Advanced framework for Displaying Power, Performance and Thermal Data for Intel CPU DFX

Submitted By

Patel Hetashvi Anilbhai

17MCEI09

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2019

Advanced framework for Displaying Power, Performance and Thermal Data for Intel CPU DFX

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (INS)

Submitted By

Patel Hetashvi Anilbhai

(17MCEI09)

Internal Guide

Prof. Kruti Lavingia

Guide & Associate Professor,

Institute of Technology,

Nirma University, Ahmedabad.

External Guide

Mr. Raghav Rao

Engineering Manager,

Intel Technology Pvt Limited,

Bengaluru, Karnataka.

May 2019

Certificate

This is to certify that the major project entitled **"Advanced framework for Displaying Power, Performance and Thermal Data for Intel CPU DFX"** submitted by **Patel Hetashvi (Roll.No. 17MCEI09)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Specialization in title case, if applicable) of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Kruti Lavingia
Guide & Associate Professor,
CSE / IT Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sharada Valiveti
Associate Professor,
Coordinator M.Tech - CSE (Specialization)
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri D Bhavsar
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Patel Hetashvi**, **17MCEI09**, give undertaking that the Major Project entitled "**Advanced framework for Displaying Power, Performance and Thermal Data for Intel CPU DFX**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (INS - Information and Network Security)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. Kruti Lavingia

(Signature of Guide)

Acknowledgements

I would like to express my gratitude towards my Internship Manager and External Guide **Mr. Raghava Rao**, Intel India Lmt. I am also grateful to the Team I was in. It gives me immense pleasure in expressing thanks to my Internal Guide **Prof. Kruti Lavingia**, Associate Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Madhuri D Bhavsar**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

Patel Hetashvi

17MCEI09

Abstract

Computers are gadget in mundane life working pattern. Probably everyone can use computers because integration of hardware and Operating System is already done by the merchant companies. OS helps Application Program to execute itself by providing APIs and libraries needed. Background processes are helping OS for continuous execution. These background processes and application program use CPU. To understand the Inside computer, System architecture is used. Parts of Computer architecture can be modulated in Processor, bus system, memory, Input/Output. All this things might be implemented on one SoC(System on chip). With the help of better integration Framework it is possible to analyze what is going on inside the System and could be possible to configure the system. A framework in computer programming is an abstraction in which common code generic functionality that can be selectively overridden.

This framework is use for configuring IPs in System Architecture. It can be used for enabling and disabling IPs and for analyzing behavior of the IPs. As a first part of the project, the framework is used exploration of GbE(Giga bit Ethernet) Enabling. GbE is PCIe(Peripheral Component Interconnect Express) device which is slot for connection to the SoC and advanced version of PCI slots(Peripheral Component Interconnect). These slots are used for connection of GbE device and enabling it for stress testing, to check the connectivity PCIe Loopback test is used which can be applied on OSI Models Data link layer (Link Layer) and Physical layer (PHY Layer). As this framework is used for enabling device for stress, it is also used for monitoring of that devices effect of processor. And that is the second phase of the project Advanced framework for Displaying Power, Performance and Thermal Data for Intel CPU DFX is about a Windows GUI Program for displaying performance value of Host system with the use of advanced framework. GUI application implemented from this framework work as a Performance Monitoring tool for System architecture.

Abbreviations

PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
GbE	Gigabit Ethernet
LAN	Local Area Network
OS	Operating System
RAM	Random Access Memory
CAR	Chase as RAM
GPIO	General Purpose I/O
MTRR	Memory Type Range Register
MMIO	Memory MApped I/O
IRQ	Memory Mapped I/O Interrupt request

Contents

List of Figures

Chapter 1

Introduction

CPU is the center of the computer architecture; it assigns all tasks and devices needed to it. These devices help each other to complete their task because there might be processes which are codependent on each other. To insert external device to the system there are slots which are known as PCI/PCIe slots that allows attaching more devices to the system for example GAC (graphics adapter cards),NIC (network interface cards), storage accelerator devices (SATA) and high-performance peripheral.

IPs are Interface protocols for devices which can be connect to SoC. It is also known as Design IP because it is not essential part but it dene functional core in the SoC which are special purpose device gateways. It can be divided into 4 types: 1. Analog IP, 2. Interface IP, 3. Memory interface IP and 4. System/Peripheral IP. All PCIe devices falls under Interface IP. PCIe is mostly used because it gives external 4KB conguration space. These devices are not always device but more of a chipsets like graphics card, network interface card etc. These devices can also provide additional feature like graphic card also known as video card include a processing unit, memory, a cooling mechanism and connections to a display device. There is also a device which is which handle data transfer in chipsets more than 10 gigabit, which is known as GbE (Gigabit Ethernet). High speed Ethernet communication is becoming increasingly essential as it enables data intensive applications as part. Protocol used in local area networks (LANs), provides a data rate of s1 billion bits per second (one gigabit).

The effect of the external device on processor is noticeable. Each part of computer consume electricity in mini watts or watts according to its size and task with processing. This can be directly affect the performance of the processor and other devices. It is

also possible to monitor their characteristics and the impact of the device on processors performance. The factor which get directly affected by the power is heat. Which consume the power release some amount of heat with the output, this concept introduce the thermal parameter for the device that needed to be monitor.

1.1 Motivation

Motivation for this project is to utilize the framework in most beneficial way and make a application that can help to enable IP(in Experiment GbE in PCIe) to use it in different purpose like application of security, device validation, stress test enabling, monitoring etc. When something is work on system level the possibility of the configurations are getting broader and applications of it are getting generalized with it. In this project the IPs are enabled with framework and different configurations are done using framework. After that application area could be anything regarding device. Once the enabling process is done it is possible to monitor device IP through desktop application or console application.

1.2 Background

Computers are multifunctional electronic device. It needs power supply to work. All the connected essential parts of a computer need power supply. The power supply controls the currents flowing in computer. It controls the input and output of voltages and alternating lines and their frequency. Computers different components needs different voltage and current for function.

There are two current lines AC (alternating current) and DC (direct current).The AC has higher- frequency than the DC current. If the AC power is used in computers the strength voltage can damage the sensitive components of the computers. So power supply have to convert the voltage to a lower power and frequency. Watt is the measurements of power. The specific number of power (watts) required by a part within the computer. The power supply is located in the back of the computer. It use switcher technology for conversion from AC to DC. The power supply is used for hard drive activation, cooling system for SoC like fans to prevent overheating.

When the power supply it on, the processor is not directly get started. There is step by step process which needed to be follow. When there is process to initialize the PCIe devices, it includes all the necessary step procedure before the OS take control of the application programs. Following steps will describe the process of PCIe device initialization.

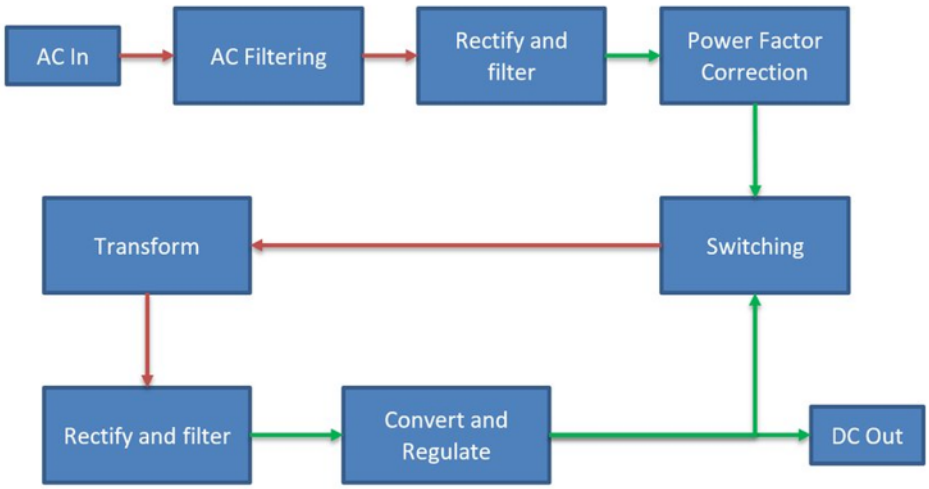

Figure 1.1: Power Supply in System Architecture

There are same process steps are followed to when the system is started. It is not possible to initialize all the devices at the same time. there is step by step process for all the elements of computer. When there is process to initialize the PCIe devices it includes all the necessary step procedure before the OS take control of the application programs. Following steps will describe the process of PCIe device initialization.

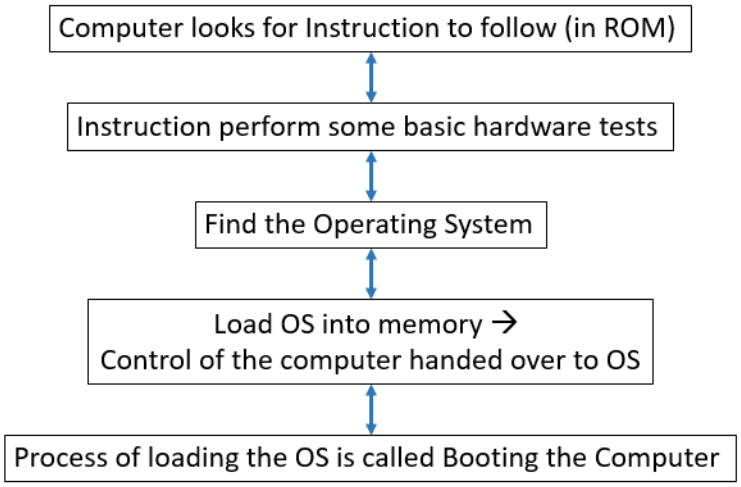

Figure 1.2: OS Initialization process

- There are many devices which are connected to the CPU to complete the computer architecture and these devices are also essential for proper computer operations. Every external device is synchronized with the boot process in the processor. Boot process starts with the platform firmware execution. Firmware execution is important to the OS boot and specification of the boot loader (which load and execute the OS). Before OS execution there is device activation process which wake ups devices connected to the computer architecture. This whole process of OS activation can be described as follow.

