M 2014

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

ADAPTIVE EQUALIZATION FOR INTERCHIP COMMUNICATION

DÉNIS GASPAR NOGUEIRA DA SILVA

DISSERTAÇÃO DE MESTRADO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
MIEEC –MESTRADO INTEGRADO EM ENGENHERIA ELECTROTÉCNICA E COMPUTADORES

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES 2013/2014

A Dissertação intitulada

"Adaptive Equalization for Interchip Communication"

foi aprovada em provas realizadas em 21-07-2014

o júri

Presidente Professor Doutor Vitor Manuel Grade Tavares
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor Luís Filipe Mesquita Nero Moreira Alves
Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Professor Doutor Henrique Manuel de Castro Faria Salgado
Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - Denis Gaspar Nogueira da Silva

Faculdade de Engenharia da Universidade do Porto

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Adaptive Equalization for Interchip Communication

Dénis Gaspar Nogueira da Silva

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor at FEUP: Prof.Henrique Salgado

Supervisor at Synopsys: M.Eng Luís Moreira

Supervisor at Synopsys: Dr.Eng Sérgio Silva

August 3, 2014

Resumo

A area da eletrónica é uma industria em constante evolução e é a fundação de novas tecnologias em outras áreas como a indústria automóvel medicina e muitas outras.
As empresas de projecto e fabrico de circuitos integrados constituem a base da evolução tecnológica fornecendo tecnologias que possam ser integradas por outras empresas em sistemas mais complexos para o desenvolvimento de um determinado produto para o consumidor .
A Synopsys é uma empresa que se dedica, além de outros temas, ao design de interfaces de altas velocidades para barramentos de comunicações digitais, fornecendo circuitos integrados para protocolos como USB, HDMI, MIPI, SATA e outros.
Actualmente a velocidade de transmissão das interfaces desenvolvidas pela Synopsys atingem a ordem dos Gbs por link, a transmissão a essas velocidades não seria possivel sem sistemas que compensem a distorção causada pelos canais utilizados na comunicação.
A distorção é causada pelo facto dos canais possuirem uma resposta em frequência cuja atenuação é dependente da frequência,o que causa que os bits transmitidos nesse canal aumentem a largura original do seu impulso e interferindo com os bits adjancentes, efeito que é conhecido como Interferência Inter-Simbólica.
A eliminação da interferência intersimbólica é conseguida através de circuitos de igualização que realizam respostas em frequência inversas à do canal para tornar a resposta em frequência total o mais plana possível.
No entanto por vezes é necessário realizar igualização sem que se saiba ao certo qual a distorção introduzida, de facto a distorção pode até mesmo mudar durante a operação da interface o que leva que a igualização se torne adaptativa.
Este assunto tem sido uma área muito estudada nos ultimos anos devido á necessidade das interfaces de comunicação de realizarem comunicações com débitos cada vez mais elevados, o que leva que a questão da eliminação da interferencia inter simbólica se torne cada vez mais importante.
O objectivo desta tese é estudar e testar métodos de adaptar os filtros de igualização presentes numa interface de alta velocidade desenvolvida pela Synopsys.
O estudo terá em conta as limitações introduzidas pelos débitos utilizados, Características dos canais e pelas restrições de complexidade impostas aos sistemas de adaptação.

Abstract

The area of electronics is in constant evolution and is the foundation of new technological breakthroughs in areas like the automobile industry, medicine and many others.
Companies dedicated to the design and manufacturing of integrated circuits are the basis of the technological advance by providing technology that can be integrated by other companies in more complex systems in the development of products for the final consumer. Synopsys is a company dedicated to the design of high speed serial interfaces to digital buses, providing integrated circuits that support protocols as USB, HDMI, MIPI, SATA and others.
Nowadays the interfaces developed by Synopsys operate at bit rates in the order of the Gbs per link, the transmission at that bit Rates would not be possible without systems that compensate the distortion caused by the channels used in the communication.
The distortion is caused by the fact that the channels have a frequency response with frequency dependent losses,this causes the spreading of the transmitted pulses to the adjacent bits causing Inter Symbol Interference.
The elimination of Inter Symbol Interference is achieved through equalizing circuits that compensate the channel by introducing a frequency response that is the inverse of the one of the channel, to make the overall frequency response as flat as possible.
Sometimes there is the need to design equalizing circuits without the knowledge of the amount of distortion that is introduced by the channel.In fact, the channel characteristics may even change during the operation of the interface leading to Adaptive equalization.
The field of adaptive equalization has been a very hot topic in research the last couple of years due to the increase in the speeds used in the communication interfaces,henceforth the topic of the elimination of Inter Symbol Interference gained relevance.
The objective of this thesis is to study and test methods that perform adaptation of the equalizing filters present in a high speed interface designed by Synopsys.
The study will take into account the limitations introduced by the bit Rates used in the interface,the characteristics of the channels and complexity restrictions imposed to the adaptation system.

