INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

MAESTRÍA EN DISEÑO ELECTRÓNICO

FREQUENCY DOMAIN OPTIMIZATION BASED METHODOLOGY TO ACCELERATE HIGH SPEED DIGITAL INTERCONNECT DESIGN

Tesis que para obtener el grado de

MAESTRO EN DISEÑO ELECTRÓNICO

Presenta: Juan Antonio Robledo Mariscal

Director de tesis: Dr. José Ernesto Rayas Sánchez

Tlaquepaque, Jalisco. Marzo de 2017.

MAESTRO EN INGENIERIA (2017) ITESO

Maestría en Diseño Electrónico Tlaquepaque, Jal., México

TÍTULO: **Frequency Domain Optimization Based Methodology to Accelerate High Speed Digital Interconnect Design**

AUTOR: Juan Antonio Robledo Mariscal
Ingeniero en Comunicaciones y Electrónica (Universidad de Guadalajara, México)

DIRECTOR DE TESIS: José Ernesto Rayas Sánchez
Departamento de Electrónica, Sistemas e Informática, ITESO
Ingeniero en Electrónica (ITESO, México)
Maestro en Sistemas Electrónicos (ITESM Campus Monterrey, México)
Doctor en Ingeniería Eléctrica (Universidad McMaster, Canadá)
Senior, IEEE

NÚMERO DE PÁGINAS: viii, 68

Dedication

I dedicate my thesis to my wife Yajaira Flores and my daughter Maria Fernanda. All time I invested in completing the Master Degree program and the thesis is time I lost sharing love with them.

I also dedicate my thesis to my mother Maria Teresa Mariscal. I could not finish listing all her attributes, all I am is because of her. During this tough year, along with my wife and daughter, they have supported me all the time, they were my loyal companion in the worst moments. For these reasons I owe them all my devotion.

I want to dedicate my thesis to my father Rigoberto Robledo as well. He taught me the value of the hard work and discipline.

Finally, I also dedicate my thesis to my grandfather Guillermo Mariscal. The wisest man I have ever met, righteous, religious and humble. Man advocated to his family during his long life.

Dedicatoria

Dedico mi tesis a mi esposa Yajaira Flores y a mi hija Maria Fernanda. Todo el tiempo que invertí en completar el programa de la Maestría y la tesis es tiempo que perdí compartiendo mi amor con ellas.

También dedico mi tesis a mi madre Maria Teresa Mariscal. No podría terminar de listar todas sus cualidades, todo lo que soy es por ella. Durante este difícil año, junto con mi esposa e hija, ellas me han apoyado en todo momento, ellas fueron mi leal compañía en los peores momentos. Por estas razones les debo toda mi devoción.

Quiero dedicar mi tesis también a mi padre Rigoberto Robledo. El me enseñó el valor del trabajo arduo y la disciplina.

Finalmente, también dedico mi tesis a mi abuelo Guillermo Mariscal. El hombre más sabio que he conocido, justo, religioso y humilde. Hombre dedicado a su familia durante toda su vida.

Summary

This thesis presents two signal integrity (SI) optimization methodologies for finding spec-compliant designs of multi-Gbps interconnects; both methods are based on classical numerical optimization formulations. The first method is a time domain (TD) implementation, suited for relatively small topologies that intrinsically require short simulation time. The second method is a novel frequency domain (FD) implementation, targeted for relatively large topologies. Long topologies demand extensive simulation time in classical TD SI methods. The FD-based optimization method is developed to reduce considerably simulation time with respect to the TD-based implementation. Following the topics presented in this thesis work, the reader is introduced with chapters that are the foundation for the discussed optimization methodologies. A general description for current SI methods is given initially. Then, a non-spec compliant topology example is presented, describing the high speed digital channel and associated simulation parameters; also, describing detailed TD SI methodology. After that, TD based optimization methodology is presented. Then, FD SI simulation is outlined, with focus centered on the reduction in simulation time over classical TD SI simulations for long topologies. Finally, a FD-based optimization methodology is discussed as a viable proposal for improving SI performance of long topologies. Channel designers, typically SI engineers, as well as companies in the computing industry may be benefited with this study. The demonstrated reduction in simulation time of the proposed numerical optimization methodologies can be exploited for engineers without access to compelling computational resources, such as robust server systems. Once a designer has a topology proposal that is close to meet passing criteria, one of these methods can be applied to fine tune the design parameters, finding a solution that satisfies the industrial specifications for a given multi-Gbps interconnect.

Resumen

Esta tesis presenta dos metodologías de optimización de integridad de señal (SI) para encontrar diseños de interconexiones de multi Gbps que cumplan con especificaciones industriales; ambas metodologías están basadas en formulaciones clásicas de optimización numérica. La primera metodología es una implementación en el dominio del tiempo (TD), pensada para topologías relativamente simples o cortas en tamaño, las cuales intrínsecamente requieren tiempos de simulación pequeños. El segundo método es una nueva implementación en el dominio de la frecuencia (FD), dedicada para topologías relativamente complejas o largas. Las topologías largas requieren tiempos de simulación extensos en métodos clásicos de SI en TD. El método de optimización basado en FD es desarrollado para reducir considerablemente el tiempo de simulación en comparación con las implementaciones en TD. Siguiendo los temas presentados en este trabajo de tesis, se presentan al lector capítulos que son básicos para las metodologías de optimización tratadas. Una descripción general de los métodos actuales de SI se da inicialmente. Después, se presenta un ejemplo de una topología que no cumple con la especificación, describiendo el canal digital de alta velocidad y sus parámetros de simulación; también se presenta la metodología SI en TD con detalle. Posteriormente la metodología de optimización numérica basada en TD es presentada. En seguida la simulación de SI en FD es mostrada, donde el enfoque se centra en la reducción de tiempo de simulación sobre métodos clásicos en TD para las topologías largas. Finalmente, el nuevo método de optimización basado en FD es tratado como una propuesta viable para mejorar el desempeño de topologías largas. Diseñadores de canales de interconexión, típicamente ingenieros de SI, así como compañías de la industria de la computación podrían ser los principales beneficiados con este estudio. La demostrada reducción en tiempo de simulación de las metodologías de optimización numérica puede ser explotada por ingenieros sin acceso a recursos computacionales de alto rendimiento, tales como sistemas de servidores robustos. Una vez que el diseñador tiene una propuesta de topología que está cerca de cumplir con el criterio de aceptación, uno de estos métodos puede ser aplicado para afinar los parámetros de diseño y así encontrar una solución que satisface la especificación industrial para una interconexión de multi Gbps dada.

Content

Summary v

Resumen vi

Introduction 1

1. An Introduction to Signal Integrity Simulations for High Speed Interconnects in Computing Platforms 3

1.1. Introduction to High Speed Digital Channel Design 3
1.2. Modeling the Interconnect 4
1.3. Time Domain Signal Integrity Simulations 5
1.4. Achieving Successful High Speed Designs 7
1.5. Conclusions 8

2. An Illustrative Signal Integrity Simulation Example of a Non Spec Compliant High-Speed Digital Interconnect 9

2.1. An Introduction to Signal Integrity Design Challenges 9
2.2. Topology Description 10
2.3. Electric Models and Associated Parameters 11
2.3.1 Transmitting Device 11
2.3.2 Packaging 12
2.3.3 Transmission Lines 12
2.3.4 Vertical Transition Vias 13
2.3.5 Connectors 13
2.3.6 Discrete Components 13
2.3.7 Cables 14
2.3.8 Generic Receiving Device 14
2.4. SPICE Simulations 14
2.5. Post-Processing Simulations 15
2.6. Eye-Based Simulations Results 16
2.7. Conclusions 17

3. A Minimax Formulation for a Time Domain Numeric Optimization of a Non Spec Compliant USB3 Topology 19

3.1. Introduction 19
3.2. Topology under Test 20
3.2.1 Simplification to Reduce Simulation Time 20
3.2.2 Passing Threshold of Simplified Topology 21
3.2.3 Input/Output Parameters 21
3.3. Time-Domain Minimax Formulation 22
3.4. Driver Function 23

3.5. Optimization Algorithm 24
3.6. Results 25
3.7. Conclusions 26

4. Frequency Domain Simulations of High Speed Digital Interconnects and Comparison with Classical Time Domain Analysis 27

4.1. Introduction 27
4.2. Topology under Test 28
4.3. FD Simulations in Hspice 28
4.4. Results 30
4.5. Conclusions 32

5. A Minimax Formulation for a Frequency Domain Numeric Optimization of a Non Spec Compliant USB3 Topology 33

5.1. Introduction 33
5.2. Topology under Test 34
5.3. Minimax Formulation 35
5.4. Driver Function 36
5.5. Optimization Algorithm 37
5.6. Results 37
5.7. Conclusions 39

General Conclusions 41

Appendices 45

References 65

Index 67

Introduction

The design of high speed digital interconnects in modern computing platforms rely on signal integrity simulations (SI) for accurate signal quality and data transmission performance predictions before prototyping. Nowadays, computing platform design is facing increasing challenges in several areas. Finding a channel design that meets the required industrial specifications is a hard engineering process that gets more complicated on each communication bus generation, in part due to the increasing bus transfer rates and decreasing driver voltage swing, which creates increasingly complex signal integrity problems. Also, new platform form factors with reduced space and higher component integration limit to designer the available board's real state area where the interconnects and components can be placed. In addition, computing industry players, looking for differentiation over competitors, push design to its limits with out of guideline topologies: non-standard connectors, new PCB materials, cost-reduced cables, etc. A description of modern computing platform design and a general view of signal integrity simulation process is presented in Chapter 1.

The most common signal integrity industrial practice is to analyze topologies in time domain (TD) transient analysis, which is described in Chapter 2. Time domain analysis is not an ideal approach when working with complex and long topologies, because they require extensive simulation time. Moreover, a comprehensive analysis of the design space requires the use of Monte Carlo based techniques; which imposes an elevated number of simulation cases. Often, even after completing the analysis of a wide design space, a solution that meet specifications cannot be found. So, the engineer needs to come up with another design space proposal for a new analysis, and so on, until a spec compliant solution is identified.

In many practical cases, for individual designers or small companies can be prohibitively expensive to follow the time domain with Monte Carlo approach. Otherwise, they need to have access to complex (and expensive) server systems with high computing processing and memory resources, powerful enough to parallelize heavy workloads.

As an alternative way to address these limitations, Chapter 3 presents a SI TD numerical optimization process to improve a topology that initially does not meet industrial specifications.

This TD-based optimization works adequately for short topologies that require relative short simulation times. However, a long topology with extensive simulation time per case is tested. An approximation is necessary in this case to speed up the simulations. In spite of that, the proposed optimization method produces a spec compliant design. An important advantage of this optimization procedure is that it follows a classical SI TD transient simulation methodology that is widely accepted in the computing industry.

Chapter 4 provides an introduction to SI simulation in frequency domain (FD) using commercial simulation tools, demonstrating that it is an effective method that reduces the simulation time as compared to classical TD simulation for the same interconnects. This makes possible to simulate long topologies in just a few minutes. However, there is room for improvements, since frequency domain simulation methods are being improved continuously in the industry for signal integrity purposes.

This thesis proposes a novel SI numerical optimization methodology developed in FD. The proposed methodology is able to identify spec-compliant multi-Gbps interconnect designs, and it is suitable for optimizing the design parameters from evaluating key S-parameters. This methodology is especially suitable for relatively large topologies that demand a long simulation time when traditional eye-diagrams are obtained from time-domain simulations. Some of these large topologies may need around one hour of simulation time per case, even if equivalent circuit models are used, however, they can be analyzed much faster in frequency domain. In order to apply the proposed FD methodology, the designer needs to select some design parameters that most likely produce a result close to fulfilling design specifications. Then, the optimization methodology finds a set of new fine-tuned design parameters that produces a spec-compliant topology. This FD SI numerical optimization methodology is described in Chapter 5.

The author wants to gratefully thank Dr. José Ernesto Rayas Sánchez, full professor and director of the Computer-Aided Engineering of Circuits and Systems (CAECAS) research group at ITESO, for his guidance and supervision. I also want to thank Dr. Chunfei Ye, technical lead of South Input Output Hub (SIOH) signal integrity team at Intel Corporation, for being so generous to share his signal integrity expertize with the author. Finally, I deeply thank Intel Guadalajara Design Center (GDC) for providing financial support through all my Master's degree program and for this thesis development.

1. An Introduction to Signal Integrity Simulations for High Speed Interconnects in Computing Platforms

An introduction to the time domain (TD) methodology of signal integrity (SI) simulations for high-speed interconnects in computing platforms is presented in this chapter. It provides an overview of the key elements of modern SI simulation in the design of multi-Gbps digital systems. It briefly touch the bases for modeling the components of the link and how models interact for the correct representation of the high-speed interconnect. It also describes the main current tools and post-processing approaches for signaling simulation. Finally, the report states how Intel supports customers in the design of their own computing platform.

1.1. Introduction to High Speed Digital Channel Design

High-speed digital system development is extensively based on computer-aided design (CAD). CAD simulations are nowadays unavoidable to provide accurate analysis of circuits, making possible design optimization before any physical prototyping. It is an area of opportunity for development of novel techniques for the modeling, design, and optimization of circuits and electronic devices [Rayas-14].

Signal integrity is the practice of ensuring sufficient fidelity of a signal transmitted between a driver and a receiver for proper functioning of the circuit, e.g., the signals over the multi-Gbps bus between a processor and its chipset. The integrity of a signal in real world applications is compromised by artifacts of the circuit's layout on a printed circuit board (PCB), the type of integrated circuit (IC) package being used, the circuit logic family, the power delivery network and other aspects of the high-speed digital design. SI simulation becomes more and more critical to the success of a computer system design; it is capable to explore electrical performance at any point in the link, and is a low cost way to incorporate various assumptions and design parameters [Fan-10], [Li-12], [Ye-12].

This chapter presents a brief introduction to the SI simulation process for multi-Gbps

interconnects of computing platforms, similar to those used in Intel architectures. First, Section 1.2 describes how components of the interconnect are modeled separately in order to set reliable and versatile building blocks. Section 1.3 mainly define the state of the art method used in time domain SI simulations. Section 1.4 describes the way Intel supports its customers to achieve a successful design and the problems associated when a design fall outside of Intel's recommendations. Finally, Section 1.5 presents some conclusions.

1.2. Modeling the Interconnect

In order to perform an accurate SI analysis, the simulation relies on electric models that characterize individual elements of the high-speed interconnect [Hall-09]. These models approximate the electromagnetic behavior of link's components. They are intended to represent their most important characteristics, such as: transfer function, parasitic effects, interference, etc. Examples of these components are: IC driver, silicon package, metallic traces on motherboard, transition vias, connectors, discrete components, cables, etc. Each model can contain specific parameters as well as user-defined variables for proper system definition.

Depending upon the nature of the component, typically there are two techniques widely used in the industry for model creation: RLGC-frequency-dependent transmission lines to model routing sections that have a uniform cross section, with loss and dispersion accounted for; and 3D full-wave modeling applied to vertical structures including vias, package, socket, connector, etc.,

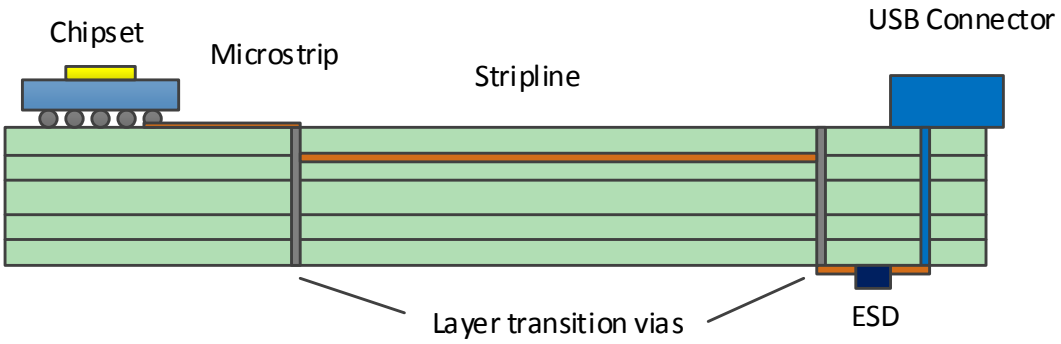


Fig. 1-1 Side view of a typical USB back panel topology, from transmitter chip to USB connector at the chassis of the computing platform. The different transmission line types and discrete components that build the interconnect are shown.

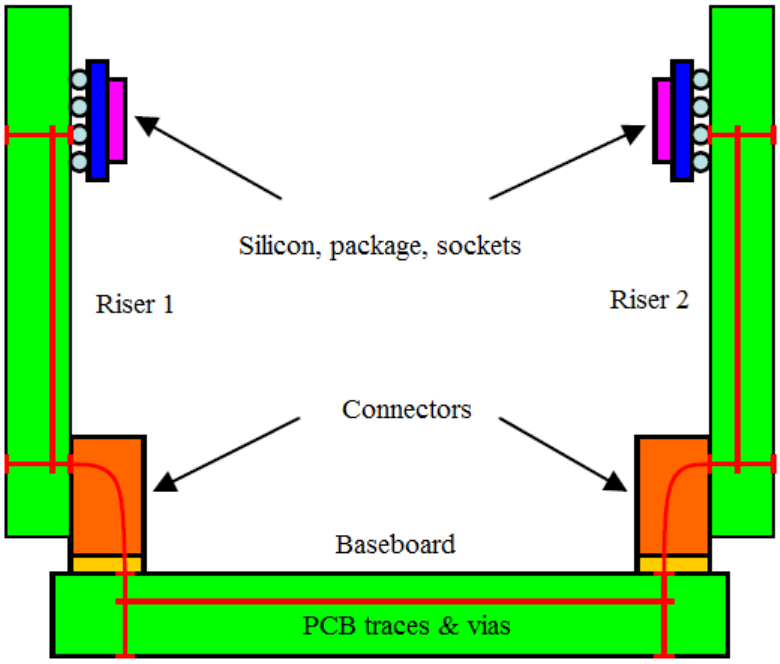


Fig. 1-2 Side view of a typical PCIe backplane topology, a baseboard connecting two riser cards. Figure taken from [Xiao-08].

where the output of the 3D full-wave model (S-parameters) is converted to an equivalent circuital model with rational polynomial function approximation algorithms [Ye-09].

The electric models mentioned above are later organized and interconnected to perform the simulation of the link. The way these models are connected depends upon the hardware description and layout of the high-speed interconnect, also known as the topology. As an example, Fig. 1-1 shows a typical USB back panel topology; it shows how the chipset is linked to the USB connector on the motherboard. This particular design uses a microstrip line close to the chipset region as escape routing, and a stripline is used as main routing; for PCB layer vertical transitions are used vias, and finally close to the USB connector, it shows the common placement of electrostatic device (ESD). Another example is shown in Fig. 1-2, where a PCIe backplane topology is illustrated; this topology features two riser cards connected through a baseboard with connectors.