- First of all when the power button is pressed the bootstrap processor start execution of the code stored at the address location known as reset vector. This is known as a CPU operating mode initialization. These modes are defined as Safe mode, Voodoo mode, protected mode. Then preparation for the memory initialization took place which start with the CPU microcode updating. After that cache as RAM (CAR) is use for faster execution, it use stack which work only in Read Only mode because it is most sensitive memory type. Tempering in this will never allow the execution of the OS. Chipset register in initialize after that which mapped the address to the system and it is important to memory initialization.

- Second phase is known as memory initialization which is done with memory controller which is a chipset. So memory controller activation and RAM initialization happened in a synchronized way to get support from each other. RAM is tested before initialization to make sure it is not broken. This is the correct setup for the initialization of the memory. After the RAM initialization the flash ROM which is attached to the motherboard is activated which is very slow process, it could be done with the CAR but it has only limited size that is why post memory initialization processes are noticeably slow. These processes contain Shadowing of the firmware into the RAM, redirecting the memory transactions, setting up the stack which is completely different from CAR.

- Third step is depends on the specification of the different systems which enable chip sets on the motherboard, in some platforms this step also consists of initializing the general purpose I/O (GPIO) registers. This step also consists of enabling of the interrupt and timer initialization. After the hardware and software both are in

almost ready to execute step there is requirement of the handler to the interrupt which tell that interrupt registers are enabled. Hardware timer is enable after that are possible used after OS activation. Some of the Os processes and application need timer. There are different purpose timers like high precision event timer for residing chipsets used by only OS, real time clock which is used by in resides in chipset and there is also local APIC in CPU.

- Memory caching controllers are initialized with the help of MTRRs(memory type range registers). Range of memory is define to respective purpose are allocated to it. At these stage all the external device get their space in memory for communication with system. This memory allocation is depend on different kind of devices for example for I/O devices PCI bus in initialized which is needed for communication with different part like memory and CPU. PCI addresses are not cached addresses. Application processor also use MTRRs to mapped it with memory before the OS boot loader took place. MTRR also connect all the CPU cores, if this is not completed properly then system will bring to the halt.

- Simple I/O devices are activated after that. It include legacy IO like key board, mouse, external memories, buttons on laptop etc. PCI device discovery is the next step which detect PCI/PCIe devices in slots. This takes huge part in resource assignment like IO space assignment, MMIO space assignment and IRQ assignment. Memory assignment done with BAR base address registers, USB initialization took place in this manner. There are also non legacy devices like SATA and SPI.

- OS boot loader take place after all these processes. This part was important because it describe the PCI and PCIe device initialization, detection and use in system architecture. After the activation of different IPs, they can communicate with different part of system including CPU or surpassing the CPU after its approval.

Example: In x64/86 there are 4 GB space for system configuration, this include PCI/PCIe configuration space which is divided into 4KB chucks for each devices. Connected GbE device with the computing system will appear as soon as it is recognize by the firmware code for device discovery module.

1.3 Purpose

Aim of this project to create a framework which help to enable IP(in Experiment GbE in PCIe) to use it in different purpose like application of security, device validation, stress test enabling, monitoring etc. When something is work on system level the possibility of the configurations are getting broader and applications of it are getting generalized with it. In this project the IPs are enabled with framework and different configurations are done using framework. After that application area could be anything regarding device.

In this report experiment on GbE controller is descried. The motto is full filed to use framework as much as possible ways and the stress test on the GbE is the purpose. It's not good to apply Stress test on device without checking the port connection of the device. In experiment Loopback is used to check connectivity, after that stress test is applied. Stress test is applied to get the monitoring results from the system under loads of task. Stress test help to set a specific limit for the system. There are different types of stress test for different type of IPs. Monitoring results of stress test could be used in many places which explains significance of this framework. To make monitoring more user friendly the concept of GFX is added. GFX is a Graphical presentation or tool for manufactured product to analyze the entities and properties of the product in a better visualized way. As the power supply is needed to make a device work as a result of it there is performance and thermal parameters are also get affected. GFX makes it easy to analyze this parameters and give values of power, thermal and performance of the processor with external device.

Chapter 2

Project Description and Goal

This chapter is about detailed description of the project and Goal. This project can be divided into main 3 elements which can be First is to understand the Significance of the framework, Second is to understand the IPs, Third is about the PCI technology and Fourth is to make GUI for the application which is the Goal of this project.

2.1 Significance of the Framework

A framework is a basic structure use for software building. It provides the basic foundation for creating the software applications. A framework includes class and functions that can be used for application development. A framework is similar to API (Application Programming Interface). APIs are more like an access elements supported by the framework, but framework may also include libraries, compiler code and other program used in software development process. For different OSs there are different type of framework that are in support for Windows development ActiveX and .NET, Mac OS use Cocoa, Touch for iOS and the Android Application Framework for Android. There is SDK support provide to the frameworks for development compatibility.

Configuration enabling framework will provide the classes and functions to modify the configuration space so that different applications can be built for purposes like testing, securing and controlling system. Native methods and classes are used for low level system communication which is provided by this framework. The purpose of creating framework is to make it more versatile with respect to the different kind of uses. A frame work in this project helps to modify the configuration space. Modification in direct memory

needs native methods from framework and that's why it is useful.

Native code can communicate with higher level languages and return values associated with respective functions called in code. Native codes are basically an interface code between the GUI or console application to the driver or to the Hardware. Native coding is a specific coding for the special purpose communication with special specification for example system like x86.

This framework is use to retrieve and set the variable which are directly associating with hardware. Basically the information what this application displaying is get from the MSRs (Model-Specific Registers) in computer. MSRs are the different control registers in the x86. It is instruction set used for debugging the applications, program execution and tracing computer performance monitoring, and toggling CPU features.

2.2 Applications of the Framework in security

As explained the IPs are generalized term for same purpose devices. So enabling the PCIe devices are open the gate of all kind of device configuration in the system. More devices get add to the system and more applications of the device can be possible with the help of framework.

- System Security from unwanted device access

  Not all the device should be allowed to communicate with the system. If some specificdevice are not grant to enable in system, the framework will allow to stop them from interacting with SoC. For Example the in school and college lab it's not allowed to insert USB drive in the system. It is possible to use the framework for that. USB might not possible to use after that. If the system wants to enable the device again then it is possible to reverse the operation, but the default value must be known.

  For this kind of application device ID vendor ID are matched with the help of framework. After OS execution the discovered devices are matched with the PCIe device repository. If device will matched the specific devices vendor ID and device ID, device will not allowed to communicate with the system.

- Device Identification

  Identification of the device can be possible with the help of framework. First of all there is need to check base memory of PCIe configure space. Identification of device is done with it's Vendor ID(VID) and Device ID(DID). DID and VID is fetched from the memory and data from the PCIe device repository are used to get understandable meaning of the data. There are repositories which are used for update device identification for framework.

  After get the value assembled devices are checked with system and device legitimatized is proved. If device is not registered then error message is raised

- System monitoring

  It is possible to monitor system performance and device performance. This type of system can be easily troubleshoot problem with device connectivity. When it is necessary for device to continuously connected till the task is over, Like transferring file. It is necessary that device stay connected with system. And if device get lost the connection that is easy to troubleshoot and identify the problem. It can be possible to use device performance checking with monitoring.

- Network monitoring

  As one system device connection is easy to check with framework, it is also possible to use framework for the network. It is possible to check which devices are connected with all host among inside of the network. Here, network is assumed to be safe, no external factor from the network can affects the system. If device is missing or external device are added that can be identify over entire network.

  To make application like this, there is need of creating log of device with each system. It is possible to keep log of connected device and check every time with previous log. That will give the missing or newly connected device with system.

  A very good place where there is need of these kind of application is in data centers. Data centers are huge and with many device interconnected with each other. If some external device is added to the network or because of some reason a device

is lost it's connection. It is hard to track the problem. In this case a framework application can be very helpful.

- Device(Hardware) security

  As we have discussed that it is possible to stress the device with required testing and measure the performance of the system. Normally the device performance are constantly monitor to get the threshold value where the device is beak and become unreliable to use. after getting the values it is possible to set a limit and stop device from bursting(stop working). For setting this limit the framework is used.

2.3 Understanding of IPs (Intellectual Properties) in SoC

IP stands for Intellectual Property. It can be consider as a gateway for the processor at SoC. In digital world, IP means a module in hardware or software form for the SoC. IPs are not the stand alone Application/Hardware, they are always associated with SoC to increase the functionality of the SoC. SoC can work without IPs but IPs cant work alone on their own. For example USB controller, Audio Codec, Video Cards, AD/DA converter, PCI controller, SDRAM Controller, etc. IPs in computer Soc (System on chip) is performance monitoring, and toggling CPU features. Ips are also known as reusable methodologies for SoC.

IPs are divided into two types. 1. Hard IPs and Soft IPs. Hard IPs are basically a hardware chip or chip-set. Soft IPs are the piece of code. Soft IP's real behavior depends on the physical implementation. Soft IPs are flexibility and portability.Hard IPs are optimized for a specific application, depending on a target technology. which makes their behaviour predictable.