Acknowledgements

I would like to express my gratitude to:

My family who always supported me through all my five years of college, i am sure that without them i would not be the person I am today. So here is a big thanks to all my five brothers Maria, Julio, Ricardo, Cláudia and Romeu, and my parents Maria and José, who sacrificed a lot to get me to finish my degree.

To all my friends who from time to time would make me shift my mind from this thesis and have some fun.

To Pedro Araújo for supporting me and dealing with my moments of insanity.

To all the people in Synopsys, specially Henrique Martins, Hélder Campos and my friends in the analogue team: Bruno, Patrício, Hugo and Nuno that always provided me with the best cookies.

To Bárbara for being my biggest supporter during this six months that I worked on my thesis.

To Célio Albuquerque for giving me the opportunity of developing my dissertation in such a great environment that is Synopsys.

To my supervisors at Synopsys, Sérgio Silva and Luís Moreira for opening a few moments on their busy schedule, to share some of their knowledge and experience. I would also would like to say that without their guidelines and advice I wouldn't have been able to finish my thesis.

To my supervisor at FEUP, Henrique Salgado for all suggestions, discussions and editing assistance during the development of this dissertation.

Finally, to FEUP for giving me the best years of my life.

Dénis Silva

"However beautiful the strategy, you should occasionally look at the results."

"Sir Winston Churchill"

Contents

List of Figures

List of Tables

Abbreviations, Acronyms and Symbols

ADS	Advanced Design System
CDR	Clock and data Recovery
CJTPAT	Compliance Jitter Test Pattern
CRPAT	Compliance Random Pattern
CTLE	Continuous Time Linear Equalizer
DFE	Decision Feedback Equalizer
FIR	Finite Impulse Response
Gbs	Gigabit per Second
IIR	Infinite Impulse Response
IP	Intellectual property
ISI	Inter Symbol Interference
LMS	Least Mean Squares
MLS	Maximum Length Sequence
MLSE	Maximum length Sequence Estimation
PHY	Physical layer in the OSI model
PRBS	Pseudo Random Binary Sequence
RF	Radio Frequency
RX	Receiver chip
TX	Transmitter chip
UI	Unit Interval
ZF	Zero forcing algorithm

Chapter 1

Introduction

The area of equalization has become a major area of research in the last couple of years due to the increase in the bit Rates used in serial communication interfaces. This increase led to the necessity to every communication interface to have some sort of equalizing circuit to compensate the distortion caused by the channel.

Many studies and articles have focused in the implementation of equalizer circuits, [3] discusses about the advantages of using a CTLE to perform cable equalization at $10Gbs$. Article [4] addresses a equalizing system with a CTLE and a DFE but both filters are static and the system does not include an adaptive algorithm. In [5] the adaptation of a DFE and CTLE for the PCIe interface at $8Gbs$ using algorithms like the Zero forcing and performing adaptation using an on chip eye monitor is discussed, however adaptation using these algorithms is usually very slow and the target bit Rate is inferior to the one used in the studied interface.

Another interesting fact is the possibility to implement a blind adaptation algorithm for the adaptation of the DFE. In these kind of algorithms the receiver has no previous knowledge of the transmitted sequence and thus has a more difficult task to estimate the amount of channel induced ISI that needs to be compensated.

Article [6], talks about the use of the constant modulus algorithm in the DFE. However the article is only focused on the mathematical implementation of the algorithm and does not give an actual example of a real implementation.

This dissertation is focused on implementing an equalization architecture combining a CTLE and a DFE to perform channel equalization at $12Gbs$ and provide algorithms to adapt both filters to the transmission environment.