1.3. Time Domain Signal Integrity Simulations

Transient SI simulation of the topology is typically performed in industry by a commercial SPICE-based simulator. SPICE is a universal circuit simulation program able to analyze electronic

connections on a chip, board or system level in the steady-state, transient, and frequency domains. It was originally developed at the University of Berkeley in 1972 and freely distributed. During a very short time SPICE became the preferred analog simulation tool at universities as well as at semiconductor companies. Accuracy of SPICE simulator depends on the quality of the models of the components that are invoked [Cuny-96]. Lately, semiconductor industry has widely adopted HSPICE[1].

The time domain waveform result of HSPICE transient simulation is post-processed to obtain the eye-based voltage and time margins at a given bit error rate (BER). The methods for obtaining the margins have evolved substantially. The first approach was a time domain simulation using random data vectors as the input, where the desired result is an eye diagram. The eye diagram is not accurately characterized by a small set of data, and on the other hand, when a very large set of random data is used as stimulus, simulation time becomes prohibitive [Casper-02]. The peak distortion analysis (PDA) was later developed to provide a worst-case eye diagram of the link. "Although PDA analysis is a computationally efficient approach to include ISI and crosstalk impact, it does not include the transmitter and receiver jitter" [Fan-10]. Lately, time domain jitter with statistic BER technique has overcome the disadvantages of previous simulation methods and it is used to obtain voltage and time margins. It uses the step response of the channel, and directly converts time domain waveforms through convolution of superposition. The method is capable of adding transmitter and receiver jitter [Fan-10], [Xiao-08].

The pass-fail indicator of a high-speed interconnect is typically a specification of eye height (EH) and eye width (EW) at certain BER, in other words, BER-eye-based voltage and time thresholds are defined as pass-fail criteria [Hall-09]. For example, USB 3.0 industrial standard specify for a successful compliance test 100 mV_{diff} and 68 ps as minimum EH and EW respectively, measured for 10^6 consecutive bits (BER $1x10^{-6}$) [USB-11]. Fig. 1-3 shows these requirements for the USB 3.0 industrial standard.

[1] HSPICE 2010.12-SP2, Synopsys, 700 East Middlefield Road, Mountain View, CA 94043. http://www.synopsys.com/home.aspx

1.4. Achieving Successful High Speed Designs

Once completed the SI analysis of the spec compliant topology, the physical and electrical characteristics of such simulated interconnect are documented in Intel's platform design guide (PDG). PDG is a document that contains an extensive set of rules and guidelines for high-speed digital design. This guide contains all the conditions to achieve a successful implementation, but not always considers the specific conditions that the customer needs [Li-12].

Traditionally, Intel supports customers via PDG. However, the customer is required to perform his own SI assessment if it is needed a link solution that is out of guideline (OOG), i.e., when it goes beyond PDG recommendations. Intel is experiencing an overwhelming number of customer request for SI support lately. More and more customers are designing OOG platforms for cost savings and better marketing competition. Therefore, cost *vs*. performance tradeoff becomes critical [Li-11].

The process of finding a successful design having some parameters out of guideline is a complicated and time-consuming simulation process. The challenge is to find a suitable tradeoff between the OOG parameters, link constrains and enablers to meet the specification. Each tradeoff implies a manual iteration in the SI methodology previously described. Moreover, when the SI result is within close proximity to the pass-fail criteria, the process gets more complicated because

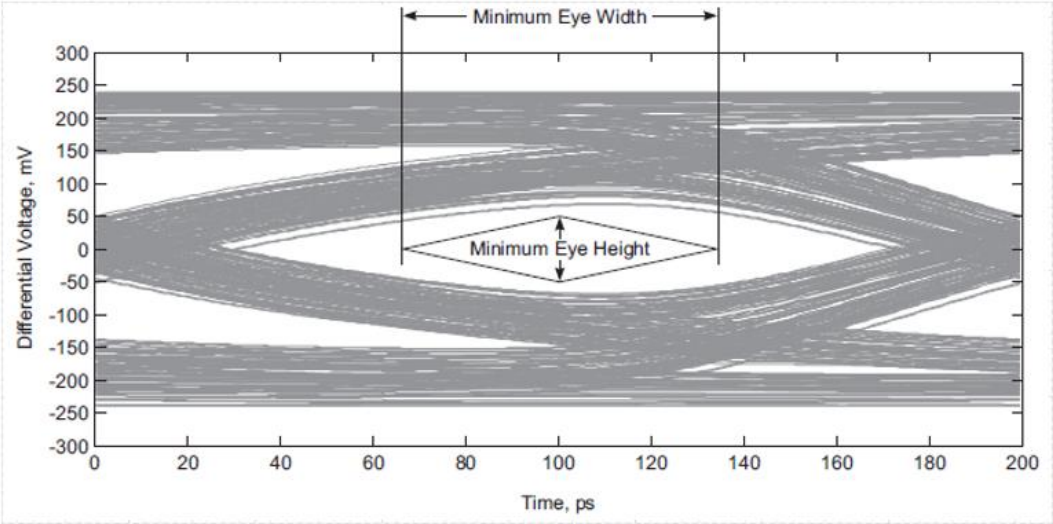


Fig. 1-3 Eye height and eye width requirements for USB 3.0 TX spec compliance test. Figure taken from [USB-11].

the number of iterations becomes larger. Finally, it is important to remark that the effectiveness of this process depend upon the experience and skills of the SI engineer.

1.5. Conclusions

An overview of SI simulation was presented in this chapter. Two mayor steps comprehend the classical SI methodology accepted in computing industry: first, to perform transient simulation using SPICE, since it is the most popular method for characterizing the high-speed interconnect; and then, time domain jitter with statistic BER for determining the resulting eye-based voltage and time margins. A successful design implies eye-based margins that satisfy the pass-fail threshold given by an industrial standard or vendor specification. In the situation of a non-spec compliant result, the task of identifying a tradeoff in the design variables to produce a successful outcome is a complicated and time-consuming process. It is identified the need to increase the efficiency in this process. Perhaps an automatic routine to efficiently analyze the design space may address this concern. The idea needs to be explored.

2. An Illustrative Signal Integrity Simulation Example of a Non Spec Compliant High-Speed Digital Interconnect

A description of a high-speed digital interconnect and its signal integrity simulation process is presented in this chapter. It provides a general description of the chosen design, corresponding circuital models, detailed simulation methodology, and interconnect parameters of USB 3.0 front panel topology example. Simulated output eye-based voltage and time margins are shown and compared against compliance test requirements. The chapter concludes that the margins are not meeting industrial specification, hence, the topology is a non-spec compliant example, which will be useful as a study case in next chapter.

2.1. An Introduction to Signal Integrity Design Challenges

Signal integrity (SI) simulations are valuable resources to accurately evaluate performance of a high-speed interconnect before prototyping any computing platform. Modern digital systems are characterized by complex design that increases data rates and clock frequencies to frontiers that a decade ago seemed to be unviable design targets. Moreover, systems form factor, circuit miniaturization, reduction of power consumption and logic level, produces additional difficulties to achieve a successful high-speed digital channel design [Fan-10], [Hall-09].

The result of this challenging scenario is the need of ever increasing gigabit link designs that hardly meet industrial specification requirements, or very demanding design guidelines provided by silicon vendors. A high-speed interconnect that fails vendor specific or industry specifications, cannot guarantee quality and fidelity of digital signals and thus, compromises the reliability of the whole computing platform. It is clear the need to ensure the adequate performance of every high-speed interconnect previous to delivering digital systems for high volume manufacturing [Li-11], [Robledo-14].

This chapter describes an example of a USB 3.0 non-spec compliant topology and its simulation methodology. Section 2.2 physically describes the complete topology from transmitting

driver to receiving end; also individual components and their characteristics are outlined. Corresponding circuital models and controlling parameters are listed in Section 2.3. Then, SPICE transient simulation and post-processing calculations needed to convert from time-domain voltage waveforms to final results in form of eye-based voltage and timing margins are described in Sections 2.4 and 2.5, respectively. Section 2.6 describes the USB 3.0 spec compliance requirements and compare them against simulation results. Finally, conclusions are given in Section 2.7.

2.2. Topology Description

The high-speed digital interconnect used in this chapter is an example of a USB 3.0 front panel topology. This kind of design implementation is widely used in workstations and server class computing platforms. The corresponding topology is shown in Fig. 2-1, it is presented as a side view in order to easily illustrate the main actual routing transition inside PCB layers. At a high level, the topology comprehends three main elements: the hub (or computing host), external long cable, and a generic USB device.

There are three sections inside the hub: a 6-inches long baseboard, a 10-inches long internal cable, and a 1-inch long daughter card. The baseboard contains the transmitting chipset packaging (PKG) with internal half inch micro routing, 0.5-inch PKG breakout routings, vertical transition

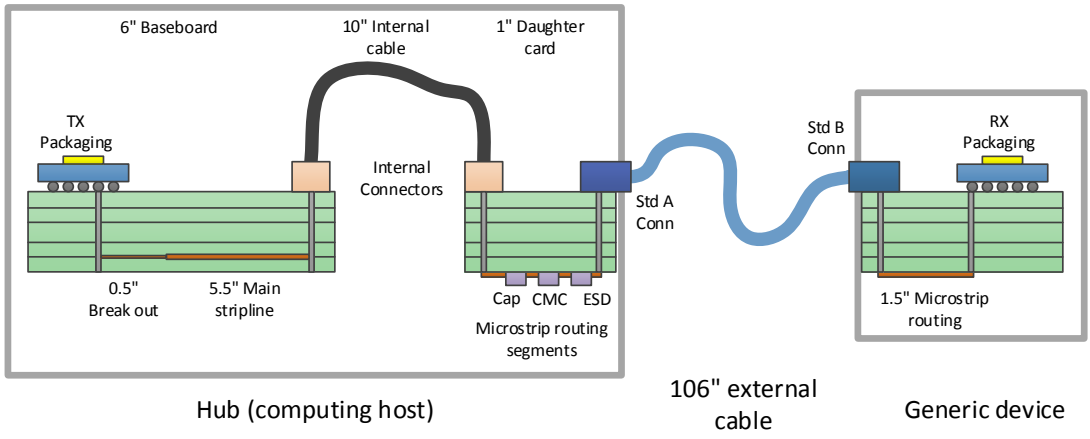


Fig. 2-1 Example of USB 3.0 front panel topology. Hub is composed internally of baseboard, cable and daughter card. External cable connects hub with a generic device. Dimensions are indicated (drawing is not at scale).

vias, 5.5-inches long main stripline routing, and pin thru hole (PTH) internal connector. The 10 inches long internal cable connects the baseboard to a daughter card. Finally, the 1-inch daughter card is located at front panel and contains discrete components such as AC coupling capacitor, common mode choke (CMC) and electrostatic discharge (ESD) component.

Outside the hub part, it is defined a 106-inches long external cable. This cable has matted connectors type Standard A (Std A) and Standard B (Std B) located at the hub and generic device sides, respectively.

At the end of this topology there is a generic device that includes a 1.5-inch long routing, and half inch long micro routing in a generic receiving package (PKG).

2.3. Electric Models and Associated Parameters

In order to perform a SI analysis, the simulation relies on electric models that characterize individual elements of the high-speed interconnect. The models approximate the electromagnetic behavior of the main topology components. They are intended to represent their most important electric characteristics, such as: transfer function, parasitic effects, and crosstalk. Semiconductor industry has adopted the use of SPICE-based circuit simulators. Accuracy of SPICE simulations depend on the quality of models that are invoked [Cuny-96], [Ye-09].

The SPICE models that represent the components of the USB3.0 front panel topology are now described.

2.3.1 Transmitting Device

USB3.0 is a Ser-Des technology. Currently, all Giga-bit serial bus technologies transmit data in form of differential signals. Transmitter (TX) driver model is a simple voltage controlled current source model including driver termination. The output is provided via three differential pairs: the central pair drives a differential step signal and the adjacent lateral pairs are kept quiet, as shown in Fig. 2-2. The model is controlled by six different parameters: differential voltage swing (vswing), signal's rise and fall times, resistive termination, capacitive parasitic, and de-emphasis value.

2.3.2 Packaging

The generic PKG model is build up by concatenating the following internal elements: bump pad where the silicon is soldered down, micro transmission lines in the substrate, PTH via that cross PKG's core, and ball grid array (BGA) terminals that join the pads in the PCB. All vertical elements comes from 3D modeling and full wave electromagnetic (EM) simulations; on the other hand, transmission lines utilize 2D solvers to create its circuital model. The next subchapters elaborate more about corresponding modeling methods for transmission lines and vertical transition vias. PKG model contains a unique adjusting knob which is the transmission line length parameter.

2.3.3 Transmission Lines

Nowadays the edge rates of digital signals are commonly found in the order of tens of picoseconds. Due to this fact, the spectrum of transmitted data spreads from DC up to several GHz; therefore, broadband transmission line models are necessary. RLGC frequency-dependent models simulate transmission line behavior accurately; these per-unit-length characterized models can predict losses and dispersion [Ye-09]. In order to account for induced crosstalk in near traces, three adjacent differential pairs are modeled in a 2D solver. Following [Li-12], it is assumed that the

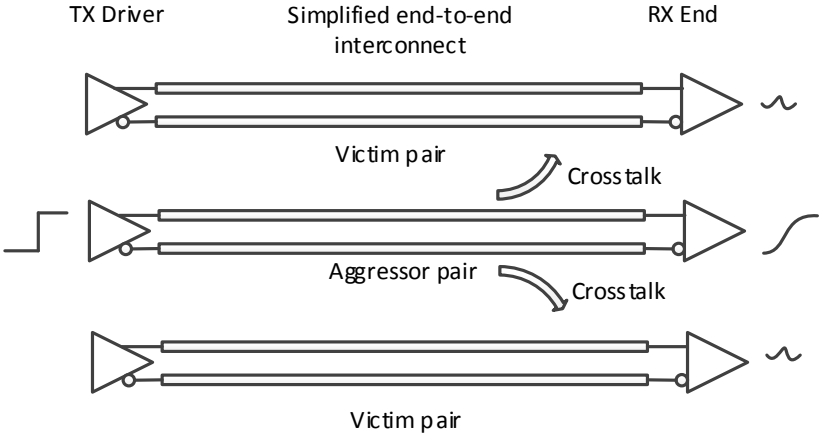


Fig. 2-2 Simplified diagram of a three differential pair configuration. Diagram shows differential TX driver and RX, the differential transmission lines represent the whole interconnect. The middle pair (aggressor) driven by step response signal and the adjacent side pairs (victims) capturing crosstalk.

middle pair is the “aggressor” and the side pairs are the “victims”. Regarding impedance corners, three models are created for each transmission line. The idea is to have a representation of a nominal impedance and tolerance, hence, there are low, nominal and high impedance models. Once selected a fixed impedance corner, length is the only parameter to be varied in the simulation deck.

2.3.4 Vertical Transition Vias

Full-wave 3D EM modeling based on Maxwell equations is the most accurate methodology to model discontinuities in high-speed interconnects. Discontinuities are structures that do not match interconnect impedance targets. Examples of discontinuities are vertical structures like PTH vias or connectors. Discontinuities have to be considered distributed elements, instead of lump elements, when the structure size is no longer negligible with respect to wavelengths at multi-GHz frequencies. S-parameters Touchstone files are the standard output of most full-wave EM solvers; they are then converted to an equivalent circuital model with rational polynomial function approximation algorithm [Ye-09]. Depending on the entry layer and exit layer for a given PCB stackup, there is a particular via model. Typically, there are no adjustable parameters for via models.

2.3.5 Connectors

Connectors follow a similar modeling method to that one for vias. There are three relevant connector models in this topology example, they are described as follows. First, Std A connector is located at external panel of hub side. Then, internal connectors are suited for cable interconnection inside computing host. Finally, Std B connector is dedicated for devices. Similarly to the via models, connector models do not have any adjustable setting for simulation.

2.3.6 Discrete Components

In this particular topology example are three discrete components: AC coupling capacitors,

an ESD device, and a CMC. The circuital models of these components account for internal parasitics and mounting path capacitances. There are not adjustable parameters in these models.

2.3.7 Cables

Cables are simply modeled as transmission lines. They have three corner impedance variations and they have an adjustable length parameter.

2.3.8 Generic Receiving Device

Receiver models are simple resistive-capacitive circuits. Both resistance and capacitance are adjustable parameters. Basically, resistance accounts for the ideal impedance termination value and capacitance represents the expected receiving parasitics in a generic device.

2.4. SPICE Simulations

After creating and validating individual models for each interconnect element, the next step is to organize and concatenate them in such way that accurately represent the topology of interest. The result is a SPICE-based netlist that is called simulation deck. Appendix A shows the USB 3.0 front panel simulation deck used in this chapter.

It was mentioned before that models and deck are arranged in a three adjacent differential pair arrangement, as shown in Fig. 2-2. In this configuration, the middle pair is used to capture how much signal is delivered from TX driver to RX end. Present signal waveforms at RX probes contain information of channel loss and reflections. The two lateral adjacent pairs are kept quiet and are meant to capture induced crosstalk [Li-12].

Time-domain transient simulation is used to solve channel step response in form of voltage waveform at RX side. Particularly, it is used the SPICE-based commercial circuit simulator HSPICE. The resulting waveform contains information that can be derived in how lossy, reflective or noisy the digital interconnect is. Fig. 2-3 shows the resulting step response voltage waveforms.

