

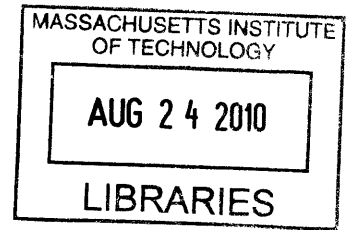
Design and Simulation of a PCI Express Gen 3.0 Communication Channel

By

Dilini Warnakulasuriyarachchi

S.B. Electrical Science & Engineering

Massachusetts Institute of Technology, 2009



ARCHIVES

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING & COMPUTER SCIENCE
AT THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY 2010
[June 2010]

© 2010 Massachusetts Institute of Technology. All rights reserved.

Signature of Author

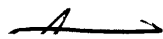
Department of Electrical Engineering and Computer Science

May 7, 2010

Certified by: _____

Scott Westbrook

Senior Member of the Technical Staff, NetApp

 VI-A Company Thesis Supervisor

Certified by: _____

Vladimir Stojanovic

Associate Professor, Department of Electrical Engineering and Computer Studies

MIT Thesis Supervisor

Accepted by: _____

Dr. Christopher J. Terman

Chairman, Department Committee on Graduate Theses

Dr. Christopher J. Terman
Department of Electrical Engineering and Computer Science
Room 38-476, M.I.T.
Cambridge, MA 02139

Subject: Master of Engineering Thesis of Dilini Warnakulasuriyarachchi

Dear Dr. Terman:

I have reviewed the attached thesis of Dilini Warnakulasuriyarachchi on behalf of NetApp. The thesis is within the scope of the thesis proposal as previously approved and does not contain any material that is objectionable to NetApp. It is also approved for its technical content.

It is understood that the actual thesis document will be the permanent property of M.I.T. and will be placed in the M.I.T. Library within one month after the date of submission. NetApp agrees that M.I.T. shall have the nonexclusive right to reproduce, publish, and distribute the thesis.

Authorized Official of Company

STEVE MILLER

SENIOR TECHNICAL DIRECTOR

NETAPP

Design and Simulation of a PCI Express Gen 3.0 Communication Channel

By

Dilini Warnakulasuriyarachchi

Submitted to the

Department of Electrical Engineering and Computer Science

May 21st 2010

In Partial Fulfillment of the Requirements of the Degree of

Master of Engineering in Electrical Engineering

ABSTRACT

PCI Express (PCIe) is a serial interconnect technology, developed by the PCI-Sig organization, which provides high bandwidth data transmission with the added benefits of reduced board space requirements, smaller connectors and simplified PCB layouts. Since faster and faster data rates are more desirable, PCIe Gen 3.0 attempts to transmit data at 8GT/s. As part of the thesis work, an existing model of a PCIe channel which connects two controller boards over a backplane, was simulated and measured under PCIe Gen 2.0 speeds (5GT/s). The resulting data from these tests were used to provide the basis for improving the model to make it function under PCIe Gen 3.0 specifications. This was achieved by exploring new receiver equalization techniques and transmitter de-emphasis and board characteristics. An integrated circuit manufacturer's model was used as the base model for PCIe Gen 2.0. This model was further developed to simulate Gen 3.0 speeds. Simulation software tools such as HSPICE, Ansoft HFSS, Ansoft Via Wizard 3.0 and MATLAB were utilized. A simulation model of the system functioning under PCIe Gen 3.0 specifications was successfully developed by using CTLE equalization technique.

MIT Thesis supervisor: Vladimir Stojanovic

Associate Professor, Department of Electrical Engineering and Computer Studies

Company Thesis Supervisor: Scott Westbrook

Senior Member of the Technical Staff, NetApp

Acknowledgement

My sincere thanks to my faculty adviser at MIT, Prof. Vladimir Stojanovic for his willingness to supervise my thesis and for his willingness to answer my questions at all times regardless of the distance or time. Thank you for always being there for your students.

I am grateful to Scott Westbrook, my VI-A company adviser who introduced me to the company and was always a great mentor. Your support and guidance were monumental in the completion of my thesis.

Thanks to Mohammad Kermani at NetApp for sharing his knowledge with me and for patiently answering my questions.

Many thanks to Steve Miller, Senior Technical Director at NetApp and my manager at NetApp Mr. Srikumar Chandran for their continuous support and guidance.

I would like to thank Richard Ely and AdityaGiry Valluri at NetApp for encouraging me to build my own probes and their willingness to help me at all times.

Last but not least, the friendship and administrative support offered by Anne Hunter, Kathy Sullivan, Vera Sayzew, and Linda Sullivan are much appreciated.

Table of Contents

	Page
Abstract	02
Acknowledgement.....	03
List of Tables	06
List of Figures	07
Chapters	
1. Introduction	09
1.1. Background.....	09
1.2. Motivation and Problem Statement	10
1.3. Objectives	11
2. Theoretical Basis	12
2.1. Differential Transmission Lines	13
2.2. Signal Integrity Issues	14
2.3. Signal Quality Measurement	16
2.4. Enhancement of Signal Quality	19
3. Technical Approach	21
3.1. Integrated circuit manufacturer’s HSPICE model	21
3.2. Gen 2.0 trace and via models	23
3.3. Integration of HSPICE file with HFSS models	26
3.4. Simulation of the new Gen 2.0 model	27
3.5. Identify changes to Gen 2.0 model for Gen 3.0 requirements	28

	Page
3.6. Development of Gen 3.0 model	29
3.7. Simulation & evaluation of Gen 3.0 model	32
4. Required Resources	34
4.1. Software Resources	34
4.2. Hardware Resources	35
4.3. Information Resources	35
5. Results	35
5.1. HFSS model analysis	36
5.2. Gen 2.0 HSPICE simulation results	37
5.3. Gen 3.0 Transmitter De-emphasis and Receiver Equalization results	39
6. Discussion	45
7. Conclusion & Future Work	48
8. References	50
9. Appendix	52

List of Tables

	Page
1. Key features of PCIe Gen 2.0 and PCIe gen 3.0	11
2. Comparison of passive CTLE methods	20
3. Via design specification	24
4. Port impedance of HFSS simulation models.....	36
5. Values of the circuit elements of the CTLE filer	41
6. Worst-case eye margin with the component tolerance	41

List of Figures

	Page
1. PCIe channel via a backplane	12
2. Differential Signaling	13
3. Effect of cross-talk between channels	14
4. Insertion loss of a transmission line 100 mils in length	16
5. A sample eye diagram	17
6. The eye template for Gen 2.0 PRBS analysis.....	17
7. The eye template for Gen 3.0	18
8. Decision Feedback Equalization	20
9. The interconnect simulation component	22
10. Trace specification developed by Mohammad Kermani employee at NetApp	23
11. Trace 3D model	24
12. Differential Impedance calculator	26
13. Gen 2.0 and Gen 3.0 simulated channel topology	28
14. Gen 2.0 eye and Gen 2.0 channel under Gen 3.0 speed	29
15. FIR filter	30
16. Transmitter waveform with de-emphasis and pre-shoot	30
17. Circuit diagram of the CTLE filter	31
18. Channel equalization and de-emphasis optimization process	33
19. Decision Feedback Equalization process	34
20. The effect of board material on the signal quality with eye mask for Gen 2.0	37
21. Via with and without backdrilling	38

	Page
22. Effect of backdrilling of the via	38
23. Effect of via stub length	39
24. Bode plot of the CTLE filter	41
25. Eye diagram at the receiver end with CTLE filtering	43
26. Eye diagram at the receiver end with DFE	44
27. DFE eye closure probability	44
28. Effect of the connector model	47

1. Introduction

Band limited wire-line interconnects can be found in various high-performance digital systems – from backplanes of high-speed internet routers, to server/rack backplanes in data-centers, to processor-memory interfaces in modern portable and desktop computers. All these devices are limited by the same channel issues (dielectric and skin loss, connector cross-talk and impedance discontinuities) as they try to reach high data rates. Various equalization techniques can be utilized to reduce these issues. This thesis attempts to do so by designing and simulating a PCIe communication channel, which fulfills the PCIe Gen 3.0 preliminary specifications that require a 8Gb/s data rate.

Chapter 1 will address the background, motivation and problem statement and the objectives of the thesis. Chapter 2 will outline the theoretical knowledge that is required for this thesis work. The method of implementation is highlighted under Chapter 3. Chapter 4 specifies the resources necessary to complete this project and Chapter 5 presents the simulation results. The results of the simulation will be discussed under Chapter 6 and possible future enhancements are detailed under Chapter 7.

1.1. Background

PCI Express (PCIe) is the third generation of multi-purpose I/O interface that was introduced by the PCI Special Interest Group (PCI-SIG) since its inception in 1992 [6]. This organization was established to develop and manage the PCI standards. In 1994 PCI-SIG introduced PCI (Peripheral Component Interconnect), later known as Conventional PCI, as its first architecture. Second introduction by PCI-SIG enhanced the performance of PCI to PCI Extended (PCI-X). PCI-X is a high performance variant of 64-bit PCI design.

In 2004 the PCIe standard was introduced by the PCI-SIG organization. This interface can be used across the computing industry from mobile through high-end servers and even communication equipment. PCIe allows each device to have its own dedicated link. Each link comprises of one-way point to point lanes. A lane contains a Transmit and a Receive with a low voltage differential signal. Two generations were introduced under the PCIe architecture. PCIe Gen 1.0, and PCIe Gen 2.0. Each generation has data transmission rates of 2.5 Gb/s and 5Gb/s respectively. Under the PCIe architecture, each generation achieves greater bandwidth by doubling the data rates [2].

1.2.Motivation & Problem Statement

A new generation of PCIe architecture; Generation 3.0 is underway to double the bandwidth to 8GT/s under the 128/130 scrambling coding system. Due to this new scrambling code, the transition from 5GT/s to 8GT/s is highly desirable since it results in an increase in the data rate without doubling the signal rate. Furthermore, under the PCIe preliminary specifications presented by the PCI-SIG organization, PCIe Gen 3.0 attempts to reduce latency and have greater power management capabilities compared to its predecessors. This new technology is desirable in a backplane environment as in the case of this thesis, since increase throughput and/or use of fewer lanes results in lower power requirement and increase throughput.

As with all new technologies, there are some key challenges engineers must face in order to transition from PCIe Gen 2.0 to PCIe Gen 3.0. Some key features of the PCIe Gen 2.0 and PCIe Gen 3.0 are given in table 1 [4].

Feature	PCI Gen 2	PCIe Gen 3
Signaling rate	5GT/s	8GT/s
Encoding	8b/10b	128b/130b
Effective bandwidth	4Gb/s	8Gb/s
Bit timing	200ps	125ps
Channel distant client	14"	14"
Channel distant server	20"	20"

Table 1: Key feature of PCIe Gen 2 and PCIe Gen 3

The key challenge of this architecture is to accommodate twice the signaling rate while maintain lower clock timing and jitter tolerance. According to the Gen 3.0 specifications, the receiver end is a closed eye [13]. In a backplane environment as in the case of this thesis, more complex equalizer techniques at the Transmitter and Receiver end, and better jitter management techniques are required to obtain the desired eye opening. Another key challenge faced while working on this thesis was that PCI-SIG is still in the process of producing final design specification for PCIe Gen3. Hence driver and receiver models which are specifically compatible with PCIe Gen 3 speeds were not readily available.

1.3. Objective

The overall objective of this thesis was to develop a model to simulate a PCIe channel connecting two controller boards via a backplane. This channel must meet the specifications of the PCIe Gen 3.0 standard. Figure 1 shows this system of interest.

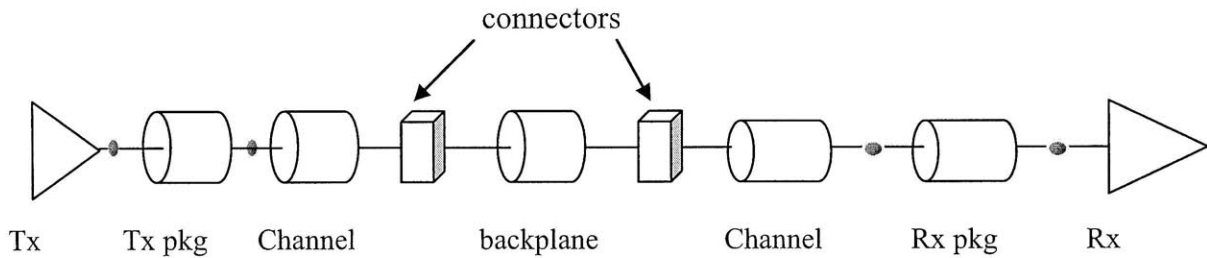


Figure 1: PCIe channel via a backplane

As part of the thesis work various board material, via breakouts and backdrilling of the vias were explored. Furthermore, de-emphasis techniques at the transmitter and receiver equalizer techniques were explored to obtain a model that functions favorably under PCIe Gen 3 speeds.

As part of the thesis work the following channel criteria were also explored:

- De-emphasis technique at the transmitter
- Behavior of the channel under different board materials and back drilling of the vias
- Behavior of the channel under different receiver equalizer techniques
- Channel lengths

2. Theoretical Basis

As part of the thesis work it was imperative to have a thorough understanding of signal integrity issues such as, cross-talk and reflection loss, frequency and time domain analysis and PCIe design principles while designing high speed communication channels. Furthermore, it was important to obtain an understanding of channel measurement techniques and various signal enhancement techniques in order to develop a working model of a particular channel. The

following section describes such topics to provide the reader with the necessary theoretical background to better understand this thesis.

2.1. Differential Transmission Lines:

Differential transmission lines are made of two single ended traces on which a complementary voltage is applied [16]. The advantage of this signaling method as opposed to a single trace is its ability to reject external common-mode interference (noise), as well as provide a tight current return path for differential mode. Since both lines will experience the same noise, subtracting one line voltage from the other line voltage, will result in noise cancelation. The second advantage to this method is that less amount of power is used in differential traces, as opposed to single ended traces to send the same signal. This is due to the fact that differential signal contains its own reference voltage that allows for much smaller swing. This ability to have small voltage swing allows for less power consumption.

