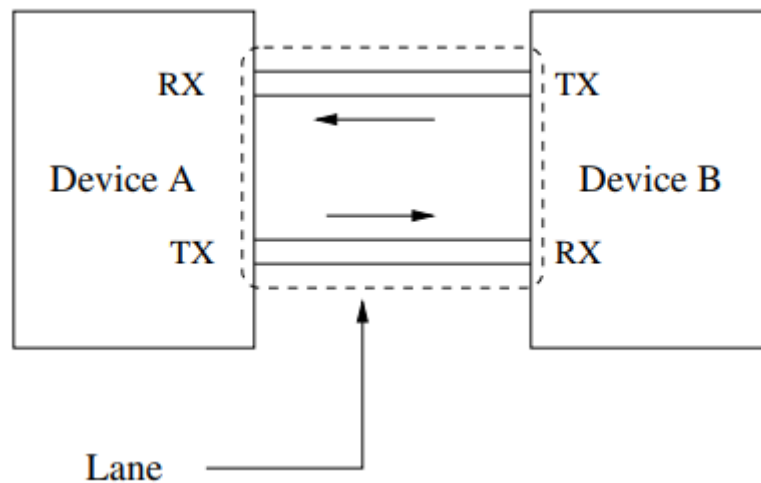


Figure2.5: X1 Link

PCI Express Size Comparison Table		
Width	Number of Pins	Length
PCI Express x1	18	25 mm
PCI Express x4	32	39 mm
PCI Express x8	49	56 mm
PCI Express x16	82	89 mm

Table2.1: PCIe Size Comparison Table

PCIe cards can fit in any of the PCIe slots on the motherboard which is at least as big as the slot itself. For example, a PCIe 1x card can fit in any PCIe 4x, PCIe 8x, or PCIe 16x slots. A PCIe 8x card can easily fit in any PCIe 8x or PCIe 16x slot.

PCIe cards which are bigger than the PCIe slot might fit in the smaller slot but only if that PCIe slot is open-ended (i.e. does not have a stopper at the end of the slot).

In general, a larger PCI Express card or slot supports greater performance, assuming the two cards or slots you're comparing support the same PCIe version.

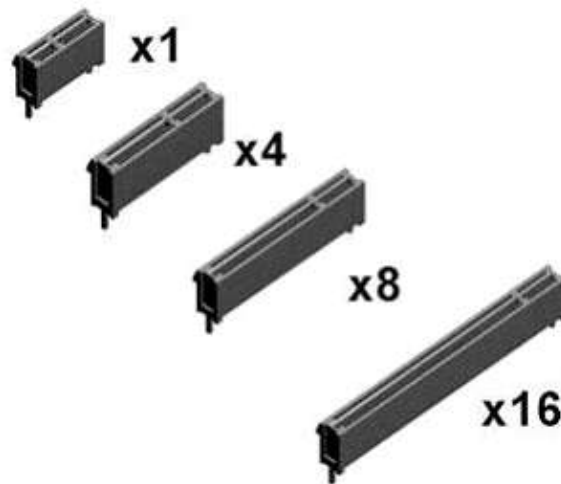


Figure2.6: The standard PCI Express slot sizes

2.4.3 PCI Express Versions

A number that you see after PCIe on a product or on a motherboard is symbolizing the current version of the PCI Express specification which is supported.

- **PCI Express 1.0:** In 2005 PCI-SIG introduced the PCI Express version 1.0. It was an update over the previous PCI Express 1.0a (launched in 2003) and it came with many improvements.
- **PCI Express 2.0:** In 2007 PCI-SIG announced the availability of the PCI Express 2.0 version that came with 2 times the transfer rate in comparison to the PCI Express 1.0. The per-lane throughput was increased from 250 MBps to 500 MBps. The PCI Express 2.0 motherboard is fully backward compatible. The PCI-SIG also mentioned several improvements in the feature list of PCI Express 2.0 from point-to-point data transfer along with the software architecture.
- **PCI Express 3:** In 2007 PCI-SIG announced the availability of PCI Express 3.0. It offered a bit rate of 8 Giga-transfers per second (GT/s). Also it was known to be backward compatible with the ongoing implementations of the existing PCI Express. PCI Express 3.0 came with an updated encoding scheme.

- **PCI Express 4:** PCI-SIG officially announced PCI Express 4.0 on June 8, 2017. There are no encoding differences from 3.0 to 4.0. But in terms of the performance, the throughput per lane of PCIe 4.0 is 1969 MB/s.
- **PCI Express 5:** Expected to arrive in late 2019 and as usual the speed will also be increased two folds.

PCI Express Link Performance Comparison Table		
Version	Bandwidth (per lane)	Bandwidth (per lane in an x16 slot)
PCI Express 1.0	2 Gbit/s (250 MB/s)	32 Gbit/s (4000 MB/s)
PCI Express 2.0	4 Gbit/s (500 MB/s)	64 Gbit/s (8000 MB/s)
PCI Express 3.0	7.877 Gbit/s (984.625 MB/s)	126.032 Gbit/s (15754 MB/s)
PCI Express 4.0	15.752 Gbit/s (1969 MB/s)	252.032 Gbit/s (31504 MB/s)

Table2.2: PCI Express version Performance Comparison Table

2.4.4 Device Types

The PCIe Base Specifications identifies 4 types of PCIe Devices. These are:

- Root Complex
- PCI Express to PCI Bridge
- Endpoint
- Switch

2.4.4.1 Root Complex

The Root complex is defined as the head or root of the connection between the I/O system ,CPU and the Memory. Each interface of the root complex defines a different hierarchy domain.

