An Automated Tool for Optimizing Margins,
Using Distributed References,
In a High-Speed Signaling Environment

Tudor Ion Secasiu
M.S., Electrical Engineering. University Politehnica Bucharest (1990)

A dissertation presented to the faculty of the
OGI School of Science & Engineering
at Oregon Health and Science University
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Electrical and Computer Engineering
September, 2004

The dissertation "An Automated Tool for Optimizing Margins, Using Distributed References, in a High-Speed Signaling Environment" by Tudor Ion Secasiu has been examined and approved by the following examination committee:

Professor Rajendra Solanki, Thesis Advisor
OGI School of Science and Engineering

Professor Shankar Ranavare
OGI School of Science and Engineering

Professor John D. Lynch
OGI School of Science and Engineering

Ajay V. Bhatt
Chief Architect, Platform Architecture and Solutions Division,
Intel Corporation

Jeff C. Morriss
Principal Engineer, Platform Architecture and Solutions Division, Intel Corporation

Table of Content

List of Figures

List of Tables

Abstract

An Automated Tool for Optimizing Margins,
Using Distributed References,
In a High-Speed Signaling Environment

Tudor Ion Secasiu

Ph.D., OGI School of Science & Engineering
at Oregon Health & Science University
September 2004

Thesis Advisor: Dr. Rajendra Solanki

Ever increasing time-to-market pressures make systemic high-speed design approaches a must, even in early design stages. However, such approaches are difficult due to lack of integrated toolsets capable of meeting both I/O and interconnect needs. Additionally, electronic design automation tools begin to have theoretical and practical limits, mainly due to approximation methods used to solve Maxwell's equations.

Gigahertz-speed data transfer rates also translate into tight integration of silicon and interconnects. This increased interdependency makes the traditional approach - designing systems components in isolation - obsolete.

To compensate for higher interconnect losses, additional capabilities, such as filtering, have to be implemented in silicon. Due to system dependencies such capabilities can only be characterized using distributed loads closely matched to operational conditions. Consequentially, realistic (idealized) distributed loads have to be used as a design reference, replacing traditional lumped elements in I/O characterization. Such

loads can also be used to increase early design integration and to bridge the existing tool gap in silicon and interconnect design capabilities.

Examples of driver and receiver optimizations from a systemic point of view based on readouts from a software enhancement, the "Eye Diagram Analyzer" (EDA) are described. The software enhancement is written in C-code and offers the advantage of providing an accurate numerical analysis and interpretation of Eye Diagram data from time domain simulations or measurements, which can be used for real-time (statistical) data analysis. The software has been written as a series of functions making it attachable through function calls to a variety of traditional high speed design tools.

As results a comprehensive methodology to characterize digital transmission paths has been achieved. The methodology has the advantage that is tool-independent, simple, and accurate up to users needs. It can be used to develop, using mathematical function, realistic transmission lines based on frequency domain measurements. It can also be used to characterize I/O enhancements such as equalization, based on real time system margining as well as incorporate I/O and interconnects into a complete system.

Chapter 1
Introduction

1.1 Motivation and Problem statement

Designing and validating high speed digital data links used to be divided into three different parts: driver, interconnect and receiver. Specifications and measurements have been usually done into lumped loads, usually 50 Ohm resistors. However recent dramatic increase in transfer rates has tightened design and validation margins and increased interactions between system components. Therefore it is increasingly difficult, or even impossible in some cases, to design and validate components in isolation.

The other emerging issue is related to design and validation tool. Because driver/receivers and interconnects have been characterized in isolation, as well as different design needs, different toolsets have been developed, with a different range of capabilities. Exchanging information between different toolsets can be a painful or even impossible task, creating difficulties when "die-to-die" simulations are desired; therefore having mathematical, tool independent models can mean a strong productivity boost, or even a pass/fail criteria.

Based on this fact new methods are created, such as the Channel-Based methodology [1], which simplifies driver and receiver designs based on performance into system loads. The Channel Based methodology asserts that a driver and receiver will be characterized into a quasi-ideal transmission line with certain characteristics. Interconnect characteristics are also compared with a reference line.

In order to use such techniques, accurate transmission line models that can be easily characterized, used and shared are required. Such models also have to be flexible, in order to match driver and receiver performance requirements in terms of patterns, number of points, maximum frequency etc. This is usually done in existing tools with interpolation and/or extrapolations, which can add errors to the model.

"Secondary" transmission line effects like dispersion, surface roughness, frequency dependent loss and crosstalk also become more significant and can affect the correct operation of the system, adding more complexity requirements to models. Consequently, conventional linear models are rapidly becoming inadequate in describing transmission line behavior.

Unfortunately, commercial simulators do not always keep-up with design needs. Transmission line models in many electrical simulation tools generally assume some primitive approximations of transmission lines that do not account for the "secondary" effects mentioned before as well as other system effects such as ground conductivity, and non-TEM modes etc. Because of this fact it is becoming increasingly important to correlate every model used with accurate measurements.

Real time measurement and evaluation instruments have to be able to sample data at much higher rates, therefore instruments capable of accurate time domain measurements are either non-existent or prohibitively expensive. Because of this measurements are performed mostly in frequency domain, using accurate Vector Network Analyzers.

It is therefore necessary to be able to transfer with ease between frequency and time domain, in order to be able to correlate simulated and measured behavior.

This investigation describes a methodology that is able to create flexible, transmission line models, based on measured data, with all the flexibility described before, as well as maintaining initial (measurements) accuracy. The models created are used to characterize drivers and receivers, as well as set initial I/O target parameters.

Independent of how we perform system characterization, eventually a receiver response has to be evaluated in time domain, so in the end a transient simulation is absolutely necessary. The most accurate way (since it has an exact mathematical equation) to transfer data from frequency domain to time domain is to use the Fourier transformation. Fourier transformations are dependent on the granularity (number of points) of the initial data, so all models have to take this aspect into account.

One of the most comprehensive ways to provide and analyze time domain results is using Eye Diagrams. Although Eye Diagrams are highly intuitive and complete, few electrical simulation packages provide options to plot and analyze them. Among those which offer plotting capabilities, very few are offering accurate mathematical ways to extract useful data, such as voltage and time openings and cycle-by-cycle jitter.

This thesis provides a methodology and a tool that allows accurate, "real time" Eye Diagram evaluations, as well as practical correlations between time and frequency domain data. While this thesis uses mainly two simulation packages, MATLAB from Mathworks and Advanced Design Simulator (ADS) from Agilent Technologies, the "Eye Diagram Analyzer" (EDA), is written in C language, therefore it can be used independently as well as implemented or used in conjunction with other tools.

1.2 Overview of the work

This thesis comprises seven chapters of which Chapter 1 is the introduction

Chapter 2 provides background information which starts with a brief overview of the limitations of the higher data links in the context of the proposed methodology.

Chapter 3 is addressing some of the above issues by providing a software package that is capable of extracting useful information from the simulated (or measured) time domain data in both "cycle-to-cycle" or "Eye Diagram" style.

Since Eye Diagrams are a very useful tool in interpreting time domain data, simulation tool vendors provide some Eye Diagram plotting capabilities, but very few of them, if any, are offering accurate "real-time" mathematical ways to extract useful data.

The "Eye Diagram Analyzer" (EDA) tool, that will be used to measure time and voltage openings at the receiver input, is a stand-alone software package that was developed and used in this investigation as an ADS "add-in" function, but it can also be used with minor modifications in any simulation package capable of understanding the C-language. Such package is provided for usage in ADS and MATLAB.

Chapter 4 describes the proposed methodology to characterize transmission lines by approximating measured scattering-parameters with mathematical functions, which are much easier to manipulate and are tool independent, hence they can be shared.

Chapter 5 and Chapter 6 describe methods to achieve an optimal driver and receiver performance using the Eye Diagram Analyzer tool and the simple interconnect models obtained in Chapter 4.

Characterization is performed in time domain using the s-parameter approximations previously described and a combination of statistical and optimization analysis. Various effects such as Inter Symbol Interference (ISI) and Duty Cycle Distortion (DCD) are quantified at receiver input using the “Eye Diagram Analyzer” (EDA).

Chapter 2
Background

2.1 Limitation for inter-chip signaling

The primary goal when designing a high speed circuit is to transmit data between system components with minimum cost per bandwidth. Besides absolute theoretical limitations described in [2] , there are additional noise sources that are interfering with the signals. In a realistic system noise sources can be divided into 3 major components:

a. Driver induced noise such as jitter, power supply noise, and improper driver termination

b. Interconnect induced noise such as reflections, inter-symbol interference, cross-talk, dispersion and frequency dependent loss and

c. Receiver induced noise, such as receiver termination mismatch and receiver sampling uncertainty.

Some of the above type of noises, such as crosstalk, are very well known and analyzed. For example crosstalk is dependent of the second power of distance, so the most effective way to reduce it is to increase spacing between victims and offenders.

Although all of the above noises can be identified using EDA, for practical considerations, only some of them are addressed in this study, mainly the ones that exercise different behaviors if characterized into distributed loads.

2.2 Interconnect noise

2.2.1 Inter-symbol interference (ISI)

ISI occurs within a serial bit stream as a result of pulse dispersion and consequential overlapping pulse edges, leading possibly to decoding errors at the receiver. ISI is actually the limiting factor within the Nyquist criteria. In a digital transmission system, distortion of the signal manifests in the temporal spreading and consequently overlapping individual pulses to the degree that the receiver can not reliably distinguish between individual signal elements. At a certain threshold, ISI interference will permanently compromise the integrity of the received data. The best way to measure ISI is using Eye Diagrams. Figure 2-1 shows ISI for an ideal pulse response. Theoretically the resultant function (cosine) spreads to infinity, however, for practical reasons the energy remains concentrated around the original bit.

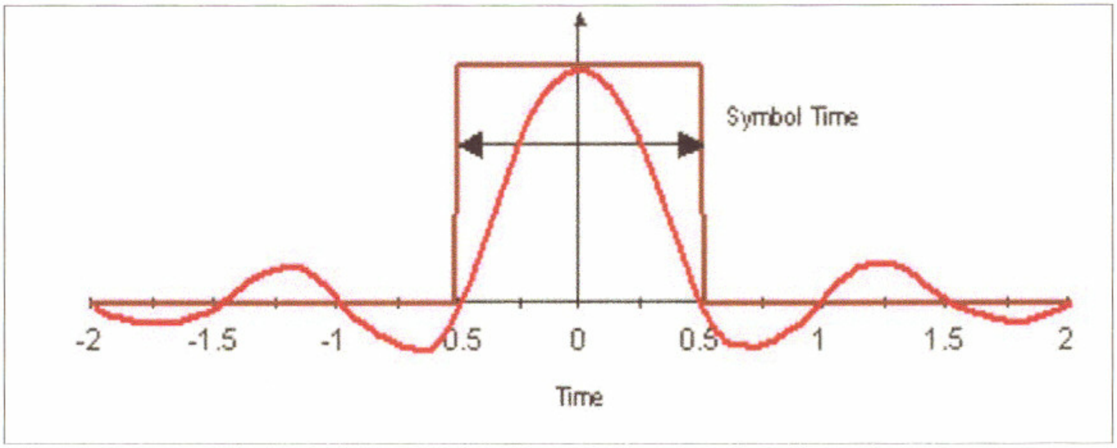


Figure 2-1 Inter-symbol interference – pulse dispersion

The only efficient way to counterbalance ISI is by filtering or compensating the resulting signal. Compensating the signal can be made at the driver, receiver or both. For the purpose of explaining the EDA tool, this investigation will emphasize driver side filtering (equalization).

Driver side equalization can be theoretically achieved by applying energy to the adjacent bits, so that it counterbalances the ISI spreading. Equalization is most effective when the total resulting energy of the side lobes is zero. Figure 2-2 is showing the block diagram of a nonrecursive FIR filter [3] similar to what will be implemented in consequent chapters.

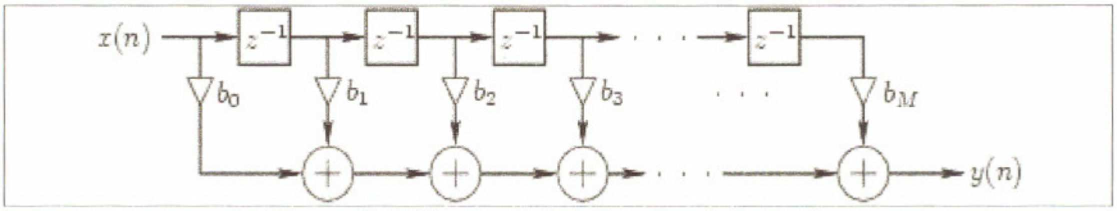


Figure 2-2 Nonrecursive filter implementation

We can see from Figure 2-1 and Figure 2-2 that determining the optimal TAP numbers (n) and TAP coefficients (bn) are critical to the nonrecursive filter.

As previously mentioned, theoretically, the nonrecursive filter has to be extended to infinity; however, the effectiveness of such filter decays rapidly. The EDA tool can be used to determine the optimal nonrecursive filter TAP numbers and corresponding coefficients.

2.2.2 Reflection

To avoid reflections in a transmission line environment, signal lines need to be terminated in the characteristic impedance of the transmission line, on either the transmitter or the receiver end of the line.

The termination impedance absorbs the transmitted signal energy and prevents it from being reflected back into the transmission medium.

Mismatches between termination and line impedance create reflected waves on the transmission line. The reflection coefficient [4] is given by:

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0}$$

Equation 2-1

Where:

Γ is the reflection coefficient

Z_0 is the line characteristic impedance

Z_L is the load characteristic impedance

Reflected waves add to the subsequent signals, thus resulting in a different form of inter-symbol interference. It should be mentioned that at lower frequencies, the line characteristic impedance is predominantly resistive; therefore the line can be properly terminated at all frequencies. However, at high frequencies, characteristic impedance is frequency dependent, therefore this line can not be perfectly matched at all frequencies.

Equation 2-1 is also valid for any electrical discontinuity within the system such as connectors, transmission line segments, packages, bond-wires, etc. Such components can create additional inductive or capacitive discontinuities, which again, degrade the signal quality by generating reflections.

Analyzing such components is critical and most of the time requires accurate field analysis using specialized field solvers.

2.2.3 Return path discontinuities

Most simulation packages that have transmission line models included assume ideal analog ground return paths. However, this is not always true. Connectors, packages, ground planes and others are introducing discontinuities that are going to influence signal integrity. As mentioned above, those are usually ignored in traditional simulators due to the mathematical difficulties they present. Unfortunately, at this time only 3D field solvers can handle return path discontinuities. Although the methodology presented here does not directly address this issue, it provided a way to overcome some of the above limitations, by using measurements (which can include return path discontinuities) as a basis for transmission line models. It is also trying to illustrate the importance of correlations between simulation models and measurements.

2.3 Driver noise

2.3.1 Jitter

Jitter is defined as a deviation from the ideal timing of an event. The reference event is usually the zero crossing for electrical systems [5]. Total jitter (TJ) includes (Figure 2-3) deterministic jitter (DJ) and random jitter (RJ).

Random jitter (RJ) is characterized by a Gaussian distribution and assumed to be unbounded. RJ can come from thermal vibrations of semiconductor crystal structures, material boundaries having less than perfect valence electron mapping due to semi-regular doping density and process anomalies, thermal vibrations of conductor atoms, and many minor contributors (e. g. Cosmic radiation, etc.). As a result, it generally affects long-term device stability. Because RJ is Gaussian in nature, the distribution is quantified by standard deviation (σ) and mean (μ). It will not be considered in this study.

Deterministic Jitter is jitter with a non-Gaussian probability density function and is characterized by a bounded peak-peak value that does not increase samples size. It is typically caused by cross talk, EMI, simultaneous switching outputs (SSO), device function dependency (pattern dependant jitter) and other regularly occurring interference signals.

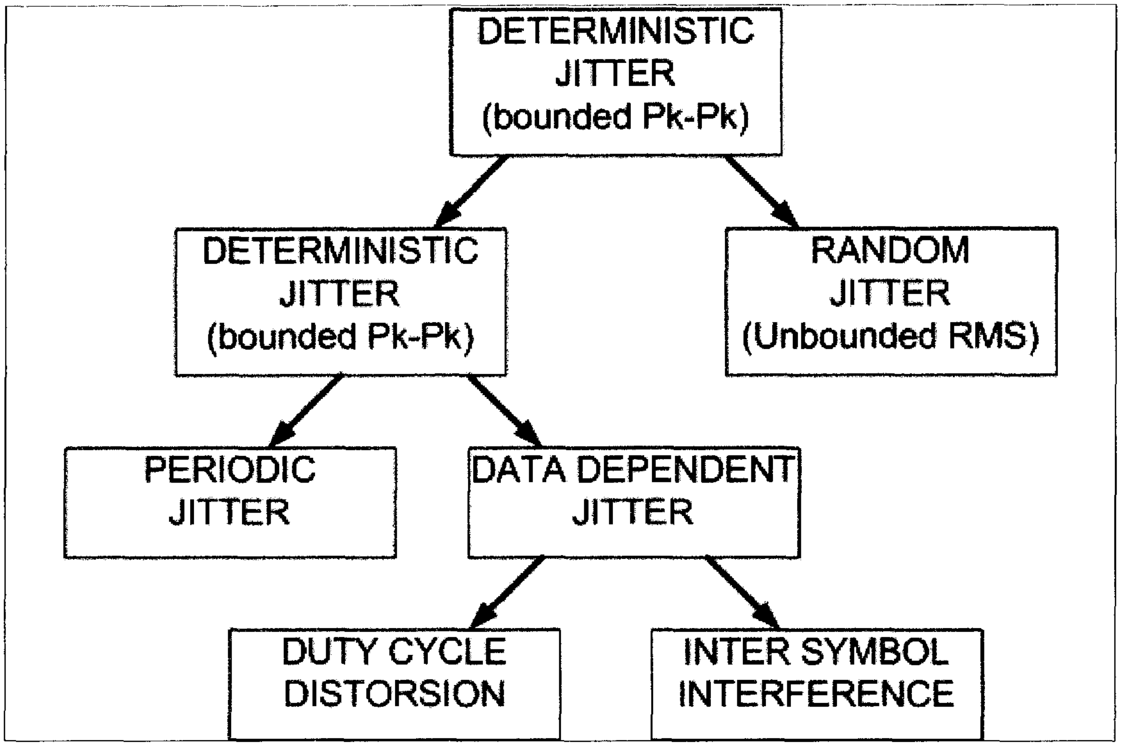

DETERMINISTIC JITTER (bounded Pk-Pk)
DETERMINISTIC JITTER (bounded Pk-Pk)
RANDOM JITTER (Unbounded RMS)
PERIODIC JITTER
DATA DEPENDENT JITTER
DUTY CYCLE DISTORSION
INTER SYMBOL INTERFERENCE

Figure 2-3 Jitter classification

DJ can be further separated into Periodic Jitter (PJ) and Data Dependent Jitter (DDJ). Furthermore Data Dependant Jitter can be also separated into Duty Cycle Distortion (DCD) and Inter Symbol Interference (ISI).

Periodic jitter (PJ), also referred to as sinusoidal jitter, has a signature that repeats at a fixed frequency. For example, PJ could be the result of unwanted modulation, such as electromagnetic interference (EMI) or crosstalk from adjacent lines. PJ is quantified as a peak-to-peak number, specified with a frequency and magnitude. This investigation will try to quantify (using the Eye Diagram Tool) the data dependent jitter.

Data dependent jitter occurs when the transmission pattern is changed from a clock-like to non-clock-like pattern. It includes ISI and DCD. The latter can be caused by non-linearity in the clock distribution, but can also be caused by the driver itself, and by other effects such as the Equalization scheme employed.

Duty Cycle Distortion is a short and medium time occurring jitter and it is known to cause a phenomenon called jitter amplification. This is the case when the driver induced jitter couples with interconnect, and suffers a time and voltage reduction. This type of jitter has to be included in the driver model in order to determine the amplification through the system. Time and voltage amplification will be determined using EDA.

2.3.2 Total driver impedance

Total driver impedance is important because of the mismatch introduced in the system as well as its contribution to ISI. The major contributor at high frequencies is the driver capacitance, because it is introducing frequency dependent impedance. Trying to match interconnect impedance, which is mostly frequency independent with driver impedance, over a large frequency range, becomes theoretically impossible.

2.4 Receiver noise

2.4.1 Receiver tracking bandwidth

One of the most important receiver characteristics must be the ability to track and filter the noise induced by the driver and interconnect. From this point of view there are two types of noise, low and high frequency dependent noises.

Tracking low frequency noise is dependent on the receiver's bandwidth and can be achieved with regular sampling techniques, whereas tracking high frequency noise is possible by using over sampling techniques.

For both types of noise it is important to study the noise distribution, so proper sampling techniques can be applied.

2.4.2 Total receiver impedance

Same considerations used for the driver can also be applied to the receiver.

2.5 Eye Diagram overview

In digital communications the "Eye Diagram" is used to visualize how the waveforms used to send multiple bits of data can potentially lead to errors in the interpretation of those bits. Conceptually, an Eye Diagram is created by “chopping” waveforms at regular intervals related to the bit time (symbol period). Each “chopped” segment is then aligned to a common timing reference and overlaid with previous segments, so that the whole waveform can be viewed in one “collapsed” time interval, usually one period.

The vertical thickness of the "collapsed line bundle" in an Eye Diagram indicates the magnitude of AC voltage noise, whereas the horizontal thickness (determined by the cross-over point) is an indication of the AC timing noise, (variance in the actual transition time from the ideal transition time) also known as jitter.

In addition to AC time and voltage noise, an Eye Diagram also produces information on the voltage swing, rise and the fall time of a signal.

The size of the eye opening indicates the amount of voltage and timing margin available to sample this signal. For a particular electrical interface, a fixed reticule called "Eye Mask" could be placed over the Eye Diagram showing how the actual signal compares to a minimum criterion of time and voltage. Depending on the interface the Eye Mask could have various shapes, the most popular being rectangular, rhombic or a combination of the above.

Figure 2-4 shows the results of the described process, as applied to the original transient data displayed in Figure 2-5. As we can see, by comparison, the Eye Diagram provides a better visual indication of the voltage and timing uncertainty associated with the signal.

For example we can have a good visual indication of the total jitter associated with the signal (labeled "Jitter" in Figure 2-4), as well at the available time and voltage margins ("Eye Opening") based on certain receiver sensitivity (V_{rec}).

Other information that can be visually extracted from the plot is the amount of overshoot (V_{over}), signal distortion or loss (V_{dist}), as well as sensitivity to timing errors based on signal rise and fall times.

Eye Diagrams have been traditionally used in optical communications, however due to the advantages they present, have been increasingly used in the more conventional "copper" data transmissions.