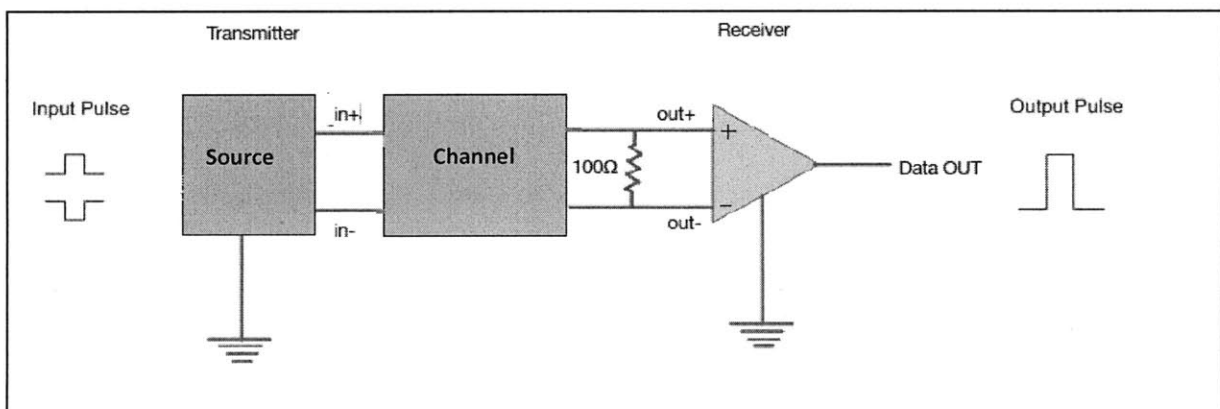


Figure 2: Differential Signaling

2.2. Signal Integrity issues:

There are many signal integrity issues that Printed Circuit Board (PCB) design engineers need to consider when designing high speed channels. An important signal integrity issue is cross-talk, which arises when there is undesirable inductive or capacitive coupling between channels. For instance, if one channel is carrying a signal and there is a nearby passive channel, a small voltage is generated in this passive channel due to change in electrical and magnetic fields [21].

The following Figure 3 depicts a signal pulse travelling down a channel shown in the middle and the voltages generated due to cross-talk on the channels on either side of this carrier. It is important to note that the pulse amplitudes on the two victim lines (shown on the top and bottom graphs) are significantly smaller than the pulse amplitude on the aggressor line.

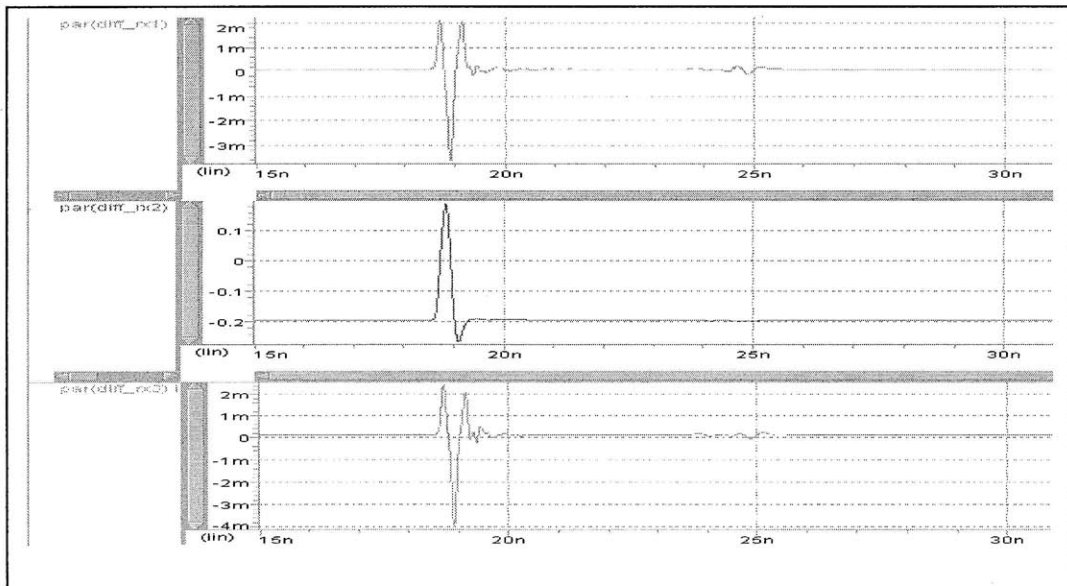


Figure 3: Effect of cross-talk between channels

Another signal integrity issue is date-dependent jitter (DDJ). DDJ is the variation in the transition time between symbols. This is caused due to Inter Symbol Interference (ISI). ISI is the result of sending high data rates down low-bandwidth material such as copper wires [3].

Two other important signal integrity issues are return loss and insertion loss of a signal. When a trace connects to cap or a via on a PCB, a signal may encounter changes in impedance. Due to these differences in impedance part of the signal may bounce back to the transmitter. This is called return loss. Return loss can be calculated by using the following equation 1.

$$RL(dB) = 10 \log_{10} \frac{P_T}{P_R} \quad \text{Equation 1}$$

P_T is the power transmitted by the source and P_R is the power reflected.

Insertion loss is the loss of the signal power as the signal travels down a transmission line or any conducting object. Insertion loss occurs due to the difference in impedances down a conducting channel and also due to skin effect and dielectric losses of the material. It can be calculated by using the following equation 2.

$$IL(dB) = 10 \log_{10} \frac{P_R}{P_T} \quad \text{Equation 2}$$

P_T is the power transmitted by the source and P_R is the power received.

The following Figure 4 shows the insertion loss experienced by a transmission line 100mil in length with FR408HR material.

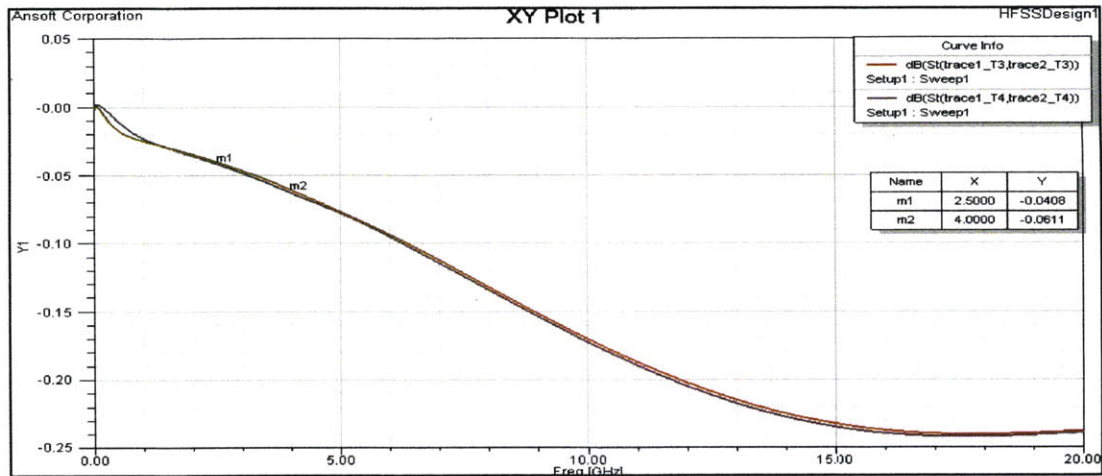


Figure 4: Insertion loss of a transmission line 100mils in length- linear scale.

2.3.Signal quality:

There are several methods to describe the quality of a signal travelling through a channel. One such method is the use of eye diagrams. An eye diagram is generated by transmitting a bit stream pattern that represents all possible bit stream patterns through the channel to be tested. Each received cycle is overlaid on the previous cycle to superimpose all the cycles. This superimposed waveform is in the shape of an open eye, hence the name eye diagrams [18]. By measuring the width of an eye and the height of an eye (measuring the eye opening) with the use of an eye mask we can determine whether a particular channel is compliant with PCIe specifications. A sample eye diagram is displayed below under Figure 5.

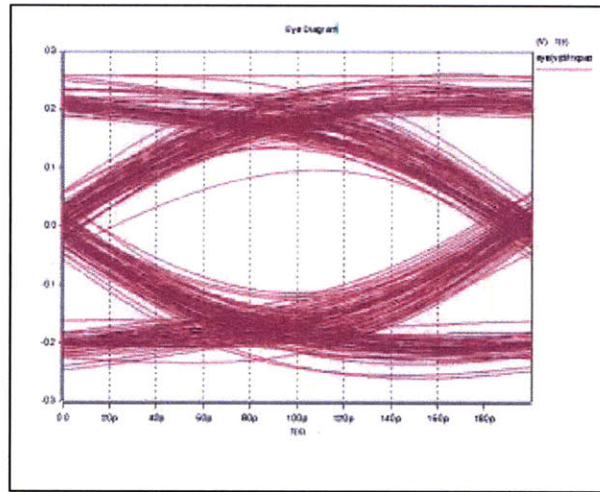
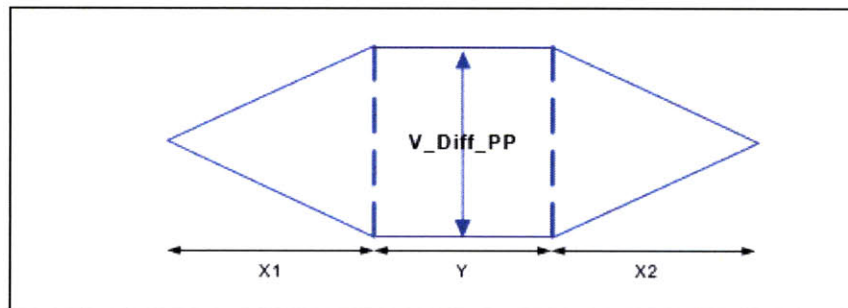


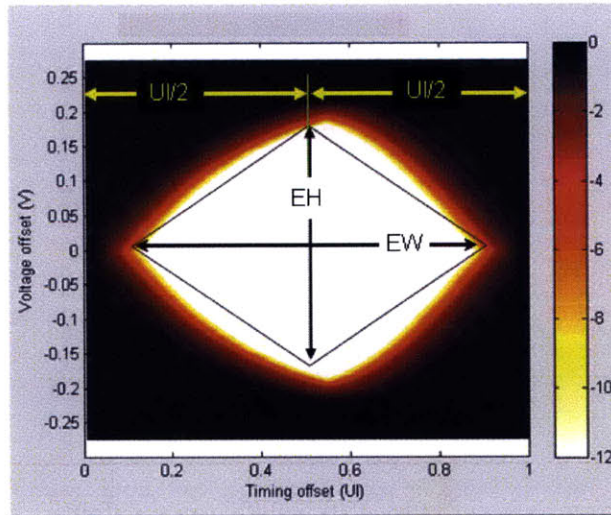
Figure 5: A sample eye diagram

The eye masks for Intel Pseudo Random Bit Stream Analysis (PRBS) for Gen 2.0 is given under Figure 6 [11].



Transfer rate	Driver	Receiver	UI	X1	X2	Y	V_Diff_PP
5.0 GT/s	IOH	Base load	200ps	45.7 ps	45.7 ps	55 ps	150 mV

Figure 6: The eye template for Gen 2.0 PRBS analysis



Eye Width (ps) after CTLE	= $\sim 0.3 \text{ UI} = 37.5$
Eye Height (mV) after CTLE	= ~ 10

Figure 7: The eye template for Gen 3.0

The Figure 7 above depicts the PCIe Gen 3.0 eye template provided in the PCIe Base Specification Rev. 03 V 0.7 [13] by PCI-SIG organization for a receiver eye after CTLE in FR4 material.

Another method to measure the quality of a channel is to measure the channel's scattering parameters. Scattering parameters includes return loss measurements (S11) and insertion loss measurements (S21) of the channel.

2.4. Enhancement of the Signal Quality:

As previously highlighted there are various signal integrity issues an engineer must consider when designing a high speed communication channel. Due to the high data rate of PCIe Gen 3.0, it is expected to have a closed eye at the receiver end of the channel. To widen this eye, Continuous Time Linear Equalization (CTLE) technique was suggested under the guidelines of the version 0.7 of the specifications [13]. Equalization is a procedure used to compensate for non-ideal channel characteristics by equalizing the frequency spectrum. As mentioned earlier, the use of low bandwidth material such as copper wires and PCB traces to process high data rates generates ISI. CTLE can be used as a method to counteract this bandwidth limitation. CTLE is a high pass / low pass filter configuration which is designed in the frequency domain. CTLE can be done actively or passively. Active CTLE can be more reliable and tunable; however, it will require high power and cause latency. Passive CTLE is non tunable yet causes less latency. For this thesis, passive CTLE implementation was used due to reduced latency and lower power requirement.

Passive CTLE can be implemented in three different ways: On die, on package or on PCB. In the thesis we used transmitter and receiver models from outside vendors. Therefore the only method available to implement CTLE is the PCB implementation. Nevertheless, a comparison of the three implementation method described by the Qualcomm Digital Signal Integrity presentation is given below under table 2 [24].

	On Die CLE	On Package CLE	On PCB CLE
Pro	Reliable	High quality and capacity of passive components	Low cost and easy to assemble
Con	Die real estate and limitation of passive component capacity	Package real estate	Limited performance due to passive component parasitic

Table 2: Comparison of Passive CTLE methods

Another equalization method is Decision Feedback Equalization (DFE), which, unlike linear equalization techniques, removes the ISI with minimal noise amplification or signal attenuation [3], Decision Feedback Equalization alters the frequency response of the equalizer to remove ISI generated by the channel at the receiver end. This non-linear equalizer performs its task by using previously made decisions to eliminate ISI on the current signal modulation. The following Figure 8 depicts a simple DFE system.

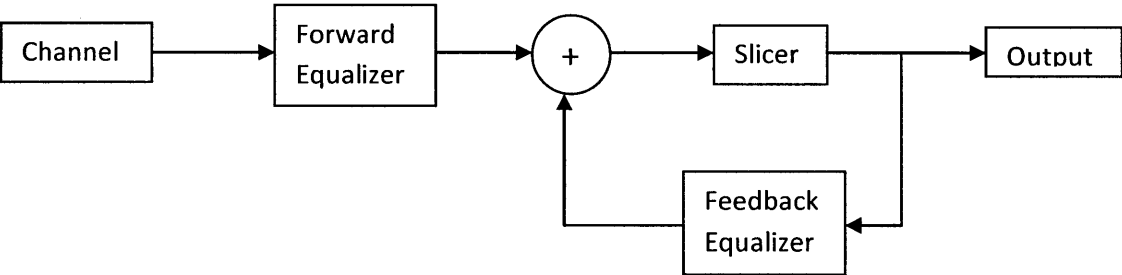


Figure 8: Decision Feedback Equalizer.