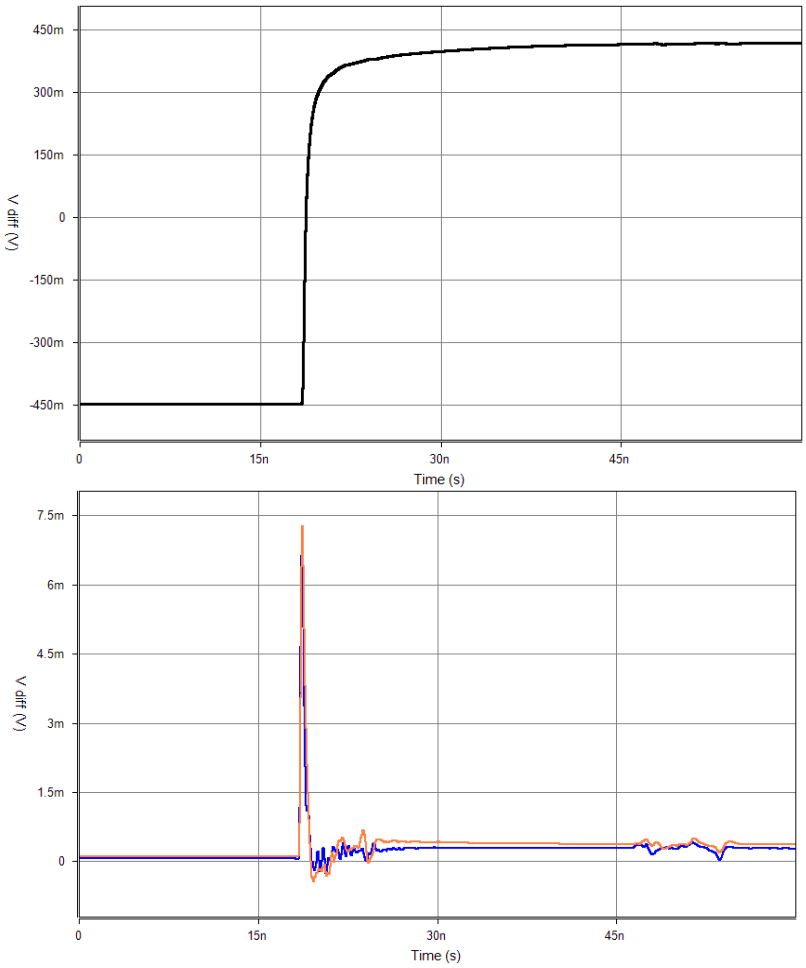


Fig. 2-3 Step response waveforms from a time-domain transient simulation of the topology in Fig. 2-1. Top waveform is the signal delivered to RX side in aggressor pair (center pair). Bottom waveforms are crosstalk present in victim pairs (adjacent pairs).

2.5. Post-Processing Simulations

Time domain jitter with statistic BER technique is the used post-processing method and it produces output eye-based metrics. It uses the step response of the channel, and directly converts time domain waveforms through convolution of superposition. Basically, step response is derived to obtain the impulse response, then this impulse response is convoluted to obtain the corresponding pulse response. Later on, statistical treatment produces eye-based voltage and time margins. Finally, the method is capable of adding transmitter and receiver jitter [Casper-02], [Fan-

10], [Xiao-08]. An Intel's internal tool is used to obtain these BER eye-based results.

2.6. Eye-Based Simulations Results

USB 3.0 specification requires a minimum eye height of 100 mV_{diff} and minimum eye width of 0.34 *UI*, where *UI* stands for unit interval, also known as bit time [USB-11]. In other words, 0.34 *UI* means 68 ps, since bit time for USB 3.0 is 200 ps. According to this specification, the eye mask is defined after measurement of 10^6 consecutives bits; this amount of bits are statistically simulated.

A topology that meets the above normative requirements is regarded as Spec-compliant, hence, able to be certified by usb.org if it passes additional test requirements [USB-09].

Simulation outcomes for this particular topology example are 97.5 mV_{diff} of EH and 0.315 *UI* (or 63 ps) of EW. Fig. 2-4 shows the resulting eye. In the figure, the brighter the color, the higher the BER. Additionally, highlighted green eye correspond to BER = 1e-6. We can conclude that this topology does not meet industrial specification requirements.

It is important to mention the computational cost that this simulation methodology required: 82 minutes took to complete the HSPICE simulation and post-processing process. The

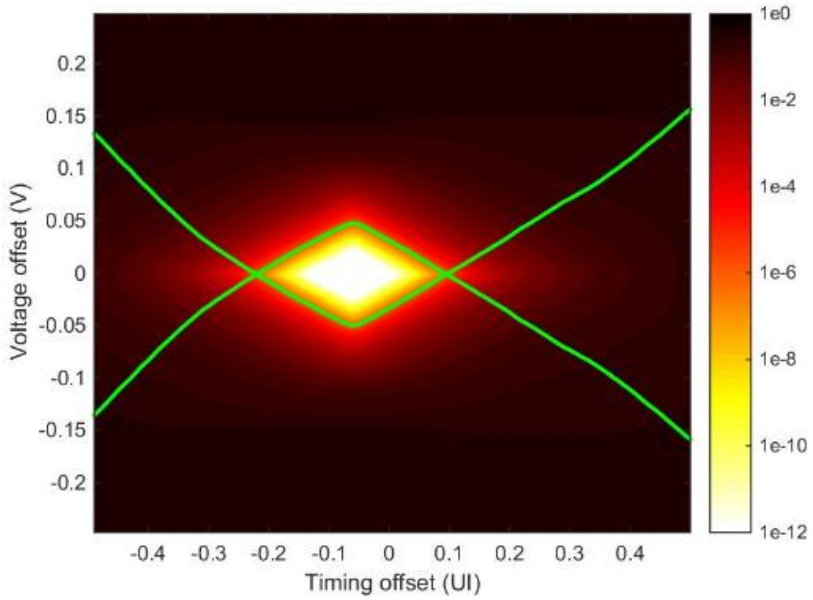


Fig. 2-4 Graphical representation of resulting eye. Highlighted green eye is the result of 10^6 statistically simulated bits, or BER $1x10^{-6}$.

computer used is a mobile workstation that comprehends an Intel® Core™ i5-4300 CPU with 4 cores @ 1.9 GHz and 4 GB of RAM memory. It is confirmed that a single test of the normative requirement has a high computational cost.

2.7. Conclusions

Description of a non-spec compliant USB3.0 front panel topology example and its SI simulation methodology were presented in this chapter. The simulation deck is a netlist created after concatenating circuital models, and a SPICE transient simulations produces time-domain voltage waveforms. Time domain channel response is statistically post-processed at BER 1e-6. Obtained eye-based voltage and time margins are violating the normative requirements, hence, the described topology fails USB 3.0 specification for compliance testing. An optimization of the interconnect parameter values could be applied in order to make this topology a spec compliant example, which is not a trivial task given the high computational cost of the simulation involved.

3. A Minimax Formulation for a Time Domain Numeric Optimization of a Non Spec Compliant USB3 Topology

A description of a minimax formulation and its implementation in Matlab for solving a non-spec compliant USB 3.0 high speed digital interconnect is presented in this chapter. The non-spec compliant topology example is described and its simulation parameters are outlined. The minimax formulation, that is the core of the optimization routine, is defined; additional functions that are building blocks are described as well. The chapter concludes that optimal design parameters found produce a spec-compliant solution for designing a successful USB 3.0 front panel topology.

3.1. Introduction

The design of high speed digital interconnects in modern computing platforms faces an increasing complexity, mainly due to industry diversification to new markets and novel form factors. In this challenging scenario, systems requirements push the performance to the limit, therefore, signal integrity of gigabit interconnects is highly impacted [Li-11], [Robledo-14].

The design of interconnect solutions that satisfy modern requirements typically need the simulation and analysis of a high number of design variables. It leads to an elevated number of test cases that requires an exhaustive engineering effort. Frequently, simulation and analysis outcomes report interconnect designs that do not satisfy industrial specifications, in which case it is required to optimize their design parameters.

This chapter proposes a time-domain numeric optimization routine to find a spec-compliant solution of a digital high speed interconnect that its initial definition does not meet the industrial specification. Section 3.2 describes the USB3 front panel topology and its design parameters used for optimization. The minimax formulation for numerical optimization is defined in Section 3.3. The driver function is discussed in Section 3.4. Then, Section 3.5 describes the optimization algorithm and its termination criteria. The evolution of design variables and results of the

optimization routine is shown in Section 3.6. Finally, conclusions are given in Section 3.7.

3.2. Topology under Test

A USB3 front panel topology with daughter card is used in this chapter. At a high level, the complete end-to-end topology comprehends three main elements: the hub (or computing host), the external long cable, and a generic USB device. This topology is shown in Fig. 2-1. A hub with the following characteristics is non-spec-compliant: 6-inches trace length baseboard, 10-inches long internal cable, and 1-inch trace length daughter card [Robledo-15a].

It was previously reported in Section 2.6 that this interconnect presents the following eye-based voltage and time margins: EH = 97.5 mV_{diff} and EW = 0.315 *UI* (63 ps). As spec-compliant requirement, USB 3.0 specification calls for a minimum EH = 100 mV_{diff} and EW = 0.34 *UI* (68 ps) [USB-09], [USB-11]. Hence, this topology is a candidate to apply a numeric optimization routine.

In Section 2.6 is also documented the computational cost of simulating this topology. The 82 minutes necessary to run a single case makes impractical to apply directly any numerical optimization routine, so a simplification to reduce the simulation time is proposed as follows.

3.2.1 Simplification to Reduce Simulation Time

USB3 specification defines length parameters of external cable and generic receiver as fixed elements for compliance testing (see Fig. 2-1). Hence, the external cable and receiver are the same in all simulating cases. In other words, USB3 compliance testing defines external elements to the computing host, so all design parameters subjected to variations reside inside the computing host [USB-09].

If the external cable and generic receiver are kept the same in all simulation cases, they can be replaced by a reasonable fixed approximation.

The complete idea to simplify the structure consists of the following. First, it is needed to know the design variable values that make this topology barely pass the specification. Generally, a validated passing point can be obtained from the silicon vendor design guidelines. Intel provides

to its customers a platform design guide (PDG) document as a design reference for spec-compliant interconnect solutions [Robledo-14]. Then, a simulation deck can be built using known passing design variables, discarding external cable and generic receiver. Next, simulations are run probing at the end of hub side, i.e. at the output of Standard A connector shown in Fig. 2-1. Finally, resulting EH and EW of this simplified version become our passing threshold.

This approach does not provides an exact equivalent between an end-to-end topology and the simplified hub topology, but produces an acceptable approximation that effectively helps to reduce simulation time. Also, this approach is not intended to override end-to-end simulations on compliance testing methodology. In this report it is only used as a practical vehicle to demonstrate the correctness of the proposed minimax formulation and its numeric optimization routine.

3.2.2 Passing Threshold of Simplified Topology

A complete end to end simulating deck featuring 5-inches long baseboard, 10-inches long internal cable and 1-inch long daughter card, including external cable and generic receiver, produces a result of 107 mV_{diff} and 0.341 *UI* (68.2 ps). This topology is in consequence spec-compliant.

The same simulating deck, but now without external cable neither generic receiver, generates an eye of 230 mV_{diff} and 0.38 *UI* (76 ps) when probing at the end of Std A connector. These results are now our passing thresholds, EH_{min} and EW_{min} respectively, of this spec-compliant simplified topology.

3.2.3 Input/Output Parameters

At a high level, the simulation of this USB topology can be viewed as an input-output system. The input parameters can be classified in three groups: design variables, pre-assigned parameters, and simulator parameters.

Design variables are those variables subjected to be optimized; pre-assigned parameters are fixed variables related to the high speed digital interconnect; finally, parameters of the simulator are settings that control simulation engine.

TABLE I
INPUT SIMULATION PARAMETERS

Vector element	Parameter name	Parameter variable	Parameter value (unit)
x_1	Baseboard's main segment length	*rtg12L*	5.5 (in)
x_2	Internal cable length	*rtg101L*	10 (in)
x_3	Daughter card's main segment length	*rtg21L*	0.4 (in)
x_{p1}	Differential voltage swing	*vswing*	1 (V)
x_{p2}	TX driver differential termination	*rterm_tx*	100 (ohm)
p_{s1}	Simulator time step	*timeStep*	10 (ps)
p_{s2}	Simulation time	*finalTime*	60 (ns)

On the other hand, the output parameters are the simulation outcomes, which in this case are EH_{sim} and EW_{sim}.

The candidate to apply numeric optimization is the simplified deck version of the non-spec-compliant topology described in [Robledo-15a]. Taking this deck as reference, the selected design variables ($\boldsymbol{x}$) are the length of main baseboard routing section (*rtg12L*), the length of internal cable (*rtg101L*) and the length of the main section of routing inside the daughter card (*rtg21L*). Pre-assign parameters ($\boldsymbol{x}_p$) are the voltage swing (*vswing*) and the resistive termination (*rterm_tx*) of the transmitting driver. Finally, the parameters of the simulator ($\boldsymbol{p}_s$) are the time step resolution (*timeStep*) and simulation stop time (*finalTime*). TABLE I shows a summary of these input parameters.

3.3. Time-Domain Minimax Formulation

In order to have a passing simulation case, the simulated eye height, EH_{sim}, should be larger than the minimum acceptable eye height, EH_{min}. Similarly, the simulated eye width, EW_{sim}, should be larger than the minimum acceptable eye width, EW_{min}. Hence, an error function can be defined as follows:

$$e_h = EH_{\text{min}} - EH_{\text{sim}} \tag{3-1}$$

$$e_w = EW_{\min} - EW_{\text{sim}} \tag{3-2}$$

$$\boldsymbol{e} = [e_h \quad e_w]^T \tag{3-3}$$

A positive error corresponds to a failing simulation case. On the opposite way, a negative error means that simulation outcomes are larger than minimum passing thresholds, corresponding to a passing simulation case. The minimax formulation is used to minimize the maximum error of a function, in this case $\boldsymbol{e}$. This report implements an unconstrained minimax formulation [Rayas-11],

$$\boldsymbol{x}^* = \arg min_{\boldsymbol{x}}\, u(\boldsymbol{x}) \text{ where } u(\boldsymbol{x}) = \max(\boldsymbol{e}) \tag{3-4}$$

with vector $\boldsymbol{x}$ containing *rtg12L*, *rtg101L*, and *rtg21L*, as defined in Table I.

The implemented objective function generates a journal file to store the evolution of some indicators. At the end of the optimization routine, it plots the indicators' evolution. Selected indicators are: design parameters *rtg12L*, *rtg101L* and *rtg21L* in linear and normalized fashion; simulation outputs EH_{sim} and EW_{sim}; HSPICE, post-processing and driver function simulation times; and finally, $u(\boldsymbol{x})$. The objective function code is shown in the Appendix B .

3.4. Driver Function

The driver is a Matlab[2] function that basically is fed with simulation parameters from objective function, calls the simulators, and reports the output back to the objective function.

Every time that the driver function is called, it builds a copy of the simplified deck version of the non-spec-compliant netlist. Then, the netlist is overwritten with the design variables values predicted by optimization method. Then, HSPICE simulator is invoked and it runs the netlist. The time domain transient simulation output is a step response waveform of the gigabit digital interconnect in the format of .tr0 file.

Next, an in-house post-processing simulator is invoked, which takes the .tr0 file as input. By using time domain jitter with statistic BER technique, the post-processing simulator produces the eye-based metrics EH and EW [Fan-10], [Robledo-15a]. The driver function is shown in

[2] MATLAB R2014b, MathWorks, 3 Apple Hill Drive, Natick, MA 01760-2098, http://www.mathworks.com/products/matlab/

Appendix C .

3.5. Optimization Algorithm

The optimization process includes three main routines: initialize optimization with seed values, evaluate $u(\boldsymbol{x})$ at predicted iterates until termination criteria are fulfilled, and report optimal design parameters found, $\boldsymbol{x}^*$.

The heart of the optimization algorithm is the Matlab function `fminsearch`, which implements the Nelder-Mead method [Matlab-15].

Seed values, $\boldsymbol{x}^{(0)}$, are the initial design parameters that start `fminsearch` function, and they are passed directly to the objective function. After first evaluation of $u(\boldsymbol{x})$, `fminsearch` determines next simulating values using a derivative-free method.

`fminsearch` function stops when termination criteria are met. In this report, three termination criteria are defined: maximum number of iterations (*MaxIter*) equal to 50, change in function evaluation (*TolFun*) is less than 1e-2, and change in next simulation inputs (*TolX*) are less than 1e-2.

Once optimization algorithm stops, $\boldsymbol{x}^*$ is reported (*xn_opt*). Additionally, this function reports optimum $u(\boldsymbol{x})$ (*feval*), termination reason (*exitflag*), number of iterations, number of

TABLE II
TIME-DOMAIN OPTIMAL DESIGN VARIABLES AND EXIT SUMMARY

Summary feature	Value
x_1^*	5.689 (in)
x_2^*	9.4424 (in)
x_3^*	0.4095 (in)
Termination reason	Satisfied Tol Func & Tol X
Optimal $u(\boldsymbol{x}^*)$	–0.0158
Number of iterations	18
Number of function evaluations	45
Total elapsed time	428 (min)

functions evaluations and total elapsed time (*output*).

Optimization algorithm is shown in Appendix D .