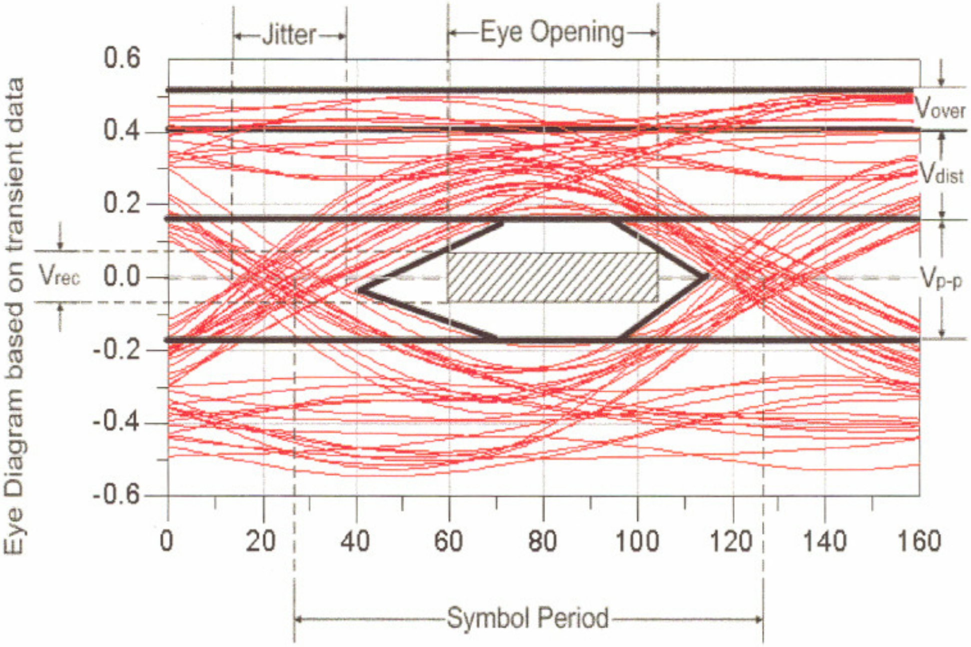


Figure 2-4 Eye Diagram based on transient V(t) data

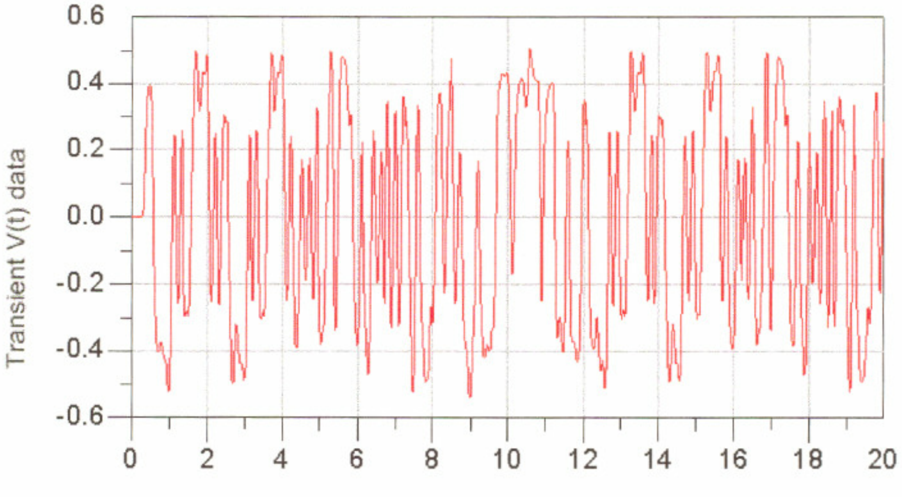


Figure 2-5 Time domain data displayed in regular V(t) format

Chapter 3
Eye Diagram Analyzer

3.1 Background

Although Eye Diagrams are one of the most comprehensive way to interpret time domain data, little effort has been put into improving this capability. One good example is that, to date, only a few real time oscilloscopes can display Eye Diagrams. Similarly, industry standard simulation tools have followed the same pattern and do not provide adequate data processing for Eye Diagram formats. Because of this, users have to create custom scripts in order to be able to evaluate basic parameters such as Eye Diagram openings, or dependent and independent variable statistics.

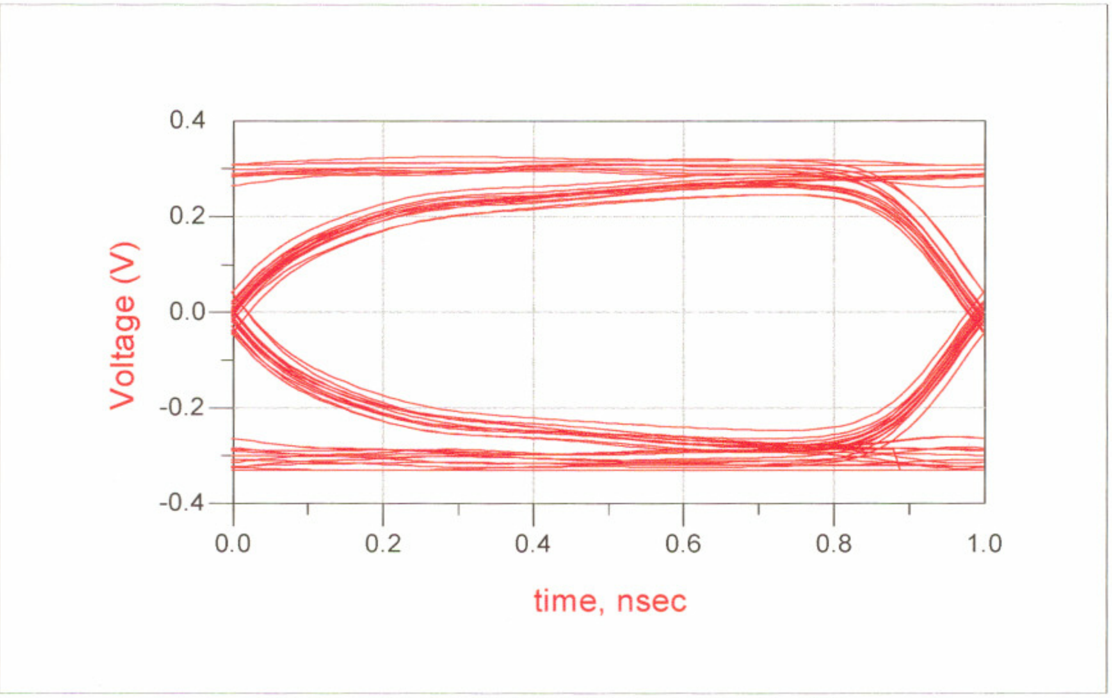


Figure-3-1 Standard Eye Diagram plot in ADS

As we can see from Figure-3-1 the Eye Diagram is providing only visual information, and makes it very difficult to determine accurate numbers from the plot. Moreover, if a large number of simulations are performed, it is almost impossible to compare results or establish trends. The other major limitation is the fact that the Eye Diagram plot is mostly a post processing tool, and can not be used as a performance criteria in any statistical or sweep analysis. A good example is trying to maximize the voltage opening at any eye point. For this, usually a parameter or multi-parameter sweep is performed and end results are visually compared. This, again, is a tedious process and subject to inaccuracies and interpretation.

The tool developed and described in this work can be used to automate the process, by being able to exactly read the useful information and being able to feed-back the results for the statistical (or sweep) analysis, so that the simulation will automatically stop when the desired criteria is met.

3.2 General description

As mentioned before, the code for the Eye Diagram Analyzer is written is C with some AEL[1] adaptations to match the ADS data format. It can also be imported into MATLAB m-file or executable (.exe) format using the proper compiler.

It will take several input parameters and it will output either absolute time or voltage at any point in the Eye Diagram, or time and voltage margins based on the difference between absolute numbers and an arbitrary hexagonal-type mask. Multiple readings can be made from the same Eye Diagram, this being equivalent of multiple mask overlay.

If simulation time points do not match with actual minimum values, it will perform a linear interpolation between points, for a more accurate value calculation. It can also be mentioned that because of the generality of the hexagonal mask that can be applied, rectangular and rhombic shapes (particular cases) are also supported (see Figure 3-2).

[1] Application Extension Language-ADS internal programming language modeled after the C programming language.

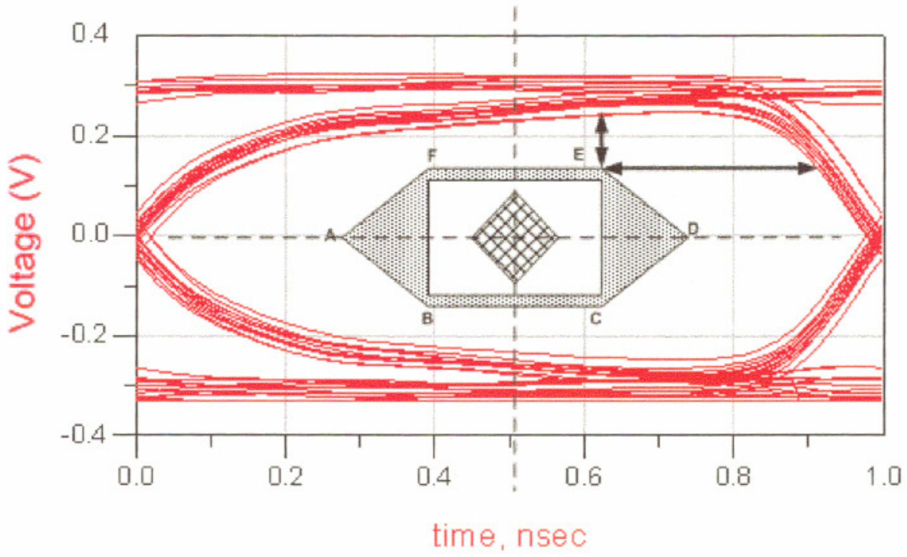


Figure 3-2 User function-defined mask overplayed over typical ADS Eye Diagram

3.3 Code description and options

The code consists of several build-in functions that take in the desired transient data, compares it with the ideal input pattern and bit rate, calculates the center of the eye (several options are available), and with the default settings outputs the minimum time and voltage from the center of the eye.

The function takes the following arguments:

a. Input voltage –this is the transient voltage on which we perform the measurement

b. Input bit sequence or piecewise linear (PWL) voltage – represents the input voltage. *Note: The tool will actually transform bit sequences into an SPICE-like ideal voltage source. The tool can also accept (with minor modifications) an ideal input directly.

c. Period – this is the input signal period.

d. Start time – this is the desired start time for measuring the input signal. Defaults to zero.

e. Stop time – this the desired stop time for measuring the input signal. Defaults to transient simulation stop time.

f. Method for determining eye center – The tool allows for several methods to determine the eye center. This is important because it allows the user to account for various methods of clocking (such as common or embedded) and various receiver implementations, such as over sampling, refresh time etc. Those settings are:

  - No adjustment (0) – in this case the middle of the eye is considered to be at exactly half of the signal period.
  - Min/Max (1) – this is the standard method for measuring diagrams where the center point is in the middle of the difference between the maximum and the minimum jitter.
  - Standard deviation(2) – the middle of the eye is calculated by performing a standard deviation of the jitter over the specified period
  - Jitter count (3) – this method counts the "positive" jitter cycles and the "negative" jitter cycles and moves the eye center based on the difference between the two numbers.
  - Other Method (4) – this option allows the user to implement a specific equation for the receiver tracking mechanism.

g. First significant bit – the first bit out of the bit sequence. This parameter is necessary to account for any initial transient conditions that we do not want to include in the Eye Diagram.

h. Time step for jitter count – this parameter is only valid when the method for determining eye center (g) is set to jitter count (3). It specifies the time step that the center of the eye is used for each movement. This is trying to closely resemble a receiver behavior that is adjusting its sampling point based on the jitter count method. Default value is 1 picosecond.

i. Voltage threshold for numerical errors – this parameter specifies the voltage threshold value that is to be considered as a numerical error. This is important because some simulation tools have some small errors when subtracting identical signals, and those errors can be both negative and positive at times, causing artificial "zero crossing" points that could be incorrectly interpreted as part of the desired bit sequence. Default value is zero.

3.4 Usage Example

A simple simulation has been performed as a usage example of a simple linear transmission line model.

The significant simulation parameters have been included in the following tables. The actual mathematical equation that has to be inserted to perform a measurement is: Output=eye measure (input_voltage, input_bit_sequence, period, start_time, stop_time, statistical_center, first_significant_bit, time_step, voltage_threshold).

The following tables are illustrating the input and output parameters for the eye measurement software. Table 3-1is showing the bit sequence that was used for transient simulations, while Table 3-2 is showing the remaining parameters used to create the Eye Diagram.

Table 3-1 Bit Sequence used for transient analysis and EDA readings

Bit Sequence
0101011001100101100000101001111101011000001010011111010110000010011111

Table 3-2 Transient Simulation parameters used by EDA

Period	*Start Time*	*Stop Time*	*Time Step*	*Bit Rate*	*Statistical Center*	*First Significant Bit*	*Time step for jitter count*	*Voltage threshold for numerical errors*
1E-9s	0s	5E-8s	1E-12s	5 GHz	1	1	NA	0

Table 3-3 Results from the Eye Diagram Analyzer Tool

Eye Opening (voltage)	*Eye Opening (time)*
0.228V	9.568E-10

Table 3-3 is showing the output data from the EDA, using only its default input parameters. As long as typical transient analysis using parameterized values are performed, no user input is necessary (beside the described function call).

In this case the output data is indicating the actual voltage value, however there are options to overlay a time-voltage "mask" that will indicate time and voltage margins compared to the overlaid mask.

Results from this table are also compared with actual measurements using ADS markers and are illustrated in Figure 3-3.

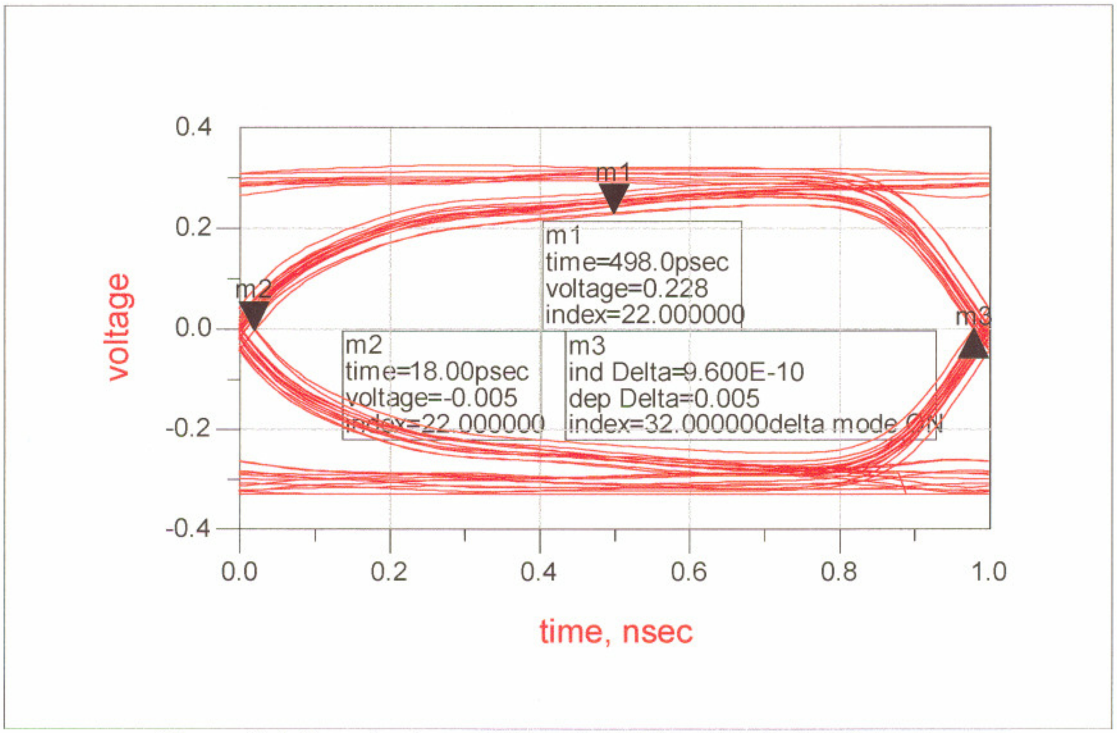


Figure 3-3 Transient results based on 10ps simulation time steps

As can be seen in Table 3-3 and Figure 3-3 there is a good correlation between the measurements using the automated tool and ADS markers. Marker m1 is the marker that indicates the voltage at the eye center, and it has exactly the same value as the automated tool.

Markers m2 and m3 are indicating the time opening of the eye. The difference is marked ind_Delta in Figure 3-3 and it has a slightly different value then the EDA. The difference is about 4 picoseconds and is generated by the fact that the ADS marker can only be moved at discrete value points, determined by the simulation time steps. So while the ADS marker precision is dependent of the transient simulation time step, the EDA tool performed a linear interpolation in order to get a better crossing point number. This example used a 10 picosecond time step.

In order to reduce the difference, a transient simulation has been performed with a finer time step at cost of increasing actual simulation time.

To illustrate this, a transient simulation using a 5ps time step has been performed, and the results are illustrated in Figure 3-4.

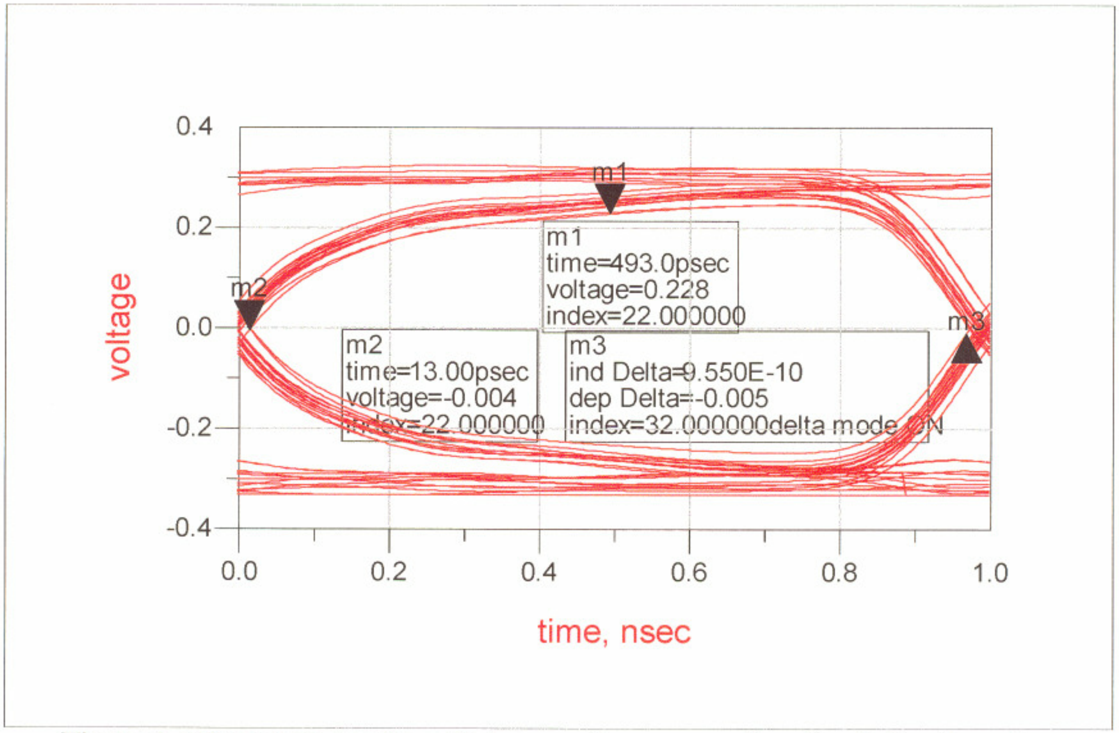


Figure 3-4 Transient results with improved accuracy - 5ps simulation time steps

We can see that in this case we get almost perfect correlation between the measurements, however, the actual simulation time (CPU time) reported by the tool has increased by 2.2 times (894 vs. 405 seconds) between the two simulations.

This example indicates that the EDA provided accurate results even when a coarser simulation time step was used, therefore saving development time.

It is to be mentioned that this is a valued feature, since simulation times tend to increase due to the fact that more and more data patterns have to be used to characterize lower frequency behavior, as well as allowing enough transient time for initial conditions to settle.

Although it appears that 4 picoseconds might not be a significant value, for a bus running at 5GT/s this is 4% of the total budget lost just because of simulation/measurement inaccuracies.

Chapter 4

Frequency domain transmission line model approximation

4.1 Background

Since the most accurate measurements for transmission lines are performed in frequency domain it can be useful if we can also translate those measurements into frequency domain transmission line models. This will not only provide an accurate one-to-one comparison between measurements and models, but could also significantly reduce simulation time in some simulators, mainly in those that use convolution to solve time domain data.

Also, as noted in the introductory chapters, transmission line models specified in time domain are very complex and do not account for "second order effects" such as "mushroom" shaped traces, and dielectric fiber-weave variations. Frequency based models by contrast are much simpler, and can be easily modified to account for the above mentioned effects.

However, a major problem faced by designers is the fact that s-parameters for lossy transmission lines are frequency dependent, so, traditionally, s-parameter modeling could only be done by large matrix manipulations with all the disadvantages that occur from that, such as matrix mismatch, non correlations between frequency steps, maximum frequency extrapolation, etc.

Another issue with s-parameters in matrix format is that Fourier and Inverse Fourier manipulation can result in aliasing and causality issues that are difficult to detect and fix.

Several approaches have been made to approximate s-parameter matrices with functions. Many of these are in fact creating complicate approximations, such as Pade

[6], or complicated functions such as Laplace and rational functions (quotients of two polynomials) used in SPICE simulators [7].

4.2 Frequency domain measurement setup

Two measurements have been performed, one of a longer microstrip (11.5 inch) and one of a shorter one (6 inch).The impedance of both traces has been calibrated to yield 60 Ohm (single ended) by using simplifying microstrip impedance formulas [11]:

$$Z0 = \frac{87}{\sqrt{(Er+1.41)}} * Ln(\frac{5.98*H}{0.8*W+T})$$

Equation 4-1

The setup used for these measurements consist of an Agilent Technologies HP 8720D Vector Network Analyzer (VNA) with Cascade Microtech 40A Picoprobes [8]. The total rated bandwidth of the system is 40GHz, but measurements have been performed up to 10GHz, which is adequate for the transfer rates seen in today's designs. The whole system was calibrated up to 10GHz using Cascade Microtech calibration kit. The Short-Open-Load-Through methodology (The standards used in this method are shorts, opens, loads, and through making this what is often referred to as a SOLT calibration). A more detailed description of the SOLT can be found in [9].

Measurements have been taken on two microstrip lines with the maximum instrument resolution of 1601 points. Microstrip lines have been chosen, because they have more complicated EM solutions (dispersion, non-TEM modes, etc), therefore are usually harder to model (and not so accurate) in current simulation tools, than similar stripline structures.