The tested algorithms should be as simple as possible to not disturb the interface specifications and should not require the insertion of many additional circuits to the interface.

1.1 Motivation

With the increase in communication speed for the serial interfaces the channel limitations become even more severe and the need of equalization becomes imperative.
One of main techniques of performing equalization in interfaces consists in placing a continuous time Linear Filter capable of compensating the high frequency losses of the channel.
The Decision feedback equalizer is also one of the equalizers that present better results in the elimination of inter symbol interference.
Although the theme of equalization of communication interfaces is a highly studied subject, the choice of the equalization architecture and adaptive algorithm are still very dependent on the interface and the channel to be compensated,with several known solutions.
Moreover, interfaces have specifications that need to be complied in order to be sold in the IC market and some equalization strategies comprise timing and power constraints that may not be compatible with the protocol specification.
So, the objective of this dissertation is to assess adaptive algorithms for the equalization filters present in the interface, these algorithms need to comply with the protocol specifications and must take advantages of specificities of the interface.
One of the main challenges of this work is the implementation of an algorithm that is capable of optimally adjust a CTLE and a DFE in the most simple fashion and with the minimum overhead possible.

1.2 Objectives

The work is focused on the study of both analogue and digital equalizers designed for the compensation of communication channels used in inter chip communications. The study will approach the following topics:

- Continuous Time Linear Equalization: Design of the filter and implementation of an adaptive algorithm
- Equalization through digital filter:study of both transversal and decision feedback equalizers.
- Study of the impact of each equalizer in the compensation of the communication channel.
- Study of algorithms and implementations for automatic equalizer adaptation. Comparison regarding performance and implementation cost.
- Requirements for training sequences
- Matlab/ADS simulations using synthetic and real signals

1.3 Structure of the document

This document is structured in the following way :

Chapter 2 Gives a theoretical approach to the problems encountered when performing transmission at high speeds. This chapter explains in detail the problem of Inter-Symbol Interference and the use of equalization for the correction of channel induced distortion.
The chapter discusses equalization with a continuous time linear equalizer and with a decision feedback equalizer explaining their working principle and their capabilities in correcting the channel Induced ISI.
The chapter also discusses techniques for the adaptation of these filters to the transmission environment.

Chapter 3 Describes the target interface that is proposed to host adaptive equalization. In this chapter are characterized the reference channels used to model the real transmission environment.
Also in this chapter it is justified the need for equalization when performing transmission at $12Gbs$ and an architecture to compensate the reference channels

Chapter 4 Studies the impact of the equalization blocks in the eye diagram opening .
In this chapter a CTLE with 16 settings of boost using a two pole one zero transfer function is developed.

Chapter 5 Presents the use of different algorithms to perform adaptation of the equalization blocks, focusing in the adaptation of the CTLE and the DFE to the transmission environment described in chapter 3.Each algorithm is compared in terms of complexity, performance and time needed to perform adaptation.

Chapter 6 presents the main conclusions of the work developed, and guidelines for future work.

Chapter 2

Digital communication systems

Modern communication systems are based on digital transmission. The primary advantage over analogue systems is that digital signals are easier to regenerate and are more insensitive to noisy communication channels. In this chapter we will study factors that restrict the speed and reduce link quality in serial communication channels, the effects of band-limited problems are described using the *Nyquist Criterium*. Studying the methods that degrade the bit error ratio in a channel we can understand ways to improve or mitigate the influence of such factors, as for example using equalization.
Equalization can be accomplished using discrete or analogue filtering combined with methods to adapt these filters to the channel.It is necessary to compare these solutions in terms of complexity and efficiency.
Some approaches already implemented are given as an example to help the reader to further comprehend ways to implement equalization filters in high speed links. Emphasis is given to implementation and adaptation of both continuous time equalizers and decision feedback equalizers.

2.1 Inter-Symbol Interference

Several aspects degrade the probability of error in a communication channel, the channel itself has associated with it a frequency response usually similar to a low pass filter. The cut-off frequency associated to the channel is a function of the channel length, temperature among other variables.
The maximum throughput that can be achieve in a communication is given by the Shannon-Hartley theorem, which states that the channel capacity C in bit/s is a function of the channel bandwidth in Hz and the signal to noise ratio:

$$C = B\log_2(1+\frac{S}{N})$$

From the previous equation we can see that we want to make the channel bandwidth as high as possible so we can achieve the maximum throughput possible.