3.6. Results

TABLE II shows optimal design variables found and exit summary. Reported optimal

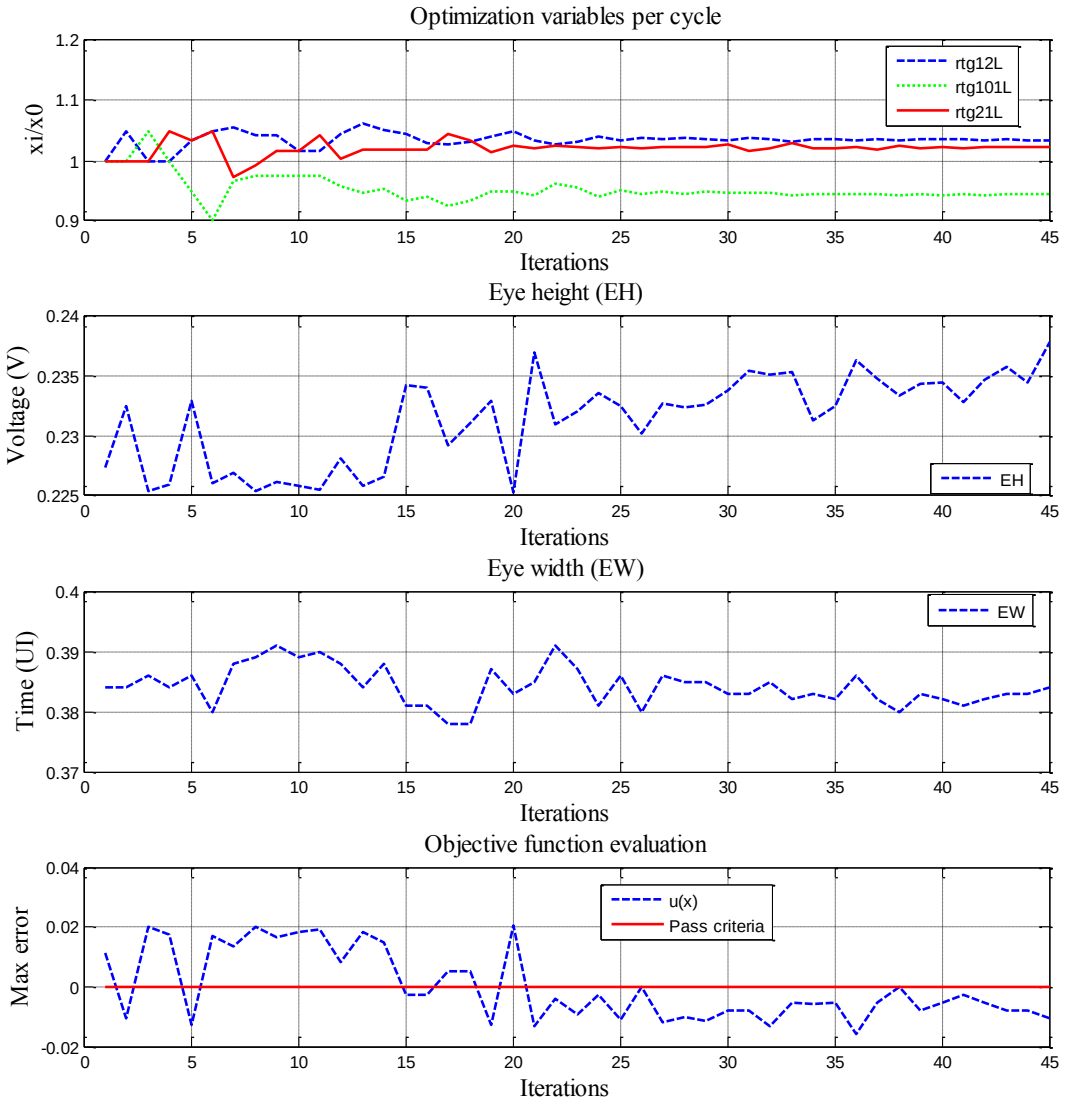


Fig. 3-1 Evolution of input design variables and outcomes per iterations. From top to bottom: first plot shows design variables normalized to seed values; second and third plots illustrate eye height and eye width respectively, outcomes from driver function; fourth plot represents objective function output $u(\boldsymbol{x})$.

design variables are $\boldsymbol{x}^* = [5.689 \;\; 9.4424 \;\; 0.4095]^T$ (inches). Notice that optimal $\boldsymbol{u}(\boldsymbol{x}^*)$ is –0.0158, meaning that both $EH_{\min}$ and $EW_{\min}$ thresholds are met. These results were obtained from the seed values $\boldsymbol{x}^{(0)} = [5.5 \;\; 10 \;\; 0.4]^T$ (inches).

As mentioned before, the numeric optimization routine is running an approximated simulation deck. In order to fully determine whether the optimal design found represents a spec-compliant solution, it is necessary to confirm it into the end-to-end simulating deck.

An end-to-end simulating deck featuring 6.189-inches long baseboard (optimal *rtg12L* + 0.5 inch of breakout routing), 9.4424-inches long internal cable (optimal *rtg101L*), and 1.095-inch long daughter card (optimal *rtg21L* + 0.6-inch of additional routing segments), including external cable and generic receiver, produces a result of 100 mV_{diff} and 0.34 *UI* (68 ps). Hence, the optimal solution found corresponds to a spec-compliant topology.

Appendix E shows a script that takes the journal file and plot its relevant contents. Plotting the journal file is an easy way to track the evolution of the optimization input and outputs, as well as an effective debugging/improving tool. Fig. 3-1 shows the evolution of monitoring parameters from journal file. From that figure, it is seen that the proposed TD-based optimization procedure focused on improving a suitable eye error function is able to find an optimal design in a relatively small number of simplified simulations.

3.7. Conclusions

Description of a minimax formulation and its implementation in Matlab for solving a non-spec compliant USB 3.0 high speed digital interconnect were presented in this report. A USB 3.0 front panel with daughter card topology was used. An approximation proposal to reduce simulation time of compliance testing was described and implemented. The proposed numeric optimization routine includes a driver function, an objective function, and an optimization algorithm based in Matlab's Nelder-Mead method. The optimization routine reported optimized design variables that were confirmed as a spec-compliant solution in an end-to-end compliance test. A journal file and a plotting script were deployed in the proposed optimization routine.

4. Frequency Domain Simulations of High Speed Digital Interconnects and Comparison with Classical Time Domain Analysis

The configuration of a SPICE-like circuital simulator to obtain S-parameter is described in this chapter. Frequency-domain S-parameters from simulation outcomes are presented and analyzed, as evidence to distinguish between a time-domain spec-compliant topology and a failing topology. This chapter concludes with a comparison of the computational cost between time domain and frequency domain simulations.

4.1. Introduction

Signal integrity is an essential step in the development of modern computing platforms. A typical multi-Gbps system comprehends a driver and receiver, each contained in separate packages, connected each other through an interconnect. The high speed interconnect is usually composed of sockets, multi-transmission line-coupled PCB traces, vias, cables and connectors. The common practice is to perform time-domain transient simulation for solving the complete high speed digital system. The use of SPICE-based circuital solvers is widely used in the computing industry [Fan-10], [Hall-09], [Robledo-14].

It was previously reported on Section 2.6 the computational cost of simulating a high speed digital interconnect with long topology. Simulation time of this order makes impractical any statistical analysis, only the use of compelling server systems can offer the computing power needed to achieve Monte-Carlo type of analysis or direct optimization by classical optimization methods.

HSPICE is a commercial CAD tool that provides expanded capabilities over traditional SPICE solvers of free distribution. One of its added frequency-domain (FD) analysis is the capability to obtain scattering (S) parameters from a given interconnect [Hspice-10].

This chapter presents an illustrative example of FD analysis that offers HSPICE to characterize a multi-Gbps long topology interconnect. The identification of FD parameters that

help identify a topology as spec-compliant is one goal of this study. Exploration of computational cost reduction over the common practice of TD transient analysis is another goal. In this chapter, Section 4.2 describes the topology used in FD analysis. Section 4.3 shows the setup needed in HSPICE to achieve FD interconnect characterization. Results are presented in Section 4.4. Finally, conclusions are given in Section 4.5.

4.2. Topology under Test

A USB3 front panel topology with daughter card is used in this study. This topology, shown in Fig. 2-1, has been described in Section 2.2.

When this topology features a 5-inches long baseboard, 10-inches long internal cable and 1-inch long daughter card, including an external cable and a generic receiver, the TD resulting eye-based parameters satisfy the USB3 spec for compliance testing. The same topology with same lengths but 6-inches in the baseboard, produces non-spec compliant solution. A TD minimax-based optimization routine was proposed in Chapter 3 to find an acceptable solution. The result is that a 6.189-inches long baseboard, 9.4424-inches long internal cable, and 1.095-inch long daughter card, including external cable and generic receiver, barely satisfy the USB3 spec-compliance requirements [Robledo-15b]. Hence, three different topologies are considered.

In order to more easily identify a particular topology, the following naming is adopted. The topology with 5-inches long baseboard, 10-inches long internal cable and 1-inch long daughter card is identified as “usb3_fp_dc_5_10_1”. This is spec-compliant topology.

The topology featuring 6.189-inches long baseboard, 9.4424-inches long internal cable, and 1.095-inch long daughter card is named “usb3_fp_dc_xOptim”. This topology is barely passing spec compliance testing.

Finally, the topology having 6-inches in the baseboard, 10-inches long internal cable and 1-inch long daughter card is named as “usb3_fp_dc_6_10_1”. This topology is non-spec-compliant.

4.3. FD Simulations in Hspice

HSPICE has extended FD analysis capabilities over SPICE free versions. When the .AC command is used in combination with .LIN analysis, HSPICE extracts linear transfer function parameters for a general multi-port network, making available multi-port S-parameters. The .LIN analysis is similar to the basic small-signal frequency sweep .AC analysis, but it also automatically calculates a set of noise and small-signal transfer parameters between the terminals identified by port (P) elements [Hspice-10]. HSPICE can output the results of group delay extraction and two-port noise analysis to a Touchstone 1.0/2.0 file [Ibis-09]. The three topologies described above are simulated in HSPICE to obtain their corresponding S-parameters.

TD simulation methodology requires 3 coupled differential T-lines for studying transmission, reflection and crosstalk phenomena [Robledo-15a]. This structure is leveraged for FD analysis. This results in a 12-port network, then, the corresponding Touchstone file is an .s12p file. Following IEEE port designation, we focus specifically on two S-parameters: insertion loss of one of the middle lines (S_{65}), and return loss of the same middle line (S_{55}). It is seen that single ended S-parameters can properly represent performance of the complete topology, so the treatment of differential S-parameters is avoided for simplicity. The simplified 3 differential interconnect topology and port assignment is shown in Fig. 4-1.

FD simulations are performed up to 10 GHz, ensuring F_{3dB} coverage given the TX driver's rise time. Specific information about driver rise time is omitted to protect Intel's IP.

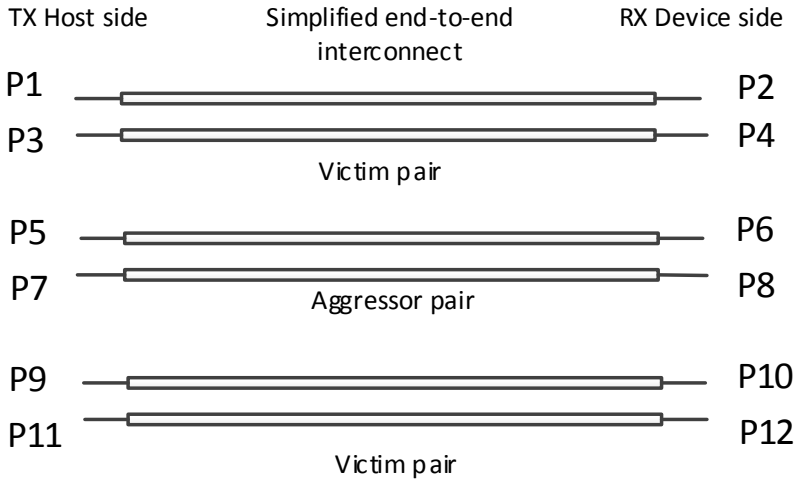


Fig. 4-1 Port designation according to IEEE standards. Insertion loss is taken from S_{65} and return loss comes from S_{55}.

USB3 technology has a bit rate of 5 Gbps, this produces a Nyquist frequency of 2.5 GHz. The highest simulated frequency of 10 GHz allows to observe up to the 4th harmonic.

Appendix F shows the HSPICE netlist of non-spec-compliant topology "usb3_fp_dc_6_10_1", configured for FD analysis as example. In order to built simulation decks for topologies "usb3_fp_dc_5_10_1" and "usb3_fp_dc_xOptim", only routing length parameters need to be adjusted accordingly.

4.4. Results

An internal tool from Intel is used to graphically plot the content of Touchstone files and report the S-parameters in dB versus frequency.

Fig. 4-2 shows the magnitude of insertion loss vs. frequency of the three topologies described in Section II. Similarly, Fig. 4-3 describes the magnitude of return loss of the same topologies.

By visual inspecting these figures, it is hard to realize conclusive differences between the

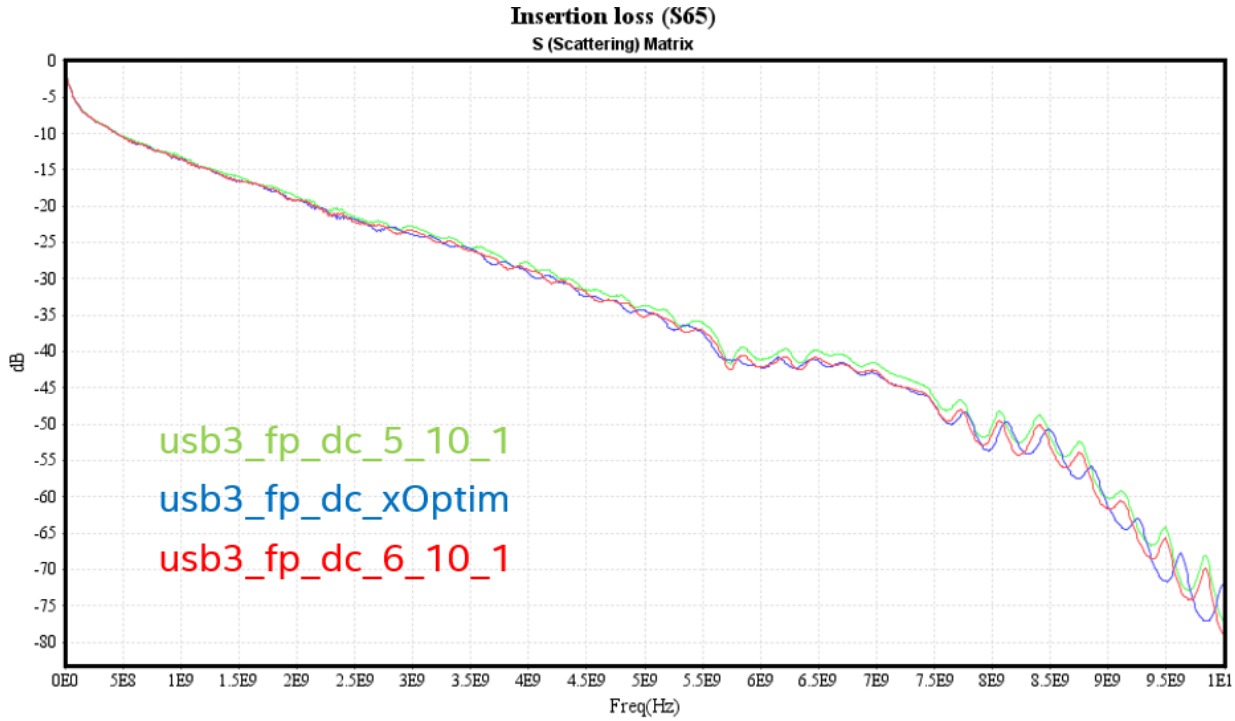


Fig. 4-2 Insertion loss curves of the three topologies under test. At full span, there is no conclusive difference among them.

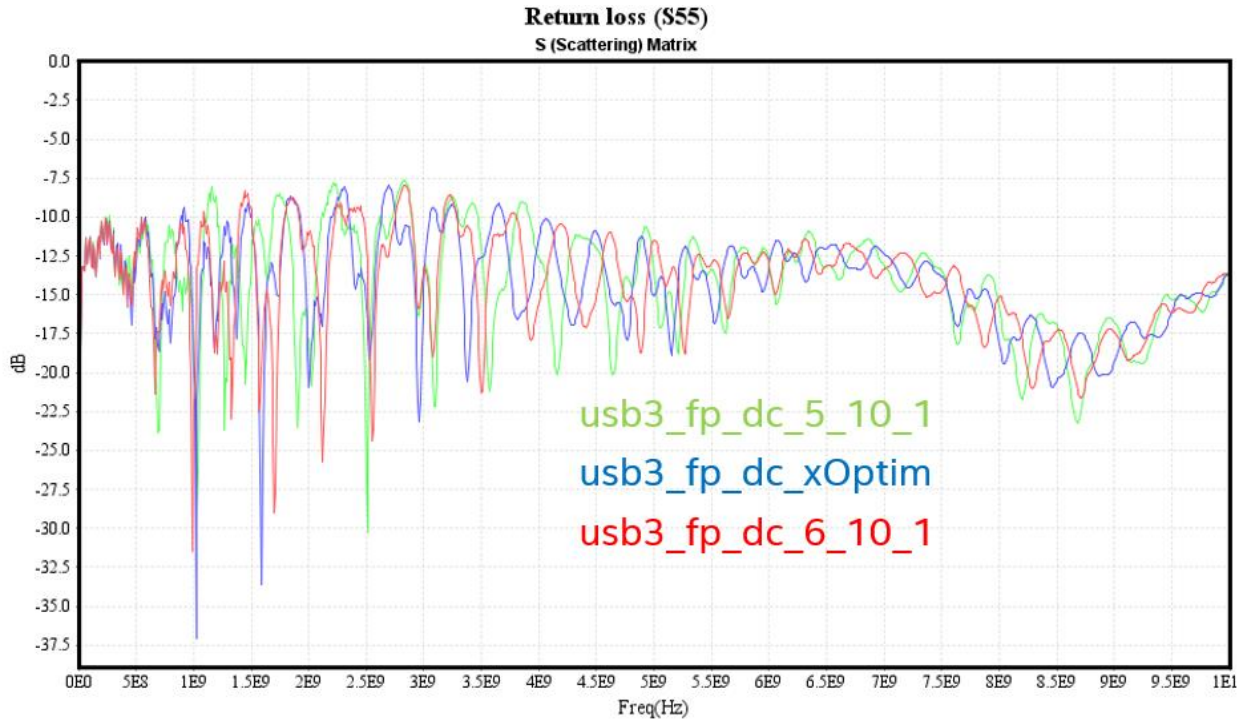


Fig. 4-3 Return loss curves of the three topologies under test. At full span, there is no conclusive difference among them.

S-parameters of each topology. Insertion and return loss curves are inconclusive when seeing the complete frequency span from DC to 10 GHz. However, when analyzing discrete frequency points, more useful information is observed.

TABLE III shows the insertion loss $|S_{65}|$ in dB of three discrete frequency points: Nyquist frequency, 2nd and 3rd harmonics. It is seen that topology usb3_fp_dc_5_10_1 has the best transmission, then topology usb3_fp_dc_xOptim, and finally usb3_fp_dc_6_10_1. These conclusions are consistent with TD simulation results observed in Section 3.6.

Similarly, TABLE IV shows the return loss $|S_{55}|$ in dB at Nyquist frequency, 2nd and 3rd

TABLE III
INSERTION LOSS $|S_{65}|$ AT NYQUIST FREQUENCY, 2ND AND 3RD HARMONICS

Frequency (GHz)	usb3_fp_dc_5_10_1	usb3_fp_dc_xOptim	usb3_fp_dc_6_10_1
2.5	-21.44 dB	-21.79 dB	-22.06 dB
5	-33.67 dB	-34.39 dB	-35.23 dB
7.5	-46.37 dB	-47.32 dB	-47.43 dB

TABLE IV
RETURN LOSS $|S_{55}|$ AT NYQUIST FREQUENCY, 2^{ND} AND 3^{RD} HARMONICS

Frequency (GHz)	usb3_fp_dc_5_10_1	usb3_fp_dc_xOptim	usb3_fp_dc_6_10_1
2.5	-26.97 dB	-15.81 dB	-11.91 dB
5	-13.42 dB	-15.02 dB	-11.51 dB
7.5	-13.41 dB	-13.79 dB	-14.80 dB

harmonics. Topology usb3_fp_dc_5_10_1 shows the best return loss at Nyquist frequency, i.e. it is better terminated in the near end. Topology usb3_fp_dc_xOptim shows a return loss in the middle. The topology that generates the highest return loss is usb3_fp_dc_6_10_1 at Nyquist frequency. This set of observations, are also consistent with TD performance reported in Section 3.6.