Based on Equation 4-1 the design parameters for the test board (FR4 material) had the following values:

$$\begin{aligned}
&Dielectric\,Height = 4mil\\
&TraceThickness = 1.5mil\\
&TraceWidth = 4.5mil\\
&E_R = 4(estimated)\\
&Tan\delta = 0.025(estimated)\\
&Conductivity = 5.3E7(estimated)\\
&Z0 = 60.51Ohm
\end{aligned}$$

Equation 4-2

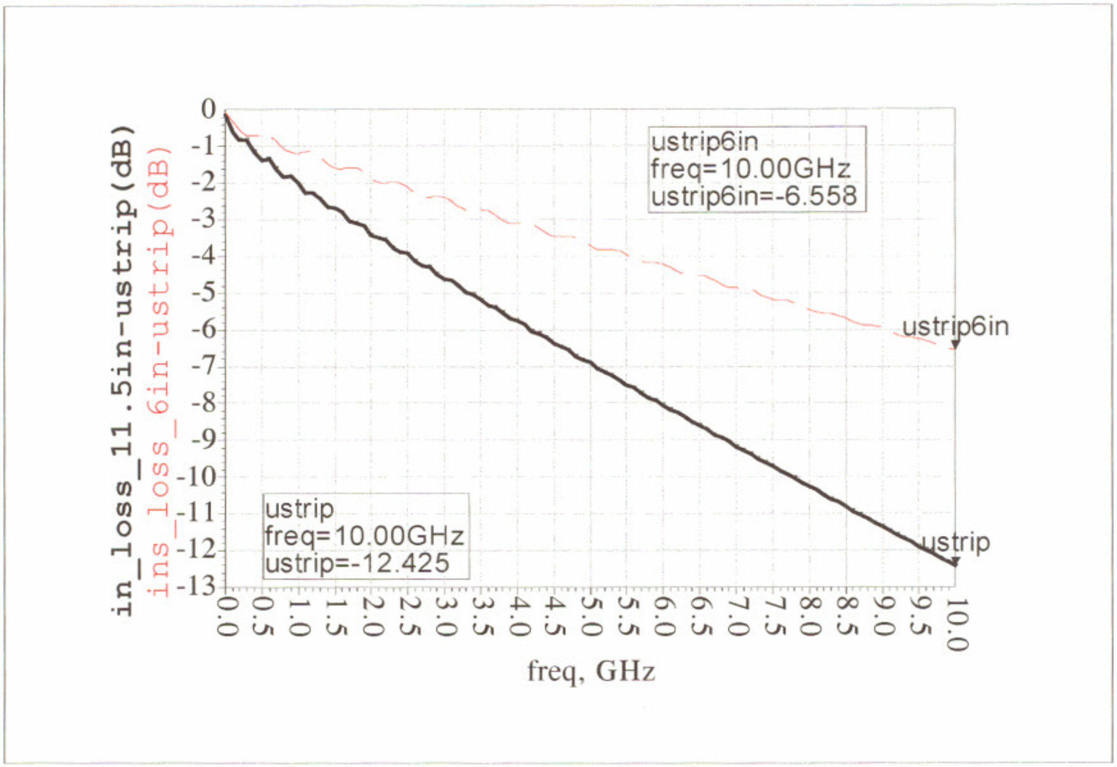


Figure 4-1 Insertion loss magnitude -VNA measurement for 6 and 11.5 inch microstrip

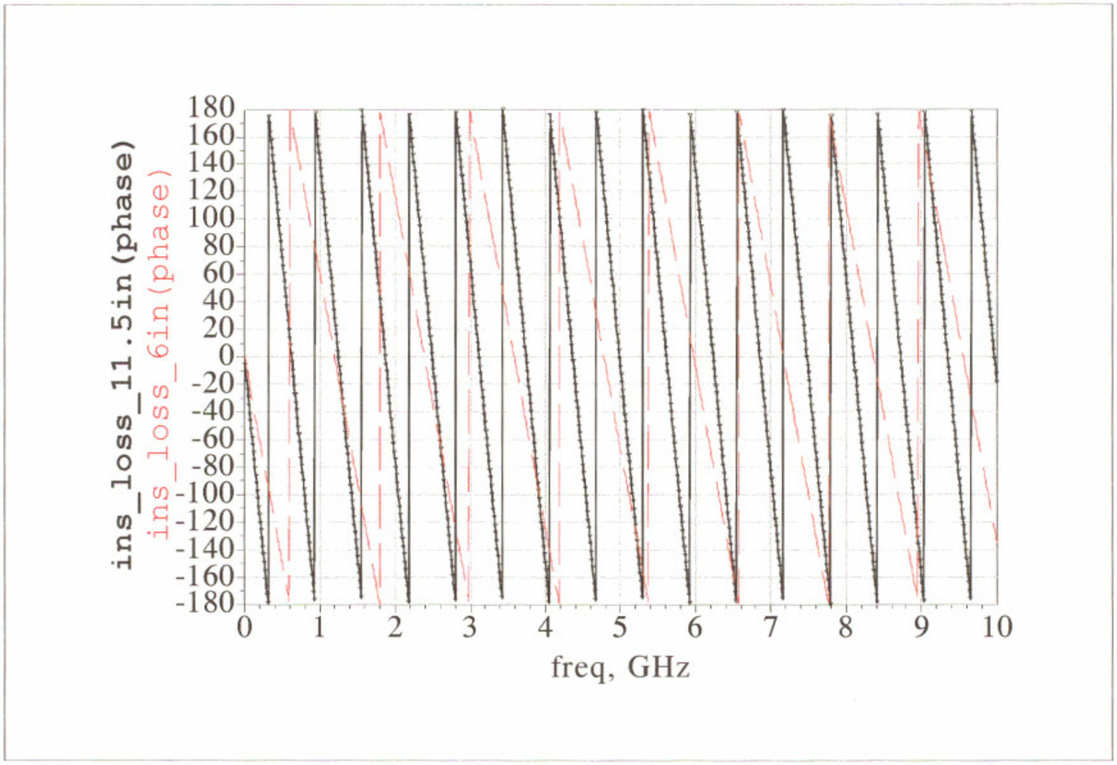


Figure 4-2 Insertion loss phase -VNA measurement for 6 and 11.5 inch microstrip

The magnitude of the frequency domain representation of forward insertion loss (S12) is shown in Figure 4-1 in (dB). Notice that the data has been extrapolated to 0 Hz based on simple resistive loss calculations. This is necessary because typical VNAs do not measure s-parameters all the way to DC, but simulators rely heavily on DC values for their calculations. Typically if a DC value is not present, simulators tend to linearly extrapolate based on the last present value and deduce the wrong values (since the loss function is obviously non-linear at low frequencies). Phase correlation for the same structures is indicated in Figure 4-2. Similarly, phase has been extrapolated to DC. Extrapolating phase to DC is a much easier task, since phase should be zero at DC.

4.3 Frequency domain approximation using linear functions

4.3.1 Insertion Loss (S12)

The approach that is presented here can be used in many simulators that support symbolically defined equations in frequency domain, such as Advanced Design System(ADS) from Agilent, as well as mathematical oriented packages such as MATLAB and Mathematica.

By looking at the magnitude of the insertion loss of the two measured microstrip structure (Figure 4-1), we observe that is has a fairly linear behavior on a logarithmic scale (dB), especially at higher frequencies. This leads us to the idea that we can represent a transmission line using a linear (logarithmic scale) function in frequency domain. Of course the actual model is not linear but we can assume that as a first approximation or for reference loading purposes and add complexity to the model as necessary. Also, it is to be noted that s-parameters are complex values; therefore, for a correct representation both magnitude and phase have to be accounted.

We know [12] that the phase of a transmission line is mathematically represented as an exponential function of the form:

$$\theta = \exp(j * 2 * \pi * f), \textit{ where } f \textit{ is the frequency}$$

Equation 4-3

In order to derive the linear model, we can combine the amplitude and phase using the most basic representation of a complex function, which is:

$$s12 = a * 10^{-f} * \exp(-j * \omega * N), \textit{ where } f \textit{ is the frequency and } \omega = 2 * \pi * f$$

Equation 4-4

In Equation 4-4 (a) and (N) represent scaling coefficients that will be determined in the following paragraphs.

First we will determine the amplitude coefficient. Since this is a logarithmic representation we will expect the coefficients to be a power of 10. We can also further assume that there is a coefficient representing the DC value and a coefficient representing the function value (which is linear) at a certain frequency. With these assumptions Equation 4-4 becomes:

$$S12 = 10^{coef\_dc\_S12+coef\_lin\_S12} * \exp(-j * omega * N)$$

Equation 4-5

In order for the coefficient to be more descriptive the following notation convention has been used:

val_type_Sxx - Value for the (type) correlation coefficient for Sxx parameter
freq_type_Sxx - Frequency for the (type) correlation for Sxx parameter
coef_type_Sxx = Coefficient for the (type) corelation for Sxx parameter

Equation 4-6

Based on the notational conventions from Equation 4-6 initial coefficients for the linear approximation can be defined as follows:

val_lin_S12 - Value for the linear correlation coefficient for S12 parameter
freq_lin_S12 - Frequency for linear correlation for S12 parameter
coef_lin_S12 = val_lin_S12 * (FREQ/freq_linS12)

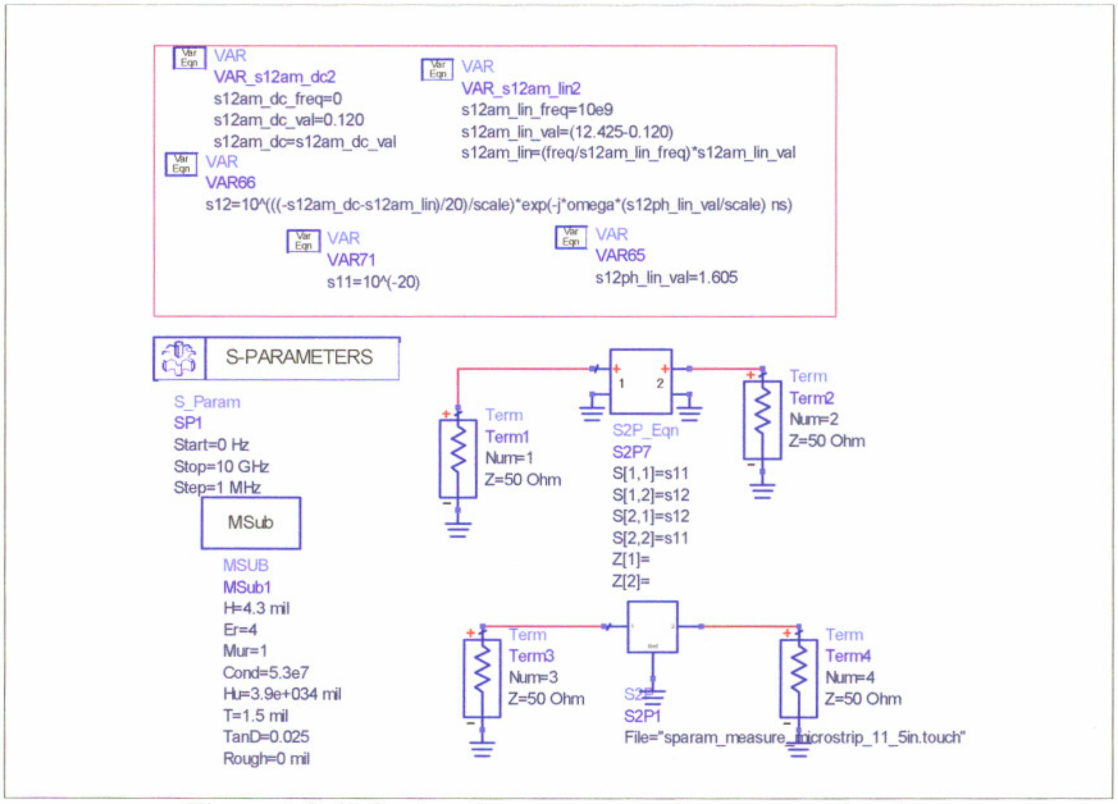
Equation 4-7

DC coefficients are also defined in a similar way:

val_dc_S12 - Value for the DC correlation coefficient for S12 parameter
freq_dc_S12 - DC correlation for S12 parameter (0Hz)
coef_dc_S12 = val_dc_S12 * 1

Equation 4-8

Setup for comparing measurements and models using ADS design tool is shown in Figure 4-3.



```
VAR
VAR_s12am_dc2
s12am_dc_freq=0
s12am_dc_val=0.120
s12am_dc=s12am_dc_val

VAR
VAR_s12am_lin2
s12am_lin_freq=10e9
s12am_lin_val=(12.425-0.120)
s12am_lin=(freq/s12am_lin_freq)*s12am_lin_val

VAR
VAR66
s12=10^(((-s12am_dc-s12am_lin)/20)/scale)*exp(-j*omega*(s12ph_lin_val/scale) ns)

VAR
VAR71
s11=10^(-20)

VAR
VAR65
s12ph_lin_val=1.605

S-PARAMETERS
S_Param
SP1
Start=0 Hz
Stop=10 GHz
Step=1 MHz

MSub
MSUB
MSub1
H=4.3 mil
Er=4
Mur=1
Cond=5.3e7
Hu=3.9e+034 mil
T=1.5 mil
TanD=0.025
Rough=0 mil

Term
Term1
Num=1
Z=50 Ohm

S2P_Eqn
S2P7
S[1,1]=s11
S[1,2]=s12
S[2,1]=s12
S[2,2]=s11
Z[1]=
Z[2]=

Term
Term2
Num=2
Z=50 Ohm

Term
Term3
Num=3
Z=50 Ohm

S2P
S2P1
File="sparam_measure_microstrip_11_5in.touch"

Term
Term4
Num=4
Z=50 Ohm
```

Figure 4-3 ADS setup for s-parameter linear approximation

Based on the measurements described before, performed on an 11.5 in microstrip line in isolation (no coupling to adjacent structures) we have determined (extracted based on measured data) the values seen in Equation 4-9 at DC and 10 GHz. We took 10 GHz as the maximum frequency over which the measurement was made, but any arbitrary value can be selected. It is desired though to select a value that is in the range of the expected time domain bandwidth, so that maximum precision is achieved in that range.

$$\text{val\_dc\_S12} = \text{-0.120 dB}$$

$$\text{freq\_dc\_S12} = 0$$

$$\text{coef\_dc\_S12} = \text{-0.120}$$

and

$$\text{val\_lin\_S12} = \text{-12.452 - val\_dc\_S12} = -12.332$$

$$\text{freq\_lin\_S12} = \text{10e9 Hz}$$

$$\text{coef\_lin\_S12} = \text{-(freq/freq\_lin\_S12)* val\_lin\_S12} = -12.332 * (freq/10e9)$$

Equation 4-9

The corresponding frequency domain results are shown in Figure 4-4. We can see that (as expected) we have perfectly matched the measured s-parameters amplitude with our linear approximation in 2 points (at 0GHz and 10GHz).

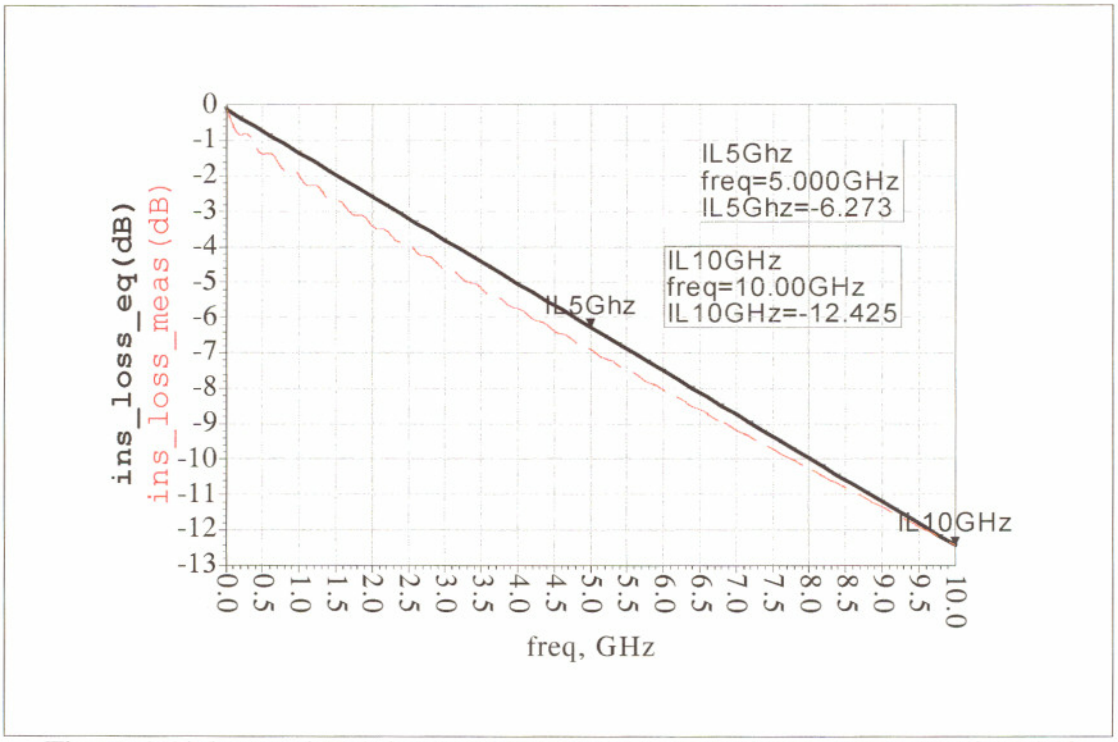


Figure 4-4 Measured and simulated insertion loss magnitude –two point (0 and 10 GHz) linear approximation

The next step is to validate our phase assumptions. Equation 4-4 assumes a linear phase variation (no dispersion). Matching the phase will be reduced, in this case, to matching the phase coefficient (N). This can be easily achieved using a linear optimization based on one value, and was performed using the ADS built-in optimizer. The result was a value of N=1.605. Based on this optimization the phase of the insertion loss is plotted in Figure 4-5. The plot shows a good phase correlation up to the desired frequency range (in this case 10GHz), so we can conclude that the line has a negligible amount of distortion.

If distortion would have been more significant a phase drift would have been present at higher frequencies. In that case a frequency dependent term can be added to the phase.

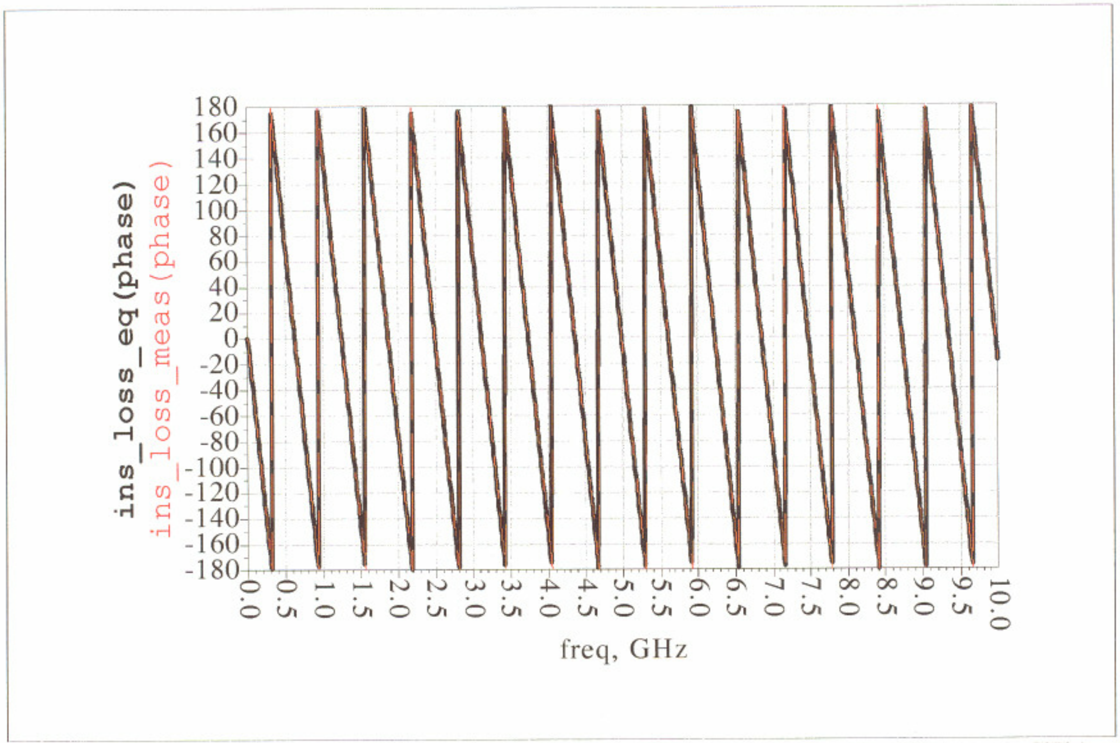


Figure 4-5 Measured and simulated insertion loss phase - two point (0 and 10 GHz) linear approximation

4.3.2 Return Loss (S11)

Before we can use the model in time domain and perform a transient or harmonic balance correlation, we have to define a value for the return loss. Similarly with Equation 4-5 we can define a linear approximation for S11:

$$S11 = val\_dc\_shift\_s11 - 10^{coef\_dc\_S11 + coef\_lin\_S11} * \exp(-j * omega * 2 * N)$$

Equation 4-10

However, there are some differences between the formulas for S12 (Equation 4-5) and S11 (Equation 4-10). The first one is that we have to add a DC value since the reflection coefficient should initially have a real magnitude value close to one. The other difference is in the phase. Since the distance for the reflected wave is twice the distance of the initial wave, the phase coefficient should be twice as much as the S12 phase coefficient.

Using the same approach we can also derive a formula for the reflection coefficient (S11). We can notice that the magnitude of the reflection loss is several orders lower than the one of the insertion loss, so we do not need the same amount of granularity for a good fit. We define the coefficients, using the same notational conventions.

$$val\_dc\_shift\_S11 = 0.13$$
$$val\_dc\_S11 = -19dB$$
$$freq\_dc\_S11 = 0$$
$$coef\_dc\_S11 = -19$$
$$val\_lin\_S11 = -28dB$$
$$freq\_lin\_S11 = 10e9Hz$$
$$coef\_lin\_S11 = -(freq / freq\_lin\_S11) * val\_lin\_S11 = -28 * (freq/10e9)$$

Equation 4-11

Using the values derived (as shown in Equation 4-11) we can obtain a good approximation of the reflection coefficient in both amplitude (Figure 4-6) and phase (Figure 4-7).

4.3.3 Fast Fourier Transform (FFT) consideration

We have determined so far the first order approximations of magnitude and phase for both inserted and reflected waves for the measured transmission line. Next step is to determine the accuracy of those models in time domain.

For this, the impulse response for both measured and deduced transmission lines has to be calculated. Impulse response is derived by performing a Fast Fourier Transform (FFT) of the s-parameters. As mentioned before this is performed in ADS using their FFT function, but can be performed in a variety of simulators.

ADS and MATLAB results have been validated and correlated for this study.

The MATALB implementation of the FFT function [13], is based on the on the Cooley-Turkey algorithm [14]. ADS developers have not provided documentation on their FFT implementation algorithm.