3. Technical Approach

To achieve the desired goal of developing a PCIe Gen 3.0 channel, several tasks were performed. These tasks are described as follows:

3.1. Integrated Circuit Manufacturer's HSPICE model:

The integrated circuit manufacturer's model is a readily available HSPICE simulation model which depicts various topologies that connect two cards across a backplane. The basic interconnect topology of the simulation given under this model is depicted in the Figure 9 below. The red boxes depict the models directly provided by the integrated circuit manufacturer. The blue boxes represent the connector models provided by an external manufacturer. The green boxes depict the model created specifically for this thesis.

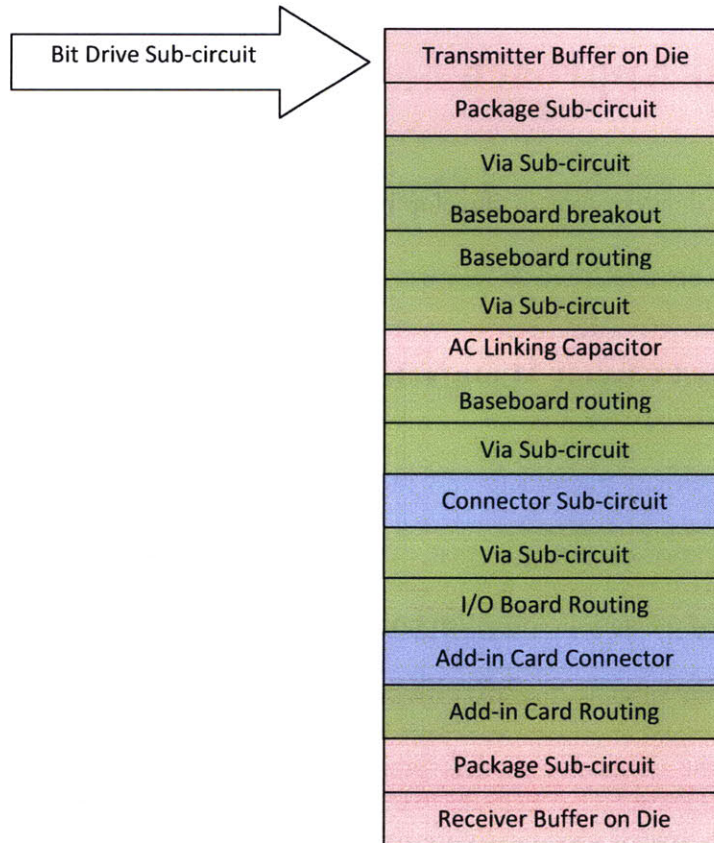


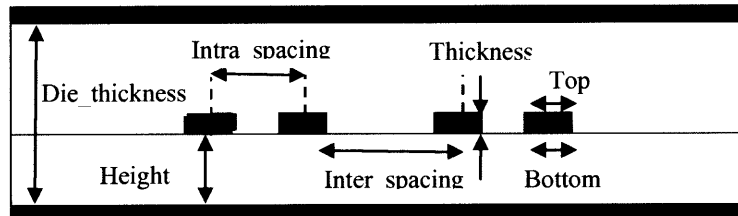
Figure 9: The interconnect simulation component

These simulation models were used to understand how to create HSPICE models which can replicate physical systems that connect two controller cards via a backplane. In the HSPICE simulations, the transmitter can be excited by either a Pseudo Random Bit Stream (PRBS), a Card Electromechanical (CEM) bit pattern or a Peak Distortion Analysis (PDA) data pattern. The HSPICE programs were thoroughly read and various simulations were run on them to understand how each component interacts with each other. After each simulation the data was analyzed in eye diagram format and evaluated against the PCIe Gen 2.0 specifications [11]. For the purpose of this thesis and due to time constraints, only PRBS analysis was performed for the spice simulations and PDA analysis was used to estimate the worst case eye opening in MATLAB.

3.2. Gen 2.0 trace and via models:

Since the system used in the company is different from the model presented under Integrated circuit manufacturer’s chipset, new simulation models for the traces, vias, and the connectors were created. This is mainly due to the different stack layout of the PCBs and the different specifications used for routing. To create these models Ansoft HFSS and Ansoft Via Wizard was used.

To create the required models certain pre-defined specifications were used. These specifications were developed by another colleague, Mohammad Kermani, from NetApp, Inc. The trace specifications are depicted under the Figure 10 below [25].



Impedance (ohms)	Top (mils)	Bottom (mils)	Die_thickness (mils)	Height (mils)	Intra_spacing (mils)	Inter_spacing (mils)	Thickness (mils)
85.2 FR408HR	6	6	12.2	6	18	5*height=30	1.2

Figure 10: Trace specification developed by Mohammad Kermani, NetApp, Inc.

The trace model was created using Ansoft HFSS. It is a 3D model with three differential pairs depicting two aggressor lines with a victim line in the middle. The cross section of the 3D model was shown in the Figure 10, where the trace width, trace thickness, trace spacing and die thickness were specified. The dielectric material used for the stack up was FR408HR with

permittivity of 3.75 and loss tangent of 0.01. The FR4 material has a permittivity of 4.4 and a loss tangent of 0.021. The impedance of the trace model was designed to be 85.2 in order to match the impedance of the connector model, which was 85Ω . The Figure 11 below shows the 3D model of the trace model.

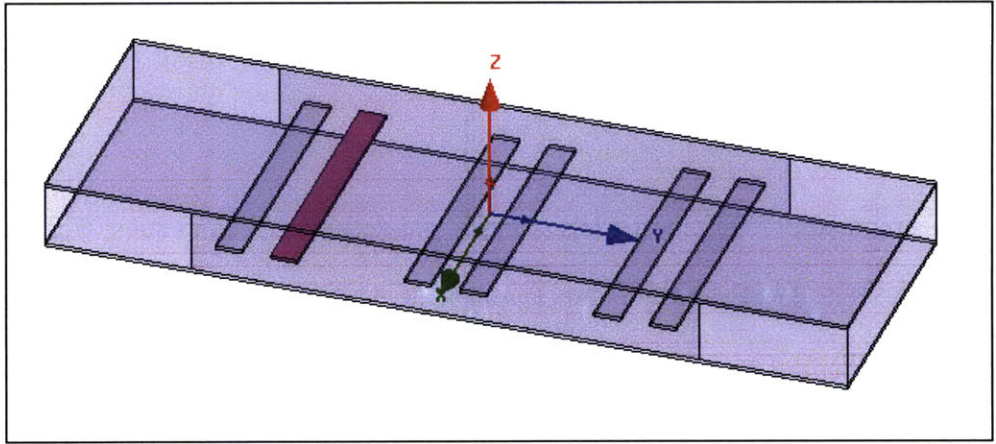


Figure 11: Trace 3D model

Via models which connect the traces from one layer to another layer on the PCB was created using the HFSS software and Via Wizard 3.0. The via specifications are given in the following table 3 [25].

Via Type	Pitch Size	Diameter	Pad Size	Anti-Pad Size
IOXM board via at connector	55.1 mils	23.6 mils	36 mils	58 mils
IOXM board via at capacitor	1mm	12 mils	20 mils	25 mils
Backplane via at connector	55.1 mils	23.6 mils	36 mils	58 mils

Table 3: Via design specifications

After each model was created, they were individually tested using HFSS analysis tool. Ansoft HFSS performs frequency domain analysis by first creating an adaptive mesh that conforms to the electrical performance of the device. Once an accurate mesh has been created, the HFSS uses this mesh to perform a desired frequency sweep. Generally this adaptive meshing is performed on the maximum frequency of the desired frequency range. For further details of this process, please refer to the HFSS manual mentioned under references [9].

An interpolating frequency analysis of the traces and vias were conducted ranging from 0 GHz to 20 GHz with a linear step size of 0.02 GHz. Interpolating sweep analysis solves at discrete frequency points that are fit by interpolation. The frequency range and frequency step size was determined by first selecting the time domain simulation window and then determining the appropriate frequency range. The following equations 3 and 4 depict the relationship between time domain analysis and frequency domain analysis.

$$\text{Maximum simulation time} = \frac{1}{\text{Frequency step}} \quad \text{Equation 3}$$

$$\text{Time step} = \frac{1}{\text{Maximum simulation Frequency}} \quad \text{Equation 4}$$

According to the frequency domain values mentioned above, the time domain analysis would range from 0ns to 50ns in 50ps steps. Since this step size is not appropriate to analyze data rates of 5Gbps and 8Gbps, a step size of 2ps was selected. This time step indicates the maximum frequency to be 500GHz. Since signal loss beyond 20GHz is quite significant, the HFSS frequency sweeps were only performed up to 20GHz.

The generated Z matrix was analyzed to determine whether the correct differential impedance was achieved on the excitation ports. The expected differential impedance was

calculated using a differential impedance calculator available online on the “Ideas Consulting” website [17]. This online calculator is designed for a differential stripline model where the user can input the trace width, thickness, spacing, distance between the planes and the relative dielectric constant. The following diagram 12 displays the web interface of this calculator.

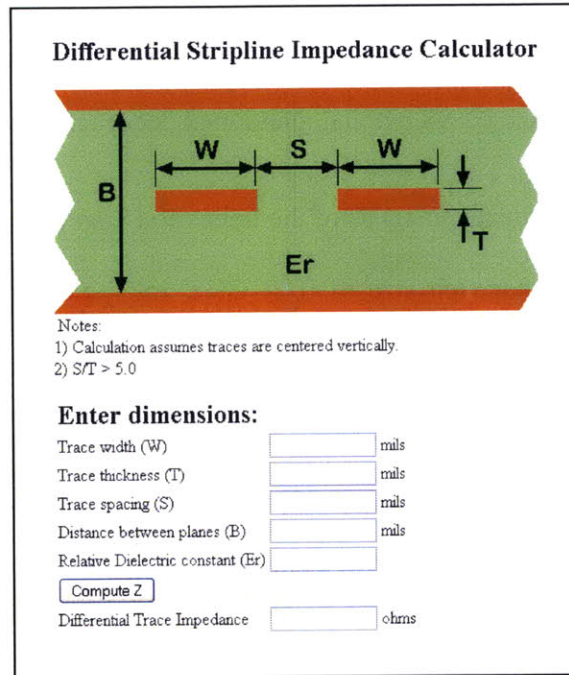


Figure 12: Differential impedance calculator

Another validation method was to graph the insertion loss and return loss of each model. These graphs were used to ensure that the traces functioned as expected.

3.3. Integration of HSPICE file with HFSS models:

The trace and via models were created using Ansoft HFSS. However, the entire Gen 2 system was simulated using the HSPICE simulation software. Therefore, the new HFSS models were integrated into the HSPICE model prior to simulation.

The trace model's resistance, impedance, conductance and capacitance (RLGC) data was extracted in a tabular format. This table file (.TAB file) was incorporated into HSPICE using the w-element syntax [Appendix - 3]. The reason for selecting this technique is because it is straightforward and because of its ease of implementation in HSPICE.

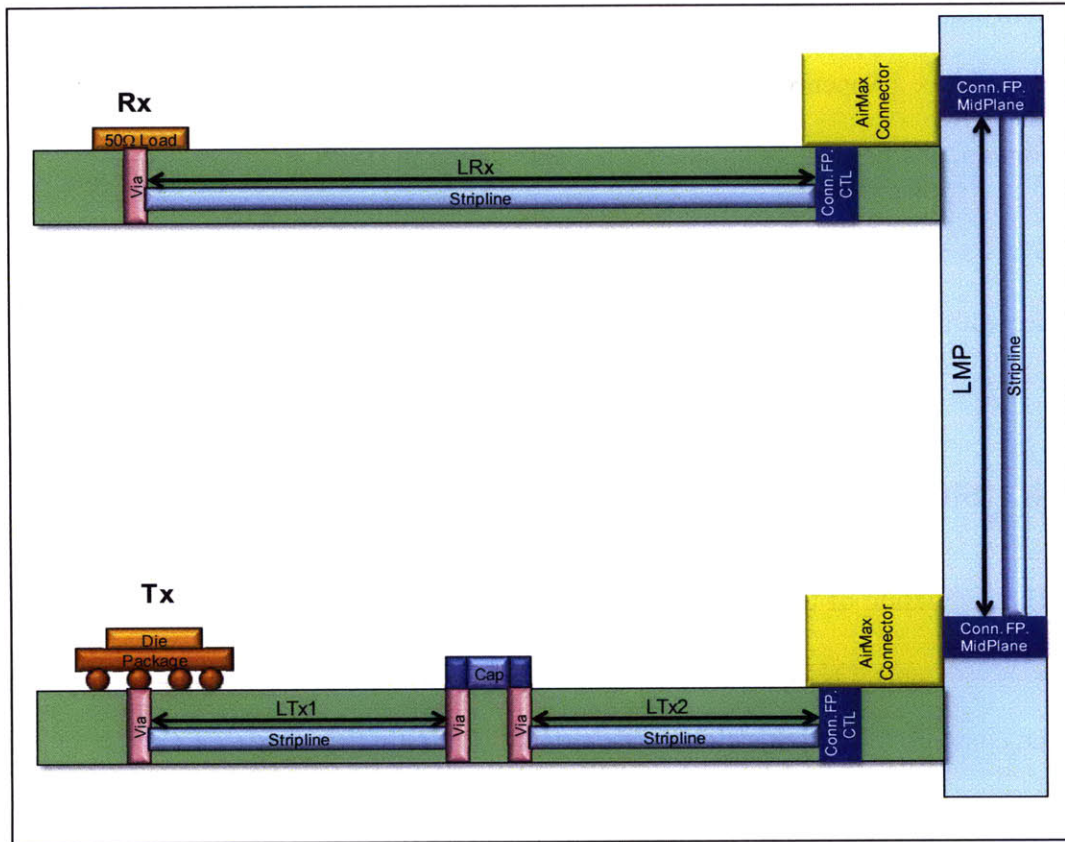
Via's scattering parameter data was extracted by using the touchstone file format. It was summoned within HSPICE as an S-element [Appendix – 4].

3.4. Simulation of the new Gen 2 model

The simulation of the new system in HSPICE was conducted in several stages. The first stage was to test each trace model individually in HSPICE, generate the transfer functions of the model and compare them to the HFSS transfer function.

The second stage comprised of incorporating the via models one by one into the HSPICE model and running the simulation under PRBS bit pattern

Finally the overall system was simulated up to 50 ns time interval (derived under section 3.2 equation 3) with all the vias, trace models and vendor specific connector models to imitate the system designed by the company. The IOXM board which is an IO expander board [Appendix - 2] has 14 layers on the stackup and the backplane stackup [Appendix - 1] contains 18 layers. The following Figure 13 displays a channel topology used for Gen 2.0 and Gen 3.0 simulations [25].