IPs are divided into 4 parts:

1. Analog IP This type of IP contain device port for analog-to-digital converter (ADC), digital-to-analog converter (DAC), and analog front-end (AFE) IP solutions, as well as timing and monitoring IP. This devices need digital conversion of the current passing through. An ADC converts an input analog voltage or current to a digital number

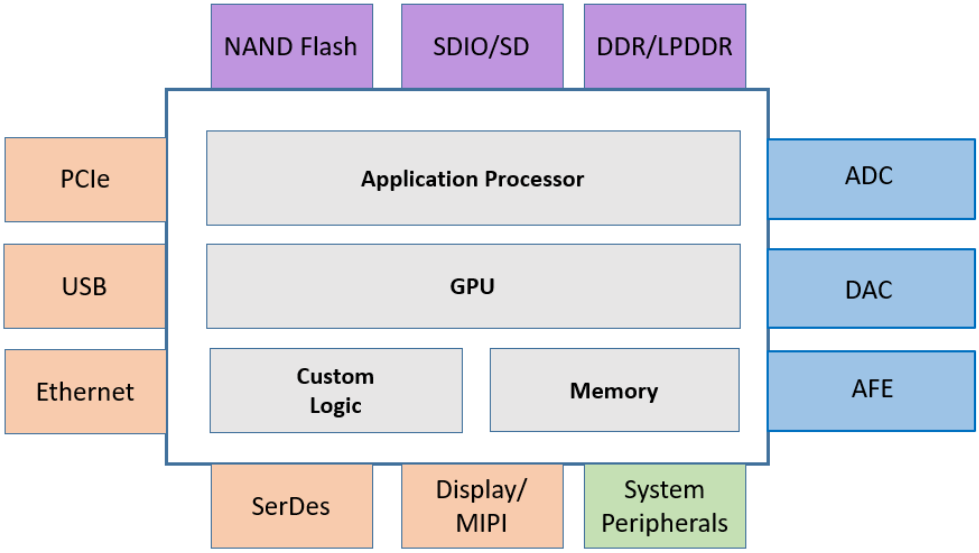

Figure 2.1: Different IP devices on SoC

representing the magnitude of the voltage or current, It is needed in computer because it works on AC power. IT also converts analog signal to digital which are used in cameras and microphones where light or sound converted into digital signals. A digital-to-analog converter is a circuit that converts digital data into an analog signal (current or voltage). DAC is used in Audio Amplifier, Video Encoder, Display Electronics and Data Acquisition Systems. AFE is used in oscilloscope meter for getting readings from the attached ICs.

All the information related this devices are flow through this ports. Analog IP is provided as hard IP. They are fixed in size, tied to a particular foundry, process and are often difficult to integrate. The specific characteristics of analog design make the development of flexible analog IP modules for reuse a much more difficult task.

2. Memory interface This interface have the interfacing for only storage devices. It hast faster interaction and efficient bus system because it is flowing data. Intellectual Property (IP) provides the important logic required to efficiently manage memory‘ devices. It gives interface for following devices latest DDR, LPDDR, NAND Flash, SPI and SD/SDIO standards. There are different IPs for different kind of memories for example for NAND based memory flash or NOR based memory flash. Soft PHY enables the highest speed clock rates 4 times (4x) evaporates which makes faster memory device interaction and transfer. For devices like SD and SDIO it is necessary to be a fast and less power consuming architecture to support the smallest devices. It’s also used in IoT

devices and smart devices.

3. Systems/ Peripherals IP

This IP is essential because without this IP SoC with processor wont work. This IP gives interfacing for processor. This IP gives interface for Audio controllers, microprocessors and system bus peripherals.

Microprocessor IPs helps the communication between SoC parts and processor's communication faster which eventually makes faster processor. They are fast, configurable, compact and compatible. System Bus Peripherals provide important timing and serial connectivity interfaces for design which is essential for every SoC. The only complexity in this IP is to manage compatibility with processor. It is depend on configuration of processor and it can be very from one architecture to another architecture. Audio controllers are very popular application. It support hundreds of devices. Because of a large application device domain the IP of audio controller should be as flexible to adjust the system application. They have to support Single Channel and Multi-Channel variants.

4. Interface IP

This IP includes following interfaces for SoCs Ethernet, MIPI, PCI Express, USB, and IP for Display Port solutions. There are verity of devices in this category that's why it need highly configurable architecture. Because of so many different interfaces IP helps to manage them. Interface IP is available in both hardware IP and Soft IP according to it's application.

Application of this IP for Ethernet is used in almost all the fields. For example automotive, industrial controller, backplane, and datacenters. Other application like Display port provide the High Definition (HD) Display support for mobile and consumer applications Whether you are working with Display Port or Embedded Display Port. For USBs there are need of certified controller which is combination of wide range of product(H/W) like SSD for mobiles and IoT devices and end users. For USB only it is categorized in many groups according to its controller type (Host device, server device, dual mode or OTG) and its specifications (USB 3.0,USB 2.0 and USB 1.1). PCIe type IP helps to enable Gigabit of data in the SoC which is not always available with normal SoC. That is the extra feature added to deal with large number of bits at a time because the normal

SoC wont allow to process large number of data at a time. Which helps in heterogeneous computing to deal with the data. SerDes is helping for serialization and deserialization in the computer. It is used in high speed communication over net, it helps to make communication speedy. Both the functions are execute as a pair of function in single line which decrease the I/O pin in device. Affecting factor on Ser/Des is clocking, parallel processing and serial processing. MIPI is a display interface for mobile devices. For example cell phones, tablets. It is also known as a serial interface for the display.

2.4 Basics of PCIe (Peripheral Component Interconnect express)

PCIe (peripheral component interconnect express) are interfaces for connecting high-speed device components. There are more than onle PLIe slots in PC motherboard that are used for GPU like video card also known as graphics cards, add on SSD cards, NICs and WiFi cards, RAID cards etc. PCIe slots take power supply from the PC mother board. It allows the hot plugging enabling and disabling feature. If it is enable it is possible to directly pick or insert device to the SoC. If hot plugging is disabled it shout downs the system and give blue screen errors in windows.

There are variations in physical configuration of PCIe slots. It come five size x1, x4, x8, x16, x32. The number after x tells how many lans are supported by that PCIe slot and how data will flow from or flow to. For example PCIe x1 can transfer data of one bit per cycle, and x4 will transfer 4 bits of data per cycle from 4 lanes and so on.

PCIe slots auto adjust power supply for the different device. For example a x1 card can sit on a x64 PXIe slot but it will need less power as well as less bandwidth. Similarly a x16 bit device ca sit in x8 PCIe slot will work in half of its bandwidth and necessary power it needs to work. The divides which need more bandwidth like video cards sits in higher configured PCIe slots like x32, x64. Devices which don need higher bandwidth sits in rest of the slots. There are further details about PCIe in the next Chapter.

2.5 Building Rich and interactive user interfaces

The UI module in this project is prepared in Windows Form application using C sharp (c) language. Windows Forms is a Graphical User Interface (GUI) class library which is

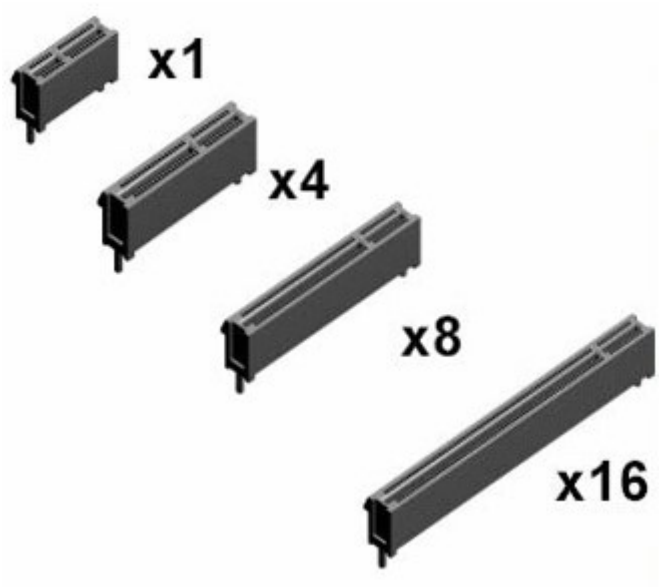

Figure 2.2: PCIe slots (physical appearance)

in .Net Framework. GUI is for making human interaction with application more easily. Which is work on PCs, Laptop and tablets. It is also known as the WinForms. The applications developed by using Windows Forms are known as the Windows Forms Applications that runs on the desktop computer. WinForms can be used only to develop the Windows Forms Applications not web applications. WinForms applications can contain the different type of controls like labels, list boxes, tooltip which gives more input options for user and give better intractable environment.

The work flow of the GUI is given if following diagram. Where user can interact with UI and the following steps for the user are totally hidden and done by windows OS with the use of windows API calls. with the help of windows APIs the UI generation is very easy because it provides all the needed elements of the UI already developed and available for developers use. The second step of the UI development is totally depend on the infrastructure of the application. This block contain the logistics need for the applications and the use of the UI element object. These objects help to make UI more interactive and user friendly. The application core is about the APIs, libraries and frameworks use to make the application and helps to target an aim. Application core files are used for handling the calls of objects and classes from GUI elements and they return the appropriate values with it.