Regarding to computational cost, a comparison is made by measuring simulation time of classical TD transient simulation *vs*. FD S-parameters simulation. Completing TD transient simulation of usb3_fp_dc_5_10_1 takes 55.34 minutes, while completing similar topology for FD analysis only takes 2.23 minutes. All simulations are done using a server-class workstation with two CPU Intel® Xeon ® E5-2630 @ 2.3 GHz, and 128 GB of DDR3 memory.

4.5. Conclusions

This chapter presented an HSPICE setup to obtain S-parameter from FD simulations of a high-speed interconnect. The same topology structure from classical TD transient analysis is implemented in FD simulations, resulting in a 12-port network. From S-parameter outcomes, measurements of insertion loss and return loss at specific discrete frequency points seem to indicate that we can distinguish between a spec-compliant solution and non-spec compliant topology. This aspect requires further research. Clearly, a very significant reduction in computational cost is observed when simulating in FD.

5. A Minimax Formulation for a Frequency Domain Numeric Optimization of a Non Spec Compliant USB3 Topology

Typical time-domain signal integrity simulations of long multi Gbps interconnects are impractical for implementation on numerical optimization routines due to the simulation time per case. Frequency-domain circuital simulations over a broad bandwidth provide a more efficient analysis method than classical time-domain circuital simulations. This can be especially useful for simulating high speed interconnects, considering that most of them can be treated as linear circuits. In this chapter, a frequency domain optimization methodology is presented to improve a high speed digital channel design. Time domain simulations are performed to verify that the optimal topology found is a spec compliant solution.

5.1. Introduction

Time-domain and frequency-domain signal integrity simulations are two methodologies that analyze the same electromagnetic phenomena on high speed digital interconnects, but from different perspectives. In the past, the development of these methods have followed totally separated paths. Moreover, the outputs of TD and FD methodologies are given in different formats, and there is not direct translation between them [Hall-00].

TD transient simulations with the use of SPICE-based circuital solvers, and jitter analysis with statistic BER technique for post-processing voltage waveform from circuital solvers are the most popular methodology for designing a high speed digital interconnect. Eye diagram parameters are the final outcome. Industrial specifications usually use eye parameters as channel quality and compliance metrics [Fan-10], [Hall-09], [Robledo-14], [Robledo-15a].

FD simulations are available on some commercial circuital solvers. HSPICE can extract linear transfer function parameters for a general multi-port network, making available multi-port S-parameters. The frequency domain parameters are delivered in Touchstone file format [Ibis-09], [Robledo-16].

In Chapter 4 is also reported that FD simulation significantly reduces simulation time over TD simulation for the same interconnect. This meaningful reduction in computational resources enables FD simulations for direct implementation in an optimization methodology. Recent investigations are being done in the industry to shift multi-Gbps channel design from TD to FD simulations [El-10], [Kim-09], [Win-15].

This chapter presents a FD numerical optimization routine based on a minimax formulation. The topology used in this report is described in section 5.2. Section 5.3 discusses the minimax formulation applied in the optimization routine. The function that drives the circuital simulator and deliver output S-parameters is presented in Section 5.4. Section 5.5 describes the optimization algorithm along with its termination criteria. Results are outlined in Section 5.6. Finally, conclusions are presented in Section 5.7.

5.2. Topology under Test

A non-spec compliant USB3 front panel topology with daughter card is used in this chapter. This topology, shown in Fig. 2-1, has been described and used in previous chapters. Appendix F contains the HSPICE netlist suited for frequency-domain simulation of this topology.

For optimization, selected design variables, pre-assigned parameters, and simulator parameters for FD are described as follows. Design variables ($\boldsymbol{x}$) are the length of main baseboard routing section (*rtg12L*), the length of internal cable (*rtg101L*) and the length of the main section of routing inside the daughter card (*rtg21L*),

$$\boldsymbol{x} = [rtg12L \;\; rtg101L \;\; rtg21L]^T \tag{5-1}$$

Pre-assigned parameters (x_p) is the routing length of the package (*len_pkg1*),

$$x_p = len\_pkg1 \tag{5-2}$$

Finally, parameters of the simulator ($\boldsymbol{p}_s$) are the initial frequency point (*iniFreq*) and final frequency (*finalFreq*),

$$\boldsymbol{p}_s = [iniFreq \;\; finalFreq]^T \tag{5-3}$$

TABLE V shows initial values of all these input parameters.

In Section 4.2, three channel lengths associated to front panel topology are discussed: spec-compliant channel length, non-spec-compliant channel length, and barely spec-passing channel

TABLE V
INPUT SIMULATION PARAMETERS AND INITIAL VALUES

Parameter variable	Parameter name	Parameter identifier	Parameter value (unit)
x_1	Baseboard's main segment length	*rtg12L*	5.5 (in)
x_2	Internal cable length	*rtg101L*	10 (in)
x_3	Daughter card's main segment length	*rtg21L*	0.4 (in)
x_{p1}	Package routing length	*len_pkg1*	0.5 (in)
p_{s1}	Initial frequency point	*iniFreq*	10e6 (Hz)
p_{s2}	Final frequency point	*finalFreq*	10e9 (Hz)

length; the last one is product of TD minimax optimization. This channel features 6.189-inches long baseboard, 9.4424-inches long internal cable, and 1.095-inch long daughter card, and it is named as "usb3_fp_dc_xOptim".

These three channels are simulated in frequency domain and their characteristic S-parameters are outlined in Section 4.4. It is concluded that performance in FD is consistent with TD observations. Since this channel barely passes TD spec compliance testing, it is valid to consider its S-parameters as minimum acceptable FD metrics.

5.3. Minimax Formulation

For FD numerical optimization implementation, specific S-parameters of the "usb3_fp_dc_xOptim" channel are considered as pass-fail thresholds, SP_{thld}. TABLE VI describes these selected S-parameters. After a simulation iteration in the optimization routine, a set of S-parameters are reported, SP_{sim}. Then, SP_{sim} are compared against SP_{thld}. Finally, an error vector is calculated, containing four parameters: return loss error at Nyquist frequency ($errorRL_{h1}$), insertion loss error at Nyquist frequency ($errorIL_{h1}$), insertion loss error at 2nd harmonic ($errorIL_{h2}$) and insertion loss error at 3rd harmonic ($errorIL_{h3}$), as follows

$$errorRL_{h1} = (SP_{\mathrm{sim1}}/SP_{\mathrm{thld1}}) - 1 \tag{5-4}$$

$$errorIL_{h1} = 1 - (SP_{\text{sim2}}/SP_{\text{thld2}}) \quad (5\text{-}5)$$

$$errorIL_{h2} = 1 - (SP_{\text{sim3}}/SP_{\text{thld3}}) \quad (5\text{-}6)$$

$$errorIL_{h3} = 1 - (SP_{\text{sim4}}/SP_{\text{thld4}}) \quad (5\text{-}7)$$

$$e = max[errorRL_{h1} \;\; errorIL_{h1} \;\; errorIL_{h2} \;\; errorIL_{h3}] \quad (5\text{-}8)$$

A positive error in (5-8) corresponds to a failing simulation case. In contrast, a negative error in (5-8) means that simulation case meets pass-fail thresholds, hence, it is a passing simulation case. The minimax formulation is used to minimize the maximum error of a function, in this case e. This report implements an unconstrained minimax formulation [Rayas-11],

$$\boldsymbol{x}^* = \arg min_{\boldsymbol{x}}\, u(\boldsymbol{x}) \text{ where } u(\boldsymbol{x}) = e \quad (5\text{-}9)$$

with vector $\boldsymbol{x}$ containing *rtg12L*, *rtg101L*, and *rtg21L*.

In the optimization routine, (5-4)-(5-8) are implemented and described in the objective function. Appendix G shows the objective function.

5.4. Driver Function

The driver function is fed with design variables, pre-assigned parameters and parameters of the simulator from the objective function. Then, with all these inputs, it builds a copy of the netlist that is described in Appendix F , overwriting the design variables values predicted by the optimization method. Later, HSPICE simulator is invoked and runs the netlist. Frequency-domain simulation output is a Touchstone file. Finally, S-parameters of interest, SP_{sim}, are extracted from Touchstone file and reported back to objective function. Appendix H presents the driver function.

TABLE VI
PASS-FAIL THRESHOLDS, SP_{THLD}, FOR FD OPTIMIZATION

Threshold variable	Threshold name	Threshold value
SP_{thld1}	Return loss magnitude at Nyquist frequency ($\|S_{55}\|$ @ 2.5 GHz)	0.161867
SP_{thld2}	Insertion loss magnitude at Nyquist frequency ($\|S_{65}\|$ @ 2.5 GHz)	0.0813516
SP_{thld3}	Insertion loss magnitude at 2nd harmonic ($\|S_{65}\|$ @ 5 GHz)	0.019056
SP_{thld4}	Insertion loss magnitude at 3rd harmonic ($\|S_{65}\|$ @ 7.5 GHz)	0.0043013

5.5. Optimization Algorithm

Nelder-Mead method implemented in Matlab `fminsearch` function is used as the selected optimization algorithm [Matlab-15]. Optimization is initialized with seed values, $\boldsymbol{x}^{(0)}$, and they are passed directly to the objective function. After first evaluation of $u(\boldsymbol{x})$, `fminsearch` determines next simulating values using a derivative-free method.

`fminsearch` function stops when termination criteria are met. Three termination criteria are defined: maximum number of iterations (*MaxIter*) equal to 50, change in function evaluation (*TolFun*) is less than 1×10^{-2}, and change in next simulation inputs (*TolX*) are less than 1×10^{-2}.

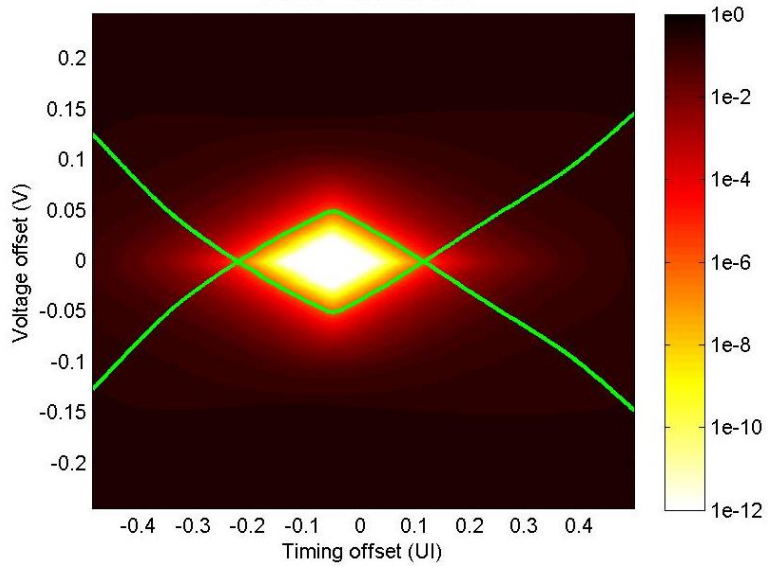
Once optimization algorithm stops, $\boldsymbol{x}^*$ is reported in vector *x_opt*, $u(\boldsymbol{x}^*)$ is shown as well (*feval*), and active termination criterion (*exitflag*). Furthermore, some additional parameters are reported inside *output* variable: number of iterations (*ni*), number of functions evaluations (*nfeval*) and total elapsed time (*tsimtime*). Optimization algorithm is shown in Appendix I .

5.6. Results

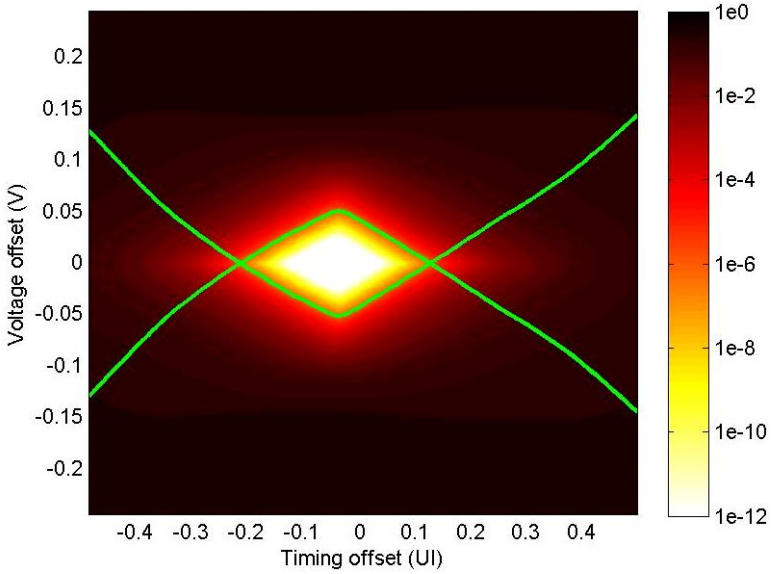
The first tested seed values, $\boldsymbol{x}^{(0)} = [5.5 \quad 10 \quad 0.4]^T$ (in), corresponds to non-spec compliant topology described in Section 5.2. Optimization algorithm finds a local minimum, reporting at the end a function value of $u(\boldsymbol{x}^*) = 0.0214$, which does not meet all FD specifications SP_{thld}.

A second vector of seed values is tested, $\boldsymbol{x}^{(0)} = [5 \quad 10 \quad 1.4]^T$ (in), where the initial function evaluation is $u(\boldsymbol{x}^{(0)}) = 1.3159$. After completion of the optimization routine, the optimal design variables reported are $\boldsymbol{x}^* = [4.509 \quad 10.9434 \quad 1.4065]^T$ (in) and a resulting $u(\boldsymbol{x}^*) = -0.0127$ is found. This result meets all design specifications SP_{thld}.

Since this FD simulation methodology is not a direct translation to TD eye diagram parameters, TD compliance testing simulations are performed for verification. The second seed values $\boldsymbol{x}^{(0)} = [5 \quad 10 \quad 1.4]^T$ (in) produced the following non passing eye parameters $EH = 99\ \text{mV}_{\text{diff}}$ and $EW = 0.339\ UI$ (67.8 ps). Later, TD simulations are done at $\boldsymbol{x}^* = [4.509 \quad 10.9434 \quad 1.4065]^T$ (in), where eye height is $EH = 102\ \text{mV}_{\text{diff}}$ and eye width is $EW = 0.342\ UI$ (68.4 ps), meeting industrial specification for compliance testing requirements. Fig. 5-1 shows both resulting eye diagrams. At a glance, it is difficult to notice a difference between eye diagrams, this is due to the



a)



b)

Fig. 5-1 a) Resulting eye diagram of seed design variables $\boldsymbol{x}^{(0)} = [5 \quad 10 \quad 1.4]^T$ (in), as function of BER. Highlighted green eye measures EH = 99 mV_{diff} and EW = 0.339 UI, located at BER = 10^{-6} bits.

b) Resulting eye diagram of optimized design variables $\boldsymbol{x}^* = [4.3142 \quad 10.514 \quad 0.4332]^T$ (in). Highlighted green eye measures EH = 102 mV_{diff} and EW = 0.342 UI.

scale. However, the numeric differences between seed values and optimum values make a distinction between failing and passing spec compliance testing respectively.

TABLE VII shows a summary of results after testing several seed values. This table shows

TABLE VII
FD OPTIMIZATION PERFORMANCE FOR SEVERAL SEED VALUES

x_1^0	x_2^0	x_3^0	x_1^*	x_2^*	x_3^*	$u(\boldsymbol{x}^0)$	$u(\boldsymbol{x}^*)$	*ni*	*nfeval*	*tsimtime (min)*
5.5	10	0.4	5.4675	10.4713	0.403	0.5671	0.0214	37	71	170
5	10	1.4	4.509	10.9434	1.4065	1.3159	-0.0127	37	69	214
5.5	10	0.9	5.5537	10.2498	0.8297	0.4223	0.0486	35	68	163
6	10	0.4	5.8215	9.7347	0.4221	0.8472	0.0217	21	42	108

for each case the optimal design parameters, the initial and optimal function values, the number of iterations required, the number of function evaluations required, and the total optimization time (*tsimtime*). The objective function seems to have plenty of local minima; this can be concluded after observing that only one out of four seeds leads to a final negative function value.

All simulations are done using a server-class workstation with two CPU Intel® Xeon ® E5-2630 @ 2.3 GHz, and 128 GB of DDR3 memory.

5.7. Conclusions

This chapter proposed a FD numerical optimization methodology to find a spec compliant design of a high speed digital interconnect, based on a minimax formulation. Pass-fail thresholds are defined based on S-parameters of a known topology that barely meets compliance testing; also, these thresholds are used to calculate the maximum error in the minimax FD formulation. The implementation of the optimization routine successfully finds a topology that meets FD design requirements. Finally, TD compliance testing simulations are performed on the optimal topology found to validate that it actually is a spec compliant solution.

General Conclusions

The modern design for computing platforms has become exponentially more complicated from previous generation's designs, where even signal integrity (SI) was not considered as a relevant phenomenon in digital interconnects; worst case signal's voltage and timing analysis used to be sufficient in warranting a successful design. Increasing clock frequencies, higher bus data transfer rates and elevated chip integration, as well as smaller signal voltage swing and lower platform spacing for signal routing have set the scenario where RF and microwaves related phenomena describe entirely the behavior of a signal traveling in a digital interconnect. SI is now an obligatory analysis stage for platform design. Even though SI is a developed area, research is still widely current the computing industry.

An effort to assist in the challenging SI process improvement of non-spec-compliant high speed digital interconnects is presented in this thesis through the implementation of numerical optimization methodologies. A brief introduction to the topic of SI simulation for computer platforms was first presented in Chapter 1.

Classical time domain (TD) SI process was presented in Chapter 2. A detailed description of the methodology is given through the simulation of non-spec-compliant USB3 topology example, the same topology used as a seed in the TD optimization implementation. Two mayor steps conform classical TD simulation process: first, to perform transient simulation using a SPICE based circuital solver, where the output is a step response voltage waveform versus time; and then, post-process the waveform using TD jitter with statistic BER technique, it uses the step response of the channel and directly converts TD waveforms through convolution and superposition to eye based voltage and time margins. The method is capable of adding transmitting and receiving jitter.

A time domain based numerical optimization implementation that involve classical SI methods was presented in Chapter 3. The core of the implementation is the Minimax formulation built upon eye voltage and time measurements directly. Starting with design parameters as a seed from a non-spec compliant USB3 topology, the TD implementation found an optimized set of design parameters that produced satisfactory compliance testing results, meeting industrial USB3 specification.