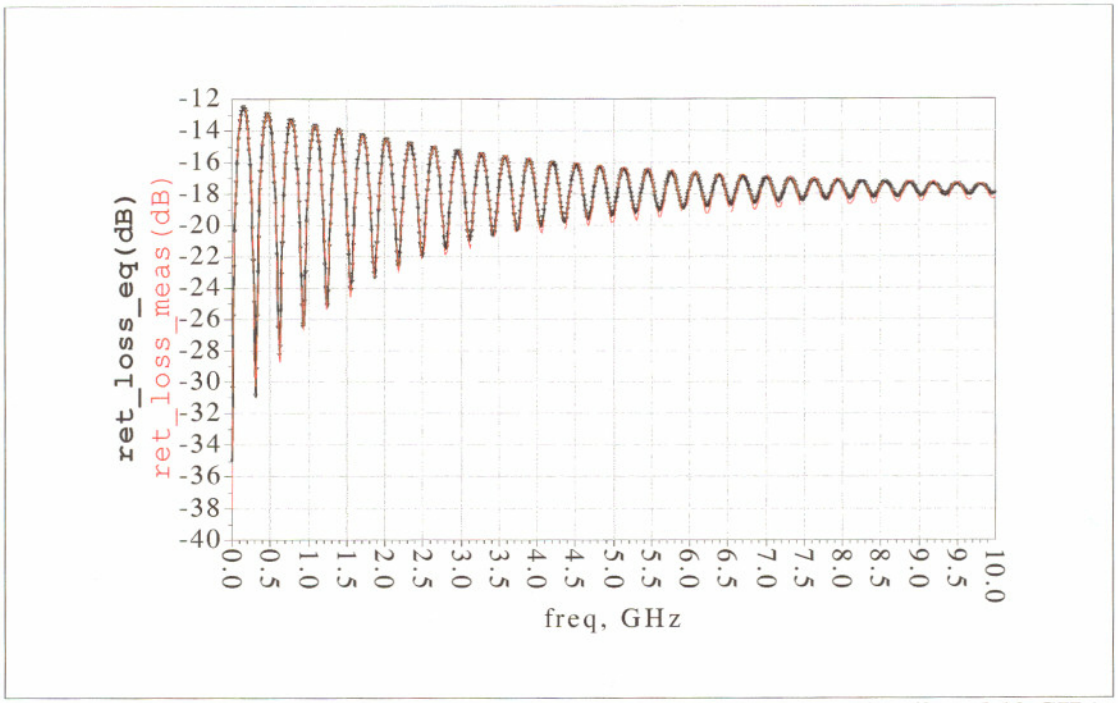


Figure 4-6 Measured and simulated return loss magnitude - two point (0 and 10 GHz) linear approximation

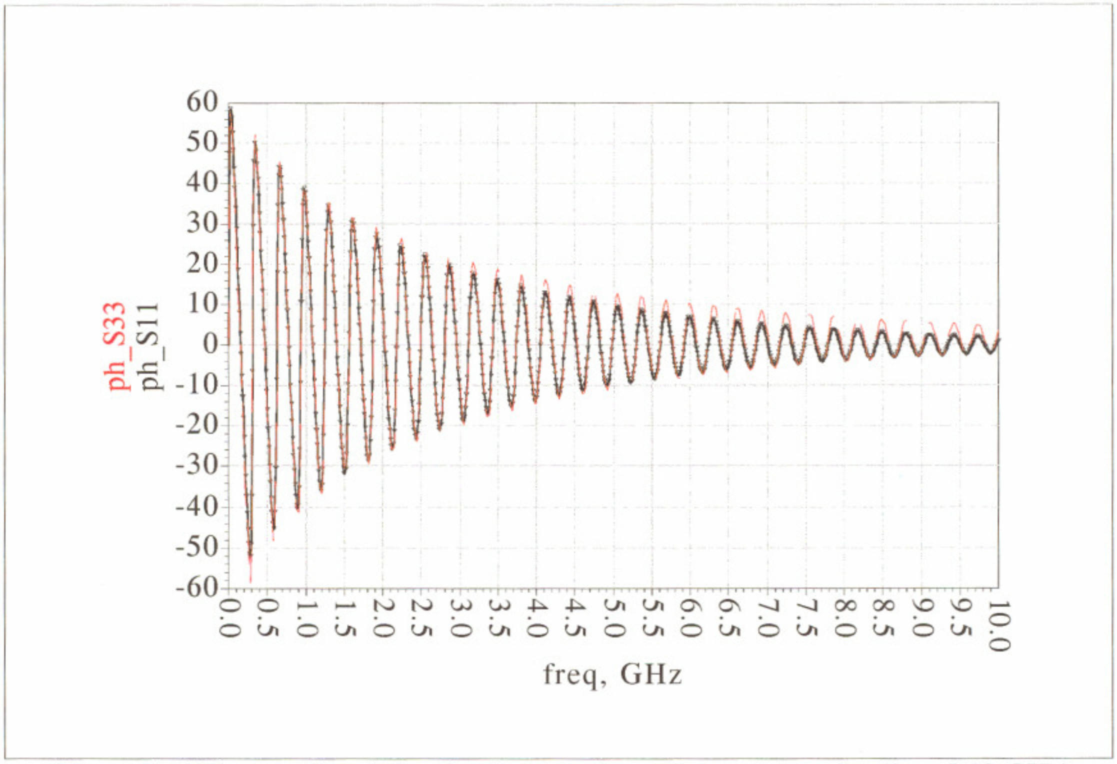


Figure 4-7 Measured and simulated return loss phase - two point (0 and 10 GHz) linear approximation

The functions X=fft(x) and x=ifft(X) implement the transform and inverse transform pair given for vectors of length (N) by:

$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)}$$

$$x(j) = (1/N)\sum_{k=1}^{N} X(k)\omega_N^{-(j-1)(k-1)}$$

$$where:$$

$$\omega_N = e^{-(2*\pi*i)/N} \quad is\ the\ N^{th}\ root\ of\ unity$$

Equation 4-12

We can see that the FFT function is dependent on the number of points N, which represents the Nth root of unity. The Cooley-Turkey algorithm uses a composite in which (N) is divided into smaller numbers by (N=N1*N2). The algorithm first computes N1 transforms of N2 sizes and then N2 transforms of N1 sizes. The decomposition is then recursively applied to N1 and N2 until the problem is solved. When N is a prime number additional algorithms have to be used before the decomposition can be performed.

This brief description of the algorithm illustrates the importance of choosing the right N value, for a fast computation. Choosing the right N value is almost impossible when s-parameters are in matrix format with a pre-determined number of points, therefore interpolation and extrapolations, which add errors, have to be performed.

This is another advantage of using mathematical continuous functions which introduce no such errors.

The ADS FFT function [15] has several options, including several filtering methods, but in order to be compliant with the MATLAB representation of the FFT, no such options have been selected.

4.3.4 Time domain correlation – linear models

The first step that has to be performed for the time domain correlation is the FFT transform of the frequency domain models. The transform has been applied to both measured data (s-parameter matrix format) and the linear model derived in the previous sections. The results are compared in Figure 4-8.

The first noticeable fact is that the impulse response of both measurement and approximation is (relatively) symmetric with respect to the main lobe. This confirms our initial indications (based on the observed phase linearity) that we do not have significant dispersion on the transmission line.

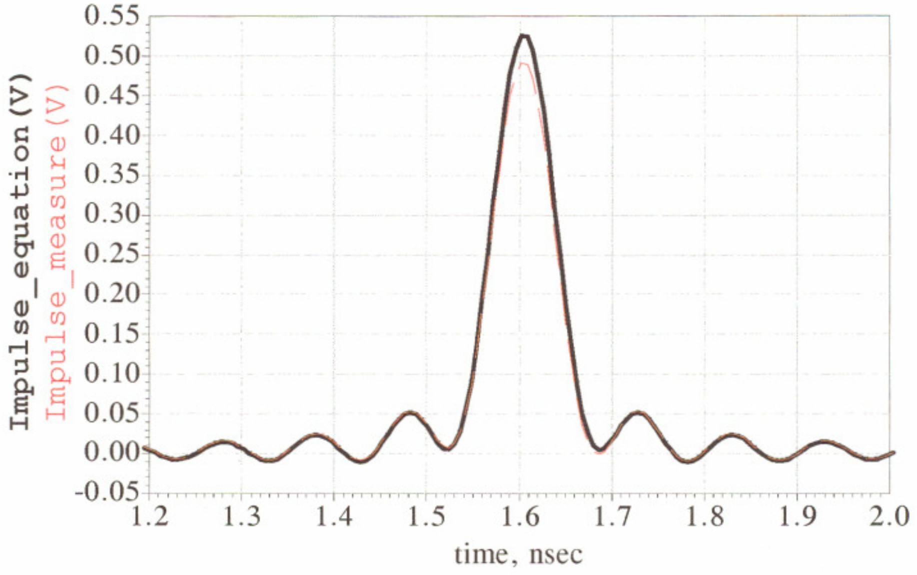


Figure 4-8 Impulse response correlation using measured and two point (0 and 10 GHz) linear approximated s-parameters

We can also see that the impulse derived from the equation is higher in amplitude than the one derived from measurements. The fact that we have less loss from the equation models is obvious if we take a look again at Figure 4-4. The linear model derived (straight line) is always higher (less loss) that the one from measurements.

Based on this amplitude mismatch we can expect some errors time domain simulation using the linear model. However this might not always be the case. A better correlation can be obtained if we are trying to match the measurements at a different frequency.

This way we will get less loss at lower frequencies and more loss at higher frequencies, therefore balancing the overall loss amplitude difference.

An example of that is shown in Figure 4-9 where the linear match is at 5GHz. Based on that we are able to better match the impulse response of the two functions, as plotted in Figure 4-10.

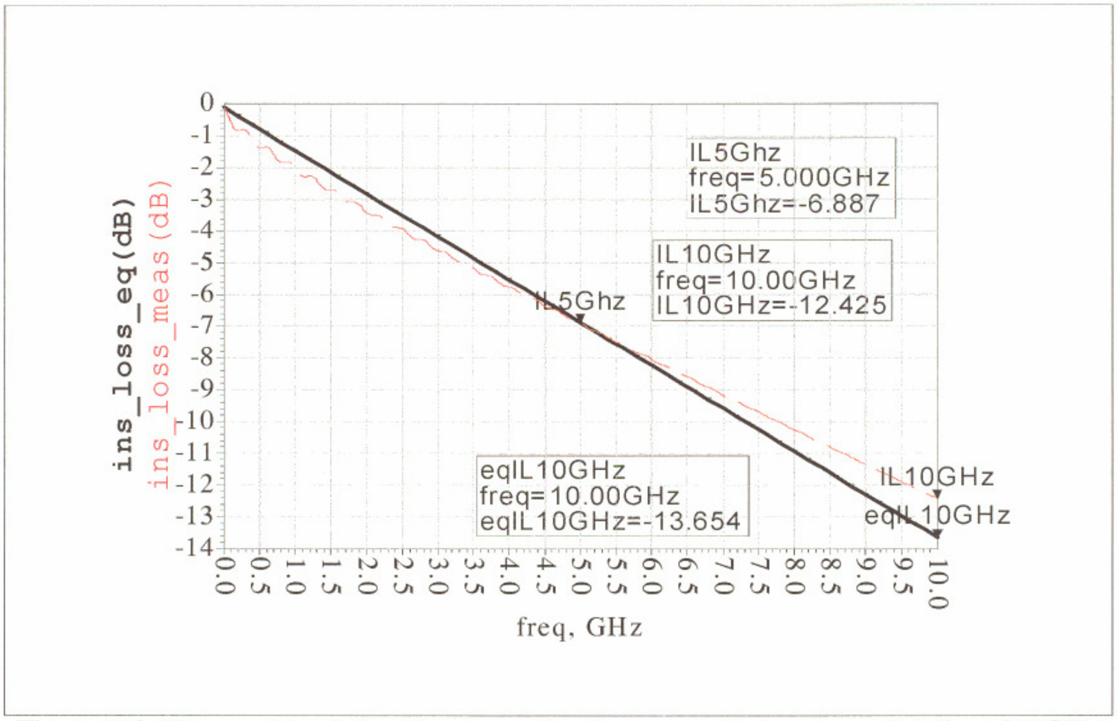


Figure 4-9 Measured and simulated insertion loss magnitude –two point (0 and 5 GHz) linear approximation

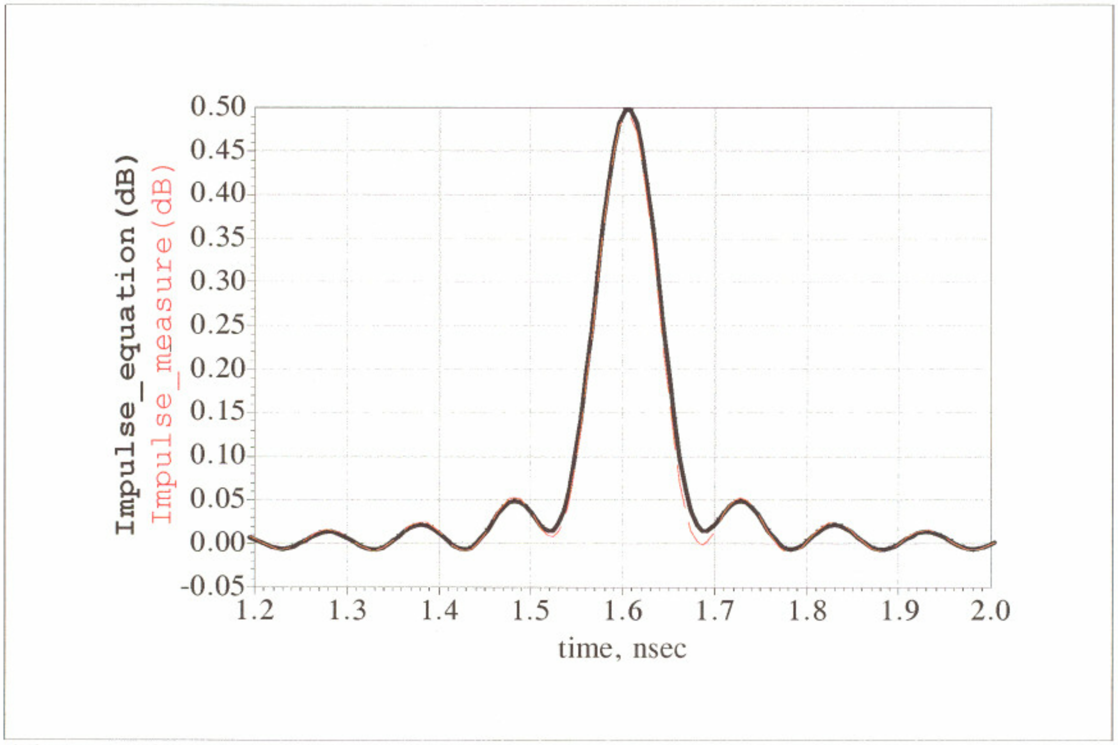


Figure 4-10 Impulse response correlation using measured and two point (0 and 5 GHz) linear approximated s-parameters

It is of course up to the designer to furthermore quantify this type of approach and determine if such models can be used in actual simulations or for other purposes such as Reference/Specification models.

In order to correctly characterize the time domain response for a specific system, we have to take our correlation one step further and convolve the impulse response with an actual driver time domain sequence. Using an ideal driver (perfect pulse) instead of a system specific one, would yield identical results, since they will be convolved with the full spectral content, uniformly distributed in our case from DC to 10 GHz. Therefore matching time domain response for ideal drivers, can be somehow misleading, and would not give accurate data for specific cases. It can be, however, used as reference designs or specifications.

Real systems have finite edges and finite spectral content, usually non-uniform distributed, therefore, using realistic driver models, is the only accurate way to characterize a particular system.

The full system characterization mentioned above can be performed in several ways. One of them is to have a traditional transient simulation, using a long random bit pattern that encompasses the full expected spectral content and read the results using the EDA. This is the only accurate way for non-linear systems.

An alternate method, applicable for linear systems only, is to use Peak Distortion Analysis (PDA) [16]. PDA uses the system response of a lone pulse to replace the random bit patterns, therefore saving computation time. PDA responses can be analyzed either mathematically, by summing all amplitudes of the TAP cursors or using the EDA. For linear system all the above approaches lead to similar results (as shown in the following experiment), however for non-linear systems only the first method (transient simulation with random bit sequence and EDA readings) can be used.

To determine the system pulse response, a linear current source driver is assumed. The final stage is driven by a 2.5 GHz (5 GT/s) pre-driver. The total capacitance of the driver used during the entire experiment is set to 1pF. Driver rise and fall times (dependent on the total driver capacitance) are close to 100 ps (measured 10% to 90%) Figure 4-11 shows the driver and pre-driver characteristics as measured into a 50 Ohm load.

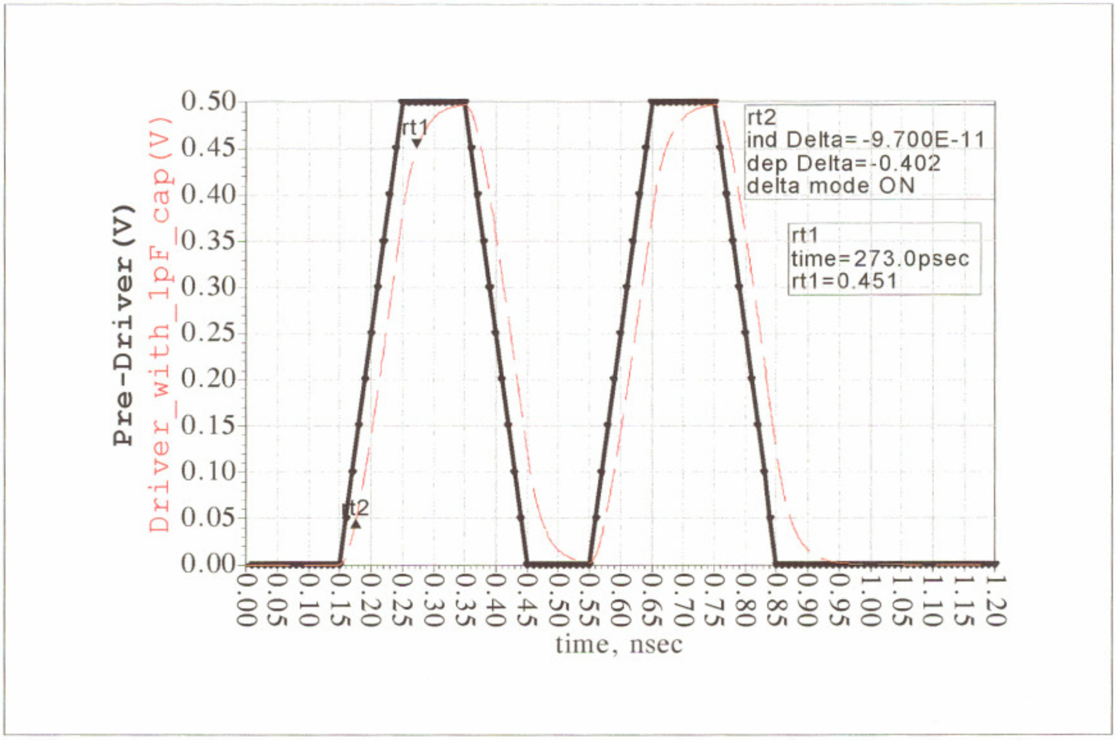


Figure 4-11 50 Ohm linear current pre-driver and driver with 1 pF total capacitance

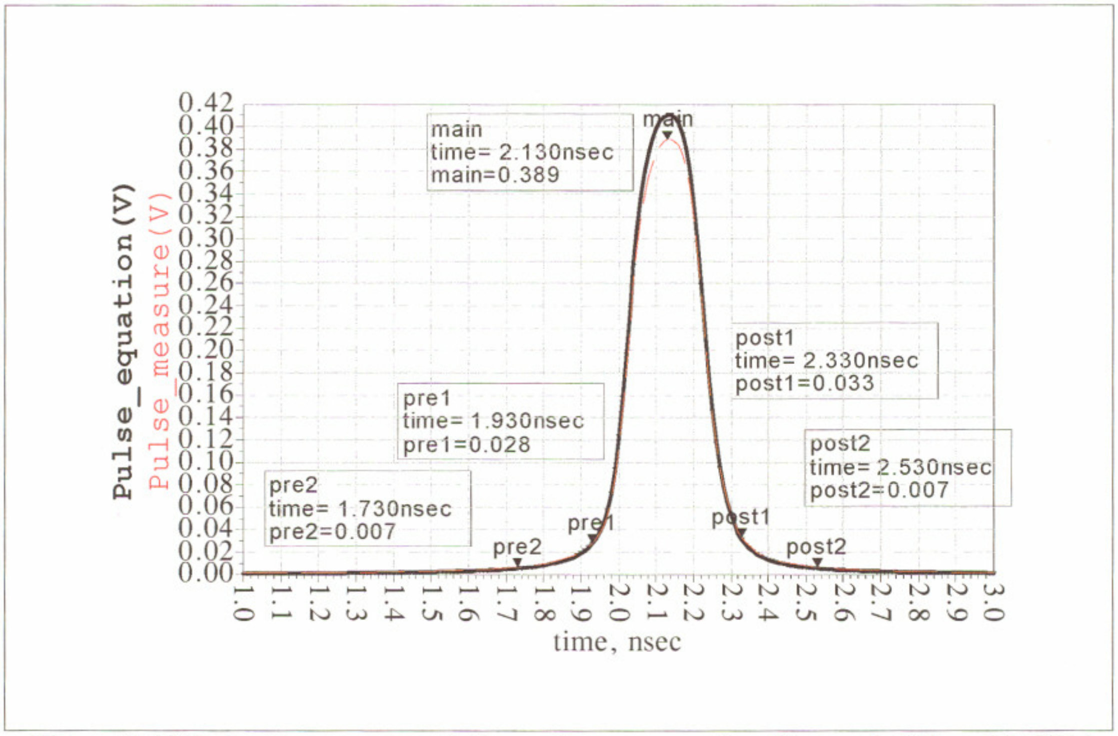


Figure 4-12 Pulse response correlation using measured and two point (0 and 5 GHz) linear approximated s-parameters

It is to be noted that even if the PDA analysis applies only to linear systems, the worst case pattern determined using a PDA approach (analyzing the pulse of the system), will be accurate even for non-linear systems. However, for non-linear systems, the absolute worst case voltage and time margins have to be calculated using transient simulations and EDA based on the PDA-derived pattern.

The pulse response obtained by using the above described driver (no equalization) for both measured and derived s-parameter data is shown in Figure 4-12. It also shows the actual cursor values used for the PDA analysis for both data sets.