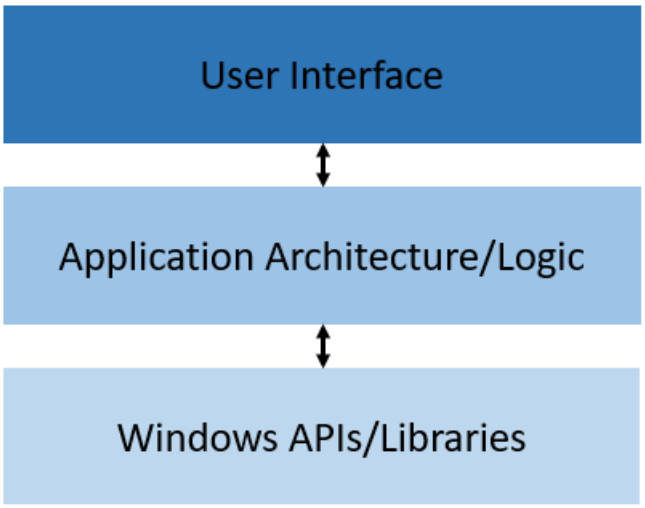

Figure 2.3: Windows form application Layers

Chapter 3

Technical Specification

There are different methods for PCIe configurations. Most of the time PCIe devices are configure by the system itself with the use of firmware. It is done by the booting process.[11] To enable special services, there are fields(group of bits) which can be set or reset the values by overwriting fields. This framework will help to set device's field values. This values could be different from the default values set by the firmware.[11][12]

3.1 History of PCI/PCIe

There is certain information IPs need to collect from the device connected to it for IP compiler like, Version, Release Date, Ordering Codes, Product ID, Vendor ID this is described in following subsection.[6][7][8]

The PCI/PCI Express (PCIe) bus are the part of the computer architecture. PCI bus structure is a multi-lane I/O interconnection to the providing LPC (low pin count), reliable and fast data transfer at rates of up to 10 Gbps per lane per direction, for serial links on back planes and printed wiring boards. It is a 3GIO (3rd Generation I/O) and PCI bus that is designed to be used as a general-purpose serial I/O interconnect, super I/O, generic-application memory. That include desktop, keyboard, mobile, mouse, server, storage (nonvolatile BIOS memory, firmware hubs) and embedded devices/ controllers. GbE (Gigabit Ethernet) is part of these connection between the chipset and architecture. It has capability allow the transactions of data up to 10/100/1000 Gbps.[6]

In past and current there are several PCI versions are existing. PCI is the first version then it is replaced by the PCI X and currently mostly PCI Express (PCIe) is used. There

are also different link width for PCIe like x1,x2,x4,x8,12,x16,x32 these include PCI 33 MHz bus, PCI 66 MHz bus, PCI-X 66 MHz/133 MHz buses, PCI-X 266/533 MHz buses and finally PCI Express.[14]

There are different types of bus according to its size (in bit).Table shows the different version of PCI.

Bus Type	**Clock Frequency**	**Peak Bandwidth**	**Card Slots per Bus**
PCI 32-bit	33 MHz	133 MBps	4-5
PCI 32-bit	66 MHz	266 MBps	1-2
PCI-X 32-bit	66 MHz	266 MBps	5
PCI-X 32-bit	133 MHz	533 MBps	1-2
PCI-X 32-bit	266/533 MHz	1066 MBps	1
PCI Express 32-bit	600 MHz	2131 MBps	1

Table 3.1: Different version of PCI

All this bandwidth will be doubled if the bus type is 64 bit.[4] Following table shows the fundamental different between PCI,PCI-X and PCIe. PCIe structure of architecture in figure 1 is showing components connected to it. There could be more than one CPU. Chipset is the connection component which differentiate connection between Memory, I/O devices, Legacy end, Bridge device and End point devices because each has different purpose to serve. Chipset can also contain part of PCIe called root complex. [4]

PCI	**PCI-X**	**PCIe**
Half duplex	Half duplex	Full duplex
Longer slot size	Longer slot size	smaller slot size
Parallel interface	Parallel interface	Serial interface
Slower	Slower	Faster
Unidirectional	Unidirectional	Bidirectional

Table 3.2: Comparison between PCI,PCI-X,PXIe

Following is the image of PCIe topology which shows how each component are connected to each other which have different components connected to the SoC though PCIe Switch. The main part of the topology is Root complex, Switch and Endpoint Device.

PCIe root complex

The root complex is act as northbridge in PCI system. It works as the connection logic to connect the PCIe device tree to main memory and the CPU. The root complex provide

high speed PCIe connection. Especially to the GUP connection. It is the connection of the two physical chips for the chipset. In current version of the root complex is integrated into the CPU chip, it is the hostbridge integrated into the CPU. The root complex connects to the PCIe device tree through root port.[17][18] The root port is implemented physically in a chip, the root complex. That's why it is known as logical port. The Intel 8-series CPU PCH implements root port arrangement. The root complex can have more than one root port.

PCIe switch

A PCIe switch connects two or more PCIe links. A switch contains connected virtual PCI-to-PCI bridges internally. To understand the PCIe device tree, it is necessary to understand PCI-to-PCI bridge topology. The root complex contains a switch. [13][17]

PCIe endpoint device

A PCIe endpoint device is a terminates the PCIe topology. There is one connection to the PCIe tree topology, it can have connection to another bus. Most of the times, A PCIe network card is an endpoint device, just like PCIe storage controller, etc. [13][16] PCIe endpoint device also act as a bridge to legacy bus just like a PCIe-to-PCI bridge or a bridge to a low pin count (LPC) bus.[12][9]

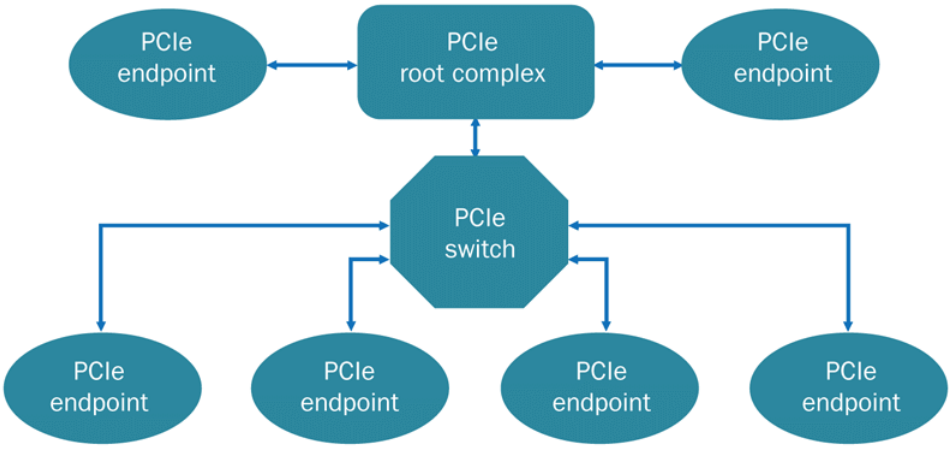

Figure 3.1: PCIe topology

3.2 Parameterize the IP for PCI Express

IP compiler starts with setting the following parameters: PCIe Core Type, PHY type, PHY interface, Configure transceiver block Lanes, Application interface, Port type, PCI

Express version, Application clock, Max rate, Test out width, CHIP reconfig.To enable all these values there have to be some values in PCIe register.

As it is mention earlier that PCI have less space then PCIe, the both structure are same structure but different memory size 256B and 4KB respectively. In this form the device information is stored in allocated device space. [10][7]

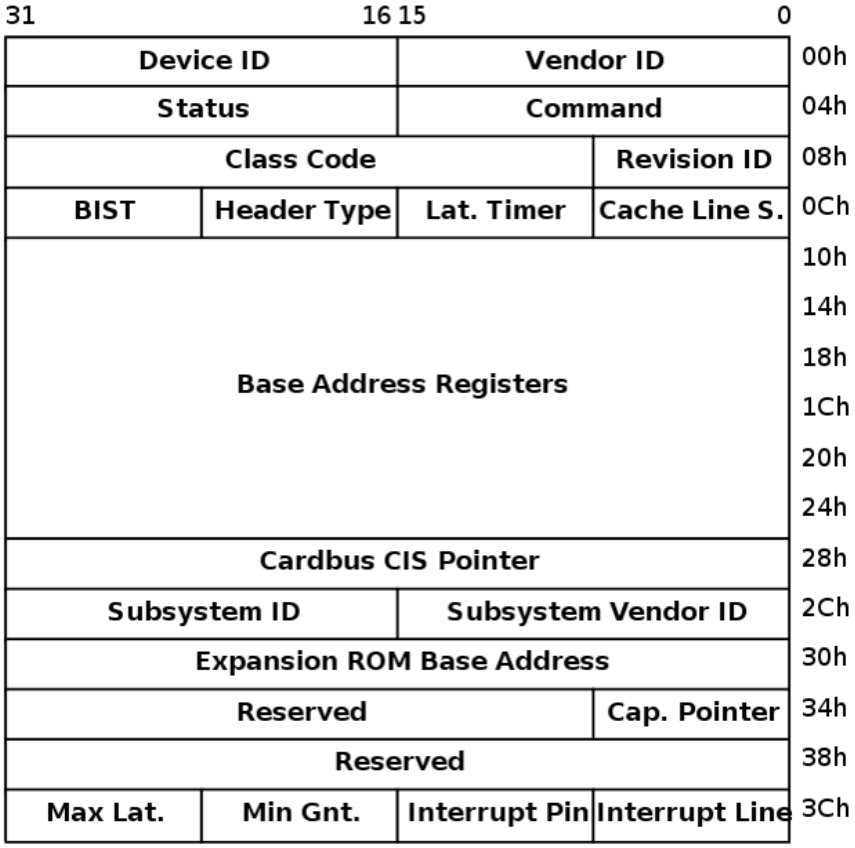

Figure 3.2: PCI Configuration Space Structure

3.3 Transaction in PCIe

PCIe followers the layered structure for the packet transmission. It is almost same with standard OSI model but for only PCIe device and indlude only three layers.[3]

Transaction Layer:

The upper Layer is Transaction Layer. Its task is to assemble disassemble Packets known as Transaction Layer packet (TLPs). Transactions are about read and write, as well as certain types of events. The packet format supports different forms of addressing depending Memory, I/O, Configuration, and Message.