The channel transfer function is usually obtained by means of the Fourier transform, in the next figure we can view the usual low pass characteristic of a communication channel.

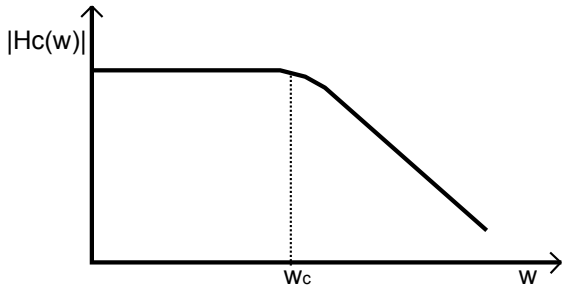


Figure 2.1: Channel Low pass Transfer Function

The channel can both introduce Amplitude and Phase distortion altering the transmitted bits. For no amplitude distortion the channel transfer function $H_c(j\omega)$ must be flat for the entire signal spectrum.

$$| Hc(j\omega) | = K$$

For no phase distortion the channel group delay must be constant:

$$-\frac{\partial H_c(j\omega)}{\partial \omega} = K$$

As the channel introduces distortion the transmitted waveforms become attenuated, and dispersion causes the duration of the bit to be extended beyond T seconds. Dispersion of the signal can be caused by the distortion or by Multipath inside the channel.

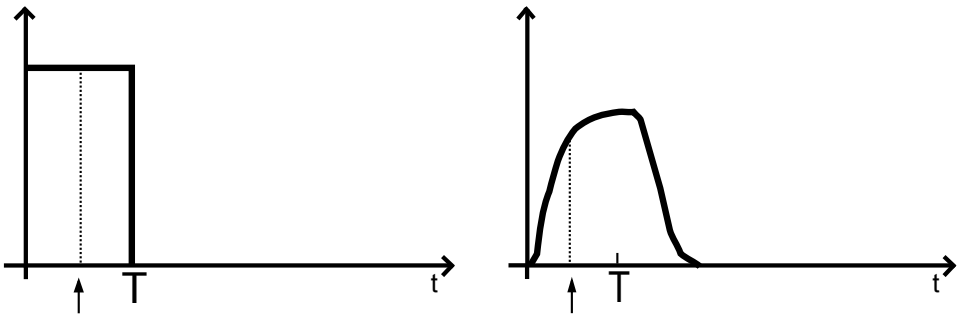


Figure 2.2: Bit distortion

When a sequence is transmitted through a channel that introduces dispersion inter symbol occurs.

Inter symbol interference results from the distortion of the waveform of the transmitted bits. Because bits interfere with the amplitude of neighboring bits in the sampling instant. ISI can cause the decision element to make incorrect decisions on the detection of a symbol.

2.1.1 Nyquist condition

Lets us consider that the channel is linear and time invariant, so it can be represented by its impulse response:

$$H_c(n) = \sum_{i=0}^{+L} h_k \delta(n-k)$$

$\delta(n)$ represents the Dirac pulse. The sequence observed after the channel results from the linear convolution of the input sequence and the channel impulse response:

$$y(n) = x(n) * h_c(n)$$

$$y(n) = \sum_{k=-\infty}^{+\infty} x(n) h_c(n-k)$$

For no ISI one must satisfy the Nyquist criteria:

$$y(nT) = \begin{cases} c & \text{if } n = 0 \\ 0 & \text{if } n \neq 0 \end{cases}$$

Nyquist criterium states that for no ISI the bit waveform can only be different from zero at is own sample time, this presents the **Time domain Nyquist Criteria**.

Considering that the received waveform $y(t)$ equals:

$$y\delta(t) = \sum_{n=-\infty}^{+\infty} y(nT)\delta(t-nT)$$

$y\delta(t)$ represents the sampled received signal. Taking the Fourier transform we get:

$$Y\delta(f) = \frac{1}{T} \sum_{n=-\infty}^{+\infty} Y(f - \frac{n}{T})$$

Because $Y\delta(t) = \delta(t) \longrightarrow Y\delta(f) = 1$

$$\sum_{n=-\infty}^{+\infty} Y(f - \frac{n}{T}) = T$$

This equation represents the **Frequency domain Nyquist Condition** and it states that for no ISI the folded spectrum of the received signal must be constant.