2.4.4.2 PCI Express to PCI Bridge

A PCI Express to PCI Bridge consists of one PCIe port and one or more PCI bus interface. This bridge allows PCIe to co-exist on a stage with existing PCI technologies. This device must fully support all PCI transactions on its PCIe interface. It must follow a bunch of rules for properly converting PCI transactions into PCIe transaction.

2.4.4.3 Endpoint

An Endpoint is a device which can request/complete PCIe transaction by itself or on behalf of some non PCIe device. There exist two types of Endpoints and they are separated by the types of transactions they perform.

2.4.4.4 Switch

A Switch is used to fan out a PCI Express hierarchy. It is responsible for efficiently forwarding transactions to the required link. Unlike a root complex, it should always manage peer-to-peer transactions in-between down stream devices.

2.4.5 PCI EXPRESSES TRANSACTIONS

Transactions form the basis for the exchange of information between different PCI Express devices. PCI Express uses a split-transaction protocol. This implies that there are 2 parts to the transaction i.e. the request and a completion. The transaction initiator is referred to as the requester, it sends out the request packet. It makes its way towards the desired target of the request, referred to as the completer.

2.4.5.1 Transaction Types

The PCI Express architecture defines four transaction types' memory, input output, configuration and message.

Memory Transactions

Transactions target in the memory space and transfer data to or from a memory mapped location. There are several types of memory transactions: Memory Read Request, Memory Read Completion and Memory Write Request. Memory transaction use either 32 bit addressing or 64 bit addressing.

I/O Transactions

Transactions targeting input output space transfer data to or from an input output mapped location. There are several types of input output transactions: Input Output Read Request, Input Output Read completion, Input Output Write Request and Input Output write Completion. I/O transaction use only 32-bit addressing.

Configuration Transactions

Transactions targeting the configuration space are used for device configuration and setup. These transactions use the configuration register of PCI Express devices. There are several types of configuration transactions: Configuration Read Request, Configuration read Completion, Configuration Write Request, and Configuration Write Completion.

Message Transactions

PCI Express includes this new transaction type to transfer a variety of miscellaneous messages to and from the PCI Express devices. These transactions are responsible for functions like interrupt signalling, error signalling or power management this transaction type is important as these functions are no longer available through sideband signals such as PME, SERR etc.

2.4.6 Architecture Build Layers

The specification defines three different layers that form a PCI Express transaction. These are:

- **Transaction Layer:** The main functionality of this layer is to start the process of forwarding requests or completion data from the device core into a PCI Express transaction.
- **Data Link Layer:** The main functionality of this layer is to make sure that the transaction going back and forth through the link is received correctly.
- **Physical Layer:** This layer is responsible for the real transmitting and receiving of the transaction through the PCI Express link.

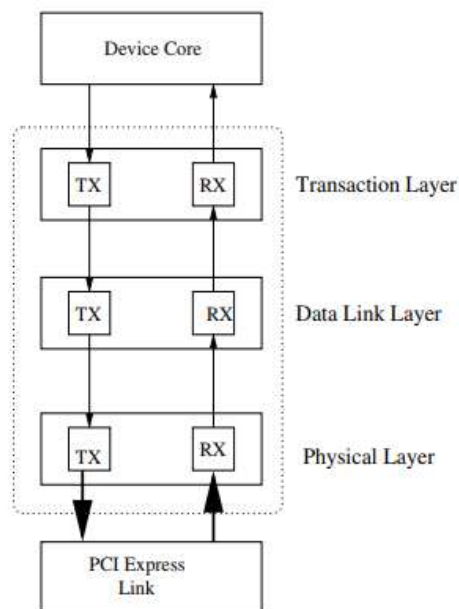


Figure2.7: The Three Architecture Build Layers

CHAPTER 3

XILINX VIVADO TOOL

3.1 XILINX VIVADO

Vivado Design Suite is a software package made by Xilinx for synthesis and analysis of HDL designs. The Vivado System Edition consists of an in-built logic simulator and also performs high-level synthesis, with a tool chain that converts C language code into programmable logic.

3.1.1 FEATURES

Vivado allows developers to simulate their designs, perform clock analysis, examine RTL diagrams, simulate a design's response to different stimuli, and configure the target with the programmer. The Vivado High-Level Synthesis compiler enables C, C++ and SystemC programs to be automatically targeted into Xilinx devices without the requirement to manually create RTL.

The Vivado Simulator is a feature of the Vivado Design Suite. It is a compiled-language simulator that works perfectly with mixed-language, TCL scripts and enhanced verification.

The Vivado IP Integrator allows engineers to easily integrate and configure IP from the giant Xilinx IP library.