Trace	LTx1	LTx2	LMP	LRx
Length	2"	2"	15"	4"

Figure 13: Simulated Gen 2.0 and Gen 3.0 channel topology

3.5. Identify changes to Gen 2 model for Gen 3 requirements

As mentioned under the introduction the main difference between the PCIe Gen 2.0 and Gen 3.0 is the speed of data transmission. In Gen 3.0 we attempt to transmit data at 8Gbp/s. To understand why any design changes are required to switch from Gen 2.0 speeds to Gen 3.0

speeds, the Gen 2.0 model was simulated at Gen 3.0 speeds. The eye diagrams of the Gen 2.0 channel made of FR408HR material with backplane channel length of 15” and the same Gen 2.0 channel simulated under Gen 3.0 speed are depicted below under Figure 14.

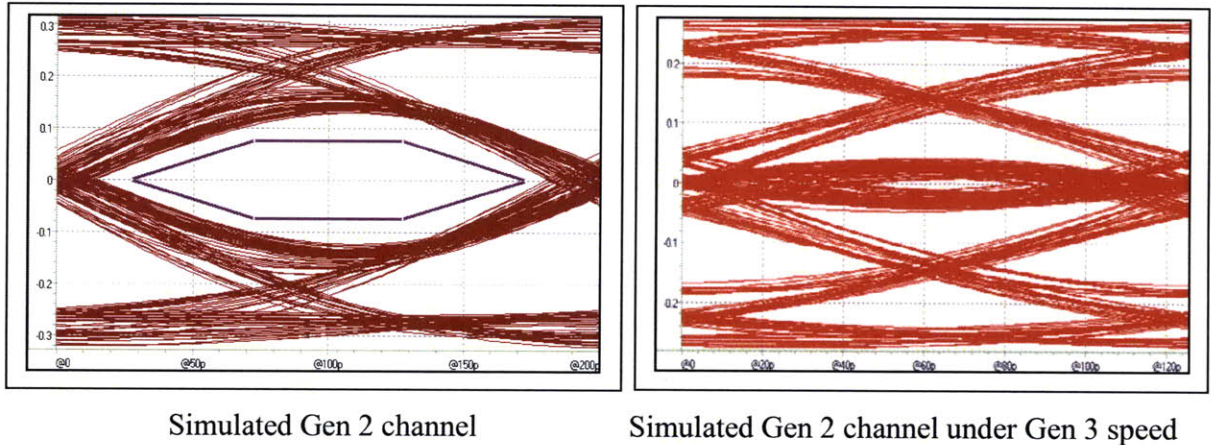
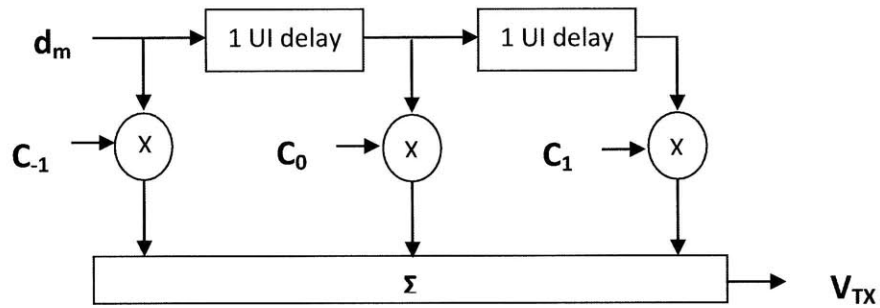


Figure 14: Gen 2.0 eye and Gen 2.0 channel under Gen 3.0 speed

As shown in the Figure above, the receiver eye under Gen 3.0 speed is far more distorted than the receiver eye of Gen 2.0. The width and the height of the eye are dramatically reduced. At the Gen 3.0 speed the eye height is 20.4 mV and eye width is 35.5 ps. Therefore, it is apparent that some design changes are needed to improve the channel.

3.6. Development of Gen 3 model

There are several channel enhancement suggestions provided by the PCI-SIG organization for PCIe Gen 3.0 channels. The first recommendation is the use of a 3-tap Finite Impulse Response (FIR) filter to provide transmitter equalization as shown in Figure 15 below [13].



$$V_{Tx} = \sum_{n=\{-1,0,1\}} c_n d_{m-n}$$

Figure 15: FIR filter

The values of the filter coefficients are determined by the transmitter de-emphasis and the pre-shoot values. De-emphasis is the process of changing the amplitude of the low frequencies or changing the attenuation of the non-transitioning bits. The following Figure 16 depicts a transmitter waveform with de-emphasis and pre-shoot.

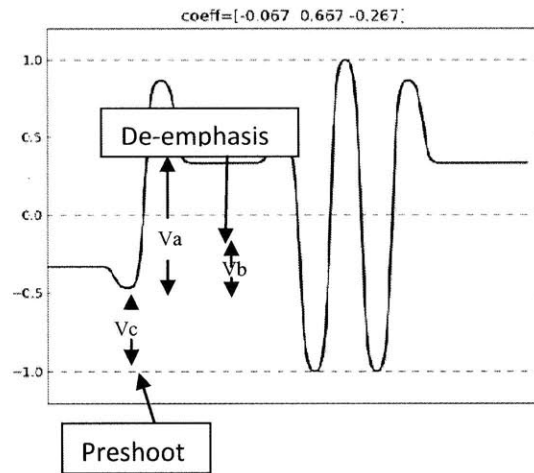


Figure 16: Transmitter waveform with de-emphasis and pre-shoot

De-emphasis and pre-shoot can be calculated by using the following equations:

$$\text{de - emphasis} = 20 \log_{10} \frac{V_b}{V_a} \quad \text{Equation 5}$$

$$\text{pre - shoot} = 20 \log_{10} \frac{V_c}{V_a} \quad \text{Equation 6}$$

Once the tap values are determined, the 3-tap FIR filter can be easily implemented by altering the transmitter code found in the integrated circuit manufacturer's chipset. The available code is designed for a 6-tap FIR filter. By setting the first three taps of the filter to our desired values we can implement this equalization technique quite conveniently.

The second recommendation made by PCI-SIG organization is to implement receiver equalization by using a Continuous Time Linear Equalizer (CTLE). The suggested circuit implementation is a high pass/ low pass filter system to enhance the amplitude of the high frequency components of the received signal. The suggested circuit diagram is shown in Figure 17 [10].

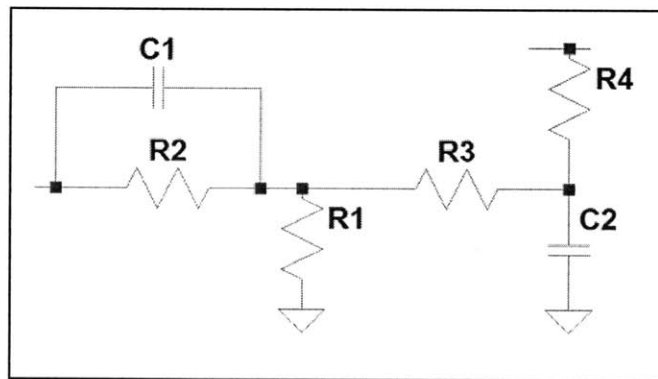


Figure 17: Circuit diagram of the CTLE filter

The transfer function of the filter is given below equation 7:

$$\frac{\text{Output}}{\text{Input}} = G_{dc} \frac{\left(\frac{s}{\omega_{p1}} + 1\right)}{\left(\frac{s}{\omega_{p1}} + 1\right)\left(\frac{s}{\omega_{p2}} + 1\right)} \quad \text{Equation 7}$$

$$G_{dc} = \left(\frac{R_2}{R_1 + R_2}\right) \quad \text{Equation 8}$$

$$\omega_{p1} = \left(\frac{R_1 + R_2}{R_1 R_2 C_1}\right) \quad \text{Equation 9}$$

$$\omega_{p2} = \left(\frac{1}{R_3 C_2}\right) \quad \text{Equation 10}$$

If we assume that resistor R1 and resistor R2 values are equal then we can observe that the location of the first pole should be at twice the frequency of the zero location. The resistor R4 is the termination resistance at the receiver end which is 50 Ω .

3.7. Simulation & evaluation of Gen 3 model

To develop Gen 3.0 channel we need to determine the FIR tap values for transmitter de-emphasis and the pole locations, gain and zero locations of the CTLE filter for receiver equalization. The FIR tap values can be determined by first obtaining an insertion loss of the channel (S21 parameters) and by extracting the impulse response of the channel from this data. From the impulse response, we can obtain a bit spaced sampled pulse response of the channel by convolving the impulse response with a pulse train. This pulse response is used to generate zero-forcing equalization coefficients for transmitter de-emphasis with peak power limited to 1 [15]. The MATLAB code for this process, developed in this thesis, extends the code initially written by Prof. Vladimir Stojanovic and students in his Integrated Systems Group at Massachusetts Institute of Technology [Appendix - 5].

Once the tap values are determined using the MATLAB code, FIR 3-tap filtering process was implemented by changing the tap values of the Verilog-A code provided by integrated circuit manufacturer's simulation set [Appendix -7]. In the code provided to the user, the transmitter had a 6-tap FIR filter already incorporated into the Verilog-A code. However, for the Gen 2.0 model, four of the taps were set to zero and only two were used which meets the Gen 2 specification [12].

The determination of the pole/zero locations and the gain of the CTLE filter are done by using an optimization method developed as a part of this thesis. Initially the pole/zero locations and the gain of the CTLE will be pre-determined. Channel insertion loss will be filtered by the CTLE filter and FIR tap values are calculated. This data is used to produce an equalized pulse sequence by convolving the tap values with the pulse response. The worse-case eye margin is calculated by determining the value of the equalized pulse sequence at the location of the main tap. This process is repeated to maximize the eye margin by changing the location of the poles/zero and the gain of the CTLE filter. The following flow chart under Figure 18 shows this repetitive optimization process.

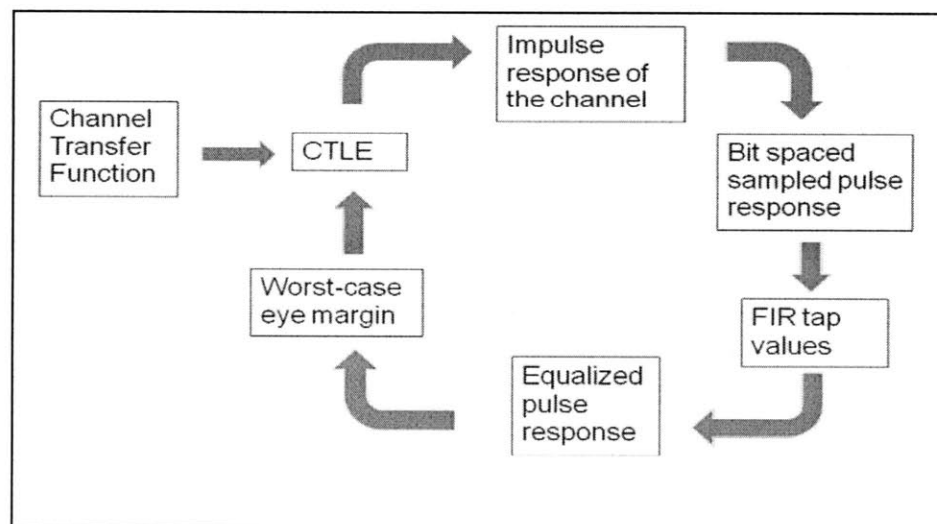


Figure 18: Channel equalization and de-emphasis optimization process

As a comparison to the CTLE equalization method, DFE equalization method was also performed using MATLAB code provided by Prof. Stojanovic [Appendix - 6]. The following flowchart in Figure 19 shows the Decision Feedback Equalization process.

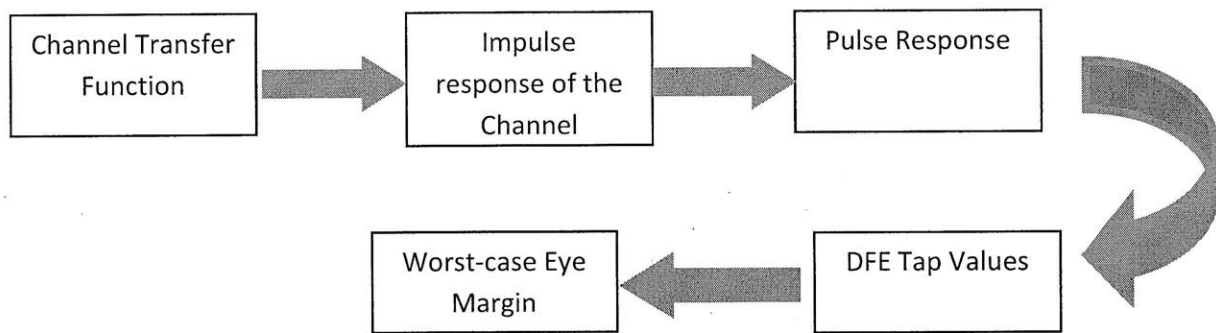


Figure 19: Decision Feedback Equalization process

4. Required Resources

For this thesis topic the following software, hardware and information resources were required:

4.1. Software Resources:

- HSPICE - HSPICE is a device level circuit simulator. HSPICE takes a spice file as input and produces output describing the requested simulation of the circuit.
- Ansoft HFSS – This simulation software allows users to precisely model and simulate 3D electromagnetic fields of high speed, high frequency components.
- Ansoft Via Wizard – Software tool used to create models of vias in a fast and efficient

manner. This software tool generated models which are easy to integrate with the Ansoft HFSS for field simulation.

- MATLAB – MATLAB is a high level language and interactive environment that allows the user to perform computationally intensive tasks in a fast and efficient manner.

4.2. Hardware Resources:

- Network Analyzer – To perform scattering parameter and channel eye measurements
- Two line cards and backplane.

4.3. Information Resources:

It is important to have some understanding of signal integrity issues when working on this proposed thesis. Several books were read to obtain a clearer understanding of return and insertion losses, jitter tolerance, cross-talk etc [18-22].

5. Results

Various results were generated throughout the course of the thesis work. These results can be categorized under several groups: The HFSS model analysis, Gen 2.0 HSPICE simulation results, Gen 3.0 transmitter de-emphasis and receiver equalization results. Each section will be presented in detail below.

5.1. HFSS model analysis:

As mentioned under the technical approach of this thesis, HFSS trace and via models were evaluated in frequency domain and the insertion loss and the port impedances were analyzed. The following table 4 displays the port impedance of the simulated model along with the expected port impedances.