Data Link Layer:

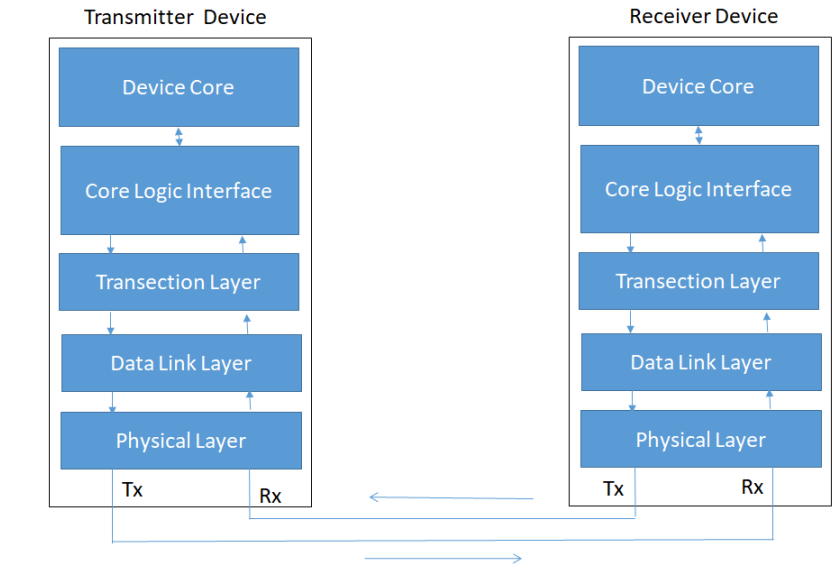

Figure 3.3: Packet Transaction in PCIe

Serves as an intermediate stage between the Transaction Layer and the Physical Layer. It manages the data and check integrity, including error detection and error correction. The transmission side of the Data Link Layer accepts TLPs assembled by the Transaction Layer and submits them to Physical Layer for transmission the reverse process is followed for receiver side.

Physical Layer:

The Physical Layer has interface operation like driver and input buffers, parallel-to-serial and serial-to-parallel conversion, PLL(s), and impedance matching circuitry. The Physical Layer exchanges information with the Data Link Layer. This Layer is responsible for converting data received into an appropriate serialized format.

The PCIestructure has hooks to support future enhancements through speed upgrades and advanced encoding techniques. The future speeds, encoding techniques or media may only impact the Physical Layer definition.

3.4 Address space

TLP transactions use PCIe addresses. There is a mapping requirement between a PCIe address and a local internal bus address and to accommodate this address mapping, built-in hardware address translation mechanisms exist. There are four types of address spaces 1.Memory, 2.I/O, 3.Configuration, and 4.Message.[2]

I/O addressing is used for legacy device operation capabilities. The device address space is divided into two spaces Range 0 and Range 1. Range 0 is used for PCIe configuration task is referred to as Configuration space, and Range 1 is used for accessing memory (non-configuration related) is referred to as Memory space.

So according to this if some 32 bit is in range 0 will indicate an address for configuration space specific purpose depend on the device and Range 1 will indicate accessing directly to the memory.

3.5 Software structure of application

The architecture of the application can be shown as the flow chart given below. The user interface is interacting with the Windows API for GUI component of the application which is made in Windows Form Application. The application is returning values available in MSRs with the framework which is using Intel specific Device driver interface. [19][20]

Drivers are use to interact with the hardware component connected with the SoC architecture. In this project that hardware components are MSRs (model-specific registers). [19][20] Each time when the value is return from MSR it is interacting through the UI and goes down to the driver interface and hardware MSR and returns the values to the UI.

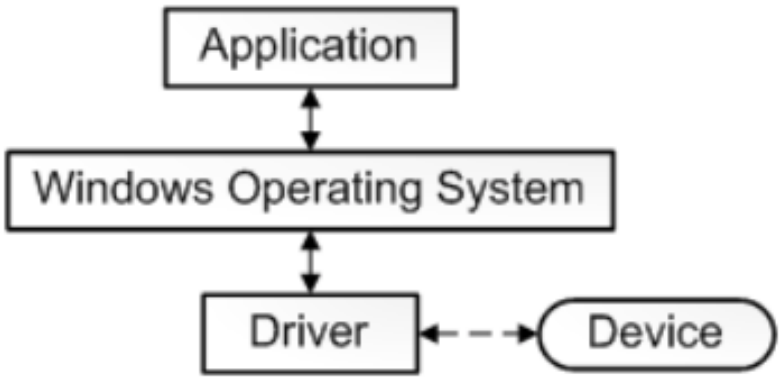

Figure 3.4: Accessing Driver

3.6 MSRs in Details

MSRs are control registers in x86 instruction set. The purpose this registers is to provide helping feature of OS in the execution of the different applications. It is used in application execution, debugging and tracing which tells the problem where the application is stopped. It also give performance monitoring and toggling for the processor.[20][21] It starts with the two test registers introduce in 80386 which are known as T6 and T7 that are used to allow the speed up the processors virtual-to-physical address conversions. After that another three registers are introduce in the the updated version of 80386 to enable test on chase of the processor and code (or data). After that these registers were not implemented in Pentium processor.

In Pentium processor two new feature are added to make the current access of the MSRs easier and faster. They are known as pair of instruction rdmsr and wemsr. Many of these registers proved usefulness of it. Intel has classified these as architectural MSRs their inclusion in future product lines.

rdmsr and wrmsr instructions are used in reading and writing of the MSRs. They are not accessible is user mode there are access privileges that allow only Operating System to execute this instructions. MSRs are Memory Type Range Registers (MTRRs) and Address Range Registers (ARRs). Following subsection describe the two most important operation possible on MSR register. [20][21]

3.6.1 rdmsr Read from Model Specific Register

It is used to read 64-bit model specific registers (MSRs) in ECX register into EAX register. The 32-bit information is loaded ton to the EAX registers. Only 64-bit or less values are read buy the EAX because of its size. They are read in block of two 32-bit of the register. The higher 32 bits are copied in EDX and the lower 32 bits are copied in EAX. If the values are less than the values cannot be implemented and location cannot be defined. The execution of this instruction requires higher level of privilege level.[23] Changing the values of a reserved or unimplemented MSR address in ECX will also cause a general protection exception. As MSRs are used for execution of the programs the reading operation can be used to test, execution tracing, performance monitoring and error checking of the programs. It gives values and address of the errors. The CPUID

instruction is used to determine the MSRs are supported before using the instruction. If the execution of the CPUID instruction indicate that MSRs are not supported then the rdmsr will not work in that processor.[23][24]

3.6.2 wrmsr Write to Model Specific Register

writes in the MSRs with the use of EDX:EAX registers. The instruction is writing in two 32-bit of the register. The higher 32 bits are copied from EDX and the lower 32 bits are copied from EAX. There is reserved bit or already written bits in MSRs are load somewhere else before write. This instruction executed at privilege level 0 or in real-address mode. Or a general protection exception will stop the execution of the application.[23][24] The CPUID instruction is used to determine the MSRs are supported before using the instruction. If the execution of the CPUID instruction indicate that MSRs are not supported then the wrmsr will not work in that processor.[23]

A Computer system can measure and monitor various parameters that contribute to the performance of the processor. These parameters can be used for compiler and memory system tuning. That also include measurement of data or instruction cache hit rates and time spent waiting for the external bus, external bus can connect the system to the connected device. An instruction scheduling algorithms can also measure address generation interlocks and parallelism. The performance monitoring features that are provided by the Pentium processor(Intel's processor micro architecture) are model specific and available only to privileged software. The Pentium processor provides an architectural Time Stamp Counter that is available to the user. The performance monitor features and the events are implementation dependent and consequently. That's why they are not considered part of the Pentium processor architecture.[21][22][23]

3.7 System Parameters displayed in DFX

DFX is Design for Excellence where the X is a variable which can have one of many possible values. DFX is a design for all attribute for the manufacturability, assembly, Inspection of the product. In this project DFX is considered as a Monitoring tool for

CPU performance which can display parameters which are depend on CPU performance. For example parameters like Power, Performance and Thermal. There are more than one components which shows the fluctuation of power, thermal and performance while something affecting things are occurs to the system. Following figure sows the system architecture. It is a microprocessor architecture of Sandy Bridge. Which is shows the different component connection between SoC and external device. This figure is for a reference that how a real System have so many external device and how they interact with each other through buses and process information to fulfill their purpose.
Following is the list of parameters which are needed to be display on GUI. These parameters are also known as system parameters. This parameters are display different Power, thermal and performance values according to different circumstances.

Core:

Small instruction processing cores inside of the processors. Core is a processing unit which reads in instructions to perform specific actions.[26] A smallest unit of instruction known as a task. Cores are the located in processors to process instruction and give appropriate output. Actions are instructions chained together at real time. In Intels processors there are more than one core in the processors. Working of Core is stares with the following basic instruction of Fetch, Decode, Execute, Writeback. [25][26]

- Fetch : The processor core retrieves instructions that are waiting for it, from Memory. Memory could RAM, but instructions are usually already waiting for the core inside the processor cache. The program counter which acts as a bookmark, they are step ahead of the last instruction ended/ executed and the next one begins.

- Decode : After fetching the immediate instruction, decoding of instruction is started. Instructions involve more than one areas of the processor core. For example, arithmetic and the processor core needs to work together for decoding instruction. There is opcode which tells the processor core what should be done with the information. Opcode are like the small mathematical or logical operation. Opcode tells which part of the core will execute the instruction.

- Execute : The execute is where the processor know what needs to do and goes ahead and does opcode. It is depend on which areas of the processor core are being used

to and what data is put in. For example, processor can do arithmetic inside the Arithmetic Logic Unit (ALU). Which is connected to different inputs and outputs to get the desired result.

- Writeback : The last step is writeback. It is to write back a result of instruction execution into memory. The output address is depend on the needs of the running application. For quick access processor puts output in registers or cache. From Registers, itll get to parts of that output need to be processed which can mean that it goes into the RAM.

After the following steps One Cycle is done processing on core. It is called an instruction cycle. These instruction cycles processes are very fast, especially with powerful processors and high frequencies. CPU with multiple cores does the cycle on every core, so data execute as many times faster as CPU has cores.