The Vivado TCL Store is a system for developing add-ons to Vivado, and can be used to add and to modify its capabilities.

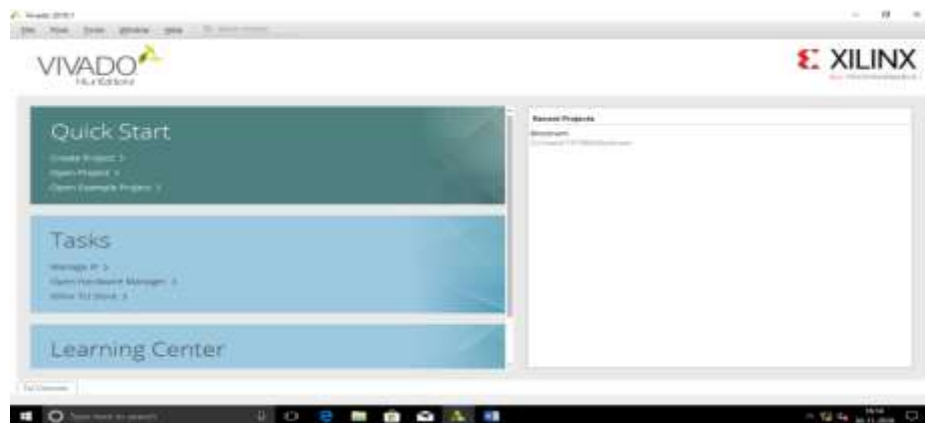


Figure3.1: Vivado Home Page.

CHAPTER 4

EXPERIMENTS AND RESULTS

After complete study about the various aspects of PCIe we simulated the PCI express design block using Xilinx Vivado. For the simulation purpose we used Kintex Ultrascale FPGA. Fig.9.1 shows the design block and Fig.9.2 shows the simulated result.



Figure4.1: PCIe Design Block

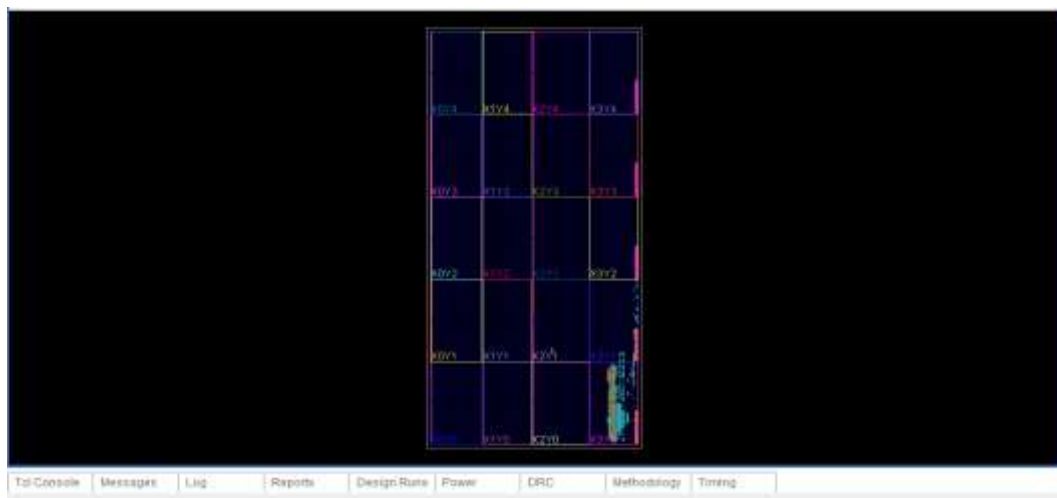


Figure4.2: PCIe Chip Mapping

Now that we have implemented the basic PCIe model we tried to get different results by changing the number of lanes in our PCIe model keeping rest of the factors same to get the efficiency and power utilization values.

We performed the simulation for X1, X4 and X8 lanes and the results are shown in the following figures:

For X1 Lanes:

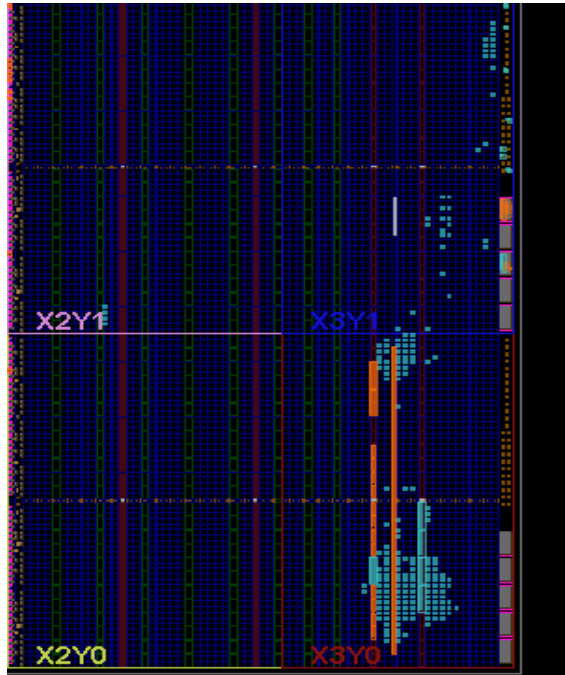


Figure4.3: Chip Mapping for X1 lanes

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	1.304 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.9°C
Thermal Margin:	73.1°C (50.1 W)
Effective θ_{JA} :	1.4°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