An important limitation of the TD SI methodology is the extensive simulation time that long high speed interconnects typically require, becoming TD numerical optimization implementation unfeasible in many practical cases.

HSPICE's frequency domain (FD) analysis capabilities were explored in Chapter 4 as an alternative SI method to reduce simulation time for long topologies. An HSPICE setup to obtain S-parameters in form of a Touchstone file was described. Three USB3 front panel topologies were tested: first, the non-spec-compliant topology example from Chapters 2 and 3, that fails the spec requirements; then, the optimized topology that resulted from the TD based implementation described in Section 3.7, that marginally passes spec requirements; and finally, a spec-compliant topology described in Section 3.2.2 that passes spec requirements with good margins. As the main conclusion, significant reduction in simulation time over classical TD SI methodology was observed, from roughly 55 minutes in TD to only 2 minutes in FD for the same interconnect. Additionally, S-parameters measurements of insertion loss and return loss at specific frequency points indicated that a distinction between spec-compliant and non-spec compliant topologies in FD is possible.

After reduction of simulation time was verified for FD SI process over classical TD SI procedure, a second optimization methodology was presented in Chapter 5, main contribution of this present thesis work. The FD based numerical optimization methodology contains the same building blocks used for TD based implementation, which are the driver function, the objective function and the optimization algorithm. The core of the optimization routine is now an S-parameter based minimax formulation, where insertion loss and return loss thresholds were defined from the topology that marginally meets compliance testing. The implementation of the optimization routine found a topology that meets FD design requirements, however several local minima were detected for this topology. Finally, classical TD SI simulations were performed on the optimal design to validate USB industrial compliance.

The present thesis may be subjected to further improvements. Although the main investigation goals were met and demonstrated, some future research work for those interested in the topic is suggested. Firstly, additional research could be deployed in improving the correlation between TD and FD optimization implementations; optimized design parameters that lead in a spec-compliant design were found in the first try when using TD optimization implementation, in contrast to several local minima found while working with FD optimization implementation.

Additionally, neither differential signaling nor crosstalk effects were included in FD optimization implementation; this was done for simplicity purposes accounting only for insertion and reflection losses effects. Finally, other CAD FD simulation tools could be investigated for better SI performance; although HSPICE is widely accepted by industry as the leader TD simulation tool, other commercial tools may be better suited for FD simulations. The author believes that additional effort in these areas will substantially improve the proposed methodologies.

Appendices

A TIME DOMAIN HSPICE SIMULATION DECK OF NON-SPEC COMPLIAN USB3 FRONT PANEL WITH DAUGHTER CARD TOPOLOGY

```
* Options*********************************************************************
.tran 2ps 120ns

* Tx Model*********************************************************************
.include '..\..\models\buffer\usb3_device_tx\dev_usb3_tx_r0p5a_SD.inc'
.param Vswing = 1
.param dtr = 35p
.param dtf = 35p
.param rterm_drv = '90/2'
.param cdie_drv = 1.35p
.param de_emp = 0
xpad1 pad1_pos1 pad1_neg1 pad1_pos2 pad1_neg2 pad1_pos3 pad1_neg3  gnd Tx_3pairs_SD

* pkgnew 1*********************************************************************
.lib '..\..\models\package\usb_device\dev_usb3_r0p5a_SD.lib' pk1_3typ
.param len_pkg1 = '0.5*.0254'
xpkgnew1 pad1_pos1 pad1_neg1 pad1_pos2 pad1_neg2 pad1_pos3 pad1_neg3
+pkgnew1_pos1 pkgnew1_neg1 pkgnew1_pos2 pkgnew1_neg2 pkgnew1_pos3 pkgnew1_neg3  gnd PKG1_SD

* Via 1************************************************************************
.lib '..\..\models\via\lbg_r0p5a_SD.lib' lbg_16L
xvia1 pkgnew1_pos1 pkgnew1_neg1 pkgnew1_pos2 pkgnew1_neg2 pkgnew1_pos3 pkgnew1_neg3
+            via1_pos1 via1_neg1 via1_pos2 via1_neg2 via1_pos3 via1_neg3  gnd cpu093_16L_L1L14_ent0_ext10_lbg_r0p5a_3p

* Routing 11*******************************************************************
.lib '..\..\models\routing\cpu_xxd_asl_bopf_4h15_xz_xxxdb_r0p5a_SD.lib' cpu_xxd_asl_bopf_4h15_xz_xxxdb_r0p5a_SD
xrtg11 via1_pos1 via1_neg1 via1_pos2 via1_neg2 via1_pos3 via1_neg3
+            rtg11_pos1 rtg11_neg1 rtg11_pos2 rtg11_neg2 rtg11_pos3 rtg11_neg3  gnd cpu_xxd_asl_bo_4h15_nz_0p72db_r0p5a
+            Length = '0.5*.0254'

* Routing 12*******************************************************************
.lib '..\..\models\routing\cpu_85d_asl_xx_4h15_xz_xxxdb_r0p5a_SD.lib' cpu_85d_asl_xx_4h15_xz_xxxdb_r0p5a_SD
xrtg12 rtg11_pos1 rtg11_neg1 rtg11_pos2 rtg11_neg2 rtg11_pos3 rtg11_neg3
+            rtg12_pos1 rtg12_neg1 rtg12_pos2 rtg12_neg2 rtg12_pos3 rtg12_neg3  gnd cpu_85d_asl_5x_4h15_nz_0p72db_r0p5a
+            Length = '5.5*.0254'

* Connector 1******************************************************************
.lib '..\..\models\connector\internal_conn_fpHdr\usb3_fpHdr_rev05_thm_93mils_r0p5a_SD.lib' USB3_int_conn_stubs_14L_93mils
xcon1 rtg12_pos1 rtg12_neg1 rtg12_pos2 rtg12_neg2 rtg12_pos3 rtg12_neg3
+            con1_pos1 con1_neg1 con1_pos2 con1_neg2 con1_pos3 con1_neg3  gnd USB3_int_conn_L12_entry_RxTxTx

* Routing 101******************************************************************
.include '..\..\models\cable\usb3_xxx_90d_xx_awg28_unc_r0p5a_SD.inc'
xrtg101 con1_pos1 con1_neg1 con1_pos2 con1_neg2 con1_pos3 con1_neg3
+            rtg101_pos1 rtg101_neg1 rtg101_pos2 rtg101_neg2 rtg101_pos3 rtg101_neg3  gnd ext_cable_skew_0ps_Z0_typ_SD
+            Length = '10*.0254'

* Connector 2******************************************************************
.include '..\..\models\connector\internal_conn_fpHdr\usb3_fpHdr_rev15_vh_thm_62mils_r0p5a_SD.inc'
xcon2 con2_pos1 con2_neg1 con2_pos2 con2_neg2 con2_pos3 con2_neg3
+            rtg101_pos1 rtg101_neg1 rtg101_pos2 rtg101_neg2 rtg101_pos3 rtg101_neg3  gnd int_usb3rx_usb3tx_usb2

* Routing 20*******************************************************************
.lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
```

```
xrtg20 con2_pos1 con2_neg1 con2_pos2 con2_neg2 con2_pos3 con2_neg3
+            rtg20_pos1 rtg20_neg1 rtg20_pos2 rtg20_neg2 rtg20_pos3 rtg20_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p72db_r0p5a
+            Length = '0.2*.0254'

* CAP 1*********************************************************************************
.include '..\..\models\cap\cap_novalue_0402_novoid_r0p5a_SD.inc'
xcap1 rtg20_pos1 rtg20_neg1 rtg20_pos2 rtg20_neg2 rtg20_pos3 rtg20_neg3
+            cap1_pos1 cap1_neg1 cap1_pos2 cap1_neg2 cap1_pos3 cap1_neg3  gnd cap_novalue_0402_novoid_r0p5a

* Routing 21*****************************************************************************
* .lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg21 cap1_pos1 cap1_neg1 cap1_pos2 cap1_neg2 cap1_pos3 cap1_neg3
+            rtg21_pos1 rtg21_neg1 rtg21_pos2 rtg21_neg2 rtg21_pos3 rtg21_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p72db_r0p5a
+            Length = '0.4*.0254'

* CMC 1*********************************************************************************
.include '..\..\models\misc\usb3_common_mode_choke_SD.inc'
xcmc1 rtg21_pos1 rtg21_neg1 rtg21_pos2 rtg21_neg2 rtg21_pos3 rtg21_neg3
+            cmc1_pos1 cmc1_neg1 cmc1_pos2 cmc1_neg2 cmc1_pos3 cmc1_neg3  gnd common_mode_choke_rev1p0_sd

* Routing 22*****************************************************************************
* .lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg22 cmc1_pos1 cmc1_neg1 cmc1_pos2 cmc1_neg2 cmc1_pos3 cmc1_neg3
+            rtg22_pos1 rtg22_neg1 rtg22_pos2 rtg22_neg2 rtg22_pos3 rtg22_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p72db_r0p5a
+            Length = '0.2*.0254'

* ESD 1*********************************************************************************
.include '..\..\models\misc\esd_SD.inc'
xesd1 rtg22_pos1 rtg22_neg1 rtg22_pos2 rtg22_neg2 rtg22_pos3 rtg22_neg3
+            esd1_pos1 esd1_neg1 esd1_pos2 esd1_neg2 esd1_pos3 esd1_neg3  gnd esd_SD

* Routing 23*****************************************************************************
* .lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg23 esd1_pos1 esd1_neg1 esd1_pos2 esd1_neg2 esd1_pos3 esd1_neg3
+            rtg23_pos1 rtg23_neg1 rtg23_pos2 rtg23_neg2 rtg23_pos3 rtg23_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p72db_r0p5a
+            Length = '0.2*.0254'

* Connector 3*****************************************************************************
.include '..\..\models\connector\std_a_conn\usb3_StdA_rev12_thm_TopPort_BtmEntry_r0p5a_SD.inc'
xcon3 rtg23_pos1 rtg23_neg1 rtg23_pos2 rtg23_neg2 rtg23_pos3 rtg23_neg3
+            con3_pos1 con3_neg1 con3_pos2 con3_neg2 con3_pos3 con3_neg3  gnd usb3rx_usb3tx_usb3tx

* Routing 102*****************************************************************************
* .include '..\..\models\cable\usb3_xxx_90d_xx_awg28_unc_r0p5a_SD.inc'
xrtg102 con3_pos1 con3_neg1 con3_pos2 con3_neg2 con3_pos3 con3_neg3
+            rtg102_pos1 rtg102_neg1 rtg102_pos2 rtg102_neg2 rtg102_pos3 rtg102_neg3  gnd ext_cable_skew_0ps_Z0_typ_SD
+            Length = '106*.0254'

* Connector 4*****************************************************************************
.include '..\..\models\connector\std_b_conn\usb3_StdB_rev12_thm_r0p5a_SD.inc'
xcon4 con4_pos1 con4_neg1 con4_pos2 con4_neg2 con4_pos3 con4_neg3
+            rtg102_pos1 rtg102_neg1 rtg102_pos2 rtg102_neg2 rtg102_pos3 rtg102_neg3  gnd stdB_usb3tx_usb3rx_usb3rx

* Routing 31*****************************************************************************
* .lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg31 con4_pos1 con4_neg1 con4_pos2 con4_neg2 con4_pos3 con4_neg3
+            rtg31_pos1 rtg31_neg1 rtg31_pos2 rtg31_neg2 rtg31_pos3 rtg31_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p88db_r0p5a
+            Length = '1.5*.0254'

* Via 2*********************************************************************************
```

```
.lib '..\..\models\via\brd_r0p5a_SD.lib' brd_14L
xvia2 rtg31_pos1 rtg31_neg1 rtg31_pos2 rtg31_neg2 rtg31_pos3 rtg31_neg3
+              via2_pos1 via2_neg1 via2_pos2 via2_neg2 via2_pos3 via2_neg3  gnd cpu093_14L_L1L14_ent0_ext0_brd_r0p5a_3p

* pkgnew 2*****************************************************************************
.lib '..\..\models\package\usb_device\dev_usb3_r0p5a_SD.lib' pk2_3typ
.param len_pkg2 = '0.5*.0254'
xpkgnew2 pkgnew2_pos1 pkgnew2_neg1 pkgnew2_pos2 pkgnew2_neg2 pkgnew2_pos3 pkgnew2_neg3
+              via2_pos1 via2_neg1 via2_pos2 via2_neg2 via2_pos3 via2_neg3  gnd PKG2_SD

* Rx Model 1*****************************************************************************
.include '..\..\models\buffer\usb3_device_rx\dev_usb3_rx_r0p5a_SD.inc'
.param Rterm_rec = '90/2'
.param cdie_rec = '1.35p'
xpad21 pkgnew2_pos1 pkgnew2_neg1 pkgnew2_pos2 pkgnew2_neg2 pkgnew2_pos3 pkgnew2_neg3  gnd general_rec_SD
.PROBE
+ diff_rx1=par('v(pkgnew2_pos1)- v(pkgnew2_neg1)')
+ diff_rx2=par('v(pkgnew2_pos2)- v(pkgnew2_neg2)')
+ diff_rx3=par('v(pkgnew2_pos3)- v(pkgnew2_neg3)')

              .end
```

B OBJECTIVE FUNCTION OF TIME-DOMAIN NUMERIC OPTIMIZATION IMPLEMENTATION

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Ing. Juan Robledo              INTEL/ITESO                    Jul 20, 2015
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Objective function for simulation deck:  usb3_fp_dc_6_10_1
% Usage: function U=usb3_fp_dc_obj_funct(Xofn)
% X = [rtg12L rtg101L rtg21L], Design variables.
% U is eye heigh (EH) and eye width (EW) for optimization.
% Functions required: None.

function U=usb3_fp_dc_obj_funct(Xofi)
% X = [rtg12L rtg101L rtg21L], Design variables.---------------------------
rtg12L = 5.5;               % baseboard routing length (in)
rtg101L = 10;               % internal cable length (in)
rtg21L = 0.4;               % daughter card routing length (in)
x0 = [rtg12L rtg101L rtg21L];

% Unnormilizing design variables.------------------------------------------
% Xofi=Xofn.*x0;
Xofn=Xofi./x0;

% Xp = [vswing rterm_tx], Pre-assigned parameters.-------------------------
vswing = 1;                 % Buffer vswing (V)
rterm_tx = 100;                 % Tx driver differential termination (ohm)
Xp = [vswing rterm_tx];

% Ps = [timeStep finalTime], Parameters of the simulator.------------------
timeStep = 10;              % Time step (ps).
finalTime = 60;             % Final time (ns).
Ps = [timeStep finalTime];

% Calculate responses------------------------------------------------------
[EH, EW, HST, SST, LST] = hspice_postproc_driver(Xofi,Xp,Ps);

errorEH = (0.23-EH)/0.23;     % Minimum relative-compliant EH is 230 mV diff
errorEW = (0.38-EW)/0.38;   % Minimum relative-compliant EW is 0.38 UI
error = [errorEH errorEW];
U = max(error);

% Writing journal file-----------------------------------------------------
fid_oj = fopen('optim_journal.csv','a');
fprintf(fid_oj, '%s', [num2str(Xofi(1)) ',' num2str(Xofi(2)) ','
num2str(Xofi(3)) ',' ...
    num2str(Xofn(1)) ',' num2str(Xofn(2)) ',' num2str(Xofn(3)) ',' ...
    num2str(EH) ',' num2str(EW) ',' ...
    mat2str(HST/60,3) ',' mat2str(SST/60,3) ',' mat2str(LST/60,3) ','
num2str(U)] );
fprintf(fid_oj, '%s\r\n','');
fclose (fid_oj);
end
```

C DRIVER FUNCTION OF TIME-DOMAIN NUMERIC OPTIMIZATION IMPLEMENTATION

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Ing. Juan Robledo              INTEL/ITESO                   Jul 29, 2015
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% This function drives Hspice netlist and postprocessing files from Matlab,
% and returns the eye heigth and eye width results as scalars.
%
% Usage: [EH, EW, HST, SST, LST] = hspice_postproc_driver(X,Xp,Ps)
% X = design variables.
% Xp = pre-assigned parameters.
% Ps = parameters of the simulator.
% Functions required: spReader2Driver.

function [EH, EW, HST, SST, LST] = hspice_postproc_driver(X,Xp,Ps)

lti = clock;

% X = [rtg12L rtg101L rtg21L], Design variables.---------------------------
rtg12L = num2str(X(1)) ;        % baseboard routing length (in)
rtg101L = num2str(X(2)) ;       % internal cable length (in)
rtg21L = num2str(X(3)) ;        % daughter card routing length (in)

% Xp = [vswing rterm_tx], Pre-assigned parameters.-------------------------
vswing = num2str(Xp(1));        % Buffer vswing (V)
rterm_tx = num2str(Xp(2));      % Tx driver differential termination (ohm)

% Ps = [timeStep finalTime], Parameters of the simulator.------------------
timeStep = num2str(Ps(1));      % Time step (ps).
finalTime = num2str(Ps(2));     % Final time (ns).

% Buiding array of strings that contain netlist info.----------------------
arrStr=spReader2Driver('usb3_fp_dc_tx_6_10_1_input_drv.sp');

arrStr{14}= ['.OPTION runlvl=6 wdelayopt=3 delmax=' timeStep 'p risetime=100p
METHOD=GEAR MAXORD=3'];
arrStr{15}= ['.tran ' timeStep 'ps ' finalTime 'ns'];
arrStr{19}= ['.param Vswing = ' vswing '    $ Differential Vswing'];
arrStr{22}= ['.param rterm_drv = ''' rterm_tx '/2'' $ Tx termination'];
arrStr{49}= ['+ Length = ''' rtg12L '*.0254'''];
arrStr{60}= ['+ Length = ''' rtg101L '*.0254'''];
arrStr{82}= ['+ Length = ''' rtg21L '*.0254'''];

% Saving Hspice netlist in Matlab Working Directory-----------------------
hs_fileName='usb3_fp_dc_tx_iter_drv';

rows= length(arrStr);
fp = fopen(sprintf('%s',hs_fileName,'.sp'),'w+');
for i = 1:rows
```

```
    fprintf(fp,'%s', arrStr{i} );
    fprintf(fp,'%s\r\n','');
end
fclose(fp);

% Hspice Executable File in Command Line Mode-----------------------------
ExecProg= 'C:\synopsys\Hspice_E-2010.12-SP2\BIN\hspice.exe ';
ProjFile= sprintf('%s','-i ',hs_fileName,'.sp -o ',hs_fileName,'.lis -mt 4');
thsi = clock;
system([ExecProg ProjFile]);
thsf = clock;
HST = etime(thsf,thsi);

% Run Sigsim 3.7.0---------------------------------------------------------
work_dir='J:\07_master_degree_thesis\projects\150820_usb3_fp_dc_stdA_td_optim
_ndrive';
ss_xls='\sigsim370_usb3_generic.xls';
ss_col=1:1;
ss_row=1:1;

cd ..\..\..\04_sigsim_370_source_code\;
tssi = clock;
sigsim_wrapper(sprintf('%s',work_dir,ss_xls), ss_col, ss_row, 'spreadsheet');
tssf = clock;
SST = etime(tssf,tssi);
cd (sprintf('%s',work_dir));

% Read Sigsim Output File--------------------------------------------------
cd ss_results\
rng='B40..B41';
out=csvread('1_1.csv',39,1,rng);
EH=out(1);
EW=out(2);

% Erase Unwanted Output Data Files-----------------------------------------
delete rx_aperture_distribution*.jpg;
cd ..\
delete (sprintf('%s',hs_fileName,'.ic0'));
delete (sprintf('%s',hs_fileName,'.pa0'));
delete (sprintf('%s',hs_fileName,'.st0'));
delete (sprintf('%s',hs_fileName,'.lis'));

% Reporting Simulation Times-----------------------------------------------
ltf = clock;
LST = etime(ltf,lti);
% disp(['Hspice simulation time = ' mat2str(HST/60,5) ' minutes']);
% disp(['Sigsim simulation time = ' mat2str(SST/60,5) ' minutes']);
disp(['Loop simulation time = ' mat2str(LST/60,5) ' minutes']);

end
```