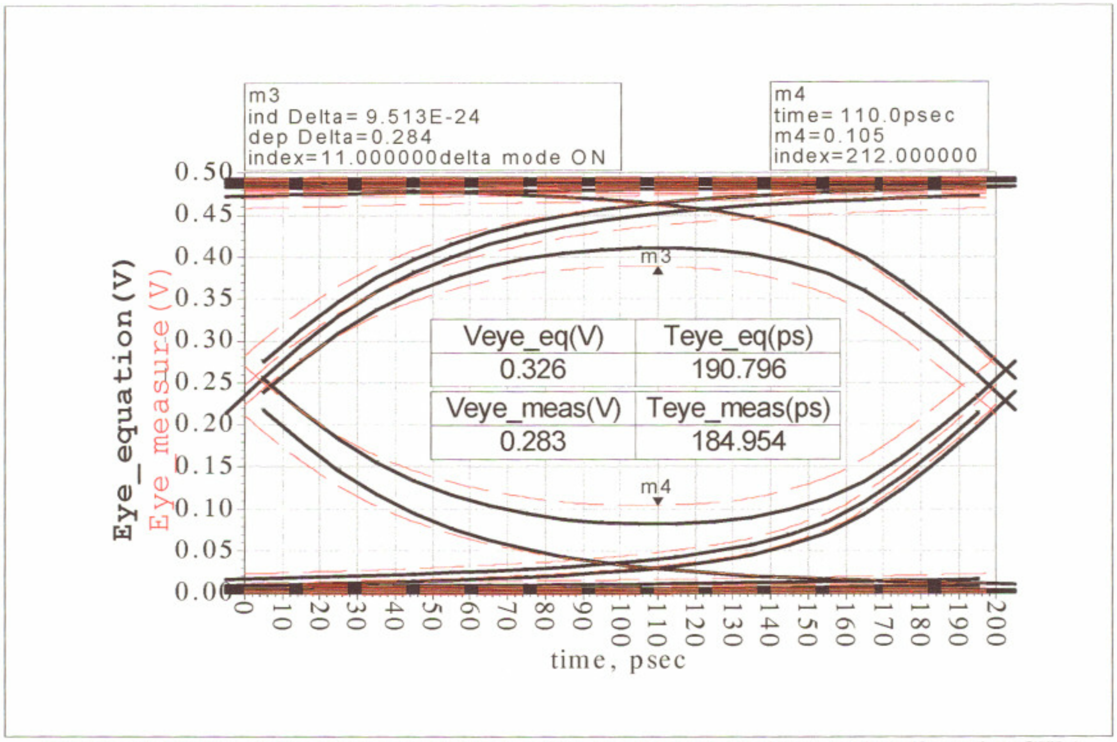
To perform an accurate PDA analysis cursors have to be accounted for until they reach 0 (no energy left on the tails). This can be done with the EDA reader which can also provide all cursor values. Values obtained with the EDA (see Table 4-4) can be compared for reference with the one indicated in Figure 4-12.By looking at the cursor values the first relevant issue is the fact that there are no negative coefficients. The fact that there are no negative cursors makes determining the worst case pattern an easy task. If only positive cursor values are present a single (lone) one will be the worst case pattern.

Table 4-4 also shows the mathematical calculations using PDA that would give us the worst case eye opening (in voltage) for the linear system. This calculation involves adding all the ISI values (ISI_SUM) and subtracts that from the main pulse (EYE_OPEN).

Table 4-4 Pulse response cursors using measured and two point (0 and 5 GHz) linear approximated s-parameters

Number	time	isi_meas		time	isi_eq
		CURSOR			**CURSOR**
	2.13600E-09	**0.38912**	\\	2.13400E-09	**0.41093**
		PRE-CURSOR			**PRE-CURSOR**
1	5.36000E-10	0.00038	\\	5.34000E-10	0.00020
2	7.36000E-10	0.00081	\\	7.34000E-10	0.00042
3	9.36000E-10	0.00104	\\	9.34000E-10	0.00056
4	1.13600E-09	0.00141	\\	1.13400E-09	0.00081
5	1.33600E-09	0.00205	\\	1.33400E-09	0.00127
6	1.53600E-09	0.00337		1.53400E-09	0.00228
7	1.73600E-09	0.00698		1.73400E-09	0.00538
8	1.93600E-09	0.03072	\\	1.93400E-09	0.02930
		POST-CURSOR			**POST-CURSOR**
1	2.33600E-09	0.03064	\\	2.33400E-09	0.02937
2	2.53600E-09	0.00651	\\	2.53400E-09	0.00519
3	2.73600E-09	0.00309	\\	2.73400E-09	0.00222
4	2.93600E-09	0.00184	\\	2.93400E-09	0.00124
5	3.13600E-09	0.00122	\\	3.13400E-09	0.00080
6	3.33600E-09	0.00087	\\	3.33400E-09	0.00056
7	3.53600E-09	0.00065	\\	3.53400E-09	0.00041
8	3.73600E-09	0.00051	\\	3.73400E-09	0.00032
9	3.93600E-09	0.00040	\\	3.93400E-09	0.00026
10	4.13600E-09	0.00034	\\	4.13400E-09	0.00021
11	4.33600E-09	0.00030	\\	4.33400E-09	0.00018
12	4.53600E-09	0.00029	\\	4.53400E-09	0.00015
13	4.73600E-09	0.00034	\\	4.73400E-09	0.00013
14	4.93600E-09	0.00059	\\	4.93400E-09	0.00012
15	5.13600E-09	0.00233	\\	5.13400E-09	0.00010
16	5.33600E-09	0.00566	\\	5.33400E-09	0.00009
17	5.53600E-09	0.00016	\\	5.53400E-09	0.00009
18	5.73600E-09	0.00016	\\	5.73400E-09	0.00008
19	5.93600E-09	0.00023	\\	5.93400E-09	0.00007
20	6.13600E-09	0.00021	\\	6.13400E-09	0.00007
21	6.33600E-09	0.00019		6.33400E-09	0.00006
22	6.53600E-09	0.00017		6.53400E-09	0.00006
23	6.73600E-09	0.00016		6.73400E-09	0.00006
24	6.93600E-09	0.00014		6.93400E-09	0.00002
25	7.13600E-09	0.00013		7.13400E-09	0.00000
		ISI SUM			ISI SUM
		0.10388			0.08208
		EYE OPEN			EYE OPEN
		0.28525			**0.32885**

To compare the (above) eye opening based on PDA cursor measurement with results obtained using the EDA tool we can apply a single (lone) one and a single zero pattern and run the Eye Diagram Analyzer. Results (numbers) obtained using the EDA tool are overlaid over the traditional Eye Diagram simulators cursors (m3 and m4) provided by the simulators graphical interface, and can be seen in Figure 4-13:



m3
ind Delta= 9.513E-24
dep Delta=0.284
index=11.000000delta mode ON

m4
time= 110.0psec
m4=0.105
index=212.000000

Veye_eq(V)	Teye_eq(ps)
0.326	190.796
Veye_meas(V)	Teye_meas(ps)
0.283	184.954

Figure 4-13 Eye Diagram correlation using measured and two point (0 and 5 GHz) linear approximated s-parameters

We can see that the EDA gives us the same results as the PDA methodology. This is expected, considering that we have only linear components in the system.

Figure 4-13 also shows that there is around 15% error in the voltage margin. This outlines the necessity to correctly evaluate linear s-parameter models for specific applications. For our particular system, even if the impulse response was perfectly matched we are seeing significant differences in eye margins. However the 15% error quoted is the absolute worst case and will significantly improve if patterns that reduce the total bandwidth are used (such as balanced 8bit10 encoding [17].

4.4 Frequency domain approximation using non-linear functions

4.4.1 Insertion Loss

To provide a better approximation in time domain we have to compensate for the actual non-linearity of the insertion loss. The actual non-linearity is in the lower frequencies, so we would like to add a function that has a strong impact at those lower frequencies and fades at higher ones. Such a function is the logarithmic function. With this observation the magnitude term in Equation 4-5 becomes.

$$S12 = 10^{coef\_dc\_S12+coef\_lin\_S12+coef\_\log\_S12} * \exp(-j * omega * N)$$

Equation 4-13

The lower frequency to be chosen as a reference is somewhat arbitrary, but in general is should coincide with the highest deviation from linearity. In our case we have chosen to optimize that frequency using the ADS optimizer, so:

$$\log\_comp\_freq = c1$$

Equation 4-14

Since the compensating frequency can be in some cases non-intuitive, it can be left to the compensation algorithm, but this will actually increase the complexity of the compensation process as well the necessary convergence time.

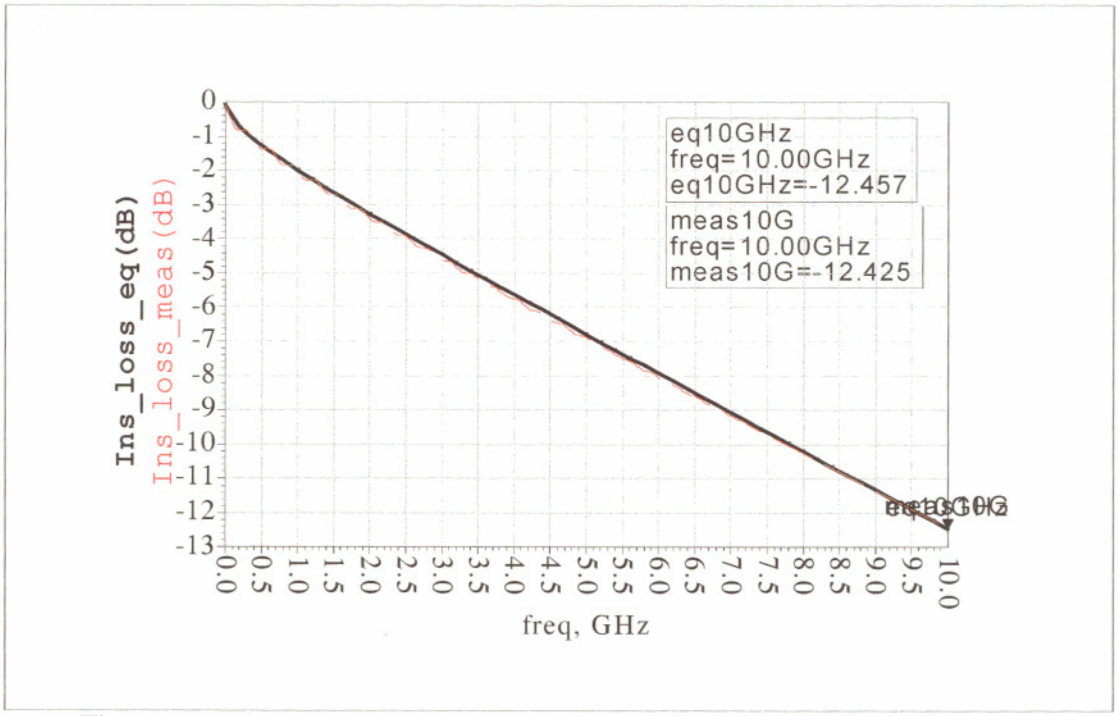


Figure 4-14 Insertion loss correlation using measured and mixed (linear and logarithmic) approximated s-parameters

With the above observation the frequency dependent term for the insertion loss becomes:

$$coef\_\log\_S12 = c1 * \log(\log\_comp\_freq * freq)$$

Equation 4-15

With those adjustments we are able to obtain a very good correlation between measurements and equation as seen in Figure 4-14. Some small mismatch is seen at lower frequencies, but as mentioned before those are most likely due to measurement inaccuracies (such as calibration errors).

$$\begin{gathered}\mathrm{val\_dc\_S12} = -0.0945\,\mathrm{dB}\\ \mathrm{freq\_dc\_S12} = 0\\ \mathrm{coef\_dc\_S11} = -0.0945\\ \mathrm{val\_lin\_S12} = -10.98\\ \mathrm{freq\_lin\_S12} = 10\mathrm{e}9\,\mathrm{Hz}\\ \mathrm{coef\_lin\_S12} = -10.98 * (\mathrm{freq}/10\mathrm{e}9)\\ \mathrm{val\_log\_S12} = -0.6\,\mathrm{dB}\\ \mathrm{freq\_log\_S12} = 5\mathrm{e}7\,\mathrm{Hz}\\ \mathrm{coef\_log\_S12} = \mathrm{val\_log\_S12} * (\log(\mathrm{freq}/\mathrm{freq\_log\_s12}) + 1) = -0.6 * (\log(\mathrm{freq}/5\mathrm{e}7) + 1)\end{gathered}$$

Equation 4-16

By looking at the above coefficients we can notice that the initial linear coefficient had to be (minor) adjusted to compensate for the added logarithmic terms. Also, a small number has been added to the coefficient of the logarithmic function to avoid evaluating it at zero (log function is not defined there).

Since we are finally interested in transient results for our specific system, a convolution between impulse and pulse responses will be performed. The same linear driver (as in the previous paragraph) will be used.

Impulse response of the FFT transform, for both cases (measured and equation-based s-parameters) is plotted in Figure 4-15.

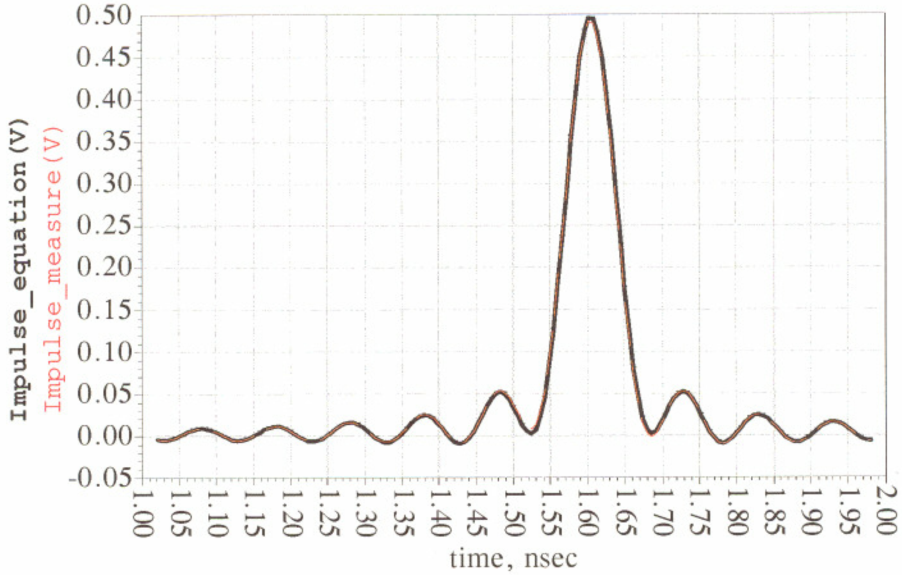


Figure 4-15 Impulse response correlation using measured and mixed (linear and logarithmic) approximated s-parameters

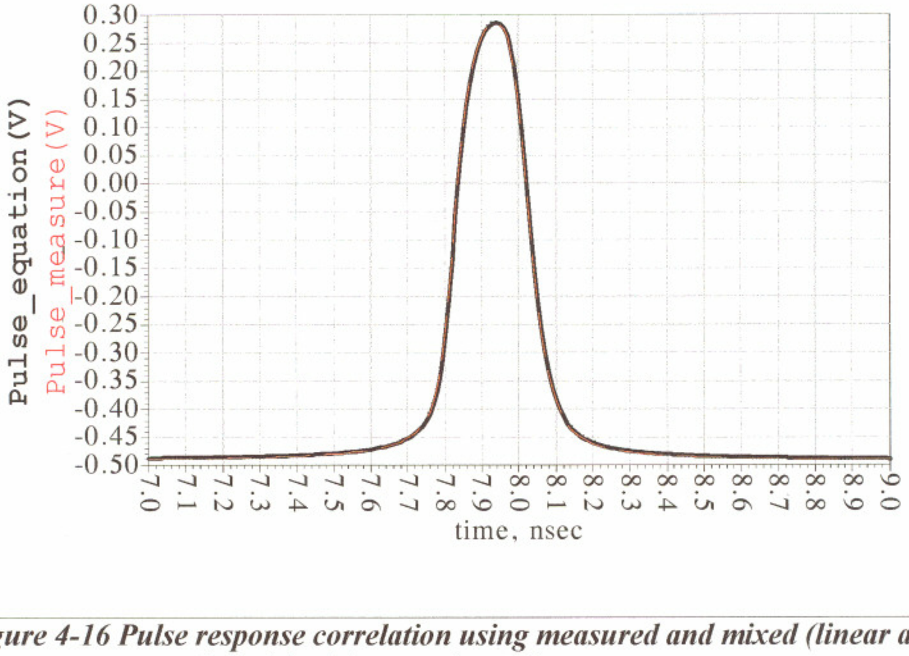


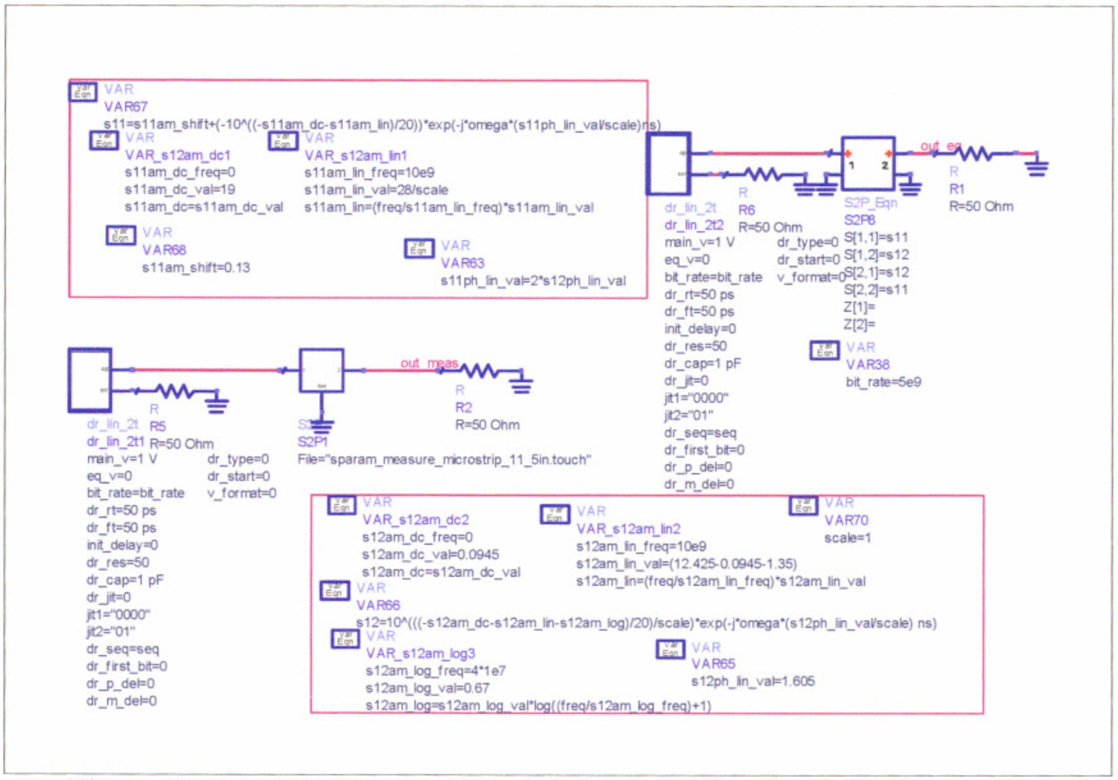
Figure 4-16 Pulse response correlation using measured and mixed (linear and logarithmic) approximated s-parameters

4.4.2 Return loss

As mentioned before, return loss is several orders of magnitude smaller than the insertion loss. We have also seen in paragraph 4.4.2 that we have got a good approximation using only linear coefficients; therefore no additional improvements should be necessary for the return loss.

4.4.3 Time domain correlation – logarithmic approximation

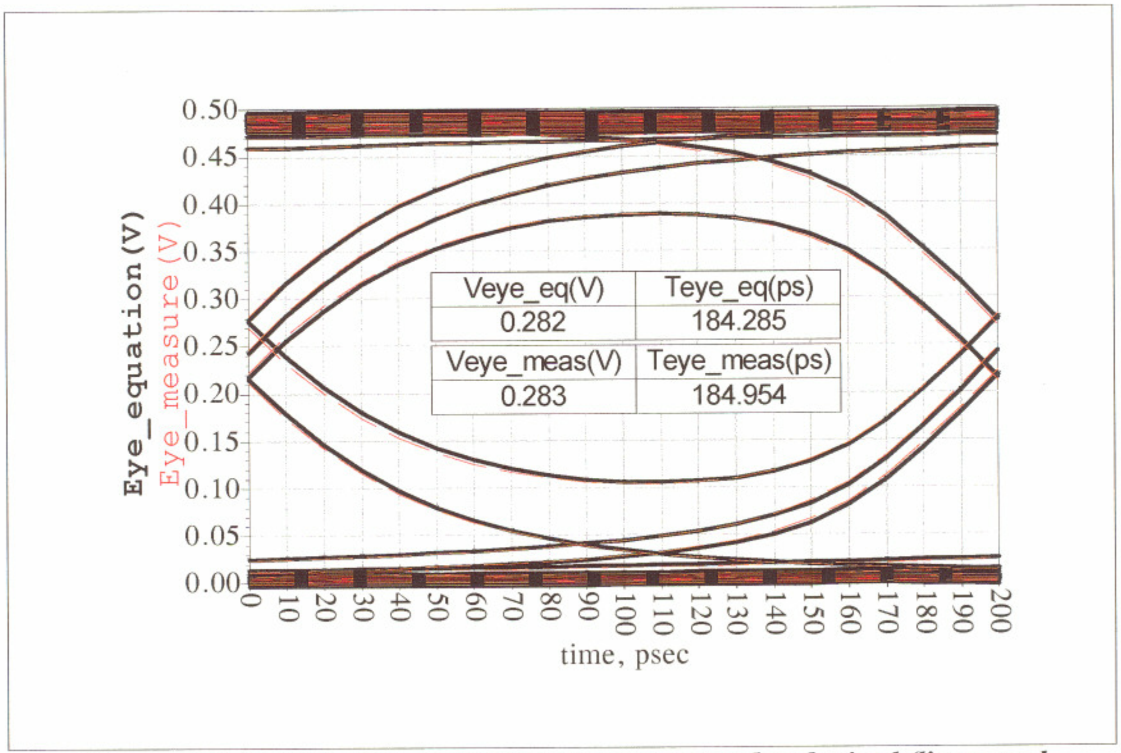
Based on the new formula for the insertions (S12) and using the same linear approximation for return loss (S11) we can now determine the pulse response for the system using the same driver as before (Figure 4-11). Based on the degree of fit for both s-parameters we can expect a good correlation in time domain. It can also be noted that theoretically the pulse response is the only necessary element to completely characterize a linear system, so this will fully quantify the validity of the approximations used. The time domain correlation using the pulse response of the system is shown in Figure 4-16. Final simulator settings are shown for reference in Figure 4-17.



```
VAR
VAR67
s11=s11am_shift+(-10^((-s11am_dc-s11am_lin)/20))*exp(-j*omega*(s11ph_lin_val/scale)ns)

VAR
VAR_s12am_dc1
s11am_dc_freq=0
s11am_dc_val=19
s11am_dc=s11am_dc_val

VAR
VAR_s12am_lin1
s11am_lin_freq=10e9
s11am_lin_val=28/scale
s11am_lin=(freq/s11am_lin_freq)*s11am_lin_val

VAR
VAR68
s11am_shift=0.13

VAR
VAR63
s11ph_lin_val=2*s12ph_lin_val

dr_lin_2t
dr_lin_2t2
main_v=1 V
eq_v=0
bit_rate=bit_rate
dr_rt=50 ps
dr_ft=50 ps
init_delay=0
dr_res=50
dr_cap=1 pF
dr_jit=0
jit1="0000"
jit2="01"
dr_seq=seq
dr_first_bit=0
dr_p_del=0
dr_m_del=0
dr_type=0
dr_start=0
v_format=0

R
R6
R=50 Ohm

S2P_Eqn
S2P8
S[1,1]=s11
S[1,2]=s12
S[2,1]=s12
S[2,2]=s11
Z[1]=
Z[2]=

R
R1
R=50 Ohm

VAR
VAR38
bit_rate=5e9

dr_lin_2t
dr_lin_2t1
main_v=1 V
eq_v=0
bit_rate=bit_rate
dr_rt=50 ps
dr_ft=50 ps
init_delay=0
dr_res=50
dr_cap=1 pF
dr_jit=0
jit1="0000"
jit2="01"
dr_seq=seq
dr_first_bit=0
dr_p_del=0
dr_m_del=0
dr_type=0
dr_start=0
v_format=0

R
R5
R=50 Ohm

S2
S2P1
File="sparam_measure_microstrip_11_5in.touch"

R
R2
R=50 Ohm

VAR
VAR_s12am_dc2
s12am_dc_freq=0
s12am_dc_val=0.0945
s12am_dc=s12am_dc_val

VAR
VAR_s12am_lin2
s12am_lin_freq=10e9
s12am_lin_val=(12.425-0.0945-1.35)
s12am_lin=(freq/s12am_lin_freq)*s12am_lin_val

VAR
VAR70
scale=1

VAR
VAR66
s12=10^(((-s12am_dc-s12am_lin-s12am_log)/20)/scale)*exp(-j*omega*(s12ph_lin_val/scale) ns)

VAR
VAR_s12am_log3
s12am_log_freq=4*1e7
s12am_log_val=0.67
s12am_log=s12am_log_val*log((freq/s12am_log_freq)+1)

VAR
VAR65
s12ph_lin_val=1.605
```

Figure 4-17 Mixed (linear and logarithmic) approximation –simulator settings

We can see that we have got a very good (close to perfect) correlation. This actually means that our equation based model is going to perform in both time and frequency domain in a similar manner as the measured transmission line. The Eye Diagram of the system with the worst case pattern is shown in Figure 4-18. As proven in paragraph 4.3.4, for a linear system the EDA approach leads equivalent results as the PDA method. As we can see in this case the total worst case error is almost undetectable.