Simulation Model	Expected Port Impedance (Ω)	Simulated Port Impedance (Ω)
Differential stripline trace: 6mil wide, FR408HR material	85.2	83.22
Differential capacitor IOXM board via: L1-L3 signal trace, FR408HR	85.2	84.64
Differential connector IOXM board via: L1-L3 signal trace, FR408HR	85.2	84.73
Differential connector backplane via: L1-L3 signal trace, FR408HR	95.9	91.67
Differential capacitor IOXM board via: L1-L12 signal trace, FR408HR	85.2	84.75
Differential connector IOXM board via: L1-L12 signal trace, FR408HR	85.2	84.65

Table 4: Port impedances of HFSS simulation models

Analyzing the insertion loss of the HFSS trace models for a frequency range of 0GHz – 20GHz with a trace length of 100 mils indicates that the insertion loss at 4GHz is 0.0611 dB for FR408HR material and 0.1297 dB for FR4 material. The insertion loss is more than double for the FR4 dielectric.

5.2. Gen 2.0 HSPICE simulation results:

Simulations for Gen 2.0 model were performed to test several design parameters: board material, via backdrilling, via breakout. These results are listed below.

- Board material: The simulations were performed using two different board materials; FR408HR and FR4. The dielectric constant of the FR408HR is 3.75 with a loss tangent of 0.01. The dielectric constant of FR4 is 4.4 with a loss tangent of 0.021. The Figure 20 depicts the eye diagrams of a 23” channel made with these two different board materials.

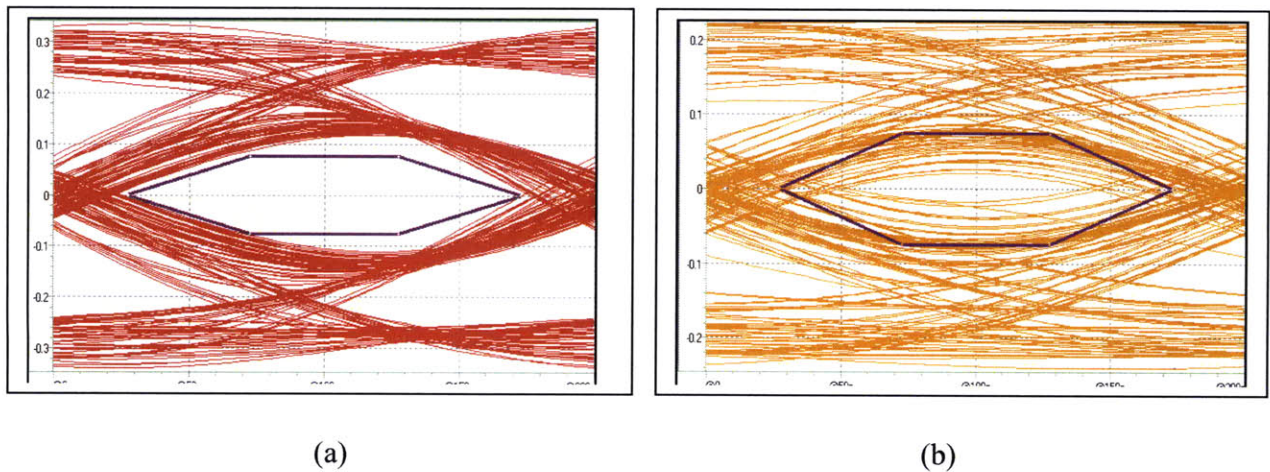


Figure 20: The effect of board material on the signal quality with eye mask for Gen 2.0. (a) The receiver eye pattern of FR408HR board material and (b) The receiver eye pattern of FR4 board material

- Via backdrilling: Backdrilling is the process of removing a part of the via stub to enhance the signal quality. The Figure 21 depicts a cross section of the via with backdrilling and without backdrilling.

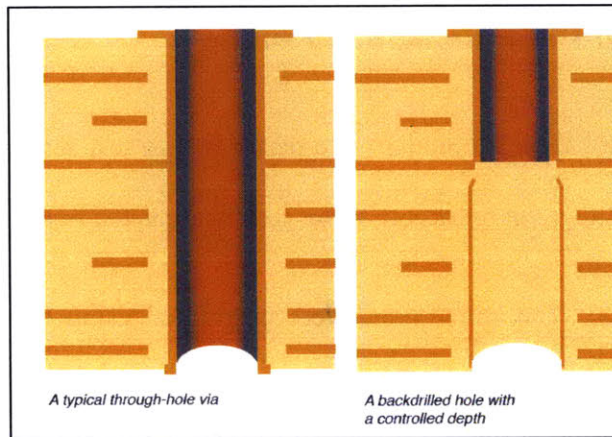


Figure 21: Via with and without backdrilling

Figure 22 shows the insertion loss of a interface with backdrilling of the vias on the backplane and one without backdrilling of the vias on the backplane for a 15" channel on the backplane made of FR408HR material.

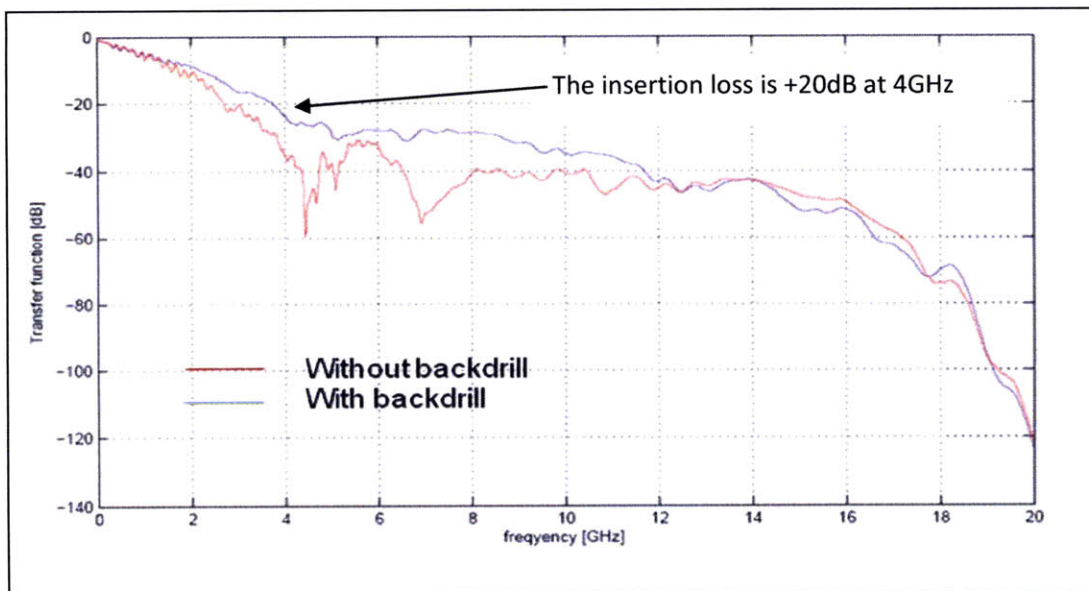
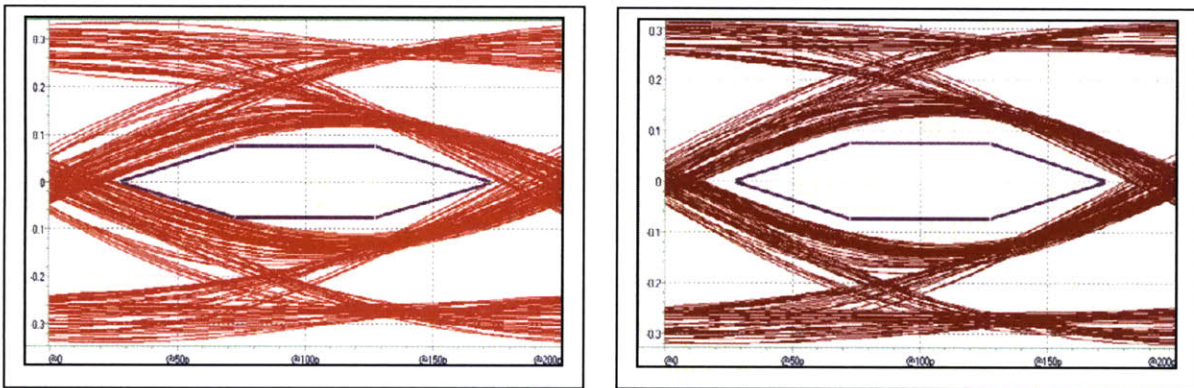


Figure 22: Effect of back drilling of the vias.

Note the significant at 4GHz and 7GHz on the channel without backdrilling.

- Via breakout: The layer on which the signal travels can have an effect on the signal quality, especially if there are vias on the path of the trace. This is due to the fact that the via stub will be greater if the signal trace is on a higher layer than at the bottom layer of the stackup. Simulations were performed to analyze which signal layer on the IOXM board and the backplane corresponds to the best signal quality. The following Figure 23 depicts the receiver eyes of two channels with 15” on the backplane made of FR408HR material. The first channel has the signal on layer 3 throughout the channel. The second channel has the signal trace on layer 12 on the IOXM board and layer 3 on the backplane.



Eye at the receiver at layer 3 throughout

Eye at the receiver at layer 12 on the backplane

Figure 23: Effect of via stub length

5.3. Gen 3.0 Transmitter de-emphasis and Receiver equalization results:

The starting point of the pole/zero locations and gain of the CTLE filter was determined by first understanding that we require a high pass filter around our frequency of interest, which is 4GHz. By observing the CTLE filter suggested by PCI-SIG mentioned under the technical

approach of this thesis, the zero should occur prior to 4GHz, the first pole should occur around 4GHz and the second pole should be located after 4GHz. The gain can be determined by observing the S21 parameter of the channel. Based on these deductions, the pole and zero locations and the gain of the CTLE filter were first determined to be at:

$$\text{Zero} = 1\text{GHz};$$

$$\text{Pole 1} = 2\text{GHz};$$

$$\text{Pole2} = 8\text{GHz};$$

$$\text{Gain} = 0.5$$

The MATLAB [Appendix -5] was run with these initial values. Then the code execution was repeated to optimize the eye margin. The final results yielded the worst-case eye margin of 10.4mV for the following variable values:

$$\text{Zero} = 0.64 \text{ GHz}$$

$$\text{Pole 1} = 4.9 \text{ GHz} \quad \text{Pole 2} = 14.7 \text{ GHz}$$

$$\text{Gain} = 3/22$$

$$\text{FIR tap values} = 0.7266, -0.0829, 0.1905$$

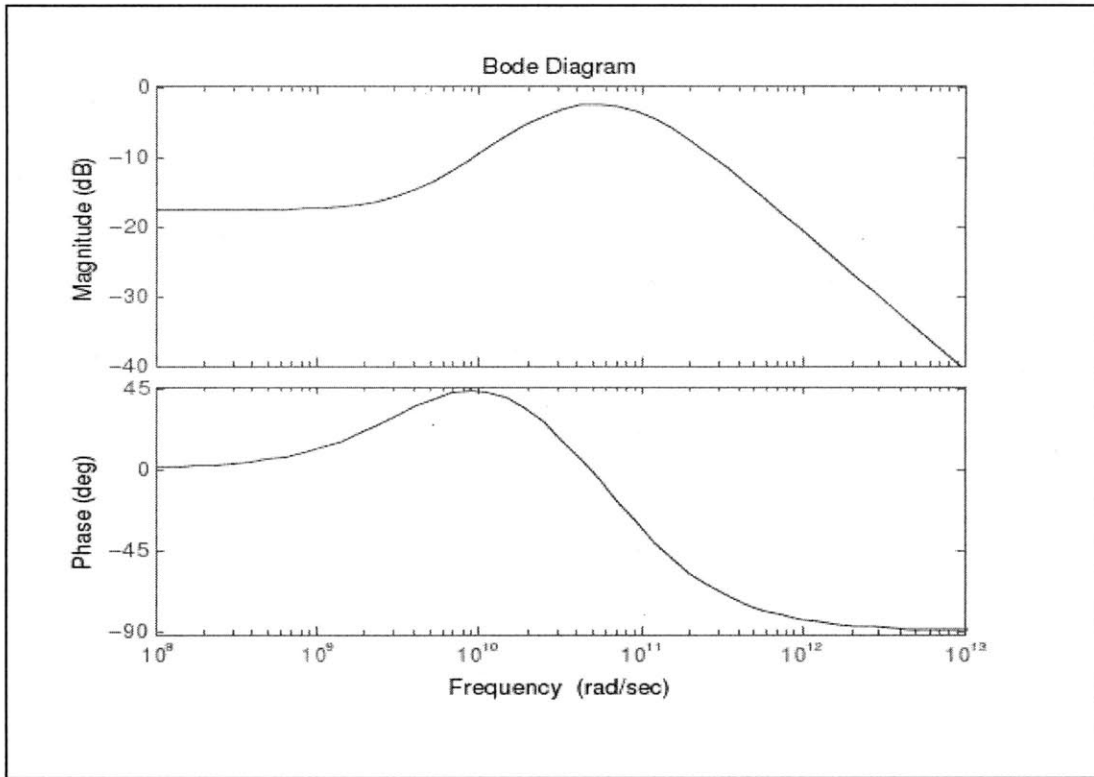


Figure 24: Bode plot of the CTLE filter

The Figure 24 shows the magnitude and the phase values of the CTLE filter. The standard values of circuit elements were calculated using the CTLE pole, zero and gain information. These values are given under table 5 below.

Element name	Standard Value	Tolerance
C1	39 pF	+/- 10%
C2	10 pF	+/- 10%
R1	6.8 Ω	+/- 5%
R2	1.0 Ω	+/- 5%
R3	1.1 Ω	+/- 5%

Table 5: Values of the circuit elements of the CTLE filter

By applying these values, the first pole is located at 4.681 GHz and the second pole is located at 14.4 GHz and the zero is located at 0.60013 GHz. A worst-case eye margin of 10.0mV can be obtained by applying a CTLE filter with the above component values. Further analysis was performed to calculate the worst-case eye margin when the component values vary according to their tolerance. Table 6 below highlights this data.

Element Name	Tolerance	Worst Case Eye Margin
C1	+ 10%	8.9 mV
	-10%	8.7 mV
C2	+10%	9.4 mV
	-10%	8.8 mV
R1	+5%	8.0 mV
	-5%	10 mV
R2	+5%	9.1 mV
	-5%	9.1 mV
R3	+5%	9.7 mV
	-5%	8.4 mV

Table 6: Worst-case eye margin with the component tolerance

The Figure 25 below depicts the receiver eye for a 23” channel with 15” on the backplane made from FR408HR material. This Figure contains the receiver eye generated using the MATLAB code with the newly derived FIR tap values at the transmitter end.