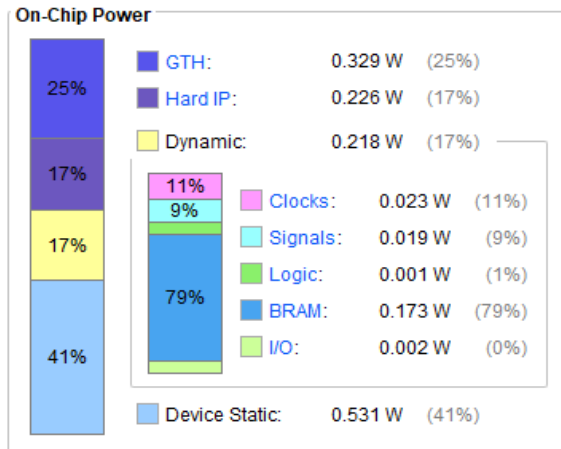


Figure4.4: Power Analysis for X1 lanes

For X4 Lanes:

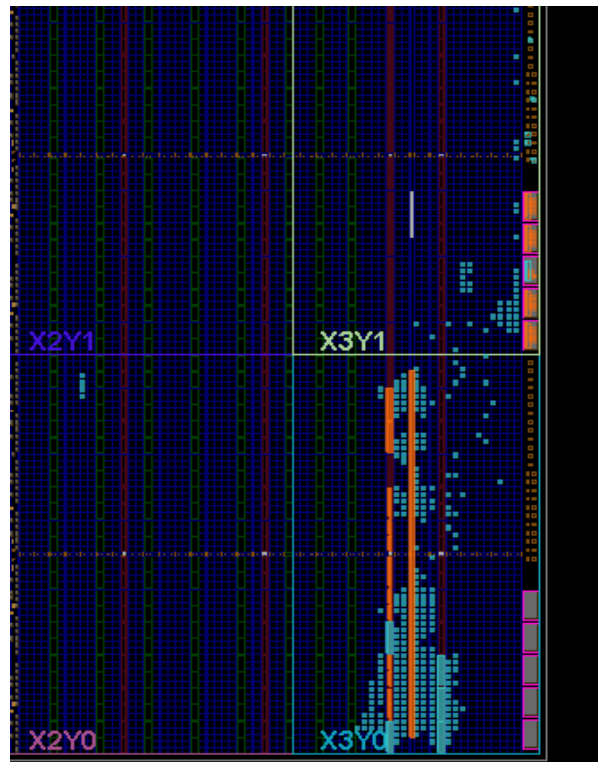


Figure4.5: Chip Mapping for X4 lanes

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	2.16 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	28.1°C
Thermal Margin:	71.9°C (49.3 W)
Effective θ_{JA} :	1.4°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

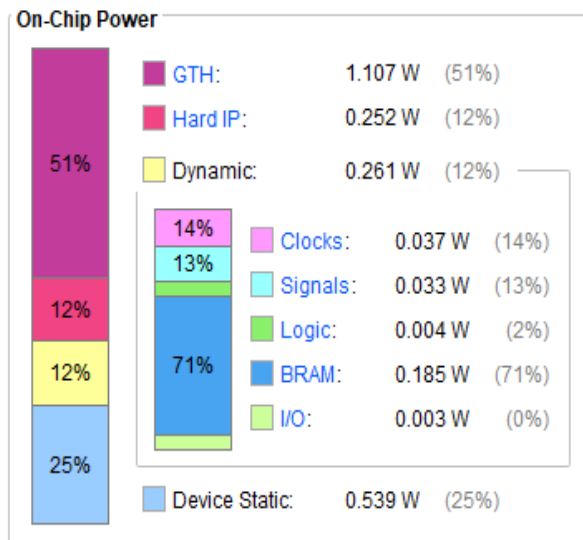


Figure4.6: Power Analysis for X4 lanes

For X8 Lanes:

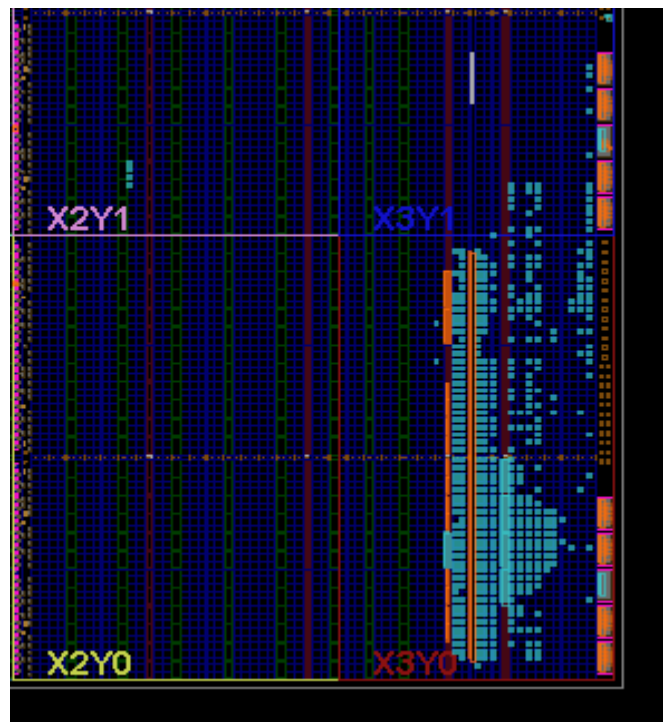


Figure4.7: Chip Mapping for X8 lanes

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	3.352 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	29.8°C
Thermal Margin:	70.2°C (48.1 W)
Effective θ_{JA} :	1.4°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

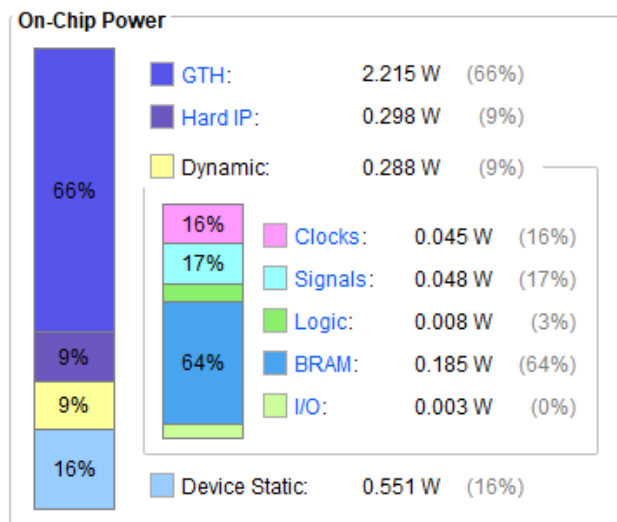


Figure4.8: Power Analysis for X8 lanes

When we analysis power consumption for X1,X4,X8 lanes then we conclude that X1 lane based result consume more power and is less efficient and X8 lane based result consume least power and is the most power efficient amongst the three. We observe that more the GTH value more is the efficiency and lesser is the power consumption.

Moving forward, we analyzed power consumption for three different FPGA architecture that is Kintex-7, Zynq 7000, Kintex Ultrascale and the results for the same are shown in the following figures:

For Zynq 7000:

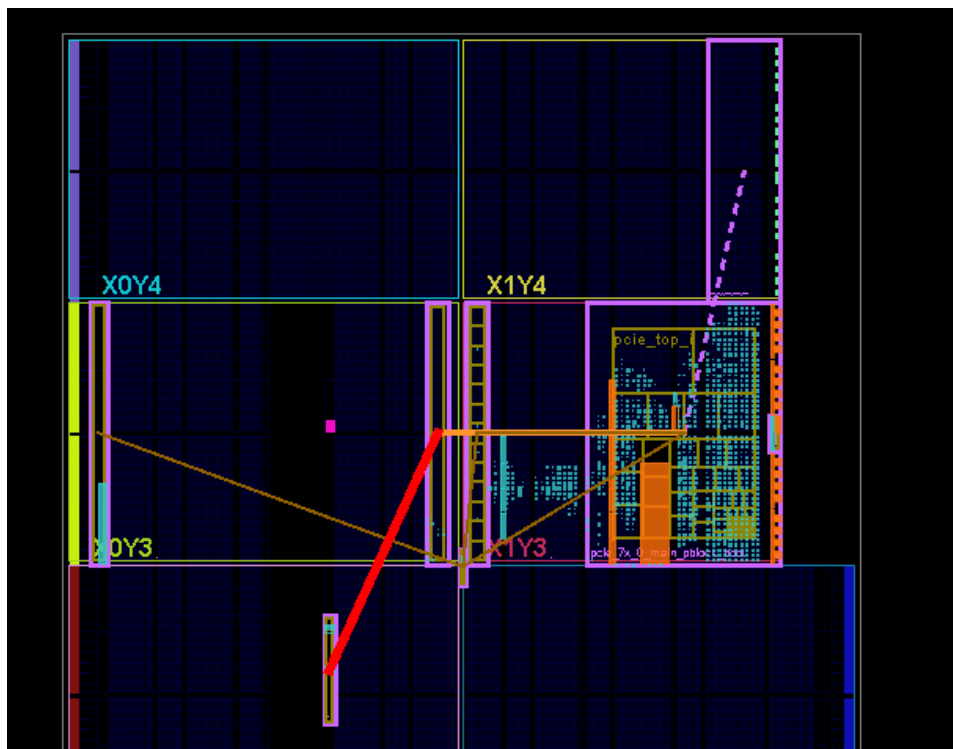


Figure 4.9: Chip Mapping of Zynq 7000

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 1.645 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 29.1°C
 Thermal Margin: 55.9°C (22.2 W)
 Effective θ_{JA} : 2.5°C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