Assuming a channel with bandwidth W, the Nyquist condition has the following implications:

- Suppose that the symbol rate is so high that $1/T > 2W$: no matter how the received spectrum looks like there will always be gaps between spectrum copies and ISI is inevitable.

- If the data rate is slower then $1/T < 2W$ the copies of X(f) will overlap and there is many options for X(f) that make the folded spectrum $\sum_{n=-\infty}^{+\infty} Y(f - \frac{n}{T})$ flat.

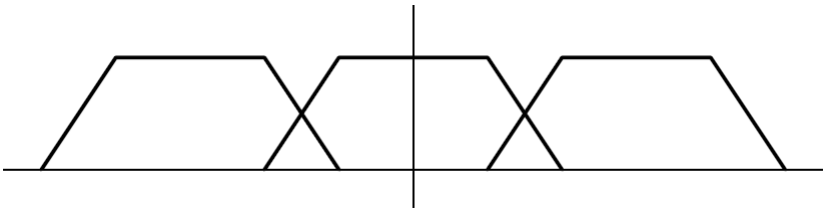


Figure 2.3: Overlapping spectrum in the case of $1/T < 2W$

- If $1/T = 2W$ Then the spectral copies of the bit waveform just touch and in order for not to exist ISI the spectrum $Y(f)$ must be rectangular. A rectangular spectrum can only be achieve by shaping the bits as sync pulses. The rate $1/T = 2W$ is know as the Nyquist Rate and imposes the maximum theoretical bit rate that can be transmitted in a channel with bandwidth W.

2.1.2 Jitter

A finite bandwidth of the communication channels causing distortion is not the only factor that degrades the quality of a binary communication. In fact even if a channel satisfies the Nyquist condition the jitter alone can be a major limiting factor to the amount of BER.
By definition Jitter is the disturbance of the periodicity of the timing signals along the transmission path. Such disturbance can make the communication nonviable as the received waveform must satisfy both voltage and timing constraints.
For example in the extreme event of the peak to peak jitter reaching $0.5UI$, meaning that each edge of a bit can drift at most 0.5UI, if the channel is ideal and does not introduce distortion the jitter alone will cause the receiver to make incorrect decisions as the sample time will drift to the adjacent bits.

Jitter can be divided into the following categories as stated in [7]

- Random jitter(RJ): characterized by a normal distribution described by its mean, usually zero, and by its RMS value
- Deterministic jitter(DJ): not described by a statistic distribution being characterized by its peak to peak value, it comprehending the following types :
  - Periodic Jitter (PJ): caused by electronic noise, thermal, shot, flicker noise

  - Data Dependent Jitter(DDJ):caused by the transmitted data patterns or due to cross talk
    * Duty Cycle distortion(DCD): the timing of each bit changes along transmission
    * Intersymbol Interference Jitter (ISIJ): the band limited channels cause that some edge are faster or appear later than they should

- Unbounded uncorrelated Jitter(UUJ): caused by cross-talk.

Jitter is a very complex phenomenon and can have many sources, and because of that its classification is many times done in a statistical way. Usually jitter is an intrinsic characteristic of every communication system and is a factor that should be expected and considered in the design of communication channels.In spite of this some jitter components can be minimized; for example ISI jitter can be reduced with TX and RX equalization

2.1.3 Scattering Parameters

Scattering parameters are used in radio-frequency engineering to describe the behavior of linear electrical networks operating at very high frequencies. The difference between S-parameters and other forms of representation like Y or Z parameters is that they do not represent the network in terms of currents or voltages, but in terms of behavior of power waves that pass through the network, another simplification introduced is that it considers that all the ports are properly terminated with a matched load (usually 50Ω). This comes from the fact that many times it is not easy to characterize a network in terms of current and voltages used in Y or Z parameters because there are not easy ways to produce an ideal short or open circuit in radio frequencies.So the S-parameter matrix describes a network expressing the behavior of a power waves that enter a port and exit a port.By convention the waves that enter a port are considered positive and the ones coming out are negative.Consider the following 4 port network:

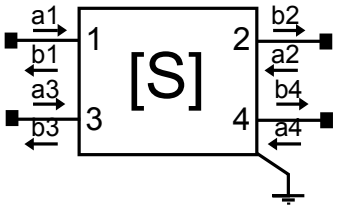


Figure 2.4: 4 port Network

The S matrix expresses the relation between the waves coming out and into a port:

$$b = Sa \text{ with } S = \begin{bmatrix} S11 & S12 & S13 & S14 \\ S21 & S22 & S23 & S24 \\ S31 & S32 & S33 & S34 \\ S41 & S42 & S43 & S44 \end{bmatrix}$$

We can define interesting quantities derived from the S parameter Matrix.