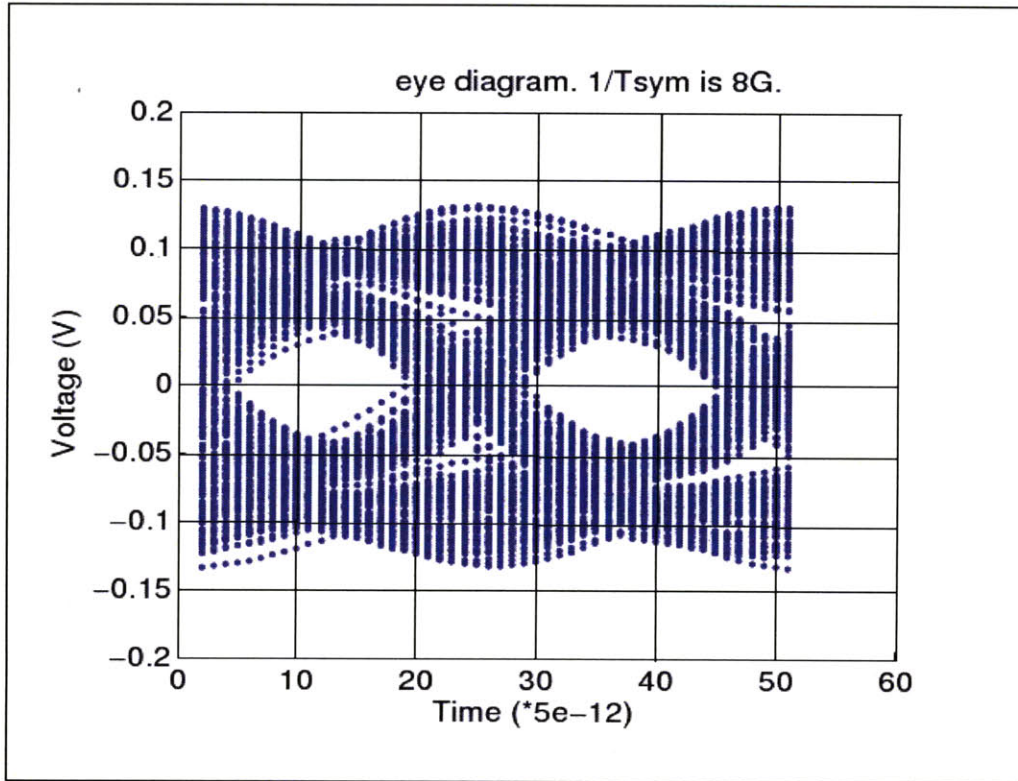


Figure 25: Eye diagram at the receiver end with CTLE filtering

The Decision Feedback equalization technique yielded a worst-case eye height of 67.3 mV for the input channel transfer function. The DFE tap values were -0.1925, 0.6223, and -0.1853. The eye diagram of the simulation is depicted in Figure 26 and the DFE eye closure probability is depicted in Figure 27.

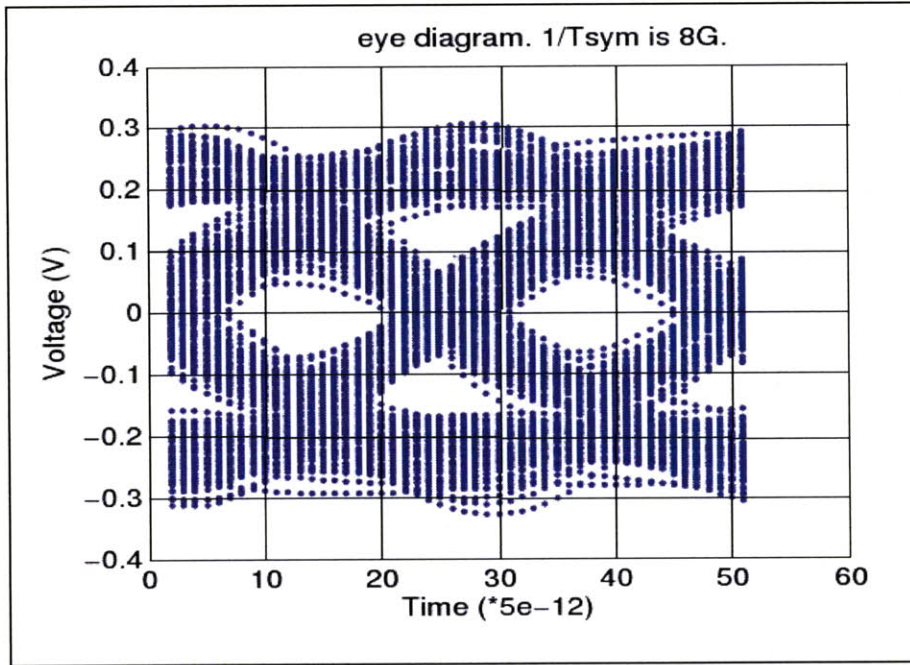


Figure 26: Eye diagram at the receiver end with DFE

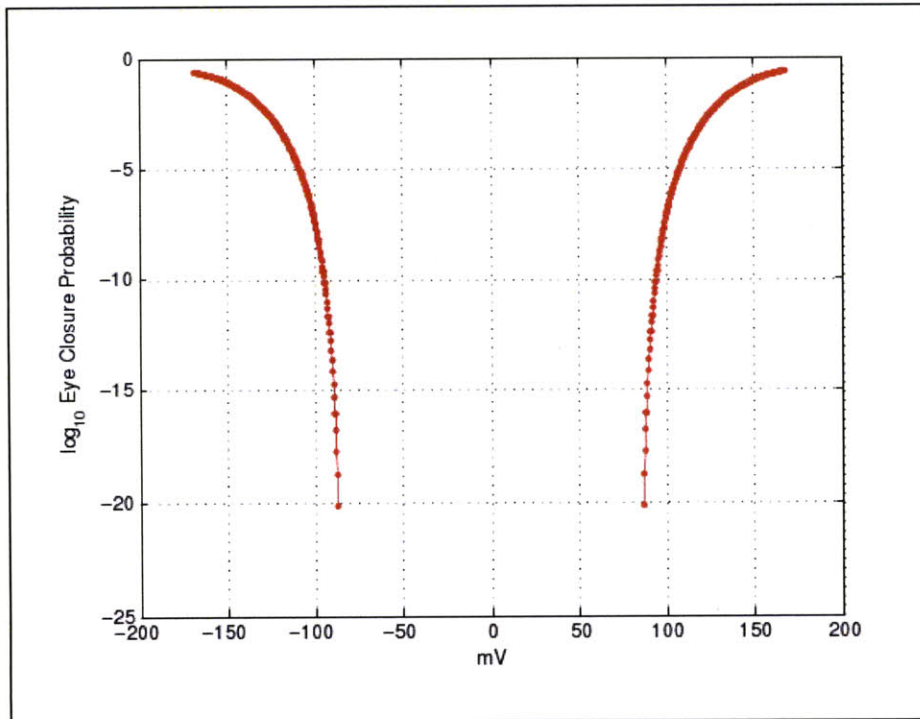


Figure 27: DFE eye closure probability

6. Discussion

The goal of this thesis was to design and simulate a PCIe Gen 3.0 communication channel. Several key lessons and observations were made in the process of developing simulation models of traces and vias and simulating them in HSPICE to achieve this end goal. Some of these key lessons will be discussed under this section.

The first observation is that the accuracy of the HFSS models is critical to generating a valid communication channel. It is imperative to select a valid range for frequency analysis in HFSS which relates to the time domain simulation the user intends to run in HSPICE. The HSPICE results are invalidated if the frequency sweeps of the HFSS models are chosen incorrectly because there is not a valid transformation from frequency domain to time domain.

The second observation is that the mode of data extraction from HFSS into HSPICE can influence the final result. In this thesis the trace models were incorporated into HSPICE as w-elements and via models were incorporated as S-elements.

The third observation is that it is important to select the right settings for the driver model in HSPICE. By selecting incorrect de-emphasis values or power values, the end result will be incorrect.

From the simulation results of Gen 2.0 several design guidelines can be made which can predict the behavior of the Gen 3.0 channel. From the results we can observe that the board material used during design stage can affect the overall length of the channel. By using the FR408HR material we can effortlessly meet the eye mask for a 15" long channel on the

backplane. However, if we use FR4 material, the 15” channel fails the eye mask test for Gen 2.0. By using FR4 material we were only able to reach a channel length of 10” on the backplane. The reason for this difference is due to the dielectric loss of the material. FR408HR has a smaller dielectric loss compared to FR4 material. Therefore it results in having an insertion loss which is approximately half the insertion loss of the FR4 material.

The second observation one can make from the simulation results is that backdrilling of the via stub can affect the signal amplitude. Since long via stubs means reflection loss due to stub capacitance, this observation is not too surprising. As seen by the insertion loss of the system shown in Figure 19 on page 33, the presence of notches in the frequency domain indicates more losses due to this stub capacitance.

From the results, it can be seen that the via breakouts determine the via lengths and stub capacitance. This can also affect the signal quality through the channel. It was observed that signal trace on layer 12 on the IOXM board and the signal trace on layer 3 with backdrilling on the backplane results in maximum eye height.

The MATLAB analysis on the Gen 3.0 channel was performed to determine the FIR tap values and the appropriate CTLE coefficients. The precursor and main tap values (-0.1007 and 0.7278 respectively) were within the range specified by the PCI-SIG organization. However the post cursor tap value of 0.1715 was not within the 0 to -0.35 range specified by PCI-SIG. One possible reason for this is that the CTLE amplifies most of the high frequency components. Therefore the transmitter tap values are not typical pre-emphasis values. Another reason for this is that the S21 parameters of the channel depicted a very high loss in the high frequency range. This loss is mainly due to the loss incurred by the connector model used in this

channel .The following Figure 28 shows a 15” backplane channel made with FR408HR material with and without the connector models. The CTLE coefficients denote a filter that shows high pass behavior around the 4GHz frequency value.

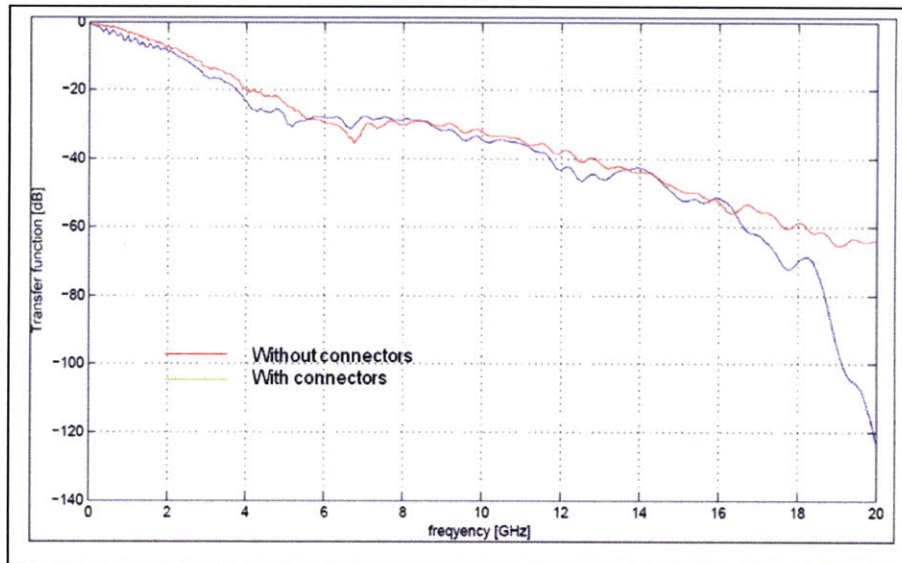


Figure 28: Effect of the connector model

The decision feedback equalization generated a worst-case eye margin of 67.3 mV, far greater than the minimum required height of 10mV. The tap precursor and post-cursor tap values were both within the expected margin of 0 to -0.25 and 0 to -0.35 given by the PCI-SIG organization. Hence DFE is the most optimum mode of equalization for this channel. However, unlike CTLE, DFE does not provide the ease of implementation.

7. Conclusion & Future Work

In this thesis, a high speed communication channel which meets the eye requirements of the PCIe Gen 3.0 standard was designed and simulated. A Gen 2.0 system was modified to function under the new Gen 3.0 speeds. Several design parameters were evaluated for the Gen 2.0 model. It was determined that using FR408HR board material as opposed to FR4 material improves the signal quality of the channel. Furthermore, it was observed that backdrilling of the via stub can improve the signal quality. The optimum via breakout was observed to have the signal trace on layer 12 on the IOXM board (backdilling depth of 85.38 mil) and on layer 3 on the backplane (backdilling depth of 19.23 mil).

While we have achieved the goal of this thesis which is to design and simulate a PCIe Gen 3.0 channel, several additions can be made to improve the performance of the model. In this work we were not able to use transmitter, receiver and connector models that were specifically designed for PCIe Gen 3.0 standards. The reason for this is that these models were not readily available in the industry at the time of this thesis work since the PCIe Gen 3.0 standards are still under development. It would greatly enhance the model if we include these models in the future and observe the functionality of the channel.

Another enhancement we can make to this model is to incorporate better receiver equalization techniques such Decision Feedback Equalization (DFE). The CTLE equalization method which is currently used in the model is a very basic equalization technique which relies on the designer to determine its pole and zero locations. By incorporating better equalization techniques we have demonstrated that we can obtain a higher worst-case eye opening of 67.3

8. References

- [1] J. Boh, "Signal integrity simulation of PCI Express Gen 2 channel," *XrossTalk Magazine*, pp. 16-20, January 2009.
- [2] M. J. Chong, "A PCI Express to PCIX Bridge optimized for performance and area." Master's Thesis, Massachusetts Institute of Technology, 2004.
- [3] Al-Dhahir, N.; Cioffi, J.M.; , "MMSE decision-feedback equalizers: finite-length results," *Information Theory, IEEE Transactions on* , vol.41, no.4, pp.961-975, Jul 1995
- [4] J. Pavlat, "PCI Express Gen2/Gen3 update: An interview with Jasmin Ajnovic and Kevien W. Bross, Intel." April 2009.
<http://www.compactpcisystems.com/articles/id/?3103>.
- [5] A. Kazmi, "PCI Express Gen 3 simplified." February 2009.
<http://embeddeddsp.embedded.com/design/testissue/214502580;jsessionid=E1CLPQC4IOMZUQSNDLPCKHSCJUNN2JVN?pgno=1>
- [6] "About PCI-SIG." 2009. http://www.pcisig.com/membership/about_us
- [7] Synopsys, "HSPICE Signal Integrity Guide." March 2005.
- [8] Synopsys, "HSPICE RF Manual." September 2005.
- [9] Ansoft, "User's Guide – High Frequency Structure Simulator." 2005.
- [10] J. Morrison, G. Talbot, "PCI Express 3.0 Electrical." PCI-SIG Developer Conference, 2009.
- [11] PCI-SIG, "PCI Express Base Specification Revision 2.0." December 2006.
- [12] PCI-SIG, "PCI Express Card Electromechanical Specification Revision 2.0." April 2007.

mV. By using this equalization technique instead of passive CTLE, we can obtain longer channel lengths.