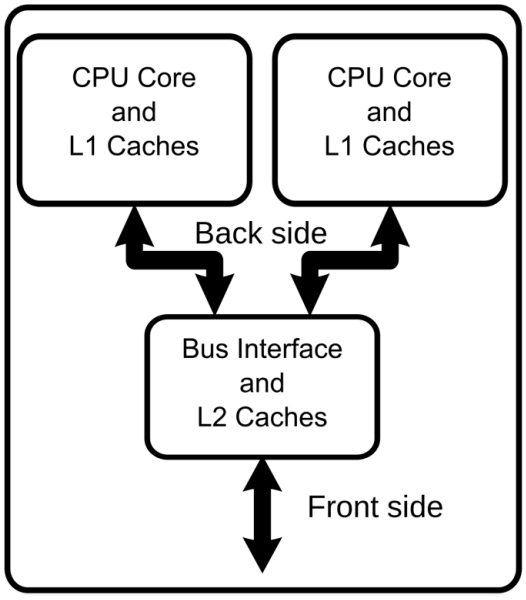

Figure 3.5: Block Diagram of Core

Uncore:

This part of processor describe the functions of processor that are not in core but closely connected to the core to reach to the high performance. They are known as an non processing part of microprocessor. They are called as "system agent" because they are not taking part in processing but they make the process faster.[27]

The microarchitecture of the Intel uncore is divided into many parts. The main uncore interface to the core is cache box, which interfaces with Last Level Cache (LLC). Multiple internal and external links are managed by physical-layer units known as PBox. Uncore includes QPI controllers. They are for point-to-point processor interconnect, developed by Intel to replaced the front-side bus which is direct core connection. The increased clock rate allows the core to access critical functions with significantly less latency Which reduce core access times to DRAM by 10 ns or more.[27][28]

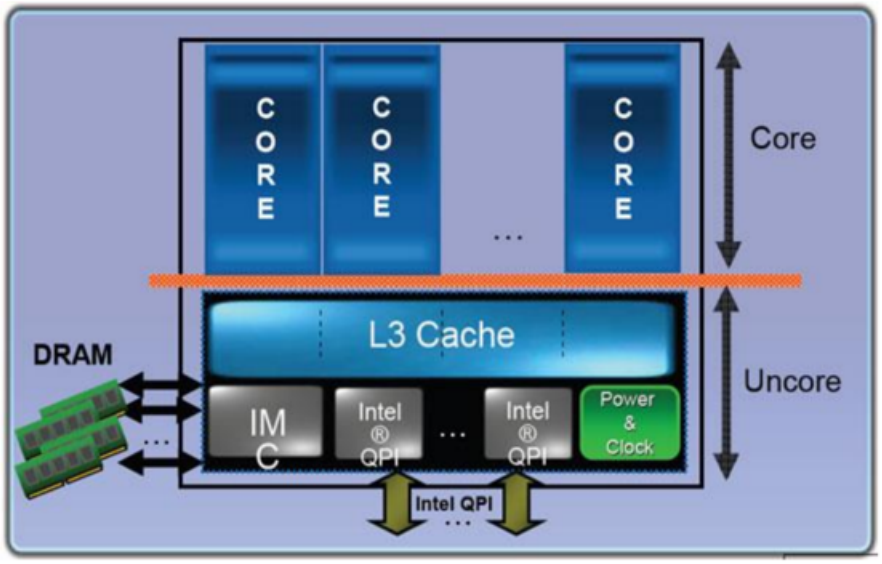

Figure 3.6: Uncore Part of Processor along with the Core

Package:

In general, a Package is a container that has one or more items in it. In context of CPU, Package is bunch of cores. It is plastic/ceramic and gold-plated contacts that match those on motherboard. [29][30] All cores are interconnected to each other.Package mounted on a substrate land-carrier.

There are Heat Spreader (IHS) is attached to the packages and core. They are the mating surface for the thermal solution such as a heatsink. That helps to tell temperature of the each core while the processes in processors are still is running. The cooling system is important because if temperature is increased than the threshold it can damage the hardware. [29][30] Package is also help in cooling system activation. A silicon physically contains the transistors implementing the processor and the package that attaches that silicon to contacts to the motherboard and IO.

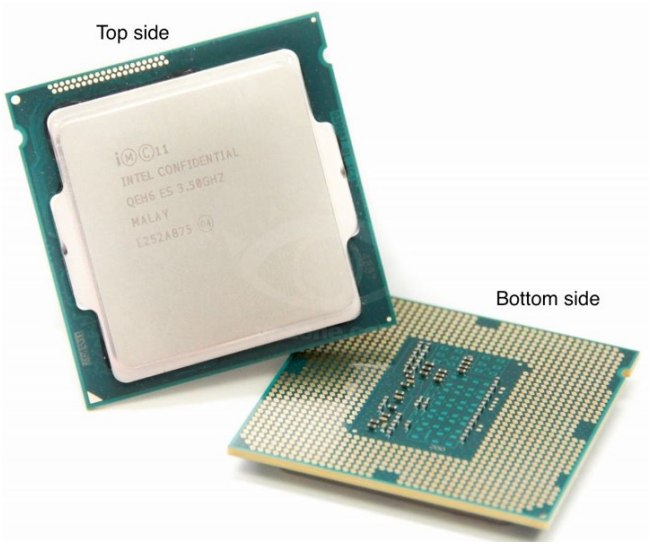

Figure 3.7: Package of Processor

Graphics:

GPU (Graphics Processing Unit) is computer chip that performs rapid mathematical calculations, for rendering images on screen. Highly complicated calculation needs extra component just to make a process a fast.[31] A processor can do GPU's work but it makes other processes slower thats why an extra component is added. It performs parallel operations. It is used for zooming and panning the screen. It is important for smooth decoding of graphics calculation and rendering of 2D/3D animations and video. The more sophisticated the GPU.[32]/ Resolution is use as the unit of processing at a time. If the resolutions higher then the processing is done faster and smoother Which makes the experience of motion in games and movies.

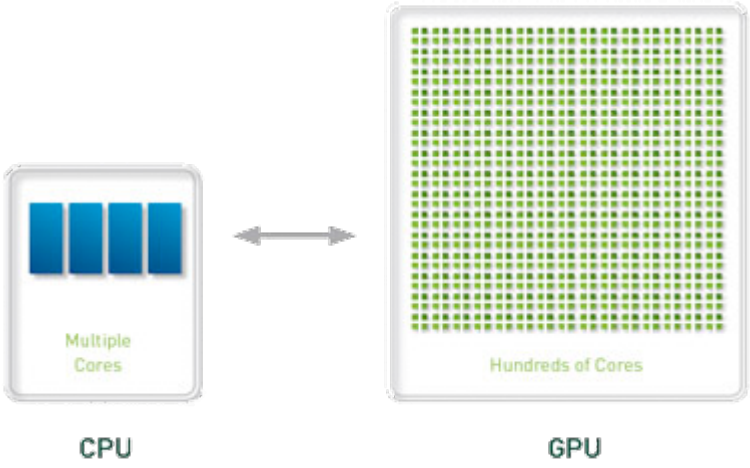

Figure 3.8: GPU compared to CPU

Memory:

A circuit that manages the flow of data going to and from the main memory that is known as MCH (memory controller hub). Older released of micro architecture used to have a separate MCH for the memory but on the latest micro architecture MCH are already installed on SoC. They are used to logical read/ write to the DRAM. As they are work separate from the processor they helps to speed up the process. Intel 3010 Chipset Memory Controller Hub (MCH) contains two components: Memory Controller Hub (MCH) and Intel I/O Controller Hub (ICH).[34][35] The MCH provides the connection between processor and main memory, PCI Express, and the ICH. The ICH is the I/O Controller Hub and provides a multitude of I/O related functions. The MCH must support multiple processor busses, memory channels and PCI-E because they are the major source of data flow in processor. FSB a front side bus is use for that. FSB have to be a very efficient bus system because it have to handle data from the GbE devcie which flow Gigabit of data per second.[35]

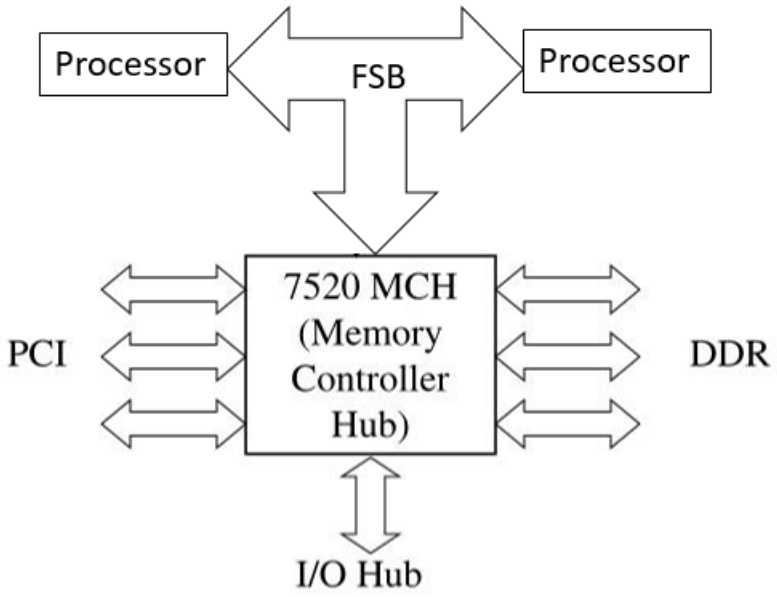

Figure 3.9: Connection of MCH with processor

PCH (Platform Controller Hub):

The PCH have special data paths and functions (these functions are also known as support function). It's working includes clocking, Flexible Display Interface (FDI) and Direct