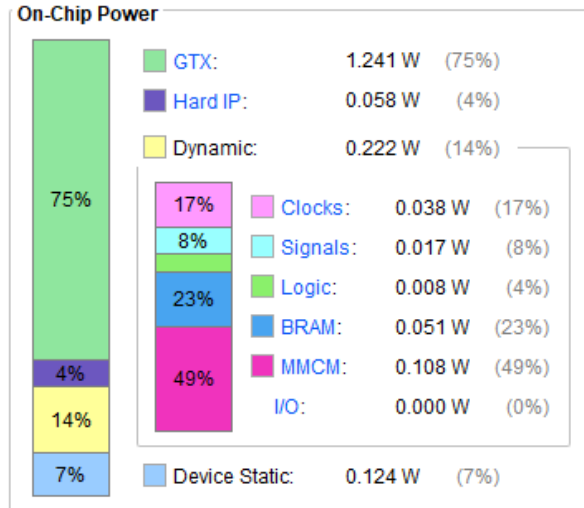


Figure4.10: Power Analysis for Zynq

Total power on-chip dissipation by PCI Express on the Zynq 7000 is 0.222 Watt and Temperature at the junction i.e. junction temp is 29.1 Celsius. The contribution of Dynamic power is 14% in the total power as shown in Fig. 4.10.

For Kintex-7:

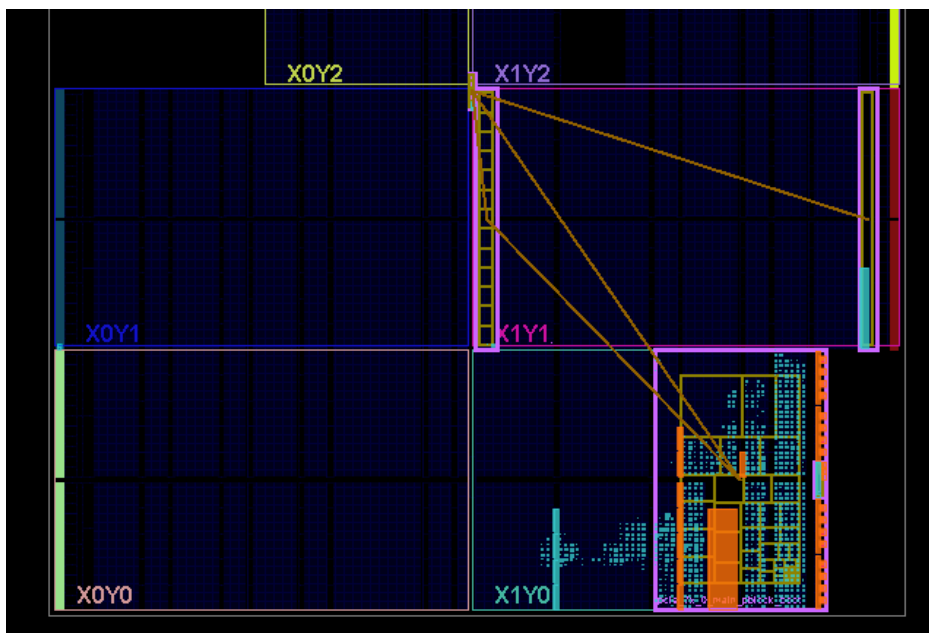


Figure 4.11: Chip Mapping for Kintex-7

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 1.655 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 29.3°C
 Thermal Margin: 70.7°C (26.6 W)
 Effective θ_{JA} : 2.6°C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

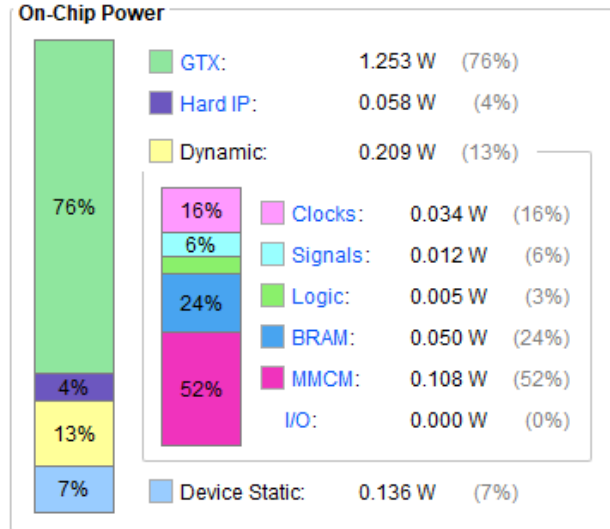


Figure 4.12: Power Analysis for Kintex-7

Total power on-chip dissipation by PCI Express on Kintex7 is 0.209 Watt and Temperature at the junction i.e. junction temp is 29.3 Celsius. The contribution of Dynamic power is 13% in total power dissipation as shown in Fig. 4.12.