Finally, another design variation to explore is to observe the effect of the trace width on the signal quality. By increasing the width of the trace we can reduce the loss in the channel. However this will create routing problems and cross-talk issues. Therefore it is worthwhile to analyze which trace widths produce the most balanced solution.

- [13] PCI-SIG, "PCI Express Base Specification Revision 3.0 Version 0.7." October 2009.
- [14] Keven R. Andryc, "A novel approach to PCI simulation using ScriptSIM." Master's Thesis, University of Massachusetts Amherst, May 2008.
- [15] Sanquan Song, Vladimir Stojanovi, "Behavioral Simulation of a High-Speed Link Transceiver Using VppSim." July 2008.
- [16] "Differential Signaling." November 2009.
http://en.wikipedia.org/wiki/Differential_signaling
- [17] Idea Consulting, "Differential Stripline Impedance Calculator." 1999.
<http://www.ideaconsulting.com/dstrip.htm>
- [18] E. Bogatin, "Signal Integrity Simplified." Prentice Hall, 2003.
- [19] H. Johnson, M. Graham, "High-Speed Signal Propagation: Advanced Black Magic." Prentice Hall, 2003.
- [20] H. Johnson, M. Graham "High-Speed Digital Design: A Handbook of Black Magic." Prentice Hall, 1993.
- [21] D. Brooks, "Signal Integrity Issues and Printed Circuit Board Design." Prentice Hall, 2003.
- [22] E. C. Ifeachor, B.W. Jervis, "Digital Signal Processing." Prentice Hall, 2001.
- [23] "Common Resistor and Capacitor values for Electronic Circuits." April 2010.
http://www.kennethkuhn.com/students/rlc_values.pdf
- [24] T. Michalka, "Digital Signal Integrity Technical & Non-Technical Considerations." Qualcomm, April 2008.
- [25] M. Kermani, "Pre-layout Simulations: PCIe Gen. 2 Interconnect with 2 Connectors." NetApp, Inc., 2009.

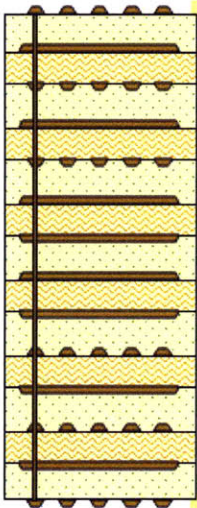
9. Appendix

[1] Backplane PCB layout

Layer Type	Copper Weight	Original Thk	DK	Finished Thk (ML)	Material Information	Vendor	Family
L 1 Signal	0.5oz	0.67			Jada HTE 1/2 oz 0.67mil 23in		FOIL
L 2 Power / Ground	1.0oz	4.10	3.64	3.98	Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
L 3 Signal	1.0oz			1.20	Isola FR408HR 0.25 mm 1/1 0"X0"(10mil) RTF 2x1652	Isola	FR408HR
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
		13.20	3.82	12.18	Isola FR408HR 2116 53% 50in x 125yds	Isola	FR408HR
L 4 Power / Ground	1.0oz	10.00	4.07	1.20	Isola FR408HR 0.25 mm 1/1 0"X0"(10mil) RTF 2x1652	Isola	FR408HR
L 5 Signal	1.0oz			1.20			
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
		13.20	3.82	12.18	Isola FR408HR 2116 53% 50in x 125yds	Isola	FR408HR
L 6 Power / Ground	1.0oz	10.00	4.07	1.20	Isola FR408HR 0.25 mm 1/1 0"X0"(10mil) RTF 2x1652	Isola	FR408HR
L 7 Signal	1.0oz			1.20			
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
		13.20	3.82	12.06	Isola FR408HR 2116 53% 50in x 125yds	Isola	FR408HR
L 8 Power / Ground	2.0oz	5.00	3.71	2.40	Isola FR408HR 0.127 mm 2/2 0"X0"(5mil) 2x1080	Isola	FR408HR
L 9 Power / Ground	2.0oz			2.40			
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
		8.20	3.86	7.72	Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
L 10 Power / Ground	2.0oz	5.00	3.71	2.40	Isola FR408HR 0.127 mm 2/2 0"X0"(5mil) 2x1080	Isola	FR408HR
L 11 Power / Ground	2.0oz			2.40			
					Isola FR408HR 2116 53% 50in x 125yds	Isola	FR408HR
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
		13.20	3.82	12.06	Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
L 12 Signal	1.0oz	10.00	4.07	1.20	Isola FR408HR 0.25 mm 1/1 0"X0"(10mil) RTF 2x1652	Isola	FR408HR
L 13 Power / Ground	1.0oz			1.20			
					Isola FR408HR 2116 53% 50in x 125yds	Isola	FR408HR
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
		13.20	3.82	12.18	Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
L 14 Signal	1.0oz	10.00	4.07	1.20	Isola FR408HR 0.25 mm 1/1 0"X0"(10mil) RTF 2x1652	Isola	FR408HR
L 15 Power / Ground	1.0oz			1.20			
					Isola FR408HR 2116 53% 50in x 125yds	Isola	FR408HR
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
		13.20	3.82	12.18	Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
L 16 Signal	1.0oz	10.00	4.07	1.20	Isola FR408HR 0.25 mm 1/1 0"X0"(10mil) RTF 2x1652	Isola	FR408HR
L 17 Power / Ground	1.0oz			1.20			
					Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
L 18 Signal	0.5oz	0.67	3.64	3.98	Isola FR408HR 2113 55% 50in x 125yds	Isola	FR408HR
					Jada HTE 1/2 oz 0.67mil 23in		FOIL

[2] IOXM board layout

Layer	Cu Thick. (mils)	Cu wt (oz)	DK (1 MHz)	Lam. Thick. (mils)	Description
1	1.80	.5 oz			Foil, .5 oz
2	1.20	1 oz	3.50	3.86	Prepreg FR408 2313
3	1.20	1 oz	3.50	6.00	Core FR408 6mils 2x1080 1 oz / 1 oz 48x36
4	1.20	1 oz	3.50	5.00	Prepreg FR408 1080/1080
5	1.20	1 oz	3.50	6.00	Core FR408 6mils 2x1080 1 oz / 1 oz 48x36
6	1.20	1 oz	3.50	5.00	Prepreg FR408 1080/1080
7	1.20	1 oz	3.50	4.00	Core FR408 4mils 1x3313 1 oz / 1 oz 48x42
8	1.20	1 oz	3.50	14.52	Prepreg FR408 2116/2116/2116
9	1.20	1 oz	3.50	4.00	Core FR408 4mils 1x3313 1 oz / 1 oz 48x42
10	1.20	1 oz	3.50	5.00	Prepreg FR408 1080/1080
11	1.20	1 oz	3.50	6.00	Core FR408 6mils 2x1080 1 oz / 1 oz 48x36
12	1.20	1 oz	3.50	5.00	Prepreg FR408 1080/1080
13	1.20	1 oz	3.50	6.00	Core FR408 6mils 2x1080 1 oz / 1 oz 48x36
14	1.80	.5 oz	3.50	3.86	Prepreg FR408 2313 Foil, .5 oz
Layers	Drill Type	Via Fill	88.64	Thickness over Laminate	
1 - 14	PTH	--	92.24	Thickness over Copper	
			93.24	Thickness of Soldermask	



[3] W-element HSPICE syntax

Wxxx input1 input2 ... inputN inputReference
+ output1 output2 ... outputN outputReference
+N=val L=val
+ <RLGCMODEL=name or RLGCFILE=name or UMODEL=name
+ FSMODEL=name or TABLEMODEL=name or SMODEL=name>
+ [INCLUDERSIMAG=YES|NO FGD=val] [DELAYOPT=0|1|2]
+ <NODEMAP=XiYj...>

Parameter	Description
N	Number of signal conductors
Input1.....inputN	Node name of the near –end reference conductor terminal
inputReference	Input reference
Output1...output	Node name for the far-end reference-conductor terminal
outputReference	Output reference
L	Length of the trace
RLGCMODEL	Name of the RLGS model
RLGCFILE	Name of the external file with the RLGS parameters
UMODEL	Name of the U model
FSMODEL	Name of the file solver model
TABLEMODEL	Name of the frequency dependant tabular model
SMODEL	Name of the S model
INCLUDERSIMAG	Imaginary term of the skin effect
FGD	Cutoff frequency of the dielectric loss
DELAYOPT	Deactivates (0), activates (1), or determines automatically (2).The default is 0.
NODEMAP	String that assigns each index of the S parameter matrix to one of the W Element terminals. This string must be an array of pairs that consists of a letter and a number, (for example, Xn), where <ul style="list-style-type: none"> • X= I, i, N, or n to indicate near end (input side) terminal of the W element • X= O, o, F, or f to indicate far end (output side)

	terminal of the W element. The default value for NODEMAP is "I1I2I3...InO1O2O3...On"
--	--

Example:

Wdimm N=6 in1_pos in1_neg in2_pos in2_neg in3_pos in3_neg gnd

+ out1_pos out1_neg out2_pos out2_neg out3_pos out3_neg gnd

+ tablemodel=cpu85_sl_10_typ L=Length

[4] S-element HSPICE syntax

Sxxx nd1 nd2 ... ndN ndRef
 + <MNAME=Smodel_name> <FQMODEL=sp_model_name>
 + <TYPE=[s|y]> <Zo=[value|vector_value]>
 + <FBASE = base_frequency> <FMAX=maximum_frequency>
 + <PRECFAC=val> <DELAYHANDLE=[1|0|ON|OFF]>
 + <DELAYFREQ=val>
 + <INTERPOLATION=STEP|LINEAR|SPLINE>
 + <INTDATYP =[RI|MA|DBA]> <HIGHPASS=value>
 + <LOWPASS=value> <MIXEDMODE=[0|1]>
 + <DATATYPE=data_string>

Parameter	Description
Nd1...ndN	Nodes of an S element
MNAME	S model name
FQMODEL	.MODEL statement of sp type which defines the frequency behavior
TYPE	Parameter type
Zo	Characteristic Impedance value
FBASE	The base frequency
FMAX	Maximum frequency used in transient analysis
DELAYHANDLE	Delay handle for transmission-line type parameters
DELAYFREQ	Delay frequency for transmission-line type parameter
INTERPOLATION	Interpolation method
INTDATYP	Data type for the linear interpolation of the complex data
HIGHPASS	Method of extrapolating higher frequency data points
LOWPASS	Method of extrapolate lower frequency points
MIXEDMODE	Set to 1 if the parameters are represented in the mixed mode
DATATYPE	A string used to determine the order of the indices of the mixed-signal incident or reflected vector

Example:

```
Svia1 pkg1_pos2 pkg1_neg2 pkg1_neg1 pkg1_pos1 pkg1_pos3 pkg1_neg3
```

```
+ via1_pos2 via1_neg2 via1_neg1 via1_pos1 via1_pos3 via1_neg3 gnd
```

```
+ mname=s_via_IOXM_cap
```

```
.model s_via_IOXM_cap S TSTONEFILE='ViaCap_6mil_IOXM_L1L3_3pairs.S12P'
```

[5] PCIe Gen 3.0 Matlab code for CTLE optimization and FIR tap coefficients for transmitter de-emphasis

Main code:

```

%%% Test bench for High-Speed Link Example %%%
clear all; close all;
% addpath('/home/sanquan/matlab/high-speed-link');
%%% Create channel response for the simulator %%%

channelName='C:\Documents                               and                               Settings\dilini\My
Documents\matlab\results.s4p';

mode='s21';
[f,Hinit]=extract_mode_from_s4p(channelName,mode);

Figure(10)
plot(f*1e-9,20*log10(abs(Hinit)), 'b');           xlabel('frequency           [GHz]');
ylabel('Transfer function [dB]'); grid on;

%%%%%%%%%%%% CTLE code - DILINI %%%%%%%%%%%%%
pole2 = 11.62e9;
pole1 = 4.3e9;
zero = 0.7e9;

ctle_filter = pole2*(zero + 1i*f) ./ ((pole1 + 1i*f) .* (pole2 + 1i*f)); %
CTLE filter
H = ctle_filter.*Hinit; % Hinit is the transfer function produced by the
extract_mode_from_s4p function

%%%%%%%%%%%% end of CTLE code - DILINI %%%%%%%%%

Tsym=1250e-12*log2(2);
Ts=5e-12;
delta=1E-4;
vari=5E-5;
imp=xfr_fn_to_imp(f,H,Ts,Tsym); %%% Ts sampled impulse response

nsym_short=500; %%% persistence of the impulse response tail in the channel
in terms of the number of symbols

imp_short=[imp(1:min(floor(nsym_short*Tsym/Ts), length(imp)))];
save imp_short.mat    imp_short;

%%%%%%%%%%%%
%%% reduce the length of imp, imp_fext, imp_next to a bearable level.
delta_num=40;
i=1;
while (i<length(imp) && max(abs(imp(i:length(imp))))>delta/delta_num)
    i=i+200;
end

```

```

i=min(i, length(imp));
imp_short=imp(1: i);
i=1;

imp=imp_short;

eq_tap_length=3;
eq_tap_pre=1;

pul=ones(1, floor(Tsym/Ts));
[eq_taps, pulse_resp_Ts]=func_zfe(imp, pul, Tsym, Ts, eq_tap_length, eq_tap_pre, 'y');
[tmp, index_max]=max(pulse_resp_Ts);

pul_resp=conv(imp, pul);
[x, index_max]=max(pul_resp);
index=max(1, index_max-floor(index_max/(Tsym/Ts))*floor(Tsym/Ts)):
floor(Tsym/Ts): length(pul_resp);
p=pul_resp(index);

figure
plot(p, 'b.-')
xlabel('Symbol time')
ylabel('Pulse response')
xlim([0 500]);
% func_eye(imp, eq_taps, pul, 1000, Tsym, Ts, 'PAM2');

eq_taps % output the FIR tap values

%%%%%%%%% eye margin calculation - DILINI %%%%%%%%%%%%%%%

peq = conv(eq_taps,p); % equalized pulse response
[maintap, index_maintap] = max(peq);
eye_seq = 2*maintap - sum(abs(peq)) % generates the worse-case eye
margin

end