Media Interface (DMI). This part is also installed in SoC in newer version of the microprocessor architecture. In the previous architecture memory controller and PCI-e lanes were integrated into the CPU while the PCH took over other functions. FSB are also use for connection between PCH and processor according to it's bandwidth. PCH's temperature can get to very critical level. it is important to maintain cooling system for that. So they have temperature sensing circuit with it.[19]

Following figure gives the information of the elements of the processor and different parameters located in processors (not physical location). And how they all are connected to each other on SoC with different Interfaces.[33]

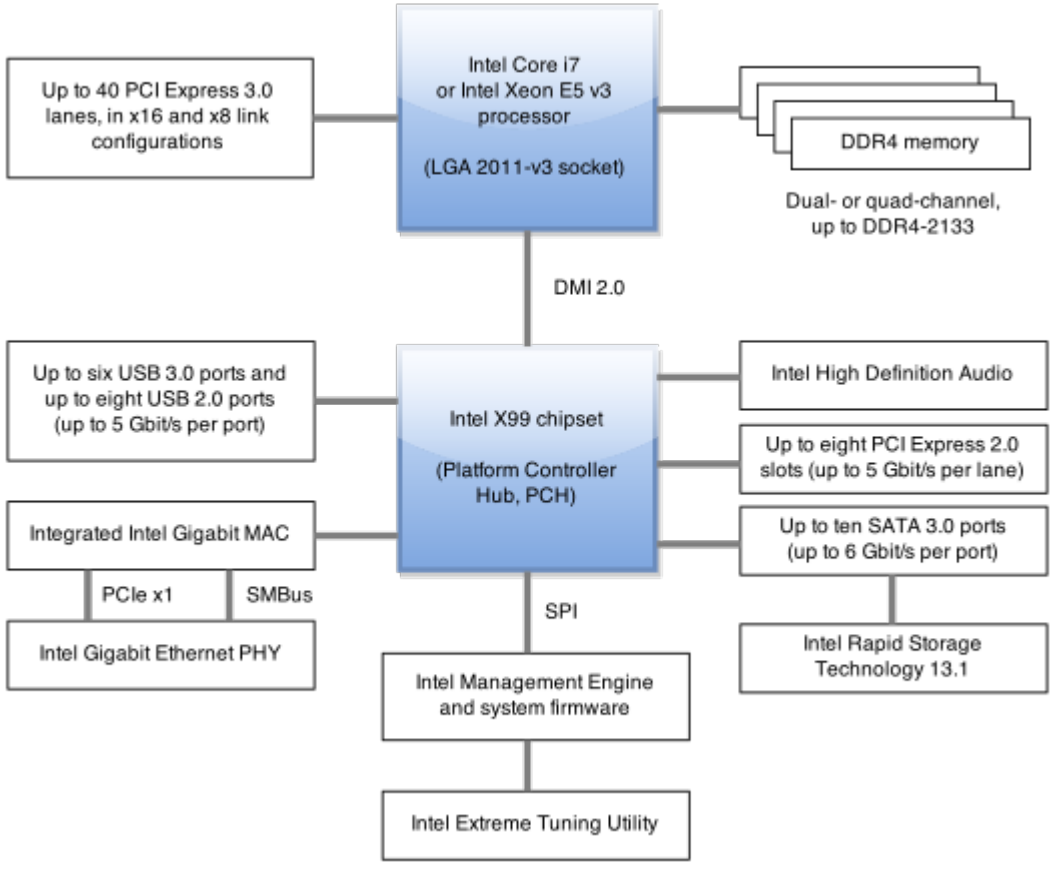

Figure 3.10: System Architecture

3.8 .NET Framework for Application GUI

3.8.1 .NET Framework Architecture

.NET framework is very vast and open source for using it for coding and language support. .NET Framework architecture is divided into 3 main blocks.[36]

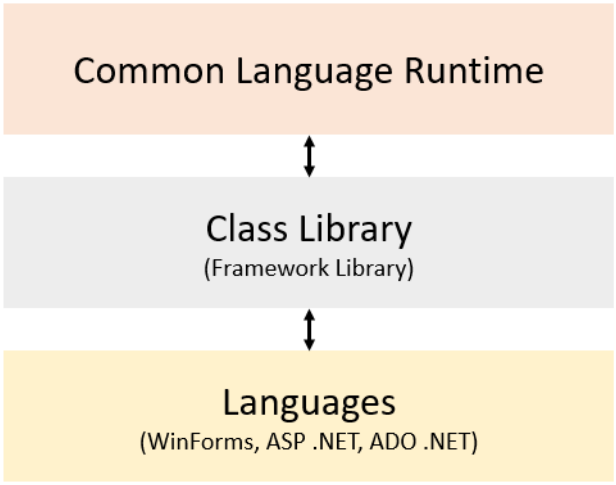

Figure 3.11: .NET Framework Architecture

1. Common Language Runtime The Common Language Infrastructure (CLI) is a platform on which programs are executed. The CLI has the following key features:[37]

Exception Handling - Exceptions are errors which occur when the application is executed. But it is handled by the system. For Example, An application tries to open a file, but the file is not present.

Garbage Collection - The process of removing unwanted resources is known as Garbage collection. It is used when resources are no longer required. For Example, A File handle which is no longer required then the file handle will be removed from the function stack.

Working with Various programming languages A variety of .Net programming languages can be possible to use on this framework. Like Native languages.

Language - The most common ones are VB.Net and C. They are consider as first level languages for .NET framework.

Compiler A compiler which will be separate each programming language. So underlying the VB.Net language. There will be a separate VB.Net and C (separate for both)

compiler.

Common Language Interpreter There is CLI so any programming language can be possible to run on .NET. So the subsequent compiler send the program to the CLI layer to run.

2. Class Library The .NET Framework includes standard class libraries which is a collection of methods and functions that can be used for the core purpose. If there is a class library with methods for file operations then there are methods for file manipulation. Like read, write, create, delete the text from a file. Most of the methods are divided into the System.* or Microsoft.* namespaces. A namespace is a separation of methods. A name space can be included in other name space to use more methods with out rewriting it over and over.[37]

3. Languages The types of applications that can be built in the .Net framework are listed below.[37]

- WinForms

  This is used for developing Forms-based applications. WinForms is a graphical (GUI) class library part of Microsoft .NET Framework providing a platform to rich client applications. This can be run on desktop, laptop and tablet PCs. It is replacing older complex C++ based Microsoft Foundation Class Library. C++ is not providing comparable platform for the user interface in a multi-tier solution.

- ASP.Net

  This is used for developing web-based applications. This application can run on any browser like Internet Explorer, Chrome or Firefox etc. The Web application needed to be processed on a server. So there is need of Internet Information Services Installed. Internet Information Services is a Microsoft component which is used to execute. The result of the execution is sent to the client machines and the output is displayed on the browser.

- ADO.Net

  This is used for development of applications to interact with Databases like Oracle, Microsoft SQL Server, etc. This Database should be supported by windows

operating system.

3.8.2 Design Principle of .Net Framework

There are Five design principles of the .Net framework. It is used because it is very relevant to create .Net based applications.[36][37]

- Interoperability - .Net framework provides a backward support. If an application built on an older version and if running the same application on a machine which had the higher version of the .Net framework then the application will still work. This is possible because Microsoft ensures that older framework versions gel well with the latest version.

- Portability - Applications built on the .Net framework can be possible to run on any Windows platform (also depend on application logic). Microsoft is also making Microsoft products work on other platforms, like iOS and Linux Which will give the a wide portability on different OS.

- Security - The .NET security feature gives the mechanism to helps in validation and verification of applications. The mechanism is used to give permission to the user access for the code and running program.

- Memory management - The .Net framework can look into memory management resources which are used to run application in actual runtime. This resources are release accordingly. This is done with "Garbage Collector". The garbage collector runs at regular intervals of time to keep checking system resources utilization and releasing.

- Simplified deployment - The .Net framework have tools, which can built a application on the .Net framework. It packaged up all modules of application. These packages distributed to clients. The packages will automatically install the application.

Chapter 4

Experiments Attempts and Result

4.1 GbE Controller in PCIe

Gigabit Ethernet is a transmission technology based on optical fiber. It carries more than one Gigabits of data per second. Ethernet LANs with 10 and 100 Mbps cards can handle Gigabit Ethernet. A newer standard, 10-Gigabit Ethernet, is also becoming available. Frame size is fixed in GbE which is in bytes, more than that size of frame slow down the network.

GbE controller is an integrated circuit chip that controls Ethernet transmission. Most computers use Ethernet to communicate with each other and connect with the Internet. Ethernet controller decodes the data so computer can use. Every packet from the internet have to pass through GbE controller. It examines the each packet from one network to another. In same network communication it simply decodes and relay it to processors. Before GbE controller there was Ethernet controller which is assembled using different IC. Now advance version of Ethernet controller, GbE comes in a single IC. Build in GbE are connected with motherboard while older computers used to have NICs for that. If Ethernet controller fails then it wont allow to connect with internet.

Before enabling GbE device there has to be assurance of the transmission. For that Loopback test is used. A framework helps to enable loopback in the GbE controller.Following section will give the detailed description of the loopback test in GbE controller.

4.2 PCIe Loopback

Loopback is for checking the connection reliability in SoC. It test whether the interfaces are properly working or not. There are 2 types of loopback

1.PIPE interface (Link Layer) and

2.PHY loopback (PHY Layer)

4.2.1 PIPE interface (Link Layer)

The PIPE loopback requires two PCIe devices connected to each other in a loopback master and a loopback slave configuration. A loopback master is requester and slave is looping back the data.

The process depends upon whether the device is RC (Root complex) or EP(End point) mode, this mode activation process is described below. Loopback mode cannot be used for looping back transactions. These modes are to be used with PCIe test equipment only for symbol level loopback. Loopback Master: Flow of the Loopback master can be described as following procedure.