D TIME-DOMAIN OPTIMIZATION ALGORITHM

```
TSTi= clock;
fid_oj = fopen('optim_journal.csv','w');
fprintf(fid_oj,'%s', ['rtg12L(in),rtg101L(in),rtg21L(in),' ...
    'rtg12L_n,rtg101L_n,rtg21L_n,EH(V),EW(UI),' ...
    'HspiceSimTime(min),SigsimTime(min),LoopSimTime(min),U(max error)']);
fprintf(fid_oj,'%s\r\n','');
fclose(fid_oj);

% X = [rtg12L rtg101L rtg21L], Design variables.--------------------------
rtg12L = 5.5;           % baseboard routing length (in)
rtg101L = 10;           % internal cable length (in)
rtg21L = 0.4;           % daughter card routing length (in)
x0 = [rtg12L rtg101L rtg21L];
% x0n=x0./x0;

% Call to optimization algorithm-----------------------------------------
MaxIter=50;
TolFun=1e-2;
TolX=1e-2;
%
options=optimset('Display','final','MaxIter',MaxIter,'PlotFcns',@optimplotfva
l,'TolX',TolX,'TolFun',TolFun);
options=optimset('Display','final','MaxIter',MaxIter,'TolX',TolX,'TolFun',Tol
Fun);
[xn_opt,feval,exitflag,output]=fminsearch('usb3_fp_dc_obj_funct', x0,
options)
% x_opt=xn_opt.*x0
x_opt=xn_opt;

TSTf = clock;
TST = etime(TSTf,TSTi);
disp(['Total simulation time = ' mat2str(TST/60,3) ' minutes or '
mat2str(TST/3600,5) ' hours']);

%exitflag---------------------------------------------------------------
%1  fminsearch converged to a solution x.
%0  Maximum number of function evaluations or iterations was reached.
%-1 Algorithm was terminated by the output function
```

E PLOTTING JOURNAL SCRIPT OF TIME-DOMAIN NUMERIC OPTIMIZATION IMPLEMENTATION

```
% Generating Matrix from journal variables---------------------------------
jrlMtx=csvread('150823A_optim_journal.csv',1,0); %Filename here
jm_rtg12l=jrlMtx(:,1);
jm_rtg101l=jrlMtx(:,2);
jm_rtg21l=jrlMtx(:,3);
jm_rtg12n=jrlMtx(:,4);
jm_rtg101n=jrlMtx(:,5);
jm_rtg21n=jrlMtx(:,6);
jm_EH=jrlMtx(:,7);
jm_EW=jrlMtx(:,8);
jm_hst=jrlMtx(:,9);
jm_sst=jrlMtx(:,10);
jm_lst=jrlMtx(:,11);
jm_ui=jrlMtx(:,12);

% Plotting Outcomes Evolution-----------------------------------------------
figure(2)
set(axes,'FontName','Times','Fontsize',12);
subplot(4,1,1),plot(jm_rtg12n,'--b','LineWidth',2)
hold on
plot(jm_rtg101n,':g','LineWidth',2)
plot(jm_rtg21n,'-r','LineWidth',2)
hold off
title('Optimization variables per cycle', 'Fontsize', 14,'FontName','Times')
legend('rtg12L','rtg101L','rtg21L','Location','Best')
xlabel('Iterations','FontSize',14,'FontName','Times')
ylabel('xi/x0','FontSize',14,'FontName','Times')
grid on

subplot(4,1,2),plot(jm_EH,'--b','LineWidth',2)
% hold on
vecL=length(jm_EH);
% EHspec=0.1*ones(vecL,1);              % Min EH is 100mV diff. USB3 spec
% plot(EHspec,'r')
% hold off
title('Eye height (EH)', 'Fontsize', 14,'FontName','Times')
% legend('EH','spec')
legend('EH','Location','Best')
xlabel('Iterations','FontSize',14,'FontName','Times')
ylabel('Voltage (V)','FontSize',14,'FontName','Times')
grid on

subplot(4,1,3),plot(jm_EW,'--b','LineWidth',2)
% hold on
% EWspec=0.34*ones(vecL,1);             % Min EW is 0.34 UI. USB3 spec
% plot(EWspec,'r')
% hold off
title('Eye width (EW)', 'Fontsize', 14,'FontName','Times')
% legend('EW','spec')
legend('EW','Location','Best')
```

```
xlabel('Iterations','FontSize',14,'FontName','Times')
ylabel('Time (UI)','FontSize',14,'FontName','Times')
grid on

subplot(4,1,4),plot(jm_ui,'--b','LineWidth',2)
hold on
deckPass=zeros(vecL,1);
plot(deckPass,'-r','LineWidth',2)
hold off
title('Objective function evaluation', 'Fontsize', 14,'FontName','Times')
legend('u(x)','Pass criteria','Location','Best')
xlabel('Iterations','FontSize',14,'FontName','Times')
ylabel('Max error','FontSize',14,'FontName','Times')
grid on
```

F FREQUENCY DOMAIN HSPICE NETLIST OF USB3 NON-SPEC-COMPLIANT TOPOLOGY

```
* Options*********************************************************************
.AC lin 1000 10e6  10e9
.LIN format=touchstone

* Tx Model*********************************************************************
P1 pad1_pos1  0 DC=0 AC=0 z0=50 port=1
P3 pad1_neg1  0 DC=0 AC=0 z0=50 port=3
P5 pad1_pos2  0 DC=0 AC=0 z0=50 port=5
P7 pad1_neg2  0 DC=0 AC=0 z0=50 port=7
P9 pad1_pos3  0 DC=0 AC=0 z0=50 port=9
P11 pad1_neg3  0 DC=0 AC=0 z0=50 port=11

* pkgnew 1*********************************************************************
.lib '..\..\models\package\usb_device\dev_usb3_r0p5a_SD.lib' pk1_3typ
.param len_pkg1 = '0.5*.0254'
xpkgnew1 pad1_pos1 pad1_neg1 pad1_pos2 pad1_neg2 pad1_pos3 pad1_neg3
+            pkgnew1_pos1 pkgnew1_neg1 pkgnew1_pos2 pkgnew1_neg2 pkgnew1_pos3 pkgnew1_neg3  gnd PKG1_SD

* Via 1************************************************************************
.lib '..\..\models\via\lbg_r0p5a_SD.lib' lbg_16L
xvia1 pkgnew1_pos1 pkgnew1_neg1 pkgnew1_pos2 pkgnew1_neg2 pkgnew1_pos3 pkgnew1_neg3
+            via1_pos1 via1_neg1 via1_pos2 via1_neg2 via1_pos3 via1_neg3  gnd cpu093_16L_L1L14_ent0_ext10_lbg_r0p5a_3p

* Routing 11*******************************************************************
.lib '..\..\models\routing\cpu_xxd_asl_bopf_4h15_xz_xxxdb_r0p5a_SD.lib' cpu_xxd_asl_bopf_4h15_xz_xxxdb_r0p5a_SD
xrtg11 via1_pos1 via1_neg1 via1_pos2 via1_neg2 via1_pos3 via1_neg3
+            rtg11_pos1 rtg11_neg1 rtg11_pos2 rtg11_neg2 rtg11_pos3 rtg11_neg3  gnd cpu_xxd_asl_bo_4h15_nz_0p72db_r0p5a
+            Length = '0.5*.0254'

* Routing 12*******************************************************************
.lib '..\..\models\routing\cpu_85d_asl_xx_4h15_xz_xxxdb_r0p5a_SD.lib' cpu_85d_asl_xx_4h15_xz_xxxdb_r0p5a_SD
xrtg12 rtg11_pos1 rtg11_neg1 rtg11_pos2 rtg11_neg2 rtg11_pos3 rtg11_neg3
+            rtg12_pos1 rtg12_neg1 rtg12_pos2 rtg12_neg2 rtg12_pos3 rtg12_neg3  gnd cpu_85d_asl_5x_4h15_nz_0p72db_r0p5a
+            Length = '5.5*.0254'

* Connector 1******************************************************************
.lib '..\..\models\connector\internal_conn_fpHdr\usb3_fpHdr_rev05_thm_93mils_r0p5a_SD.lib' USB3_int_conn_stubs_14L_93mils
xcon1 rtg12_pos1 rtg12_neg1 rtg12_pos2 rtg12_neg2 rtg12_pos3 rtg12_neg3
+            con1_pos1 con1_neg1 con1_pos2 con1_neg2 con1_pos3 con1_neg3  gnd USB3_int_conn_L12_entry_RxTxTx

* Routing 101******************************************************************
.include '..\..\models\cable\usb3_xxx_90d_xx_awg28_unc_r0p5a_SD.inc'
xrtg101 con1_pos1 con1_neg1 con1_pos2 con1_neg2 con1_pos3 con1_neg3
+            rtg101_pos1 rtg101_neg1 rtg101_pos2 rtg101_neg2 rtg101_pos3 rtg101_neg3  gnd ext_cable_skew_0ps_Z0_typ_SD
+            Length = '10*.0254'

* Connector 2******************************************************************
.include '..\..\models\connector\internal_conn_fpHdr\usb3_fpHdr_rev15_vh_thm_62mils_r0p5a_SD.inc'
xcon2 con2_pos1 con2_neg1 con2_pos2 con2_neg2 con2_pos3 con2_neg3
+            rtg101_pos1 rtg101_neg1 rtg101_pos2 rtg101_neg2 rtg101_pos3 rtg101_neg3  gnd int_usb3rx_usb3tx_usb2

* Routing 20*******************************************************************
.lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg20 con2_pos1 con2_neg1 con2_pos2 con2_neg2 con2_pos3 con2_neg3
+            rtg20_pos1 rtg20_neg1 rtg20_pos2 rtg20_neg2 rtg20_pos3 rtg20_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p72db_r0p5a
```

```
+		Length = '0.2*.0254'

* CAP 1*********************************************************************************
.include '..\..\models\cap\cap_novalue_0402_novoid_r0p5a_SD.inc'
xcap1 rtg20_pos1 rtg20_neg1 rtg20_pos2 rtg20_neg2 rtg20_pos3 rtg20_neg3
+		cap1_pos1 cap1_neg1 cap1_pos2 cap1_neg2 cap1_pos3 cap1_neg3  gnd cap_novalue_0402_novoid_r0p5a

* Routing 21*********************************************************************************
* .lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg21 cap1_pos1 cap1_neg1 cap1_pos2 cap1_neg2 cap1_pos3 cap1_neg3
+		rtg21_pos1 rtg21_neg1 rtg21_pos2 rtg21_neg2 rtg21_pos3 rtg21_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p72db_r0p5a
+		Length = '0.4*.0254'

* CMC 1*********************************************************************************
.include '..\..\models\misc\usb3_common_mode_choke_SD.inc'
xcmc1 rtg21_pos1 rtg21_neg1 rtg21_pos2 rtg21_neg2 rtg21_pos3 rtg21_neg3
+		cmc1_pos1 cmc1_neg1 cmc1_pos2 cmc1_neg2 cmc1_pos3 cmc1_neg3  gnd common_mode_choke_rev1p0_sd

* Routing 22*********************************************************************************
* .lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg22 cmc1_pos1 cmc1_neg1 cmc1_pos2 cmc1_neg2 cmc1_pos3 cmc1_neg3
+		rtg22_pos1 rtg22_neg1 rtg22_pos2 rtg22_neg2 rtg22_pos3 rtg22_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p72db_r0p5a
+		Length = '0.2*.0254'

* ESD 1*********************************************************************************
.include '..\..\models\misc\esd_SD.inc'
xesd1 rtg22_pos1 rtg22_neg1 rtg22_pos2 rtg22_neg2 rtg22_pos3 rtg22_neg3
+		esd1_pos1 esd1_neg1 esd1_pos2 esd1_neg2 esd1_pos3 esd1_neg3  gnd esd_SD

* Routing 23*********************************************************************************
* .lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg23 esd1_pos1 esd1_neg1 esd1_pos2 esd1_neg2 esd1_pos3 esd1_neg3
+		rtg23_pos1 rtg23_neg1 rtg23_pos2 rtg23_neg2 rtg23_pos3 rtg23_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p72db_r0p5a
+		Length = '0.2*.0254'

* Connector 3*********************************************************************************
.include '..\..\models\connector\std_a_conn\usb3_StdA_rev12_thm_TopPort_BtmEntry_r0p5a_SD.inc'
xcon3 rtg23_pos1 rtg23_neg1 rtg23_pos2 rtg23_neg2 rtg23_pos3 rtg23_neg3
+		con3_pos1 con3_neg1 con3_pos2 con3_neg2 con3_pos3 con3_neg3  gnd usb3rx_usb3tx_usb3tx

* Routing 102*********************************************************************************
* .include '..\..\models\cable\usb3_xxx_90d_xx_awg28_unc_r0p5a_SD.inc'
xrtg102 con3_pos1 con3_neg1 con3_pos2 con3_neg2 con3_pos3 con3_neg3
+		rtg102_pos1 rtg102_neg1 rtg102_pos2 rtg102_neg2 rtg102_pos3 rtg102_neg3  gnd ext_cable_skew_0ps_Z0_typ_SD
+		Length = '106*.0254'

* Connector 4*********************************************************************************
.include '..\..\models\connector\std_b_conn\usb3_StdB_rev12_thm_r0p5a_SD.inc'
xcon4 con4_pos1 con4_neg1 con4_pos2 con4_neg2 con4_pos3 con4_neg3
+		rtg102_pos1 rtg102_neg1 rtg102_pos2 rtg102_neg2 rtg102_pos3 rtg102_neg3  gnd stdB_usb3tx_usb3rx_usb3rx

* Routing 31*********************************************************************************
* .lib '..\..\models\routing\cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD.lib' cpu_85d_us_xx_2p7h0p3_xz_xxxdb_r0p5a_SD
xrtg31 con4_pos1 con4_neg1 con4_pos2 con4_neg2 con4_pos3 con4_neg3
+		rtg31_pos1 rtg31_neg1 rtg31_pos2 rtg31_neg2 rtg31_pos3 rtg31_neg3  gnd cpu_85d_us_9x_2p7h0p3_nz_0p88db_r0p5a
+		Length = '1.5*.0254'

* Via 2*********************************************************************************
.lib '..\..\models\via\brd_r0p5a_SD.lib' brd_14L
xvia2 rtg31_pos1 rtg31_neg1 rtg31_pos2 rtg31_neg2 rtg31_pos3 rtg31_neg3
```

```
+              via2_pos1 via2_neg1 via2_pos2 via2_neg2 via2_pos3 via2_neg3  gnd cpu093_14L_L1L14_ent0_ext0_brd_r0p5a_3p

* pkgnew 2*****************************************************************************
.lib '..\..\models\package\usb_device\dev_usb3_r0p5a_SD.lib' pk2_3typ
.param len_pkg2 = '0.5*.0254'
xpkgnew2 pkgnew2_pos1 pkgnew2_neg1 pkgnew2_pos2 pkgnew2_neg2 pkgnew2_pos3 pkgnew2_neg3
+              via2_pos1 via2_neg1 via2_pos2 via2_neg2 via2_pos3 via2_neg3  gnd PKG2_SD

* Rx Model 1*****************************************************************************
P2 pkgnew2_pos1  0 DC=0 AC=0 z0=50 port=2
P4 pkgnew2_neg1  0 DC=0 AC=0 z0=50 port=4
P6 pkgnew2_pos2  0 DC=0 AC=0 z0=50 port=6
P8 pkgnew2_neg2  0 DC=0 AC=0 z0=50 port=8
P10 pkgnew2_pos3  0 DC=0 AC=0 z0=50 port=10
P12 pkgnew2_neg3  0 DC=0 AC=0 z0=50 port=12
.PROBE
+ diff_rx1=par('v(pkgnew2_pos1)- v(pkgnew2_neg1)')
+ diff_rx2=par('v(pkgnew2_pos2)- v(pkgnew2_neg2)')
+ diff_rx3=par('v(pkgnew2_pos3)- v(pkgnew2_neg3)')
.end
```

G OBJECTIVE FUNCTION OF FREQUENCY-DOMAIN NUMERIC OPTIMIZATION IMPLEMENTATION

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Ing. Juan Robledo              INTEL/ITESO                    Jul 23, 2016
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Objective function for simulation deck:
usb3_fp_dc_tx_fd_e2e_6_10_1_input_drv
%
% Usage: function U=usb3_fp_dc_fd_obj_funct(Xofn)
% X = [rtg12L rtg101L rtg21L], Design variables.
% U is eye heigh (EH) and eye width (EW) for optimization.
% Functions required: None.

function U=usb3_fp_dc_fd_obj_funct(X)

% Xp, Pre-assigned parameters.-----------------------------------------------
len_pkg1 = 0.5;       % package routing length (in)
Xp = len_pkg1;

% Ps, Parameters of the simulator.-------------------------------------------
iniFreq = 10e6;       % Initial frequency (Hz).
finalFreq = 10e9;     % Final frequency (Hz).
Ps = [iniFreq, finalFreq];

% Calculate response--------------------------------------------------------
SPthld= [0.161867 0.0813516 0.019056 0.0043013];    % Pass-fail thresholds