```

The Extract_mode_from_s4p function code:

```

function [f,H]=extract_mode_from_s4p(filename,mode)
%%% read in s parameters from four port s4p files %%%
% input
% filename
% mode {s11,s21,s31,s41,...,s14,s24,s34,s44}
% output
% f - frequency in Hz
% H - transfer function

% Extract units and parameter types - look for the line that starts with #
[first_char]=textread(filename,'%c*[\n]');
headerlines=find(first_char=='#')-1;
[freq_unit,sim_type,param_type,z_term_type,z_term_value]=textread(filename,'#
%s%s%s%s%f*[\n]',1,'headerlines',headerlines);

```

```

freq_unit=char(freq_unit);
param_type=char(param_type);
% Position to the beginign of the number array and extract number from file
headerlines=max(find(first_char=='!'));
[col0,col1,col2,col3,col4,col5,col6,col7,col8]
textread(filename,'%f%f%f%f%f%f%f%f', 'headerlines',headerlines);

freq_index=1:4:length(col0);
s1_index=1:4:length(col0);
s2_index=2:4:length(col0);
s3_index=3:4:length(col0);
s4_index=3:4:length(col0);

f=col0(freq_index)';
switch upper(freq_unit)
    case 'HZ'
        scaleF=1;
    case 'KHZ'
        scaleF=1e3;
    case 'MHZ'
        scaleF=1e6;
    case 'GHZ'
        scaleF=1e9;
    otherwise
        error('Unknown frequency unit %s',freq_unit);
        return
end
f=f*scaleF; % frequency in Hz

switch upper(mode)
    case 'S11'
        param1=col1(s1_index)';
        param2=col2(s1_index)';
    case 'S21'
        param1=col0(s2_index)';
        param2=col1(s2_index)';
    case 'S31'
        param1=col0(s3_index)';
        param2=col1(s3_index)';
    case 'S41'
        param1=col0(s4_index)';
        param2=col1(s4_index)';
    case 'S12'
        param1=col3(s1_index)';
        param2=col4(s1_index)';
    case 'S22'
        param1=col2(s2_index)';
        param2=col3(s2_index)';
    case 'S32'
        param1=col2(s3_index)';
        param2=col3(s3_index)';
    case 'S42'
        param1=col2(s4_index)';
        param2=col3(s4_index)';
    case 'S13'
        param1=col5(s1_index)';
        param2=col6(s1_index)';

```

```

    case 'S23'
        param1=col4(s2_index)';
        param2=col5(s2_index)';
    case 'S33'
        param1=col4(s3_index)';
        param2=col5(s3_index)';
    case 'S43'
        param1=col4(s4_index)';
        param2=col5(s4_index)';
    case 'S14'
        param1=col7(s1_index)';
        param2=col8(s1_index)';
    case 'S24'
        param1=col6(s2_index)';
        param2=col7(s2_index)';
    case 'S34'
        param1=col6(s3_index)';
        param2=col7(s3_index)';
    case 'S44'
        param1=col6(s4_index)';
        param2=col7(s4_index)';
    otherwise
        error('unknown mode %s', mode);
end

switch upper(param_type)
    case 'MA'
        H=param1.*exp(j*param2*pi/180);
    case 'RI'
        H=param1+j*param2;
    case 'DB'
        H=10.^(param1/20).*exp(j*param2*pi/180);;
    otherwise
        error('unknown parameter type %s',param_type)
end

```

The xfr_fn_to_imp function code:

```

function imp=xfr_fn_to_imp(f,H,Ts,Tsym)
%%% Create impulse response from transfer function in frequency domain  %%%
%%% Impulse response is interpolated to the sample time required by the  %%%
%%% simulator
% input
% f - frequency in Hz
% H - transfer function
% Ts - simulator sample time
% Tsym - simulator symbol period
% output
% imp_response

num_fft_pts=2^12;

% set the symbol frequency
f_sym=1/Tsym;
% get the maximum sampling frequency from the transfer function

```

```

f_sym_max=2*max(f);
% stop the simulation if the symbol frequency is smaller than the maximum
% measured sampling frequency
if (f_sym > f_sym_max),
    error('Max input frequency too low for requested symbol rate, can't
interpolate!');
end

f_sym_max=f_sym*floor(f_sym_max/f_sym);
Hm=abs(H);
Hp=angle(H);

%%% need to force phase to zero at zero frequency to avoid funky behavior
if f(1)==0,
    Hm_ds=[fliplr(Hm(2:end-1)) Hm];
    Hp_ds=[fliplr(-Hp(2:end-1)) Hp];
    fds=[-fliplr(f(2:end-1)) f];
    fds_m = fds;
    fds_p = fds;
else
    Hm_ds=[fliplr(Hm(1:end-1)) Hm];
    Hp_ds=[fliplr(-Hp(1:end-1)) 0 Hp];
    fds_m=[-fliplr(f(1:end-1)) f];
    fds_p=[-fliplr(f(1:end-1)) 0 f];
end

df=f_sym_max/2/num_fft_pts;
f_ds_interp=-f_sym_max/2+df:df:f_sym_max/2;
Hm_ds_interp=spline(fds_m,Hm_ds,f_ds_interp);
Hp_ds_interp=spline(fds_p,unwrap(Hp_ds),f_ds_interp);

Hm_ds_interp_sh=fftshift(Hm_ds_interp);
Hp_ds_interp_sh=fftshift(Hp_ds_interp);

H_ds_interp_sh=Hm_ds_interp_sh.*exp(j*Hp_ds_interp_sh);

% impulse response from ifft of interpolated frequency response

imp=ifft(H_ds_interp_sh);
imp_r=real(imp);

dt_sym=1/f_sym_max;

%refit data into simulator's time step
dt_time=0:dt_sym:dt_sym*(length(imp_r)-1);
time = 0:Ts:dt_time(end);
imp = interp1(dt_time, imp_r, time, 'spline')*Ts/dt_sym;

return

```

The func_zfe code:

```

function [w, pulse_resp_Ts]=func_zfe(imp, pul, Tsym,Ts,L,num_pre, str)
%%% Computes linear zero-forcing equalizer coefficients
%%% with peak power constraint scaled to one

```

```

%%% inputs
% imp - channel impulse response
% pul - symbol shape
% Ts - impulse response time quantization
% Tsym - symbol duration
% L - equalizer length
% num_pre - number of precursors in the equalizer
%%% output
% w - equalizer coefficients

pulse_response_data=[zeros(1,floor(Tsym/Ts)) pul zeros(1,floor(Tsym/Ts))];
pulse_response = conv(imp,pulse_response_data);

[foo,index_max]=max(pulse_response);
symbol_index = round( (index_max-
floor(index_max*Ts/Tsym)*Tsym/Ts):floor(Tsym/Ts):length(pulse_response));
if symbol_index(1)==0
    symbol_index=symbol_index(2:length(symbol_index));
end
p=pulse_response(symbol_index);

if str== 'y'
    figure(11)
    hold off; plot(Ts*(1:length(pulse_response)), pulse_response,'r-');
    hold on; plot(Ts*symbol_index, p,'y.');
```

end

```

[foo,main_index]=max(p);
main_eq_index=main_index+num_pre-1;
delta_vec=zeros(1,L+length(p)-1);
delta_vec(main_eq_index)=1;
for n=1:L,
    P(n,:)=[zeros(1,n-1) p zeros(1,L-n)];
end
w=(inv(P*P')*P*delta_vec)';
w=w/sum(abs(w));
PP=w*P;

if str== 'y'
    figure(12)
    subplot(2,1,2);
    tmp=w*P;
    [x,k]=max(tmp);
    mainHmV=x*1000*0.5
    resISI=tmp;

    stem(tmp(1:min(k+100, length(tmp))));
    title('Equalized symbol spaced pulse response');
    subplot(2,1,1);
    [x,k]=max(p);
    stem(p(1:min(k+100, length(p))));
    title('Unequalized symbol spaced pulse response')
end

% save response.mat PP(1:1000);
PPshort = PP(1:200);
save response.mat PPshort;
```



```

figure(11)
plot(PPshort, 'b.-')

a=length(w);
tmp=w(1)*pul;
for i=2:a
    tmp=[tmp,w(i)*pul];
end

pulse_resp_Ts=conv(tmp, imp);

```

The func_eye code:

```

function func_eye(imp, w, pul, K, Tsym, Ts, str)
% imp:      impulse response
% pul:      pulse shape (length of Tsym/Ts )
% w:        equalizer coefficient
% K:        number of data to be simulated
% Tsym:     length of a symbol
% Ts:       length of an sample

[x,index_max]=max(abs(conv(imp, pul)));
data=round(rand(1,K)).*2-1;

data=conv(data,w);
symbols=zeros(1, length(data)*length(pul)+100);
symbols(1:length(pul))=[data(1)*pul];
for i=2:length(data)
    symbols((i-1)*length(pul)+1: i*length(pul))=[data(i)*pul];
end
%out=symbols;
out=conv(symbols, imp); %./3;
out=out(index_max+round(80*round(Tsym/(2*Ts))):index_max+ceil(K*0.99)*round(Tsym/(2*Ts)));
index=1:length(out);

figure
axes('FontSize', 15);
hold on
title([str ' eye diagram. 1/Tsym is ' num2str(1/Tsym/1E9) 'G.'], 'FontSize', 15);
plot(index-floor(index/(2*Tsym/Ts))*2*floor(Tsym/Ts)+1, out, '.');
xlabel(['Time (*' num2str(Ts) ')'], 'FontSize', 15);
ylabel(['Voltage (V)'], 'FontSize', 15);

```

[6] Decision Feedback code

```
function [w, peq]=func_zfe_p(p,L,num_pre,eq_taps_dfe,str)
%% Computes linear zero-forcing equalizer coefficients
%% with peak power constraint scaled to one
%% inputs
% p - pulse response
% Ts - impulse response time quantization
% Tsym - symbol duration
% L - equalizer length
% num_pre - number of precursors in the equalizer
%% output
% w - equalizer coefficients

% clear all; close all;clc;
% Tsym=100e-12; Ts=Tsym/10; L=30; num_pre=10;
% imp=load('link_top\link_channel.dat','ASCII');

[foo,main_index]=max(p);
% main_eq_index=main_index+num_pre-1;
main_eq_index=main_index+num_pre;
delta_vec=zeros(1,L+length(p)-1);
delta_vec(main_eq_index)=1;
for n=1:L,
    P(n,:)=zeros(1,n-1) p zeros(1,L-n);
end

%% DFE
Pd = [P(:,1:main_eq_index) P(:,main_eq_index+eq_taps_dfe+1:end)];
delta_vecd = [delta_vec(1:main_eq_index)
delta_vec(main_eq_index+eq_taps_dfe+1:end)];

w=(inv(Pd*Pd')*Pd*delta_vecd)';
w=w/sum(abs(w));
peq=w*P;
peq(main_eq_index+1:main_eq_index+eq_taps_dfe)=0;

% keyboard

if str=='y'
    figure(12)
    subplot(2,1,2);
    [x,k]=max(peq);
    mainHmV=x*1000
```

```

resISI=peq;
% resISI(k)=0;

plot(peq(1:min(k+100, length(peq))), 'r.-');
title('Equalized symbol spaced pulse response');
grid on;

subplot(2,1,1);
[x,k]=max(p);
plot(p(1:min(k+100, length(p))), 'b.-');
title('Unequalized symbol spaced pulse response');
grid on;

[pISIb,vISIb]=distISI(resISI,2);
vISI = vISIb(find(vISIb>0));
pISI = pISIb(find(vISIb>0));
% sigmaISIdata=sqrt(sum(pISI.*vISI.^2));
% index0=find(vISI==0);
% pGauss=0.5*erfc([vISI(index0:end) vISI(index0:end)+
vISI(end)]/sqrt(2)/sigmaISIdata);
% keyboard;
[foo,indexMax] = min(abs(vISI-mainHmV/1000));
pCumISI=cumsum(pISI(1:indexMax));
% keyboard

sigmaN = 10.^(-5:0.05:0);
for k=1:length(sigmaN),
    ber(k) = sum(pISI*erfc(vISI/sqrt(2)/sigmaN(k)));
end
% ber
figure(100)
hold off; semilogy(20*log10(sigmaN),ber,'g')
xlabel('Noise rms [dBV]'); ylabel('log_{10} BER');
ylim([10.^[-20 0]]);

figure(14)
plot(1000*[-fliplr(vISI(1:indexMax)) vISI(1:indexMax)], [fliplr(log10(pCumISI))
log10(pCumISI)], 'r.-');
xlabel('mV'); ylabel('log_{10} Eye Closure Probability');
grid on;
end

```

[7] Altered Verilog A code

NOTE: extracted from thPCIe Bosboro-Ex rev. 10 code

```
*****
*** Tx 1Pair Subckt
*****
.subckt Tx_1pair_bhv wdata3 wdata4 pad12_pos pad12_neg vtxpwr vtxvss

.param f3db = 70e9                $ Input to driver Bessel filter
.param zo=1
.param base= 'abs (p2)+abs (p3)+abs (p4)+abs (p5)+abs (p6)'          $ DC value
.param p1= -0.1007                $ the altered cursor value
.param p2= 0.7278                 $ the altered cursor value
.param p3= 0.1715                 $ the altered cursor value
.param p4= 0                      $ other post cursorset to 0
.param p5= 0                      $ other post cursorset to 0
.param p6= 0                      $ other post cursorset to 0
.param idrvm = 1

*****
*** 2-tap Tx Linear Equalizer
*****
Xeq_pos in_vic_pos drv1_pos vtxvss module_lib_x6tapfir1
+ eq_tdelay1='eq_tdelay1' base='base' idrvm=1 p1='p1' p2='p2'

Xeq_neg in_vic_neg drv1_neg vtxvss module_lib_x6tapfir1
+ eq_tdelay1='eq_tdelay1' base='base' idrvm=1 p1='p1' p2='p2'

*** Verilog-A Module
*module module_lib_x6tapfir1 (in1, out1, gnd1);
```