For Kintex Ultrascale:

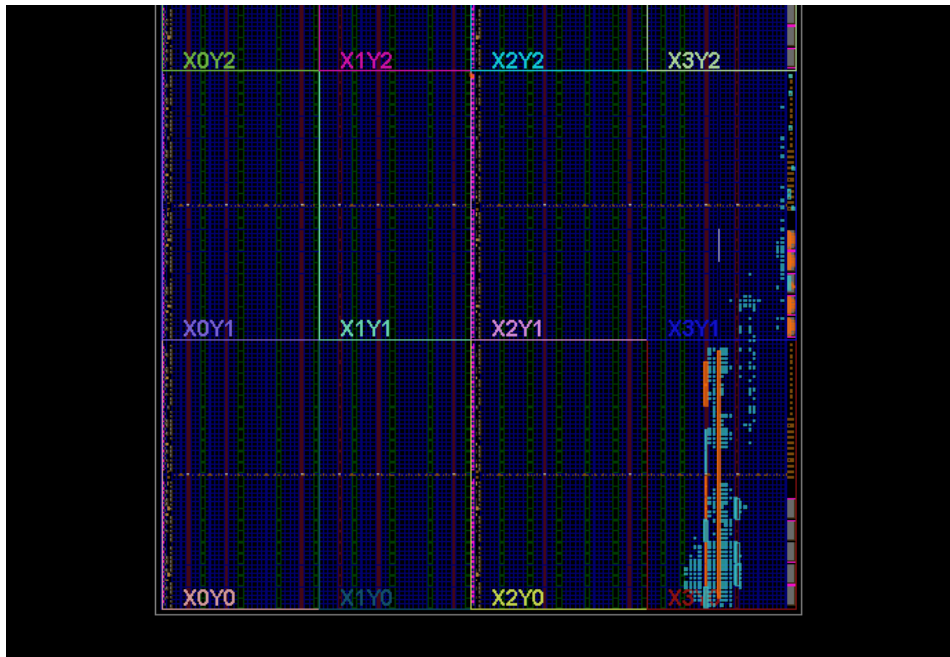


Figure 4.13: Chip Mapping for Kintex Ultrascale

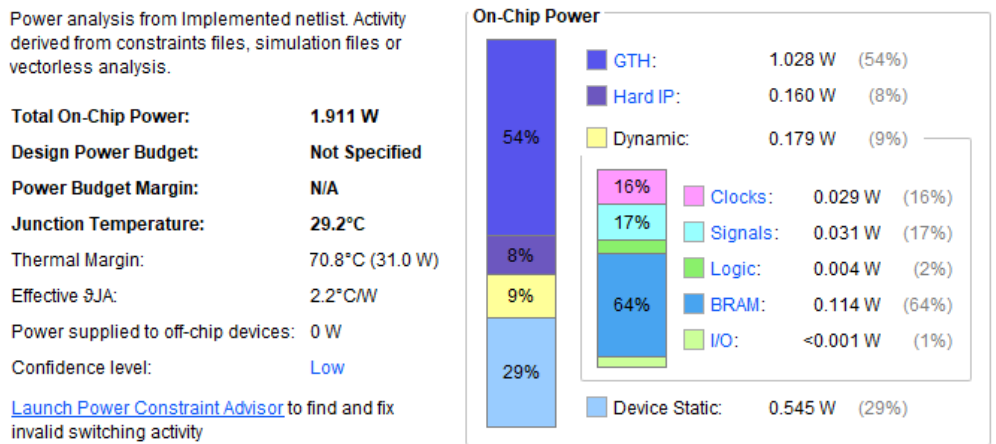


Figure 4.14: Power Analysis for Kintex Ultrascale

Total power on-chip dissipation by PCI Express on Kintex Ultrascale is 0.179 Watt and Temperature at the junction i.e. junction temp 29.2 Celsius. The contribution of Dynamic power is 9% in total power as shown in Fig. 4.14.

On comparing the three architectures we observe that Zynq is the most hungry for power architecture and Kintex Ultrascale is the most power saving architecture.

Performance analysis of PCIe on Zynq 7000: The manufacturing of Zynq 7000 is based on 28nm technology. It has 325BRAM, 202800 FF, 101400 LTU as shown in table 4.1.

Resource	Utilization	Available	Utilization %
LUT	2232	101400	2.20
LUTRAM	28	35000	0.08
FF	2865	202800	1.41
BRAM	12.50	325	3.85
IO	1	285	0.35
GT	4	4	100.00
MMCM	1	8	12.50

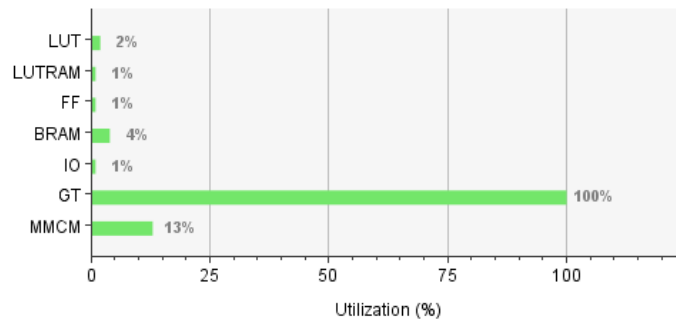


Table4.1: Resource Utilization for zynq 7000

Performance analysis of PCIe on Kintex Ultrascale: The manufacturing of Kintex Ultrascale is based on 20nm technology. It has 540BRAM, 406256 FF, 203128 LTU as shown in table 4.2.