Veye_eq(V)	Teye_eq(ps)
0.282	184.285
Veye_meas(V)	Teye_meas(ps)
0.283	184.954

Figure 4-18 Eye Diagram correlation using measured and mixed (linear and logarithmic) approximated s-parameters - 11.5 inch microstrip trace

We have been able to show that we can achieve close to perfect correlations between any well behaved transmission line and its corresponding mathematical approximation. In this case the closest approximation has been obtained using a first order linear and a logarithmic function.

Since this is a non-linear approximation based on measurements it is generally more accurate than most library models existent in current commercial simulators.

If desired, or if system simulators do not support logarithmic functions, a Taylor Series expansion can be used instead:

$$\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots$$

Equation 4-17

Equations can be also expanded to include higher order polynomials, if measured data is not well behaved (non uniform trends).

4.5 Scaling

Since the approximation was based on measured data, a very useful feature would be to be able to extrapolate the function to different trace lengths.

This can be achieved with a scaling factor "scale". The factor would represent the ratio between the initial lengths, which were 11.5in (based on measurement) and the desired new length. As an example we have used a "scale=1.9166" to validate this option against the other measured trace, the 6in microstrip line. In this case the scaling factor would be a number under unity for lengths grater than our initial measurements (11.5in) and a number greater then unity for lengths smaller that that.

Figure 4-19 is showing the magnitude (dB) for the insertion loss.

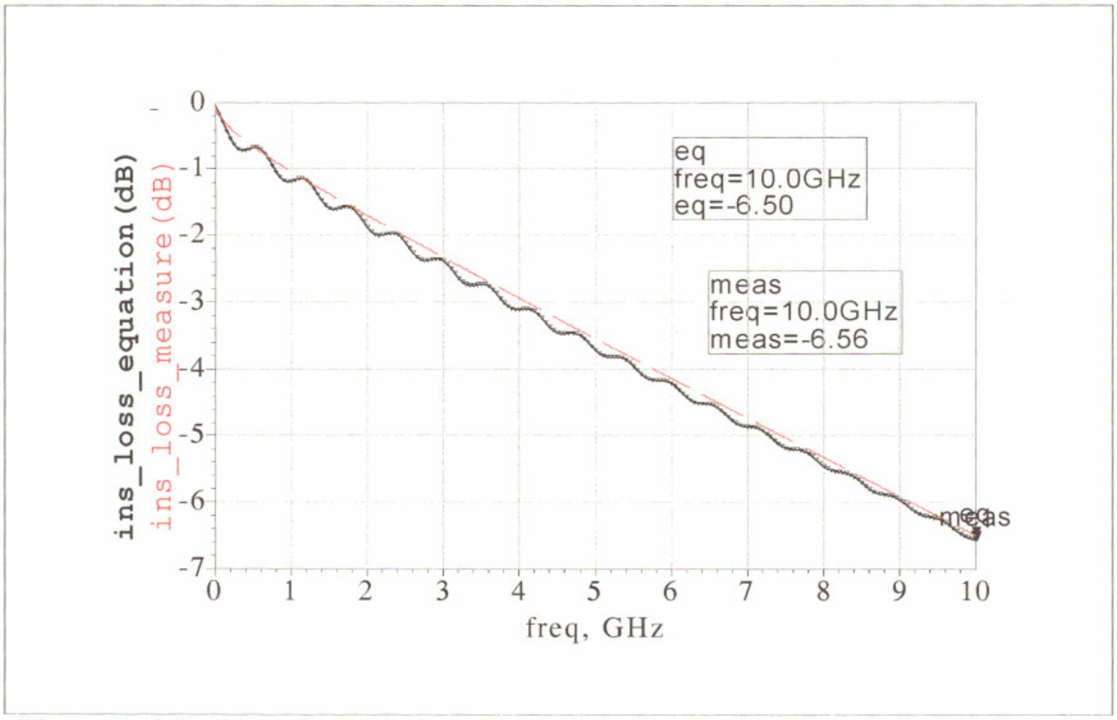
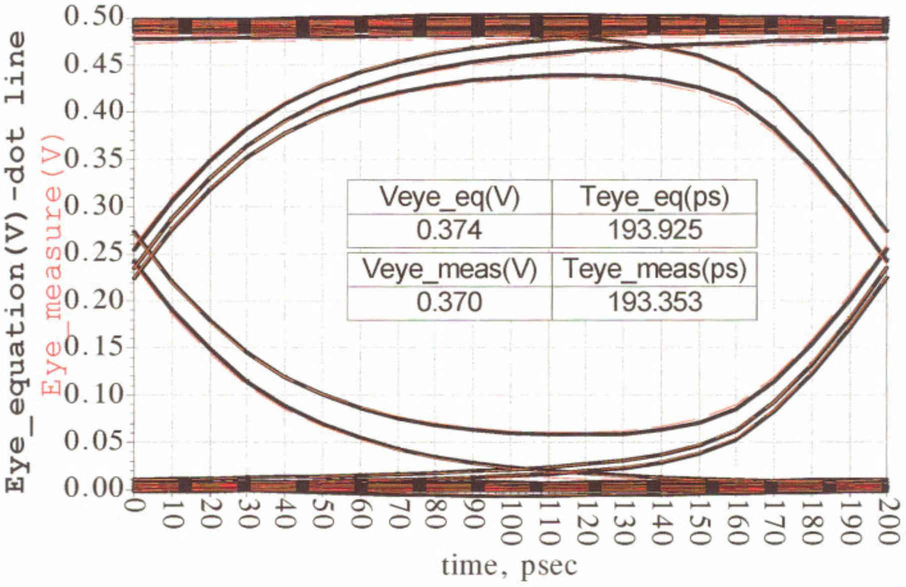


Figure 4-19 Scaling - Insertion loss correlation using measured and mixed (linear and logarithmic) approximated s-parameters – 6 inch microstrip trace

Again to validate the approximation in time domain, a pulse response is extracted for both measured and equation based data.

The results are calculated using the EDA tool and are shown in Figure 4-20.

Based on the EDA reading, we can see that even with scaling, the voltage error is 3mV (around 1%), which is negligible. Timing error is less than 1ps.



Veye_eq(V)	Teye_eq(ps)
0.374	193.925
Veye_meas(V)	Teye_meas(ps)
0.370	193.353

Figure 4-20 Scaling - Eye Diagram correlation using measured and mixed (linear and logarithmic) approximated s-parameters – 6 inch microstrip trace

4.6 Conclusions

We have been able to create a very good frequency domain approximation for a transmission line model, based on measured data. This model can be used (depending on simulator capabilities) in both time and frequency domain simulations with little or no error, since some of the error seen above can be easily assigned to measurement errors.

The other important issue to consider is simulation time. Since this approximation used a very simple equation, we will expect to perform as well or better than other models. As a matter of fact, ADS report of simulation time show a significant improvement of 25% (67seconds vs. 90seconds) even for a simple simulation structure, when compared to s-parameter data or library elements.

Chapter 5

Driver optimization using the Eye Diagram Analyzer

5.1 Problem statement

As mentioned in the introductory chapters, driver and receiver optimization used to be performed in isolation, in other words, they, and eventually the associated package, used to be measured into a standard load, usually a 50 Ohm resistor. This procedure is no longer adequate for high speed signaling. Driver and receiver characteristics such as driver/receiver de-emphasis, driver duty cycle distortion, overall jitter distribution, total driver/receiver capacitance, are strongly interacting with the interconnect.

Therefore, modern high speed specifications would have to start to require I/Os to be characterized into distributed loads, such as transmission lines. In order to do that, idealized models based on realistic (worst case) interconnects have to be used. One such model was described in Chapter 4. Since that model is independent of the simulators used, it can bridge the boundary between silicon and interconnect simulation packages, and allow a good I/O characterization. This chapter will describe some advantages that can be gained when using that model in conjunction with the EDA tool. Furthermore the driver optimization techniques described in the chapter can only be performed using distributed standards, such as the transmission line model derived in previous chapters, making traditional driver characterization not only obsolete, but in fact meaningless.

5.1.1 Methodology assessments - differences between EDA and PDA

Before we go into details on how to optimize drivers using EDA, it is useful to compare it with some of the other available methods. The first one that has been referenced throughout this thesis is the traditional brute force approach. This is the one that has the most disadvantages, because it requires a large amount of simulation and characterization in order to fully characterize the system. However, even this type of approach can be improved by using the EDA tool, because it provides for more accurate Eye Diagram measurements.

To fully benefit from EDA capabilities statistical analysis methods such as Monte Carlo (available in most simulation tools such as ADS or MATLAB), combined with real time optimization criteria based on eye measurements can be used.

To speed-up transient simulations, as well as determine the worst case data pattern, a PDA approach can be used with the EDA instead of the full random data pattern analysis. While this method had several obvious advantages, the main issue with PDA methodology is the fact that it fundamentally requires a linear system to operate. Any system non-linearity will make the method invalid. Therefore non-linear driver and receivers can not be used in conjunction with PDA. Unfortunately, today, most silicon devices and even interconnects are non-linear. Even filtering techniques, such as the FIR filter described in the introductory chapter, which itself is linear, will add non-linearity due to driver finite rise/fall times that differ between equalized and non-equalized transitions (see Figure 5-1), as well as non-linear parasitics interactions.

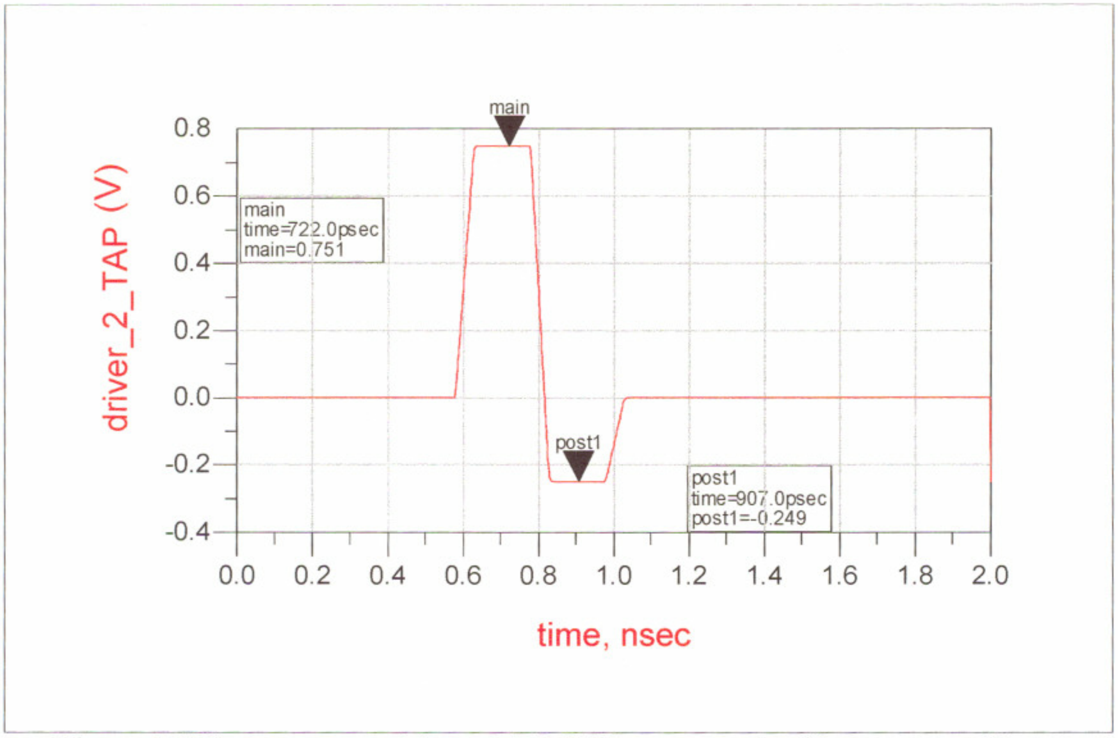


Figure 5-1 Linear current driver with 2-TAP nonrecursive filter measured into a standard 50 Ohm load

The second main issue with PDA is the inability to specify initial conditions for the system. Usually any encoding or DC balancing algorithm will invalidate PDA assumptions of initial conditions.

The third issue with using PDA for nonrecursive TAP optimization is that it actually requires a separate pulse response simulation for the channel, each time the TAP coefficients are changed, which ends up making this method having similar time and resource consuming issues as traditional methods.

The forth issue with PDA, will be described in more detail in paragraph 6.1.2 and is related to the fact that PDA does not have the notion of patterns, therefore it can not track pattern dependent jitter and cannot be used when clock recovery algorithms are analyzed for the receiver.

5.2 Driver optimization using the Eye Diagram Analyzer

End result of driver optimization can be described as a maximization of an Eye Diagram opening as perceived at the receiving end. From a designer point of view the maximization can be either at the actual receiver or into a standard load specification. It can also be either a time or a voltage eye opening or a linear combination of the two. It can be performed either at the eye center or at any point along the eye. The flexibility of the EDA tool allows real time optimizations based on all of the above criteria.

5.2.1 Voltage and ISI optimization using EDA and FIR driver filters

One of the most efficient ways to maximize driver output in high frequency design is by means of filtering. Driver and receiver filters are trying to match the loss characteristics of the distributed loads in which they operate therefore cannot be determined in isolation. The simplest filters are the one that do not depend on history, the FIR filters. One such filter is only dependent on the number of (nonrecursive) filter taps used, and the amplitude of the TAP coefficients (as described in Figure 2-2).

Since both TAP coefficients as well as number of TAPs are strongly dependent on the loss characteristics of the system, determining the correct values has to be done into an environment that closely resembles the system operating conditions. Again using EDA and the equation-based distributed model can be a good choice. Such optimization method is described in this section.

Initial optimization will consider the system without any equalization and compare that with the simplest nonrecursive filter Figure 5-1, only a 2-TAP filter.

A second optimization will compare the optimum-derived 2-TAP driver with a more complex multi-TAP filter.

Since the frequency dependence of the distributed load used in this example (based on the measured single transmission line) is fairly linear, we expect the 2-TAP filter to be quite effective; however, this is not always the case in more complex systems, therefore multiple TAP filters have to be evaluated.

All drivers are linear, and assumed to have 1pF of total capacitance, and will be driven into the channel described in the previous chapter (Chapter 4).

Initial TAP values can be identified on the markers in Figure 5-1 and Figure 5-5 in absolute values and can also be found in Table 5-5 expressed in relative value (dB) with respect to the main TAP value. Notice that we can also have positive compensation as seen in pre2 TAP (+1).

It should be mentioned that this method has none of the PDA limitations described before, so non-linear drivers can be substituted if necessary. Although a non-linear receiver can be connected at the end of the channel, this optimization will be performed into a standard 50 Ohm load, in order to be comparable with an oscilloscope measurement performed on the same channel.

There are multiple ways available to perform the optimization. Some of the most common ones are comprehensive parameter sweeps or using various statistical approaches like Monte Carlo or Design of Experiments (DOE) [18].

Table 5-5 Initial driver TAP coefficients for 6 and 2 tap FIR filter

NR of TAPS	*TAP1 (pre2) value (dB*	*TAP2 (pre1) value (dB)*	*TAP4 (post1) value (dB)*	*TAP5 (post1) value (dB*	*TAP6 (post1) value (dB*
6	+1	-2	-4	-3	-2
2			-6		

For the 2 TAP analysis a fine sweep for the TAP coefficients, has been chosen while for the multi-tap, a DOE approach has been used.

Some of the modern analog simulators (ADS included) offer some limited statistical packages to design DOE experiments, however for this study a commercial specialized software package –JMP [19] has been used. Similar experiments can also be developed using the publicly available statistical package "R". "R" is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form.

Because, as mentioned before, the system behaves fairly linear, a linear statistical interpolation based on the DOE will be acceptable. Also, since we are going to use an 8bit10 encoded data pattern, the ISI will furthermore be limited, so residual pulse response will not be able to add in an unfavorable way indefinitely, but be constrained by the pattern run length defined in the code [17]

If a more exact solution is desirable the DOE can be supplemented with actual simulations (optimizations or sweeps) around DOE predictions.

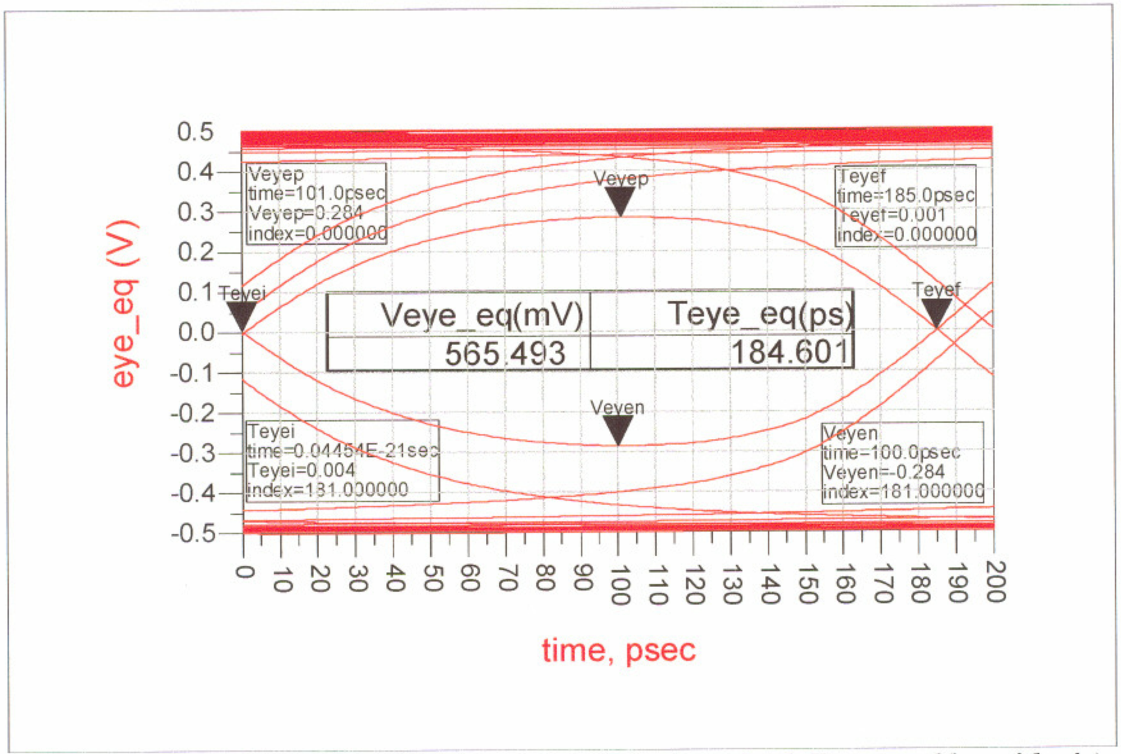


Figure 5-2 Eye Diagram Analyzer measurement using mixed (linear and logarithmic) approximated s-parameters (11.5 inch) and differential driver with no filtering

The initial EDA measurement for the full system, without any filtering is indicated in Figure 5-2. The voltage number in the center of the Eye Diagram (which is double from the number seen in Figure 4-18 because of the differential signaling used) will be used as a relative reference to determine the efficacy of the nonrecursive filter on the voltage margins. Similar approaches can be envisioned if timing margins or a linear combination of timing and voltage margins are desired.

The 2-TAP nonrecursive coefficient sweep has been performed from 0dB to 9dB in 0.5 dB increments. It is to be noted that most HVM designs have at least a (+/-) 0.5 dB tolerance in the driver filtering mechanism, so this amount of granularity is sufficient. For comparative purposes, results are scaled to the 1TAP (no equalization) value from Figure 5-2. Sweep results (plotted in Figure 5-3) are indicating that the optimal value for this transmission line is around 2.5dB.