- Voltage Reflection coefficient as *Sii*
- Complex linear Gain e.g $S13$ Represents the complex gain between port 1 and 3
- Insertion loss is defined as e.g $20\log 10|S_{21}|$
- cross talk e.g $S12$ between port 1 and 2

In this section we presented a little review on S-parameters since often channels characteristics are represented in the form of an S- matrix.

2.1.4 Eye Diagram

The best way to measure the quality in communications links is the bit error ratio, but unfortunately such a measure cannot be easily measured, Specially during normal link operation.Eye diagrams present an easy and intuitive way to evaluate communication quality as they in a visual way lets us know how the distortion and jitter affect the signal waveform, following this approach eye opening is a good way to assure a low bit error ratio as a wider eye relaxes the receivers detector operating conditions for correct operation. An eye diagram is created by overlaying sweeps of different segments of a long data stream driven by a master clock, the next figure show us an example of an eye diagram.

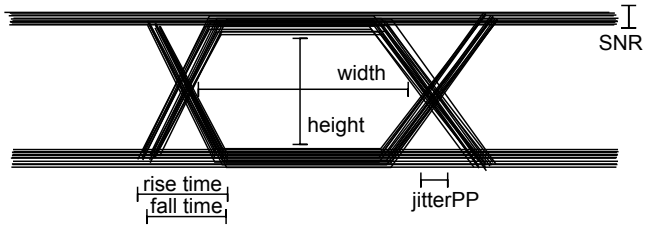


Figure 2.5: Eye Diagram example

In the figure the parameter *width* represents the minimum time between two successive passes through the threshold voltage;
Height gives us a measure of the difference between the peaks of a 0 level and a 1 level
Rise and *fall time* gives an average of the rising and fall time when the line switches voltage state
SNR is a measure of the deviation of the level 1 and 0 voltage it indicates the amount of signal to noise Ration and distortion in the signal.
JitterPP or jitter Peak to peak gives us the maximum time diference between transitions by the voltage threshold.

Usually in a serial communication protocol, some minimum values for some of the previous parameters are established forming an *eye mask* that determines the minimum quality for the communication to be considered viable. As said earlier the most important quality factor in a communication system is the BER.So how can we derive an approximation of the BER by using an eye diagram?

2.1.4.1 The Q factor

The article [8] gives a metric for the evaluation of the quality of the transmitted waveform by evaluating the properties of the eye diagram. Consider the probability density functions at mean level 1 and mean level 0 to be a Gaussian distributed with an average μ_1 and μ_0 and standard deviation σ_1 and σ_1, respectively.

Let $p(t_0)$ and $P(t_1)$ be the probability of the occurrence of a bit 0 or a bit 1, and $P(r_0)$ and $P(r_1)$ the probability of the detection of a bit 0 and 1 considering sampling in the middle of the eye opening. We can define the probability of error as:

$$P = p(r_0/t_1) * p(t_1) + p(r_1/t_0) * p(t_0)$$

Where $p(ri/tj)$ represents the conditional probability of detecting a bit i while transmitting a bit j. So the bit error ratio (BER) is given by:

$$BER = \frac{1}{2} erf(\frac{Q}{\sqrt{2}}) \text{ and } Q = \frac{\mu_1 - \mu_0}{\sigma_0 + \sigma_1} \text{ with: } erf(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x0} e^{\frac{-(x-\mu)}{2\sigma^2}}$$

2.2 Strategies to mitigate ISI

In this section we will talk about strategies to fight Inter-Symbol Interference and improve link quality.