[SPsim, iterTime] = hspice_fd_touchstone_driver(X,Xp,Ps);

errorRLh1 = SPsim(1)/SPthld(1)-1;         % S55 error @ 2.5GHz
errorILh1 = 1-SPsim(2)/SPthld(2);     % S65 error @ 2.5GHz
errorILh2 = 1-SPsim(3)/SPthld(3);       % S65 error @ 5GHz
errorILh3 = 1-SPsim(4)/SPthld(4);     % S65 error @ 7.5GHz

e = max([errorRLh1, errorILh1, errorILh2, errorILh3]);
U = e;

% Writing journal file------------------------------------------------------
fid_oj = fopen('fd_optim_journal.csv','a');
fprintf(fid_oj, '%s', [num2str(X(1)) ',' num2str(X(2)) ',' num2str(X(3)) ','
...
    num2str(SPsim(1)) ',' num2str(SPsim(2)) ',' num2str(SPsim(3)) ','
num2str(SPsim(4)) ',' ...
    mat2str(iterTime/60,3) ',' num2str(U)]);
fprintf(fid_oj, '%s\r\n','');
fclose (fid_oj);

end
```

H DRIVER FUNCTION OF FREQUENCY-DOMAIN NUMERIC OPTIMIZATION IMPLEMENTATION

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Ing. Juan Robledo               INTEL/ITESO                    Jul 16, 2016
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% This function drives Hspice netlist for FD analysis,
% and returns S-parameters of discrete frequency points as vector.
%
% Usage: [SPsim, LST] = hspice_fd_touchstone_driver(X,Xp,Ps)
% X = design variables.
% Xp = pre-assigned parameters.
% Ps = parameters of the simulator.
% Functions required: spReader2Driver.

function [SPsim, LST] = hspice_fd_touchstone_driver(X,Xp,Ps)

lti = clock;

% X, Design variables.-----------------------------------------------------
rtg12L = num2str(X(1)) ;        % baseboard routing length (in)
rtg101L = num2str(X(2)) ;       % internal cable length (in)
rtg21L = num2str(X(3)) ;        % daughter card routing length (in)

% Xp, Pre-assigned parameters.---------------------------------------------
len_pkg1 = num2str(Xp(1));      % package routing length (in)

% Ps, Parameters of the simulator.-----------------------------------------
iniFreq = num2str(Ps(1));       % Initial frequency (Hz).
finalFreq = num2str(Ps(2));     % Final frequency (Hz).

% Buiding array of strings that contain netlist info.----------------------
arrStr=spReader2Driver('usb3_fp_dc_tx_fd_e2e_6_10_1_input_drv.sp');

arrStr{12}= ['.AC lin 1000 ' iniFreq ' ' finalFreq ''];
arrStr{26}= ['.param len_pkg1 = ''' len_pkg1 '*.0254'''];
arrStr{45}= ['+ Length = ''' rtg12L '*.0254'''];
arrStr{56}= ['+ Length = ''' rtg101L '*.0254'''];
arrStr{78}= ['+ Length = ''' rtg21L '*.0254'''];

% Saving Hspice netlist in Matlab Working Directory-----------------------
hs_fileName='usb3_fp_dc_fd_iter_drv';

rows= length(arrStr);
fp = fopen(sprintf('%s',hs_fileName,'.sp'),'w+');
for i = 1:rows
    fprintf(fp,'%s', arrStr{i} );
    fprintf(fp,'%s\r\n','');
end
fclose(fp);
```

```
% Executing Hspice in Command Line Mode-----------------------------------
ExecProg= 'C:\synopsys\Hspice_E-2010.12-SP2\BIN\hspice.exe ';
ProjFile= sprintf('%s','-i ',hs_fileName,'.sp -o ',hs_fileName,'.lis -mt 4');
thsi = clock;
system([ExecProg ProjFile]);
thsf = clock;
HST = etime(thsf,thsi);

% Open Touchstone file and extract S-params--------------------------------
sparamCell=spReader2Driver('usb3_fp_dc_fd_iter_drv.s12p');

sparam2p5str=sparamCell{8986,1};
sparam5str=sparamCell{17986,1};
sparam7p5str=sparamCell{26986,1};

SPsim(1)=str2num(sparam2p5str(18:29));      % Mag S55 at 2.5GHz
SPsim(2)=str2num(sparam2p5str(46:57));      % Mag S65 at 2.5GHz
SPsim(3)=str2num(sparam5str(46:57));        % Mag S65 at 5GHz
SPsim(4)=str2num(sparam7p5str(46:57));      % Mag S65 at 7.5GHz

% Delete Unwanted Output Data Files----------------------------------------
delete (sprintf('%s',hs_fileName,'.ic0'));
delete (sprintf('%s',hs_fileName,'.pa0'));
delete (sprintf('%s',hs_fileName,'.st0'));
delete (sprintf('%s',hs_fileName,'.ac0'));
delete (sprintf('%s',hs_fileName,'.lis'));

% Reporting Simulation Times-----------------------------------------------
ltf = clock;
LST = etime(ltf,lti);
% disp(['Hspice simulation time = ' mat2str(HST/60,5) ' minutes']);
% disp(['Loop simulation time = ' mat2str(LST/60,5) ' minutes']);

end
```

I FREQUENCY-DOMAIN OPTIMIZATION ALGORITHM

```
TSTi= clock;
fid_fms = fopen('fd_optim_journal.csv','w');
fprintf(fid_fms,'%s', ['rtg12L(in), rtg101L(in), rtg21L(in),' ...
    'S55 @ 2.5GHz, S65 @ 2.5GHz, S65 @ 5GHz, S65 @ 7.5GHz,' ...
    'LoopSimTime(min), U(max error)']);
fprintf(fid_fms,'%s\r\n','');
fclose(fid_fms);

% X = [rtg12L rtg101L rtg21L], Design variables.--------------------------
rtg12L = 5.25;          % baseboard routing length (in)
rtg101L = 10;           % internal cable length (in)
rtg21L = 0.4;           % daughter card routing length (in)
x0 = [rtg12L, rtg101L, rtg21L];

% Call to optimization algorithm-----------------------------------------
MaxIter=50;
TolFun=1e-2;
TolX=1e-2;
%
options=optimset('Display','final','MaxIter',MaxIter,'PlotFcns',@optimplotfva
l,'TolX',TolX,'TolFun',TolFun);
options=optimset('Display','final','MaxIter',MaxIter,'TolX',TolX,'TolFun',Tol
Fun);
[x_opt,feval,exitflag,output]=fminsearch('usb3_fp_dc_fd_obj_funct', x0,
options)

TSTf = clock;
TST = etime(TSTf,TSTi);
disp(['Total simulation time = ' mat2str(TST/60,3) ' minutes or '
mat2str(TST/3600,5) ' hours']);

%exitflag---------------------------------------------------------------
%1  fminsearch converged to a solution x.
%0  Maximum number of function evaluations or iterations was reached.
%-1 Algorithm was terminated by the output function
```

J INTERNAL RESEARCH REPORT LIST

[1] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "An introduction to signal integrity simulations for high speed interconnects in computing platforms," Internal Report *CAECAS-14-10-R*, ITESO, Tlaquepaque, Mexico, Dec. 2014.

[2] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "An illustrative signal integrity simulation example of a non-spec compliant high speed digital interconnect," Internal Report *CAECAS-15-07-R*, ITESO, Tlaquepaque, Mexico, Jun. 2015.

[3] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "A minimax formulation for a time domain numeric optimization of a non-spec-compliant USB3 topology," Internal Report *CAECAS-15-15-R*, ITESO, Tlaquepaque, Mexico, Dec. 2015.

[4] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "Frequency domain simulations of high speed digital interconnects and comparison with classical time domain analysis," Internal Report *CAECAS-16-02-R*, ITESO, Tlaquepaque, Mexico, Jan. 2016.

[5] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "A minimax formulation for a frequency domain numeric optimization of a non-spec-compliant USB3 topology," Internal Report *CAECAS-16-07-R*, ITESO, Tlaquepaque, Mexico, Aug. 2016.

References

[Casper-02] B. K. Casper, M. Haycock, and R. Mooney, "An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes," in *Symp. VLSI Circuit Digest of Technical Papers*, Honolulu, HI, Jun. 2002, pp. 54-57.

[Cuny-96] R. H. G. Cuny, "SPICE and IBIS modeling kits the basis for signal integrity analyses," in *IEEE Int. Symp. Electromagnetic Compatibility (EMC)*, Santa Clara, CA, Aug. 1996, pp. 204-208.

[El-10] A. E. El-Rouby, "A new frequency domain based methodology for signal integrity analysis," in *IEEE Int. Conference on Computer Engineering and Systems (ICCES)*, Cairo, Egypt, Dec. 2010, pp. 342-345.

[Fan-10] J. Fan, X. Ye, J. Kim, B. Archambeault, and A. Orlandi, "Signal integrity design for high-speed digital circuits: progress and directions," *IEEE Trans. Electromagnetic Compatibility (EMC)*, vol. 52, issue 2, pp. 392-400, 2010.

[Hall-00] S. H. Hall, G. W. Hall and J.A McCall, *High-Speed Digital System Design: A Handbook of Interconnect Theory and Design Practices*. New York, NY: John Willey and Sons, 2000.

[Hall-09] S. H. Hall and H. L. Heck, *Advanced Signal Integrity for High-Speed Digital Design*. Hoboken, NJ: John Willey and Sons, 2009.

[Hspice-10] HSPICE user guide: simulation and analysis, Version E-2010.12, December 2010, [Online] Available: https://www.synopsys.com/Tools/Verification/AMSVerification/CircuitSimulation/HSPICE/Pages/WhatNew.aspx?cmp=HSPICE09-HSPICE-HL

[Ibis-09] Touchstone file format Specification, Version 2.0, April 2009, [Online] Available: https://ibis.org/touchstone_ver2.0/touchstone_ver2_0.pdf

[Kim-09] D. Kim and Y. Eo, "S-parameter-measurement-based time-domain signal transient and crosstalk noise characterizations of coupled transmission lines," *IEEE Trans. on Advanced Packaging*, vol. 32, pp. 152-163, Feb. 2009.

[Li-11] Y. L. Li, X. Ye, K. Xiao, T. Su, and K. Kang, "Channel quality comparison for OOG IO bus designs," in *12th Int. Conf. Electronic Packaging Technology and High Density Packaging (ICEPT-HDP)*, Shanghai, Aug. 2011, pp. 1-3.

[Li-12] Y. L. Li, et al, "Board-level signal integrity methodology," in *7th Int. Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, Taipei, Oct. 2012, pp. 217-219.

[Matlab-15] Matlab fminsearch function (2015), [Online] Available: http://www.mathworks.com/help/matlab/ref/fminsearch.html

[Rayas-11] J. E. Rayas-Sánchez (2011), *Methods for Constrained Optimization*. [Online] Available: http://desi.iteso.mx/erayas/documents/CirOpt_course/Lectures/Optimization/Opt_Constrained_methods.pdf

[Rayas-14] J. E. Rayas-Sánchez, "Breve revisión histórica del diseño asistido por computadora de circuitos y dispositivos electrónicos, y algunas tendencias futuras," ITESO, Tlaquepaque, Mexico, 2014.

[Robledo-14] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "An introduction to signal integrity simulations for high speed interconnects in computing platforms," Internal Report *CAECAS-14-10-R*, ITESO, Tlaquepaque, Mexico, Dec. 2014.

[Robledo-15a] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "An illustrative signal integrity simulation example of a non-spec compliant high speed digital interconnect," Internal Report *CAECAS-15-07-R*, ITESO, Tlaquepaque, Mexico, Jun. 2015.

[Robledo-15b] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "A minimax formulation for a time domain numeric optimization of a non-spec-compliant USB3 topology," Internal Report *CAECAS-15-15-R*, ITESO, Tlaquepaque, Mexico, Dec. 2015.

[Robledo-16] J. A. Robledo-Mariscal and J. E. Rayas-Sánchez, "Frequency domain simulations of high speed digital interconnects and comparison with classical time domain analysis," Internal Report *CAECAS-16-02-R*, ITESO, Tlaquepaque, Mexico, Jan. 2016.

[USB-09] USB 3.0 Electrical Compliance Methodology White Paper Rev 0.5 (2009) [Online] Available: http://www.usb.org/

[USB-11] USB 3.0 Specification Rev 1.0 (2011), ch. 6.7, [online] Available: http://www.usb.org/

[Win-15] S. T. Win *et al*, "A frequency-domain high-speed bus signal integrity compliance model: Design methodology and implementation," in *IEEE Electronic Components and Technology Conference (ECTC)*, San Diego, CA, May 2015, pp.545-550.

[Xiao-08] K. Xiao, B. Lee, and X. Ye, "A flexible and efficient bit error rate simulation method for high-speed differential link analysis using time-domain interpolation and superposition," in *IEEE Int. Symp. Electromagnetic Compatibility (EMC)*, Detroit, MI, Aug. 2008, pp. 1-6.

[Ye-09] X. Ye, K. Xiao, and C. Huang, "Building accurate SPICE models for multi-Gbps interconnect in computer systems," in *IEEE Int. Symp. Electromagnetic Compatibility (EMC)*, Austin, TX, Aug. 2009, pp. 290-295.

[Ye-12] C. Ye, K. Xiao, M. Johnston, R. Cuadras, and X. Ye, "Evaluating high-speed link performance using direct end-to-end measurements," in *IEEE Int. Symp. Electromagnetic Compatibility (EMC)*, Pittsburgh, PA, Aug. 2012, pp. 183-188.

Index

A

aggressor, 12, 13, 15

B

baseboard, 5, 10, 20, 21, 22, 26, 28, 34, 35, 50, 51, 53, 60, 62
BER, 6, 8, 15, 16, 17, 23, 33, 38, 41
bit rate, 30
bit time, 16

C

cable, 10, 11, 13, 20, 21, 22, 26, 28, 34, 35, 47, 48, 50, 51, 53, 56, 57, 60, 62
CAD, 3, 27
compliance test, 6, 7, 9, 26
compliance testing, 17, 20, 21, 26, 28, 35, 37, 39, 42
computational cost, 16, 17, 20, 27, 28, 32
computing host, 10, 13, 20
connector, 4, 5, 11, 13, 21, 47, 48, 56, 57
convolution, 6, 15, 41
crosstalk, 6, 11, 12, 14, 15, 29, 65

D

daughter card, 10, 20, 21, 22, 26, 28, 34, 35, 47, 50, 51, 53, 60, 62
deck, 13, 14, 17, 21, 22, 23, 26, 47, 50, 59
design parameters, 3, 19, 20, 23, 24, 39
design variables, 8, 19, 21, 22, 23, 25, 26, 34, 36, 37, 38, 50, 51, 60
device, 5, 10, 11, 14, 20, 47, 49, 56, 58
differential pair, 12, 14
driver function, 19, 23, 25, 26, 36, 42, 51, 60

E

electric models, 4, 5, 11
eye diagram, 6, 37, 38
eye height, 6, 16, 22, 25, 37
eye width, 6, 7, 16, 22, 25, 37, 50, 51, 59

F

`fminsearch`, 24, 37, 53, 62, 65
form factors, 1, 19
frequency domain, i, v, 27, 33, 35, 56, 63, 65
front panel, 9, 10, 11, 14, 17, 19, 20, 26, 28, 34, 42, 47

H

harmonic, 30, 35
HSPICE, 6, 14, 16, 23, 27, 29, 30, 32, 33, 34, 36, 42, 65
hub, 10, 11, 13, 20, 21

I

insertion loss, 29, 30, 31, 32, 35, 42
ISI, 6

J

jitter, 6, 8, 15, 23, 33, 41

L

layout, 3, 5

M

margins, 6, 8, 9, 10, 15, 17, 20, 41, 42
microstrip, 5
minimax, 19, 21, 23, 26, 28, 34, 35, 36, 39, 63, 66

N

netlist, 14, 17, 23, 30, 34, 36, 51, 56, 60
non-spec compliant, v, 8, 9, 17, 19, 26, 28, 32, 34, 37, 42, 63, 66
Nyquist frequency, 30, 31, 35

O

objective function, 23, 24, 25, 26, 36, 37, 39, 42, 50, 59

optimal design, 19, 24, 25, 26, 37, 39, 42
optimization, i, v, 3, 17, 19, 20, 21, 22, 23, 24, 26, 27, 28, 33, 34, 35, 36, 37, 39, 42, 50, 51, 53, 54, 59, 60, 62, 63, 66
Optimization algorithm, 25, 37
output parameters, 22

P

passing thresholds, 21, 23
PCB, 1, 3, 5, 10, 12, 13, 27
PDG, 7, 21
PKG, 10, 11, 12
post-processing, 3, 10, 15, 16, 23, 33
pre-assigned parameters, 21, 34, 36, 51, 60
processor, 3

R

Receiver, 14
return loss, 29, 30, 31, 32, 35, 42
RLGC, 4, 12
routing sections, 4

S

seed, 24, 25, 26, 37, 38
signal integrity, v, 1, 2, 3, 9, 19, 33, 41, 63, 65, 66
simulation time, v, 1, 6, 20, 21, 26, 32, 33, 34, 42, 52, 53, 61, 62
simulator parameters, 21, 34
S-parameter, 27, 32, 42, 65
S-parameters, 5, 13, 27, 29, 30, 31, 32, 33, 34, 35, 36, 39, 42, 60
spec-compliant, 16, 19, 20, 21, 22, 23, 26, 27, 28, 30, 32, 34, 41, 42, 56, 63, 66
specification, v, 6, 7, 8, 9, 16, 17, 19, 20, 37
SPICE, 5, 8, 10, 11, 14, 17, 27, 29, 33, 41, 65, 66
Std A, 11, 13, 21
Std B, 11, 13
step response, 6, 12, 14, 15, 23, 41
stripline, 5, 11
superposition, 6, 15, 41, 66

T

termination criteria, 19, 24, 34, 37
time domain, v, 1, 3, 4, 6, 8, 15, 23, 27, 41, 47, 63, 66
topology, v, 4, 5, 7, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 22, 26, 27, 28, 29, 30, 31, 32, 33, 34, 37, 39, 42, 47, 56, 63, 66
Touchstone, 13, 29, 30, 33, 36, 61, 65
transfer function, 4, 11, 29, 33
transient, 1, 6, 8, 10, 14, 15, 17, 23, 27, 28, 32, 33, 41, 65
transmission line, 4, 12, 27
transmission lines, 4, 12, 14, 65
Transmitter, 11

U

UI, 16, 20, 21, 26, 37, 38, 50, 53, 54, 55
unconstrained minimax, 23, 36
USB, 4, 5, 6, 7, 9, 10, 14, 16, 17, 19, 20, 21, 26, 66

V

victims, 12, 13
vswing, 11, 22, 50, 51

W

waveforms, 6, 10, 14, 15, 17, 41