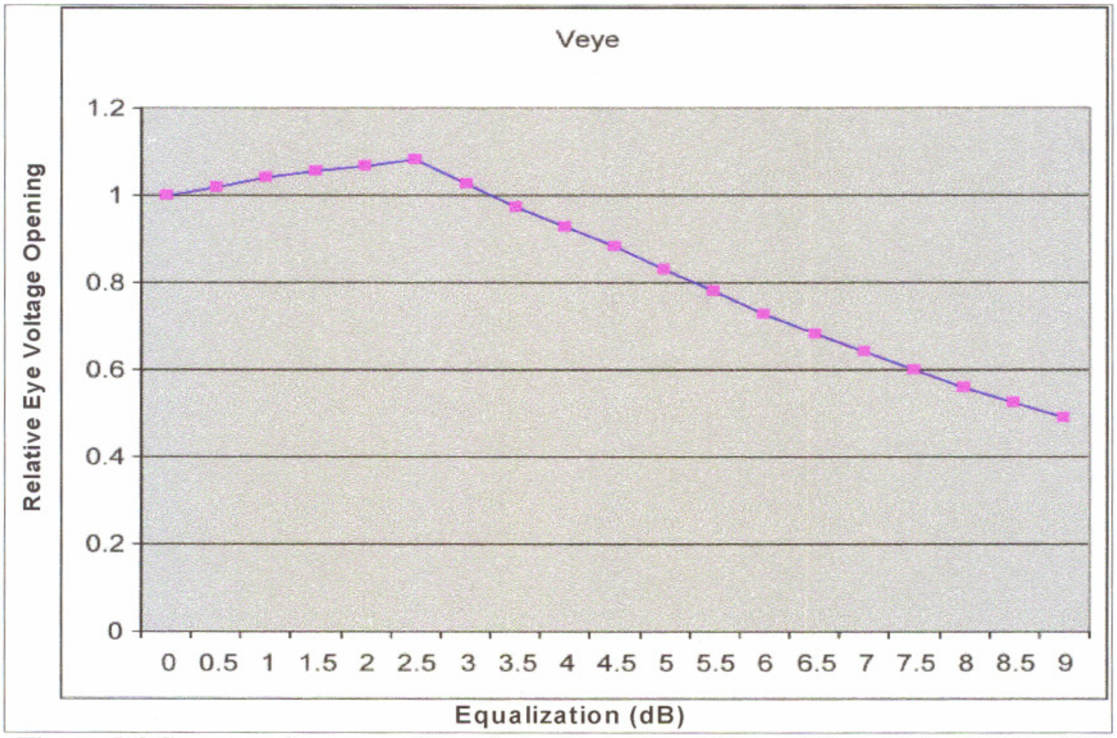


Figure 5-3 System voltage margin optimization using a linear driver with a 2-TAP FIR

Based on the optimal value identified, a system simulation is performed and EDA reading from the optimized driver is shown in Figure 5-4. We can also see that beside the voltage gain, optimizing the nonrecursive filter coefficients gives us also a significant jitter component reduction from 15ps to 5 ps. This is in accordance with the effects

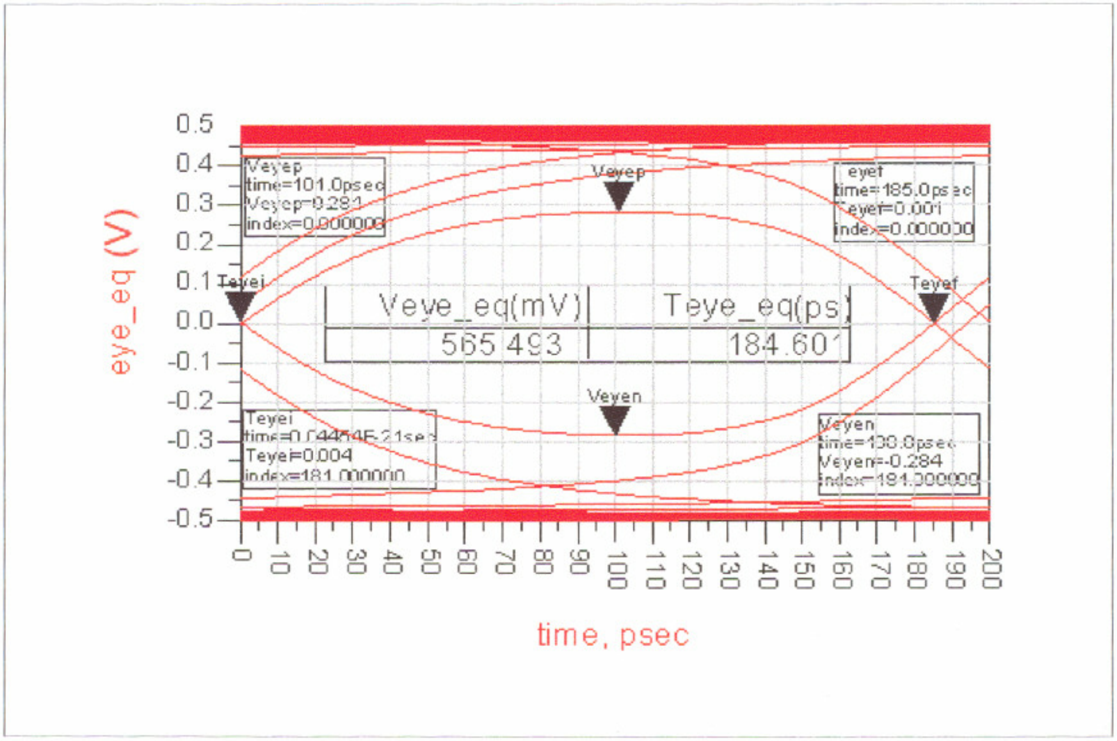


Figure 5-4 EDA measurement using mixed (linear and logarithmic) approximated s-parameters (11.5 inch) and DOE optimized 2.5dB 2-TAP FIR differential driver

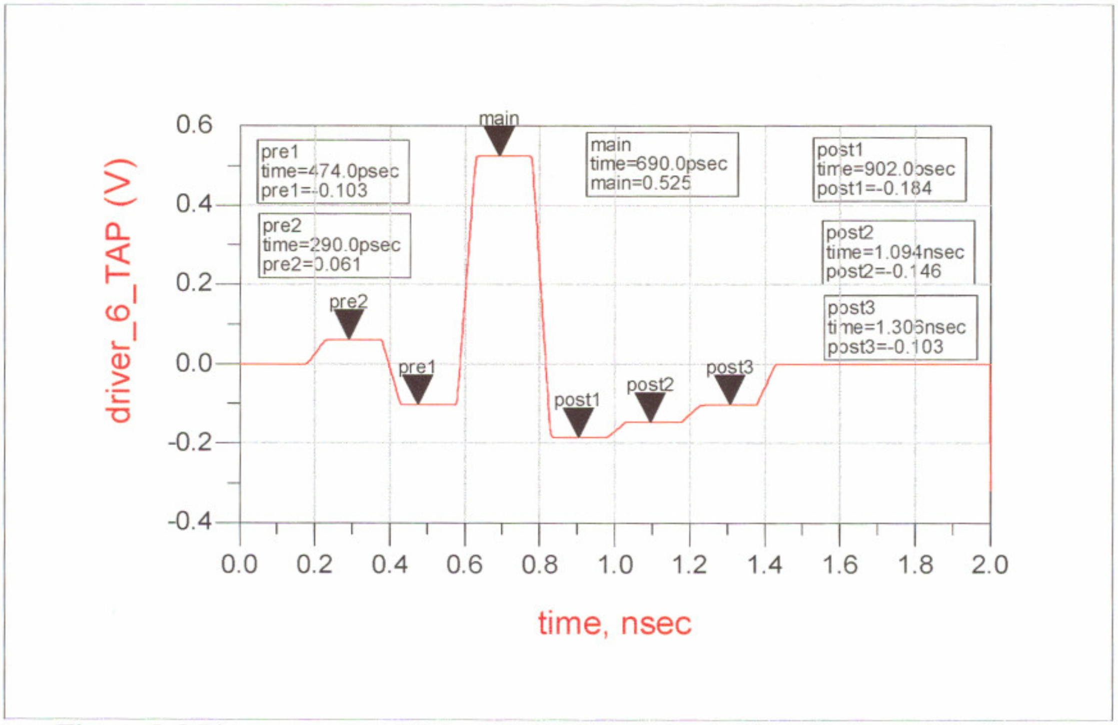


Figure 5-5 Linear current driver with 6-TAP nonrecursive filter measured into a standard 50 Ohm load

described in paragraph 2.2.1, and it can be intuitively interpreted as minimizing the energy that spreads out into adjacent pulses. As stated before, the 2-TAP filter will be evaluated against a more complex multiple TAP FIR.

For this comparison we will use a 6-TAP nonrecursive filter (Figure 5-5), with the initial coefficients visually determined by looking at the non-filtered pulse response in Figure 4-16.

Table 5-6 System voltage margin optimization DOE experiment using a 6-TAP FIR linear driver

index	POST1	POST2	POST3	PRE1	PRE2	RESULTS(Veye)
0	3	-2	-1	-2	-1	0.9777
1	3	-2	-1	-2	1	0.7405
2	3	-2	-1	2	-1	1.0509
3	3	-2	-1	2	1	0.7984
4	3	-2	1	-2	-1	0.7439
5	3	-2	1	-2	1	0.8653
6	3	-2	1	2	-1	0.8856
7	3	-2	1	2	1	0.8224
8	3	2	-1	-2	-1	0.7865
9	3	2	-1	-2	1	0.7222
10	3	2	-1	2	-1	0.7415
11	3	2	-1	2	1	0.7492
12	3	2	1	-2	-1	0.6337
13	3	2	1	-2	1	0.4929
14	3	2	1	2	-1	0.5458
15	3	2	1	2	1	0.3546
16	4.5	0	0	0	0	0.8815
17	6	-2	-1	-2	-1	0.9051
18	6	-2	-1	-2	1	0.7170
19	6	-2	-1	2	-1	0.9032
20	6	-2	-1	2	1	0.5910
21	6	-2	1	-2	-1	0.6610
22	6	-2	1	-2	1	0.7096
23	6	-2	1	2	-1	0.6242
24	6	-2	1	2	1	0.5465
25	6	2	-1	-2	-1	0.5010
26	6	2	-1	-2	1	0.5176
27	6	2	-1	2	-1	0.4115
28	6	2	-1	2	1	0.3938
29	6	2	1	-2	-1	0.3187
30	6	2	1	-2	1	0.1794
31	6	2	1	2	-1	0.0002
32	6	2	1	2	1	0.0000

The visualization is also helpful in determining initial parametric variation for the coefficients. It is to be noted that even this visualization is a useful approach that can save some computational time, it is not a necessary approach, and the optimization can be started with arbitrary values (for example zero).Due to the relative small number of coefficients to be optimized (5) and also because the linearity of the system, a full factorial DOE experiment has been chosen. The full factorial DOE based on five variables results into 32 (2^5) experiments. The full factorial DOE used minimum, maximum (corner points) and typical values (center point), so the DOE has been designed around the "expected" initial values indicated in Table 5-5. The actual DOE setup together with the EDA results (voltage in the eye center), based on ASD simulations is shown in Table 5-6. Based on the above results, JMP can interpolate the full surface response Figure 5-6):

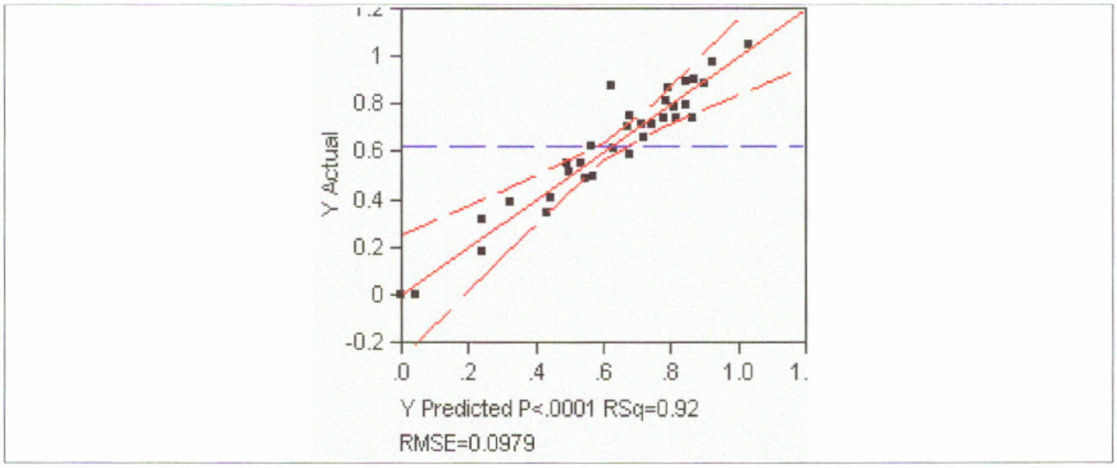


Figure 5-6 JMP full surface response and error - 6-TAP FIR linear driver DOE

We can see the prediction error RSquare (RSq) is around 9%, which is well into the HVM tolerance of (+/-) 0.5 dB for TAP coefficients.

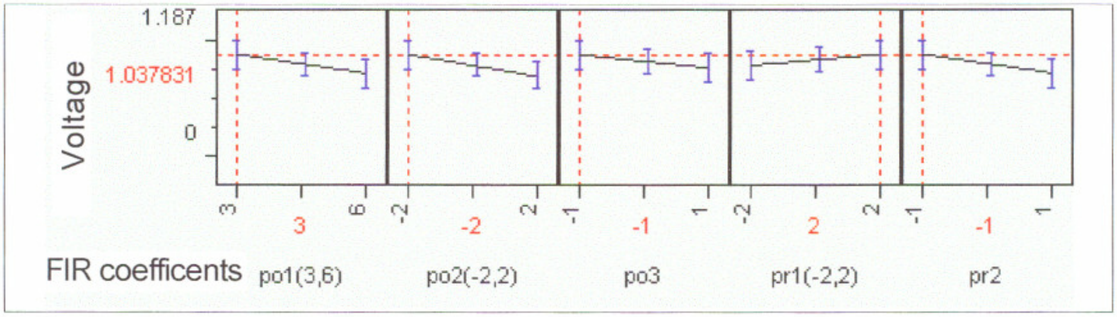


Figure 5-7 TAP coefficients adjusted to maximize the DOE response (voltage margin) using JMP graphical interface

We can also see that there is an outlier, that has been identified as the experiment number (32) which, according to Table 5-6 gave us no eye opening (0 mV eye center). If a more precise DOE is desired, the experiment can be designed (based on the previous results) with new, more restrictive ranges for the TAP coefficients. The graphical representation of the response, with the TAP coefficients adjusted to maximize the DOE response are provided in Figure 5-7.Based on the DOE results we can determine that the optimal statistical TAP values are:

$$\begin{aligned}
&post1 = 3\\
&post2 = -2\\
&post3 = -1\\
&pre1 = 2\\
&pre2 = -1
\end{aligned}$$

Equation 5-1

A numerical calculation of the voltage margin based on the DOE experiment (with the graphical representation seen in Figure 5-7) can also be established and is provided for reference in Equation 5-2:

$$\mathbf{Veye}=0.6293 -0.122 * ((\mathbf{post1} - 4.5) / 1.5) -0.162 * (\mathbf{post2} / 2) -0.097 * \mathbf{post3} -0.033 * (\mathbf{pre1} / 2) -0.046 * \mathbf{pre2} + ((\mathbf{post1} - 4.5) / 1.5) * ((\mathbf{post2} / 2) * (-0.045)) - ((\mathbf{post1} - 4.5) / 1.5) * (\mathbf{post3} * 0.020) - (\mathbf{post2} / 2) * (\mathbf{post3} * 0.045) - ((\mathbf{post1} - 4.5) / 1.5) * ((\mathbf{pre1} / 2) *0.032) - (\mathbf{post2} / 2) * ((\mathbf{pre1} / 2) * 0.026) - \mathbf{post3} * ((\mathbf{pre1} / 2) * 0.017) + ((\mathbf{post1} - 4.5) / 1.5) * (\mathbf{pre2} * 0.005) + (\mathbf{post2} / 2) * (\mathbf{pre2} * 0.012) + \mathbf{post3} * (\mathbf{pre2} * 0.019) - (\mathbf{pre1} / 2) * (\mathbf{pre2} * 0.010)$$

Equation 5-2

By comparing the simulation results from Table 5-6, experiment (1) with the DOE predictions from Figure 5-7, we can see that the actual DOE error for the optimal value is around 2%, much less than the maximum DOE error for the whole surface (RSquare) of 9%. Also we can see that, as expected, because of the shape of the pulse response, there are no benefits in using multiple TAPS on this system.

This is an important conclusion, since having just 1TAP will greatly simplify driver design, reduce die area and driver capacitance.

5.2.2 Duty Cycle Distortion (jitter amplification) using EDA

As mentioned in the introductory chapter, DOD can create jitter amplification when interacting with the system. This not only results in a larger amount of jitter at the output compared with what has been introduced at the driver (input), but also in a lower voltage at the receiver.

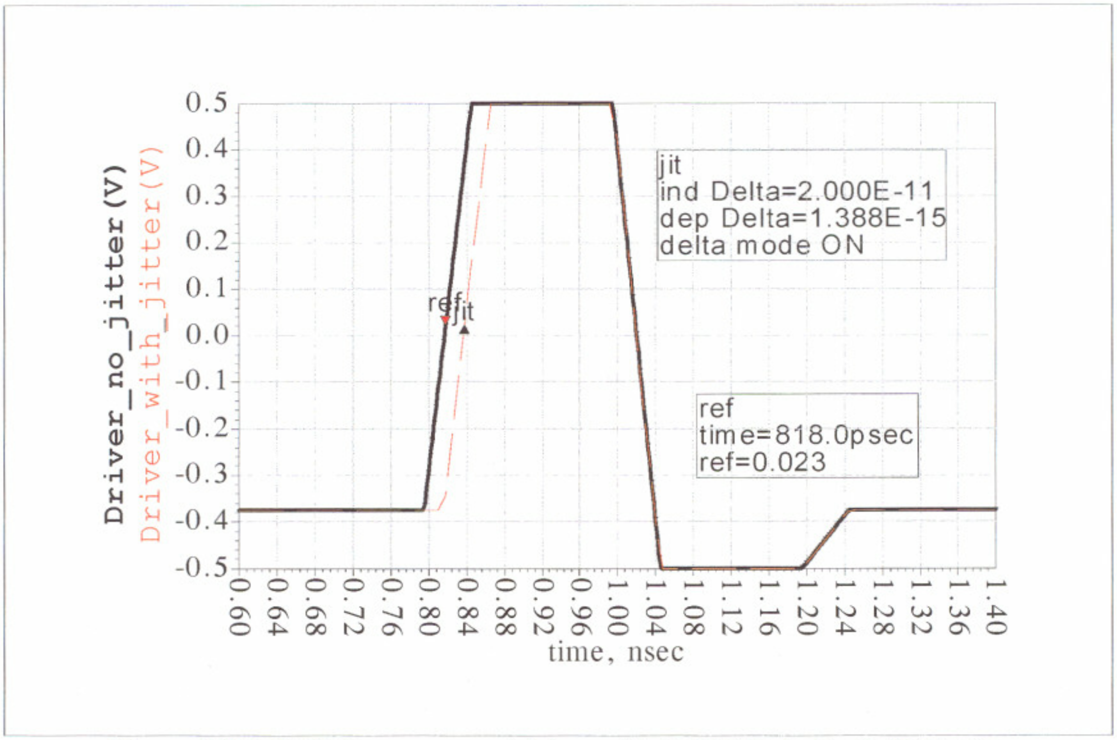
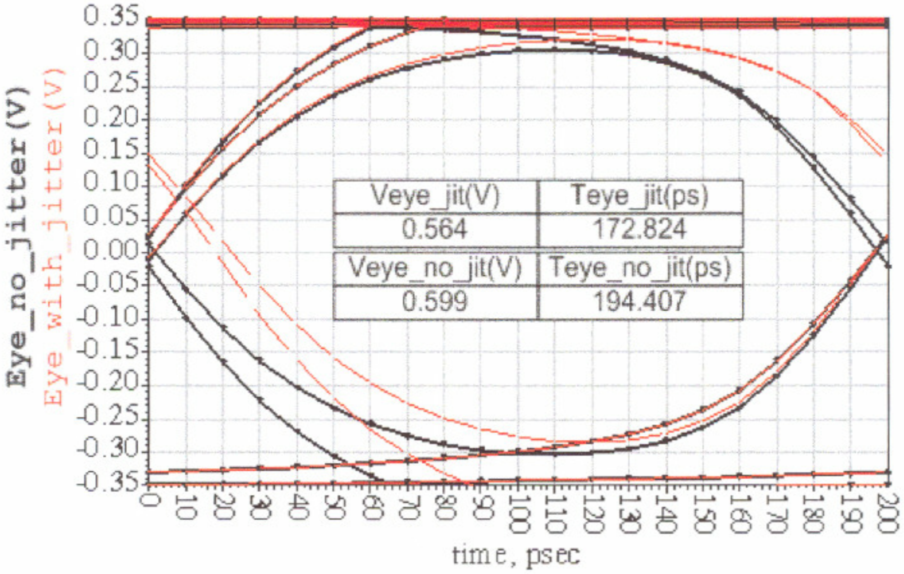


Figure 5-8 2-TAP linear current driver with 0.1UI Duty Cycle Distortion

To quantify this effect a reasonable amount of duty cycle distortion (0.1 UI) is introduced at the driver. If characterized into a 50 Ohm load, as seen in Figure 5-8, there is no difference in driver output (voltage/current and time), so no jitter amplification is observed. However when characterized into a distributed load, such as the transmission line model developed in previous chapters, we are seeing different results.

Using the same method as in previous simulations (EDA measurement of a pulse response) and the same nonrecursive filter, we compare the results of the two cases described before, one in which we have no duty cycle distortion at the drive, to the one in which we have introduced jitter (Figure 5-9).

We have inserted 20ps at the driver, and as a consequence the receiver eye has been reduced with about 22ps. More significant, the voltage in the center of the eye has been reduced form 600mV to 564 mV which is a 6% reduction in the pulse response of the system.



Veye_jit(V)	Teye_jit(ps)
0.564	172.824
Veye_no_jit(V)	Teye_no_jit(ps)
0.599	194.407

Figure 5-9 Receiver margin reduction based on 0.1UI (20 ps) driver duty cycle distortion

This jitter amplification phenomenon is another factor that shows the close interaction between I/O and interconnects, by highlighting significant voltage differences that cannot be detected if traditional driver characterization is performed (driver simulated into a lumped load and not into a distributed one).

Chapter 6

Receiver optimization using the Eye Diagram Analyzer

6.1.1 Receiver impedance optimization

As described in paragraph 2.4, there are several receiver optimizations that can be achieved using the EDA. The first one is to characterize total receiver impedance, so that an optimization based on maximum voltage and timing margins based on the Eye Diagram are achieved.

The basic transmission line theory suggests the fact that total receiver impedance has to be matched as closely as possible with the system characteristic impedance. As noted in the background chapter (Chapter 2) this is impossible to achieve even from a theoretical point of view. This is because the characteristic impedance of a transmission line stays fairly constant over frequency, while the total receiver impedance is strongly dependent on frequency. Some preliminary hand calculations can be made (to establish an initial target) by matching total impedance at a certain frequency. The parallel impedance formula (assuming a negligible reactance) is:

$$Z_{\mathrm{Re\,ceiver}} = (\frac{R}{(1 + j * \omega * R * C})$$

Equation 6-1

This formula will always lead to complex impedance, but since the transmission line impedance is a real value (or with negligible imaginary contribution) we are only concerned with the real part.

Plotting the results from the above formula for some typical receiver resistance and capacitance value will lead to Figure 6-1.

The graph indicated that we can only achieve 50 Ohm (targeted transmission line characteristic impedance) with a capacitance that is less than 0.7 pF, so the optimization will certainly lead us to a capacitance that is lower than this value. We can also see that the lower the capacitance the better, since we can achieve our target over a wider frequency range, therefore actual optimizations can be done without capacitance variation, just using the worst case design target value.