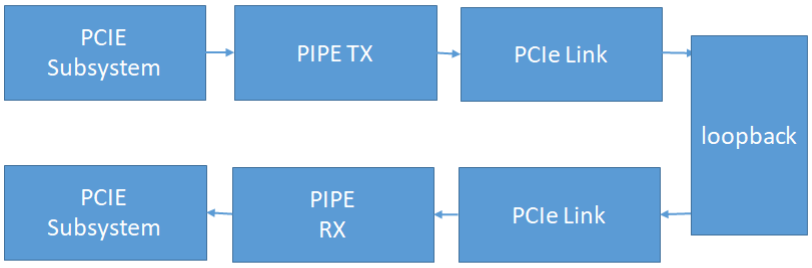

Figure 4.1: Loopback Master device

RC Mode

1. Set Loopback Enable bit (LPBK$_E N$)$in the Port Link Control Register (PL_L INK_C TRL)$.
2. The link must be initiated by setting Retrain Link bit (RETRAIN$_L INK$)$in the Link Status and Cor$

EP Mode

1. Set Loopback Enable bit (LPBK$_E N$)$in the Port Link Control Register (PL_L INK_C TRL)$.
2. Force the LTSSM to be in recovery state via the Link State field (LINK$_S TATE$)$in Port Force Link$
3. Set Force Link bit (FORCE$_L INK$)$in Port Force Link Register (PL_F ORCE_L INK)$.

Once this is done, devices at the ends of a PCIe link enter the PCIe LTSSM(Link Training and Status State Machine) loopback state. The state machine used for link connectivity and link power management is known as LTSSM. LTSSM consist of 12 main link states and their sub-states. The initiator of loopback state is the loopback master and the other device is the loopback slave. Note that it is not possible to send TLPs in this mode and return them via the loopback state of the other device.

Loopback Slave:

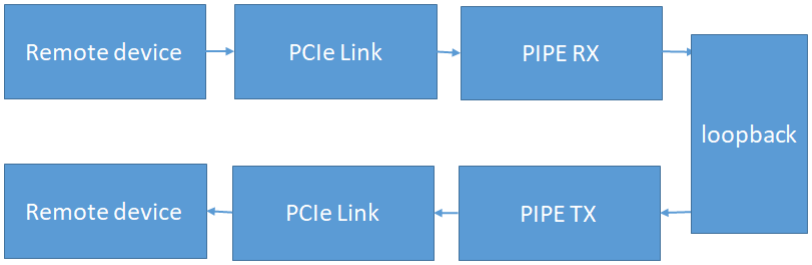

Figure 4.2: Loopback Slave device

PCIe test equipment mostly used as loopback master and it will transition the sub-system into the loopback slave state following which the inbound transactions will be looped back to the test equipment. PHY support is not required to loopback mode.

4.2.2 PHY loopback (PHY Layer)

In the PHY loopback transmitted data is looped back to the receive path at the PHY level. This can be used to perform TLP loopback even if there is no link partner. The limitation is because of link training cannot occur between two upstream ports; at least one port must be a downstream port. Both the Transmit and Receive paths must be set in loopback mode to enable PHY loopback mode.

PHY loopback configuration:

- Initialization Sequence for RC mode.
- There is need to configure the SerDes configuration registers to set Transmit and Receive paths to be in loopback mode. SerDes Configuration Lane 0 Register.
- Do the same configuration for SerDes Configuration Lane 1 Register if testing in two lanes mode.

- Before checking link training status, it need to skip Detect state in LTSSM
- With the initialization above, the PCIESS should be configured in PHY loopback-mode.

1. The PHY should be configured to operate in loopback mode before any transaction is sent out. Recommended approach is to set loopback before link training. Otherwise, any transactions that were not looped back will cause sequence numbers to increment on transmitter but not on receiver and all following transactions will be dropped because of sequence number mismatch.

2. The BAR0 and BAR1 values (if programmed) should not match the address for the transaction that is issued on slave interface. Otherwise, it will not get to the master port but to internal registers.

3. The memory base/limit registers should not be configured such that the address of the transaction lies in the range specified by the memory base/limit registers. Otherwise, the transaction will be discarded as a misrouted packet.

4. It is not possible to use configuration type transactions in this mode as RC cannot be a target of configuration transactions. Such transactions will be invalid and loopback will not work.

4.3 Experimental Result

The result of the GbE Stress testing is used to test the stability reliability of the GbE device. To check limit of device before breakdown. Stress test of a device gives a performance difference between the device under stress and device with normal load. For stress test a load generator is used which is constantly produce packets. After checking port connectivity with device with loopback, load generated by load generator apply to the ports. a very large amount of packets are generated by load generator.

It is conclude from the experiment that a device can take certain amount of stress at a time and probably device will not work efficiently. With the help of framework it is possible to put limit of data rate, which will keep away the device from the breakdown and low efficiency. It can detect suspicious activities in GbE device like suddenly device stop working, drastic change in data transmission (data rates getting high/low). The

output of the GFX which is using framework is displaying value of thermal, performance and power paramets of the different parts of the computer processor.

Chapter 5

Conclusion

The conclusion of this project is to create a framework for all IPs that helps to enable connection devices. It is possible to list out the entire PCIe device and differentiate according to different IPs. After enabling device there is need to verify port transmission which have been achieved by loopback test in case of GbE. And stress test on the device and monitor the device performance. Output of the stress test reading can help in further applications in security, device testing and chipset connection testing. GbE controller is the first from the many which is using the framework. There are other devices that can be attached with PCIe IP, the framework can be used on different PCIe devises and IPs also. The next step will be enable each device and apply different tests and monitor the performance.

Reference

[1] KeyStone Architecture Peripheral Component Interconnect Express (PCIe) User Guide SPRUGS6DSeptember 2013

[2] Performance Analysisand Evaluation of PCIe2.0 and Quad-Data Rate InniBand Matthew J. Koop Wei Huang Karthik Gopalakrishnan Dhabaleswar K. Panda

[3] May 2011 Altera Corporation IP Compiler for PCI Express User Guide

[4] Ajanovic, J., PCI Express (PCIe) 3.0 Accelerator Features, Intel Corporation, 2008, http://www.intel.com/content/dam/doc/white-paper/pci-express3-accelerator-whitepaper.pdf.

[5] Bhatt, A. V., Creating a PCI Express Interconnect, Intel Corporation, 2002, http://www.advancedaudiorentals.com/phpkb/admin/attachments/pci$_e$$xpress_w$$hite_p$$aper.pdf$.

[6] Budruk, R., Anderson, D., Shanley, T., PCI Express System Architecture, MindShare Inc., Addison-Wesley Developers Press, 2008, https://www.mindshare.com/files/ebooks/PCIExpres/SysteamArchitecture.pdf.

[7] PCI-SIG, PCI Express Base Specification Revision 3.0, November 10, 2010.

[8] Shanley, T., Anderson, D., PCI System Architecture, Fourth Edition, MindShare Inc., Addison-Wesley Developers Press, 1999.

[10] Method for configuring a Peripheral Component Interconnect Express (PCIE), Inventor Kuan-Jui HoMin-Hung ChenHsiou-Ming Chu Current Assignee VIA Technologies Inc Original Assignee VIA Technologies Inc.

[11] Host-based messaging framework for PCIe device management,r Mukund P. KhatriSurender V. Brahmaroutu Current Assignee Dell Products LP Original Assignee Dell Products LP.

[12] An open source framework for FPGA-GPU PCIe communication, 2013 International Conference on Reconfigurable Computing and FPGAs.

[13] PCI Express* (PCIe*) 3.0 Accelerator Features, Jasmin Ajanovic, Intel Corporation.

[14] Reconfigurable framework for high-bandwidth stream-oriented data processing, 2012 IEEE International SOC Conference.

[15] S. Dutta, A. Rieckmann and R. Jensen, "Viper: A Multiprocessor SOC for Advanced Set-Top Box and Digital TV Systems," in IEEE Design Test of Computers, vol. 18, no. , pp. 21-31, 2001. doi:10.1109/54.953269.

[16] Intelligent network requirements for personal communications services, IEEE Communications Society.

[17] Practical introduction to PCI Express with FPGAs, Michal HUSEJKO, John EVANS, michal.husejko@cern.ch, IT-PES-ES.

[18] PCI Overview PCI vs PCI Express, Marcel Apfelbaum Red Hat, February 09, 2016.

Websites:

[19] https://en.wikipedia.org/wiki/Software$_{a}rchitecture$

[20] https://info.lse.ac.uk/current-students/careers/resources/start-up-resources/articles/the-structure-of-a-digital-application

[21] https://en.wikipedia.org/wiki/Model-specific$_{r}egister$

[22] http://www.keil.com/support/man/docs/armasm/armasm$_{d}om1361289881714.html$

[23] https://www.felixcloutier.com/x86/rdmsr

[24] https://www.felixcloutier.com/x86/wrmsr

[25] https://www.makeuseof.com/tag/processor-core-makeuseof-explains-2/

[26] https://en.wikipedia.org/wiki/Intel$_{C}ore$

[27] https://dl.acm.org/citation.cfm?id=2747916

[28] https://en.wikipedia.org/wiki/Uncore

[29] https://www.computerhope.com/jargon/p/package.htm

[30] https://superuser.com/questions/324284/what-is-meant-by-the-terms-cpu-core-die-and-package

[31] https://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/

[32] https://en.wikipedia.org/wiki/Graphics$_{p}rocessing_{u}nit$

[33] http://www.qdpma.com/systemarchitecture/SystemArchitecture$_{2}$009.$html$

[34] https://www.mouser.in/new/Intel/intel-mch/

[35] https://www.intel.com/content/dam/doc/datasheet/7300-chipset-memory-controller-hub-datasheet.pdf

[36] https://www.guru99.com/net-framework.html

[37] https://www.c-sharpcorner.com/UploadFile/puranindia/net-framework-and-architecture