Resource	Utilization	Available	Utilization %
LUT	844	203128	0.42
LUTRAM	4	112800	0.00
FF	1820	406256	0.45
BRAM	13	540	2.41
IO	1	468	0.21
GT	4	8	50.00
BUFG	5	480	1.04

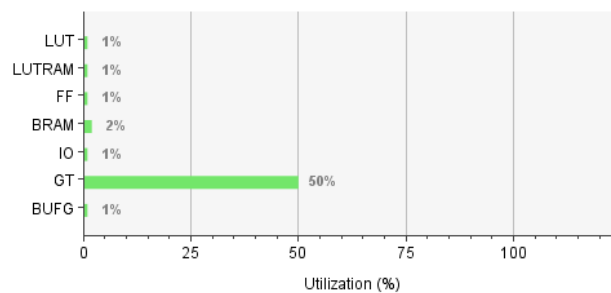


Table4.2: Resource Utilization for Kintex Ultrascale

Our design has a latency of 77-clock cycle, 77-clock cycle, and 55-clock cycle for Zynq 7000, Kintex7, Kintex Ultrascale respectively. Latency value comes out to be average for Zynq 7000 and Kintex7. Latency value of 77-clock cycles infers that it will take 77-clocks to provide the output. The interval of 78-clock cycles means that the next set of the inputs will be read after 78-clocks. For ultrascale architecture, latency value is the least. There is approximately 46.75 % reduction in the latency value when we shift from 28 nm technology based 7 series architecture to 20 nm technology based ultrascale architecture.

FPGA	Worst Case Delay	Latency	Interval
Zynq 7000	8.47	77	78
Kintex7	8.42	77	78
Kintex Ultrascale	8.26	55	56

Table4.3: Delay, Latency and Interval Comparison Table

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

When we analysis power consumption for X1,X4,X8 lanes then we conclude that X1 lane based result consume more power and is less efficient and X8 lane based result consume least power and is the most power efficient amongst the three. We observe that more the GTH value more is the efficiency and lesser is the power consumption.

After analyzing power dissipation for Artix7, Kintex7, Zynq 7000 and Ultrascale FPGA then we can make a conclusion that Zynq 7000 is a very greedy for power architecture whereas Kintex ultrascale architecture is the most power saving architecture. Ultrascale FPGA is also one of the best architecture for processing packets in 100 G networking & heterogeneous wire free infrastructure. Thus, we can conclude that ultrascale FPGA is the best architecture for power saving implementation for any communication design on FPGA. There is 46.75 % reduction in the latency value when we shift from 28 nm technology based 7 series architecture to 20 nm technology based ultrascale architecture. This Reduction in the latency values will increase efficiency of any communication design.

In the future, work could be carried out on newer and improved FPGA technologies. Similar analysis could be carried out for 14 nm, 12 nm and 9 nm technology. A dedicated cost and performance comparison is possible to determine which architecture is best suited and if power consumption and utilization is managed then PCIe technology can be a major breakthrough in mobile phone communication.

REFERENCES

- [1] *Tech-pro.net*. Retrieved 14 November, 2018, from tech-pro.net: http://www.tech-pro.net/intro_pci.html
- [2] fpga4fun. Retrieved November 14, 2018, from fpga4fun: <http://www.fpga4fun.com/PCI1.html>
- [3] *PCI Local Bus Technical Summary*. Retrieved November 16,2018, from techfest: <http://www.techfest.com/hardware/bus/pci.htm>
- [4] *PCI Bus Timing Diagram*. Retrieved November 18,2018, from silverhawk: <http://silverhawk.net/notes/tutorials/hardware/pcitiming.html>
- [5] *Figure Block Diagram of Static RAM Table Truth Table* . Retrieved November 29,2018, from docstoc: <http://www.docstoc.com/docs/4219029/Figure-Block-Diagram-of-Static-RAM-Table-Truth-Table>
- [6] *Electrofriends*. Retrieved November 29,2018, from Electrofriends: <http://electrofriends.com/articles/computer-science/protocol/introduction-to-pci-protocol/1/>
- [7] Bishwajeet Pandey, Bhagwan Das, Amanpreet Kaur, Tanesh Kumar, Abdul Moid Khan, D. M. Akbar Hussain, Geetam Singh Tomar. "Performance Evaluation of FIR Filter After Implementation on Different FPGA and SOC and Its Utilization in Communication and Network", *Wireless Personal Communications*, 2016 Retrieved January 11, 2019.
- [8] M. Aguilar, A. Veloz, M. Guzman. "Proposal of implementation of the "data link layer" of PCI express", (ICEEE). 1st International Conference on Electrical and Electronics Engineering, 2004., 2004 Retrieved February 14,2019.

[9] Muhammad Alhammami, Ooi Chee Pun, Tan Wooi Haw. "Hardware/software co-design for accelerating human action recognition", 2015 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (CSUDET), 2015 Retrieved February 24, 2019.

[10] *Techfest*. Retrieved March 20, 2019 from Techfest.com:
www.techfest.com/pcie/architecture/protocol/2/