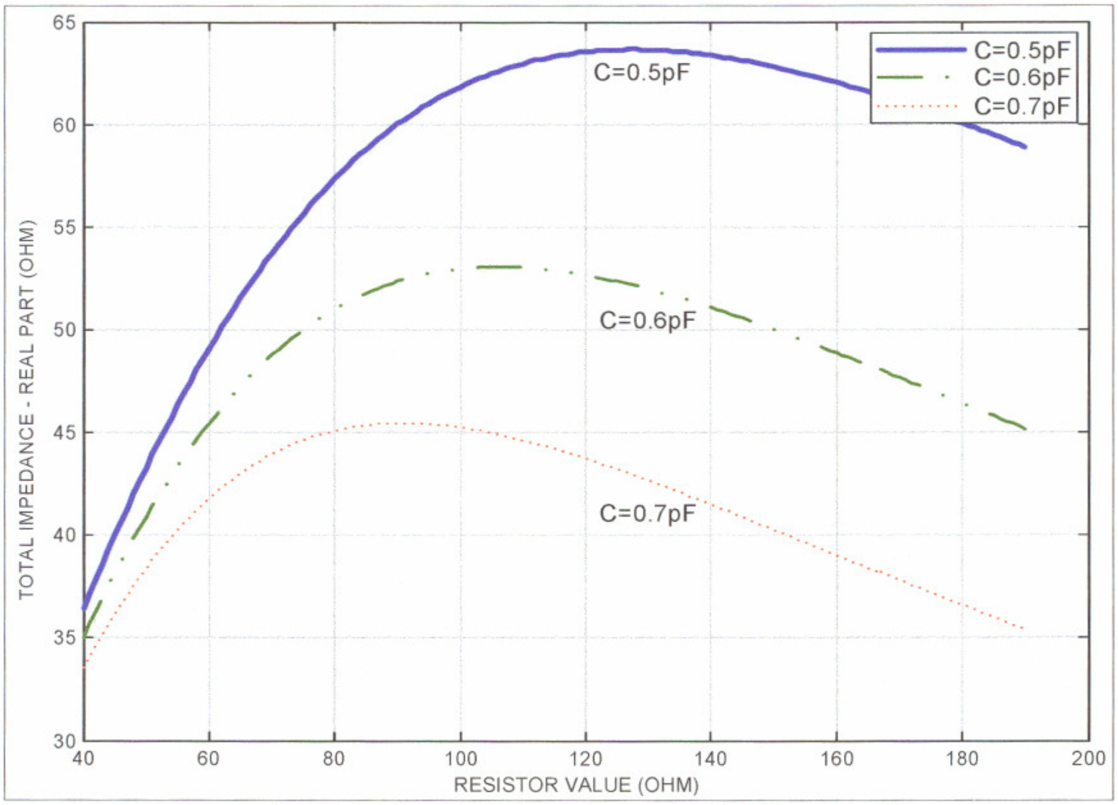


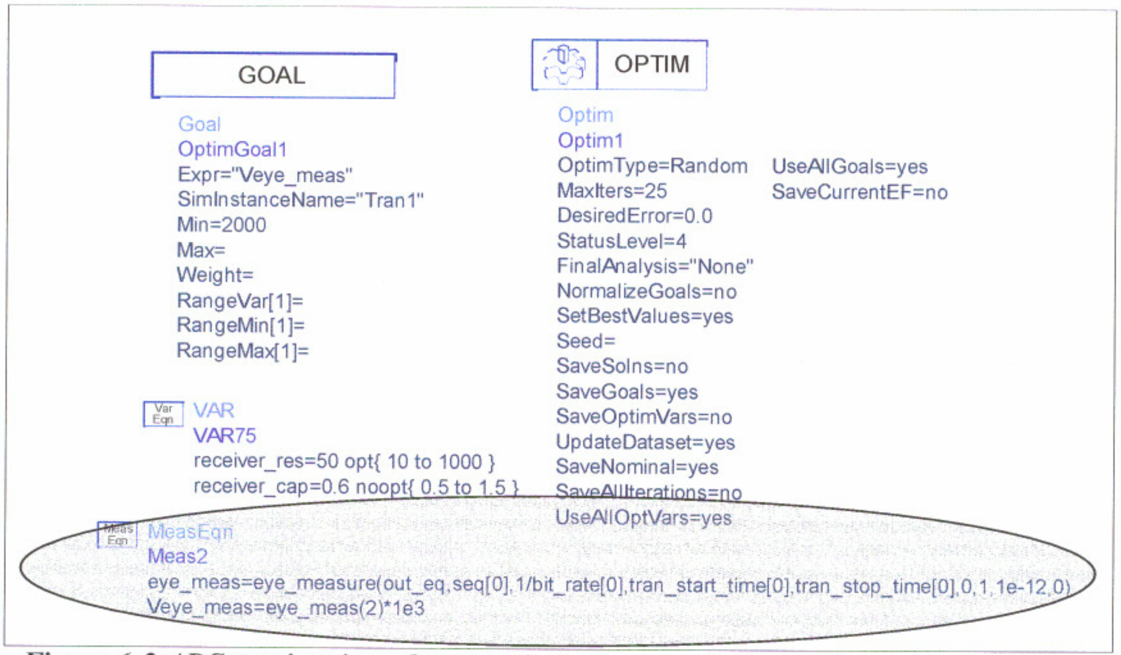
Figure 6-1 Total Receiver Impedance at 2.5 GHz – Real Part

The other thing that we have to comprehend is the fact that this formula is valid at only one frequency, so the actual optimal value can only be achieved by full system simulation, which included driver frequency content variations (including rise/fall time, equalization, driver impedance, jitter etc).

However one issue that we have to consider is the fact that the receiver impedance acts as a Thevenin voltage divider for the system, so minimizing the divided voltage should also be one of our goals. This fact creates a conflicting goal with the matched impedance, since in order to minimize the voltage divider we would have to increase as much as possible the total receiver impedance. That will of course create reflections in the system, but high loss at high frequencies will help dampen them. Since we have conflicting goals for the receiver impedance, we would have to use an optimization procedure to maximize the solution space.

The optimization can be done by a statistical method (similar with the DOE previously performed), but can also be more precisely achieved using the EDA as a criteria for a real-time optimization. Such optimization will take several user-defined parameters as an input, and will try to maximize the eye voltage or time (or a combination of both) based on the EDA readings. An example of that is described below using ADS as a simulator.

For the purpose of describing the EDA capabilities one optimization value will be used, receiver resistance, to provide us a comparison with the "hand" calculated value. The other system parameters are held constant as their nominal value Figure 6-2.



```
Goal
OptimGoal1
Expr="Veye_meas"
SimInstanceName="Tran1"
Min=2000
Max=
Weight=
RangeVar[1]=
RangeMin[1]=
RangeMax[1]=

VAR
VAR75
receiver_res=50 opt{ 10 to 1000 }
receiver_cap=0.6 noopt{ 0.5 to 1.5 }

Optim
Optim1
OptimType=Random          UseAllGoals=yes
MaxIters=25               SaveCurrentEF=no
DesiredError=0.0
StatusLevel=4
FinalAnalysis="None"
NormalizeGoals=no
SetBestValues=yes
Seed=
SaveSolns=no
SaveGoals=yes
SaveOptimVars=no
UpdateDataset=yes
SaveNominal=yes
SaveAllIterations=no
UseAllOptVars=yes

MeasEqn
Meas2
eye_meas=eye_measure(out_eq,seq[0],1/bit_rate[0],tran_start_time[0],tran_stop_time[0],0,1,1e-12,0)
Veye_meas=eye_meas(2)*1e3
```

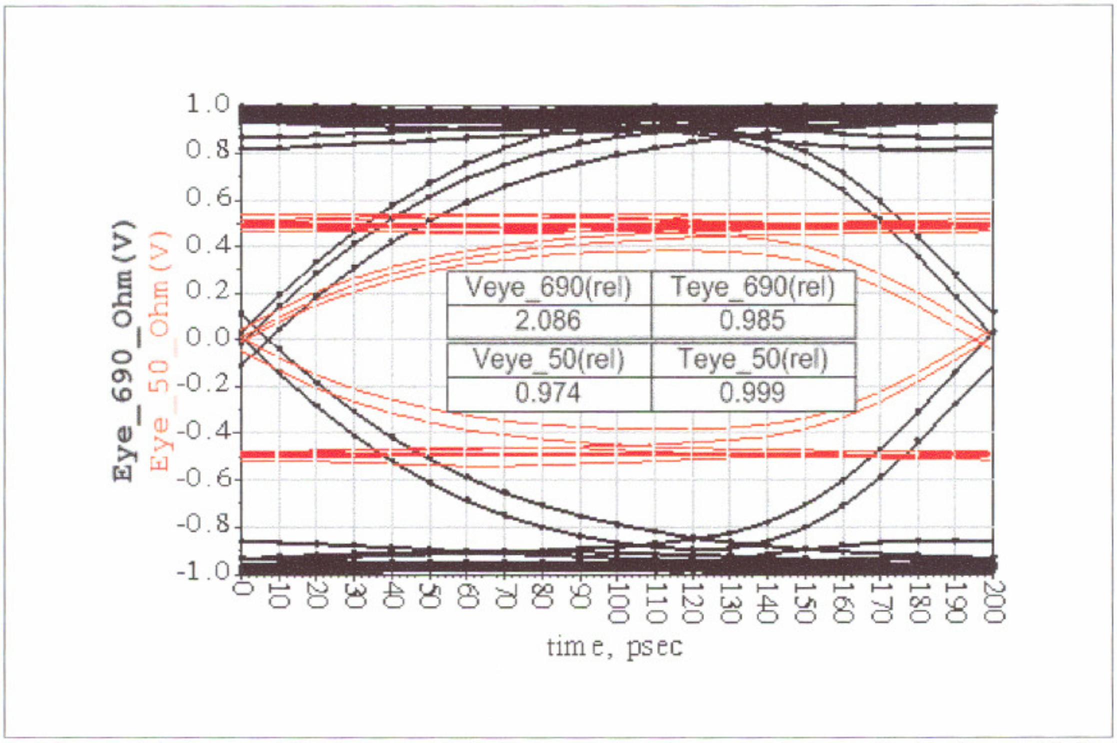
Figure 6-2 ADS receiver impedance optimization simulation settings with highlighted EDA criteria for voltage margin maximization

Receiver capacitance is set at 0.6 pF for comparison with hand calculation. Such setup assumes, of course, that the simulator has some minimal capabilities to be able to read results from the EDA and set goals based on that.

The optimization goal is set in the "Veye_meas" which is the voltage reading at the center of the eye using EDA. We have also used the "scale=1.916" option defined in Section 4.5 to match measured 6in trace, since impedance mismatch is more critical for shorter length.

Based on our goal (maximizing) eye voltage the optimal value for the resistor, based on real time eye optimization using the EDA, is 690 Ohm. We can see that this number is not at all in accordance with out initial hand calculations based only on reflection targets, so voltage divider tradeoffs are recommended at the targeted frequency.

Eye Diagram results indicating both initial (50 Ohm) and final (690 Ohm) are shown in Figure 6-3. The figure shows the absolute voltage values (y-axis) as well as the EDA readings scaled to the initial (50 Ohm) time and voltage values.



Veye_690(rel)	Teye_690(rel)
2.086	0.985
Veye_50(rel)	Teye_50(rel)
0.974	0.999

Figure 6-3 Improved voltage and timing margins due to resistor optimization – ADS graphical results and scaled EDA readings

Based on this value, a relative voltage eye improvement of 53% has been obtained. This eye improvement is, as a result of our specific goal, at the expense of smaller timing eye margin. However timing, as well as combined (weighted) voltage and timing margins, can be added, if desired, as design goals.

It is also worth mentioning that this specific optimization did not take into account any margin optimization at the driving end of the channel (such as driver operating points). Such criteria will obviously reduce the above value in order to reduce reflections seen at the driver.

6.1.2 Receiver tracking optimization

The other capability of the EDA is to characterize jitter. Determining the jitter median and deviation patterns from the median are important for a good receiver design. This has to be done based on driver input pattern and sampling numbers. They cannot be achieved with PDA-type methods, since those methods are not pattern aware.

As described in Chapter 3, there are four (4) default options that can be specified to characterize jitter distribution. Based on those results jitter median and maximum jitter can be determined. Receiver tracking capabilities can be determined based on those numbers. Optimum receiver implementations can also be determined base on a comparison of the different methods to track jitter distributions.

To exemplify this, a parallel between those results (with and without Duty Cycle Distortion), using all 4 options available in the EDA is shown in Table 6-7:

Table 6-7 Deviation from jitter median with and without DCD

Jitter Median(ps) (deviation from 1/2UI)				
	CENTER(0)	Min-MAX(1)	Standard_Dev(2)	COUNT(3)
NO Duty Cycle Jitter	0.00	4.35	2.64	6.00
With Duty Cycle Jitter	0.00	11.46	11.04	20.00

The parallel is actually telling us what the most likely placement of a receiver (that uses a specific jitter tracking function) is. Intuitively this number represents the perceived "eye center" deviation of that receiver from the true mathematical eye center. Since most likely, due to the off-center placement of the perceived "eye center" the Eye Diagram is not symmetric anymore, only the worst case number will be of interest and are reported by the EDA.

For example, a typical receiver implementation would be one using a phase interpolator to establish unit interval (UI) median. Such receivers do not actually care of the jitter amplitude, but the fact that it is positive or negative, based on an ideal cycle. They will sample each cycle and make decisions after looking at several cycles, determined by their available bandwidth.

To exemplify, a receiver with 1MHz bandwidth will have to look at 500 cycles (5GT/s data rate) to be able to correctly track jitter.

This number can be set-up in the EDA, such that the correct jitter statistic can be obtained. Such receiver should use the jitter_count method, to determine its optimum time step used for correct placement (Table 3-1-COUNT (3)).

Based on the above setting we have used the EDA and the jitter count method and determined that a 6ps "step" would be necessary for the receiver to have a proper placement in the middle of the eye, when no DCD jitter is considered. Once DCD jitter is added, that number has to be increased up to 12ps. Those numbers are necessary based on the jitter accumulated over the number of cycles of interest (in this case 500 cycles), and is telling us that in order for the receiver to be in the center of the eye for the next 500 cycles, the decision to be made has to have that granularity or better.

The other measurement that can be made is that of the voltage seen by the receiver using the above off-placement, as well as the voltage that is some distance away from the "perceived "center, in order to account for actual receiver margins

In this example a receiver margins of +/- 0.1UI and +/- 0.15 UI has been assumed. Numbers in Table 6-8 are scaled to the center value with no DCD seen in the previous case Figure 5-9 (599mV):

Table 6-8 Relative voltages based on receiver placement- NO DCD

	CENTER(Perceived)	OFF-CENTER(+/-20ps)	OFF-CENTER(+/-30ps)
NO Duty Cycle Jitter(0)	1.01	0.96	0.93
NO Duty Cycle Jitter(1)	1.00	0.95	0.91
NO Duty Cycle Jitter(2)	1.01	0.95	0.92
NO Duty Cycle Jitter(3)	1.00	0.94	0.90

The first option (0) is showing the voltage value if the receiver sampler is exactly in the center of the period, in other words is showing what voltage the receiver will probably see at the beginning of data transmission, or when the receiver has no adjusting mechanism. Accidentally, based on the asymmetric Eye Diagram (pulse response) of the system, this is actually slightly better receiver placement than the true eye center, shown in the second row (option (1)). It is also slightly better (up to 2%) if we add into account the receiver margin.

We can also see that Options (2) and Option (3) are very close the Option (1) which indicates that for this example the statistical mean of the jitter is similar with the arithmetic mean.

Table 6-9 is showing similar data only with DCD added to the driver. We can see that even with this simplifying assumption, placing the receiver in the true eye center is not the best choice. The best choice being, as expected, to place the receiver at statistical mean.

Table 6-9 Relative voltages based on receiver placement with DCD

	CENTER(Perceived)	OFF-CENTER(+/-20ps)	OFF-CENTER(+/-30ps)
With Duty Cycle Jitter(0)	0.99	0.88	0.77
With Duty Cycle Jitter(1)	0.97	0.91	0.86
With Duty Cycle Jitter(2)	1.00	0.95	0.92
With Duty Cycle Jitter(3)	1.00	0.94	0.90

By looking at both tables, we can conclude that jitter statistics, therefore receiver placement is of major importance, since we can have up to 23% eye voltage reduction based on that.

Chapter 7

Conclusions

The goal of addressing some of the limitations in today's high speed digital design has been achieved in this investigation. A new methodology and simplified process flow for the major components of a system interconnect has been developed.

Interconnect models are based on actual measurements, and have been abstracted to a simple algebraic formula. This not only makes the model tool independent, but also allows for reduced computation time and reduced errors in solving that model. It also overcomes some limitations of the models that are based on approximations of the Maxwell equations.

It is also a major improvement when compared to using and manipulating actual measured data (s-parameter matrix format) because it allows seamless interactions with other circuit elements since there are no limitations in minimum and maximum bandwidths as well as number of points. The simplified distributed load models are accurately specified by mathematical formulas, therefore they can replace traditional lumped references. Modern instruments such as oscilloscopes can take these formulas and perform internal transformations, which will accurately match actual measurements performed into corresponding loads. Also, the derived distributed model can be used "as is" with no modifications, for target silicon designs, therefore bridging the gap between the two different worlds, I/O and interconnect design. This has been illustrated in the last chapters of the thesis, in which optimizations have been performed for both driver and receiver.

Driver optimizations have been performed for parameters, such as nonrecursive filter coefficients, that simply cannot be performed in isolation, into 50Ohm loads, the way traditional design methods have characterized drivers. To furthermore illustrate the advantages of having I/O designed and characterized into distributed loads, jitter amplification, a channel dependent phenomenon, has been also addressed.

Last but not least, receiver optimization has been considered. Receiver circuits are one of the most overlooked "blocks", mainly because of the difficulty to correlate with measurements. One of the overlooked items, described in this thesis, is matching receiver sampling and interpolation methods with system jitter distribution and clock recovery algorithms. It has been shown that even for simple designs significant design margins are lost because incorrect receiver placement.

All the system characterizations described have been greatly simplified by the use of the "Eye Diagram Analyzer" software.

The tool addressed another major limitation of today's analog design toolsets, the inability to extract accurate and useful information form Eye Diagrams, as well as using the extracted data as real-time design criteria.

7.1 Future Work

The work presented in this thesis can be extended in several ways. By incorporating higher order mathematical functions, s-parameter approximations based on measurements for more complex structures (such as packages, connectors, vias) can be developed and added to a library of tool independent interconnect models.

As existing data transfer rates continue to increase, extending the existing copper communication bandwidth even further, while maintaining low production costs, will require even more complicated I/O correction techniques, such as adaptive driver and receiver equalization, pulse echo cancellation, and modulation. On and off-chip non-linear behavior will play on increased role in simulation and design accuracies.

Current mainstream characterization techniques will be difficult to extend to incorporate such effects either do to theoretical (such as PDA-linearity assumptions) or practical limitations (models based on Maxell equations), while the methodology presented here can be naturally extended to characterize the above effects as well as to incorporate advances statistical methods in conjunction with accurate time and frequency domain characterization techniques.

Due to the fact that silicon "clock" rates have been historically increasing at higher rates than the system ones, it is safe to assume that transmission line effects (which have been negligible so far) will have a preponderant influence in silicon design. Therefore the methodology presented here can also be used to accurate characterize on-chip transmission lines, analog behavior of on-chip signals (especially speed paths and clock distribution domains), as well as advanced driver and receiver correction techniques.

Appendix

Abbreviations

PDA	–	Peak Distortion Analysis
AC	–	Alternating Current
ADS	–	Advanced Design System
CTC	–	Cycle to Cycle
dB	–	Decibel
DC	–	Direct Current
DCD	–	Duty Cycle Distortion
DDJ	–	Data Dependent Jitter
DJ	–	Deterministic Jitter
DOE	–	Design Of Experiments
EDA	–	Eye Diagram Analyzer
EM	–	Electromagnetic
EMI	–	Electromagnetic Interference
FFT	–	Fast Fourier Transform
FIR	–	Finite Impulse Response
GHz	–	Gigahertz
GT/s	–	Gigatransfers / second
HVM	–	High Volume Manufacturing
I/O	–	Input/Output
IFFT	–	Inverse Fast Fourier Transform
ISI	–	Inter Symbol Interference
mV	–	Milivolt
PCB	–	Printed Circuit Board

PDA	–	Peak Distortion Analysis
PJ	–	Periodic Jitter
ps	–	Picosecond
PWL	–	Piecewise Linear
RJ	–	Random Jitter
SOLT	–	Short-Open-Load-Through
SSO	–	Simultaneous Switching Outputs
TEM	–	Transverse Electromagnetic
TJ	–	Total Jitter
UI	–	Unit Interval
VNA	–	Vector Network Analyzer

References

[1] ***Jeff C. Morriss.*** **Reference Transmission Line Theory.** Intel Corporation Internal Document.

[2] ***Jeffrey A. David, ET all.*** **Interconnect Limits on Gigascale Integration (GSI) in the 21st Century.** Proceedings of the IEEE, Special Issue on Limits of Semiconductor Technology. March 2001. Vol. 89, No. 3. Pages: 305 to 324.

[3] ***Martin S. Roden.*** **Analog and Digital Communication Systems.** 2nd ed. Prentice-Hall 1985. Pages: 341.

[4] ***Martin S. Roden.*** **Analog and Digital Communication Systems.** 2nd ed.. Prentice-Hall 1985. Pages: 105 to106.

[5] ***John Patrin, Mike Li.*** **Comparison and Correlation of Signal Integrity Measurement Techniques.** DesignCon 2002. January 28–13, 2002. Santa Clara Convention Center. Santa Clara, California USA. Pages: 5 to 19.

[6] ***Simon Ramo, John R. Whinnery, Theodore Van Duzer.*** **Fields and Waves in Communication Electronics.** 3rd ed. John Wiley & Sons: December 1993. Pages: 220.

[7] ***She Lin, Emest Kuh.*** **Transient simulation of lossy interconnects.** Annual ACM IEEE Design Automation Conference, Anaheim, California, 1992. Pages: 81 to 86.

[8] ***META Software.*** **Star-Hspice User Guide.** Release 2001.4. Pages: 20-4 to 20-11

[9] ***GGB Industries, INC***, **High Performance Microwave Probes, Model 40A.** Available: http://www.ggb.com/PdfIndex_files/mod40a.pdf [Viewed: June 20, 2004].

[10] ***Cascade Microtech***. **Technical Brief; A Guide to Better Vector Network Analyser Calibrations for Probe-Tip measurements**. Pages: 1 to 2. Available: htttp://www.cascademicrotech.com/pubs/TECHBRIEF4.pdf [Viewed: June 20, 2004].

[11] ***IPC-D-317A***, **Design Guidelines for Electronic Packaging Utilizing High-Speed Techniques**, Jan 1995. Section 5. Pages: 13 to 36

[12] ***Simon Ramo, John R. Whinnery, Theodore Van Duzer.*** **Fields and Waves in Communication Electronics**. 3rd ed. John Wiley & Sons: December 1993. Pages: 230.

[13] ***The MathWorks.*** **MATLAB Function Reference**. Version 6 - Volume 2. Pages: 2 to 23. Available: http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html [Viewed: June 20, 2004].

[14] ***J.W. Cooley, J.W. Turkey.*** **An Algorithm for the Machine Computation of the Complex Fourier Series**. Mathematics of Computation, Vol.19 April 1965. Pages: 297 to 301.

[15] ***HP Advanced Design System***. **Expressions, Measurements, and Simulation Data Processing.** January 1999. Pages: 3-194 to 3-195.

[16] ***B.K. Casper, M. Haycock, R. Mooney.*** **An accurate and efficient analysis method for multi-Gb/s chip-to-chip signalling schemes**. VLSI Circuits Digest of Technical Papers, 2002. Symposium on , 13-15 June 2002. Pages: 54 to 57

[17] ***A.X. Widmer, P.A. Franaszek.*** **A DC-balanced, partitioned-block, 8B/10B transmission code.** IBM J. Res. Develop. Sept. 1983. Vol.27.

[18] ***SAS Institute Inc.*** Cary, NC, 2004. **JMP Design of Experiment, Version 5.1.1**. Cary, NC: SAS Institute Inc. Pages: 115 to 122.

[19] ***SAS Institute Inc.*** Cary, NC, 1989-2001.**JMP, Version 4.0.4**. Available: http://www.r-project.org [Viewed: June 20, 2004].