2.2.1 Fight ISI with bit shaping

We can deduce from the frequency Nyquist condition that bit shaping can help fight the effects of ISI in communication systems. The more we compress the signalling spectrum the higher the data rate we can transmit through a channel, the spectrum compression achieved by applying a Nyquist filter to the transmitting waveform. A widely used ISI free pulse is the raised cosine:

$$X(f) = \begin{cases} T & \text{for } 0 \leq \mid f \mid \leq \frac{1-\alpha}{2T} = 0 \\ \frac{T}{2}[1 + \cos\frac{\pi T}{\alpha}(\mid f \mid - \frac{1-\alpha}{2T})] & \text{for} \frac{1-\alpha}{2T} \leq \mid f \mid \leq \frac{1+\alpha}{2T} \\ 0 & \text{for} \mid f \mid > \frac{1+\alpha}{2T} \end{cases}$$

α is the roll-of factor which determines the excess bandwidth from the original rectangular pulse 1/2T. The corresponding function is:

$$x(t) = \frac{\sin\frac{\pi t}{T}}{\frac{\pi t}{T}} \frac{\cos\frac{\pi\alpha t}{T}}{1-\frac{4\alpha^2 t^2}{T^2}}$$

$\alpha = 0$ the raised cosine pulse becomes the sync function, 2.6 and 2.7 present the pulse shape and signal spectra of a raised cosine pulse as function of α.

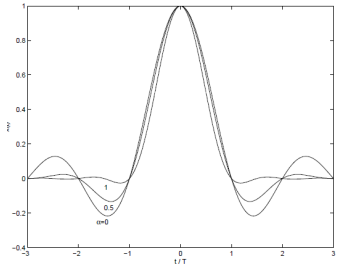


Figure 2.6: Raised cosine waveform

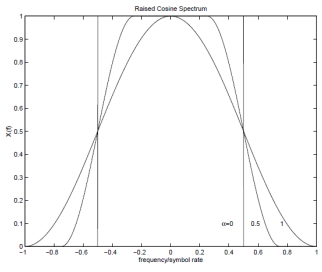


Figure 2.7: Raised cosine spectrum

Notice that in the multiples of the sample period the waveform passes through zero resulting in zero ISI. Nyquist filters are hard to realize in practice because of their infinity impulse response. The roll-of factor decreases the ondulation after the time bit, which can be harmful for ISI if the sample time suffers from jitter.

2.2.2 Maximum length Sequence Estimation

In the maximum likelihood receiver the samples are not modified or reshaped by the receiver, instead using MLSE the receiver adjusts itself to better deal with the distorted samples. The MLSE receiver uses an estimate of the channel modelled as a finite input response (FIR) filter to compute the most likely transmitted sequence.

Let us consider the following channel model:

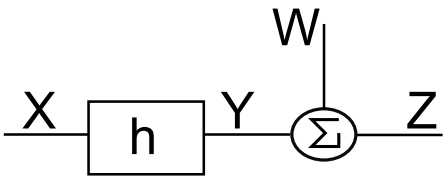


Figure 2.8: Discrete channel model.

X represents the transmitted sequence w represents white Gaussian noise $\mathcal{N}(0,\delta^2)$ added to the channel.

y represents the output of the channel with impulse response h of length L.

So from the above figure we can write $z = y + w$ and $y = x * h$ resulting in :

$$y_i = h_0 y_i + \sum_{j=1}^{L} (h_j x_{i-j})$$

The summation represents the inter symbol interference. In the MLSE receiver we want to maximize the following expression:

$$P(Z \mid U^{(m*)}) = maxP(Z \mid U^{(m)})$$

Meaning we want determine received sequence z that maximizes the probability $P(Z \mid U^{(m)})$ U(m) represents a possible transmitted sequence.In the case of binary transmission and in the case of a transmitted sequence of size M. We have 2^L possible sequences so the computational complexity increases exponentially with the sequence length M, making it impossible to use in real applications.

2.2.2.1 Viterbi's Algorithm

Viterbi's algorithm uses a simplification of the MLSE algorithm and it takes advantage of a special structure called Trellis. The advantage of a Viterbi's decoder compared with the original MLSE is that the complexity of the algorithm is not a function of the sequences length. The algorithm involves calculating a measure of similarity or distance between the received signal $Z(ti)$ and all the trellis paths entering each state at time t_i, discarding those trellis paths that could not possibly be candidates to the maximum likelihood sequence. When two paths enter the same state, the path with best metric is chosen, the method is repeated for all the received bits.

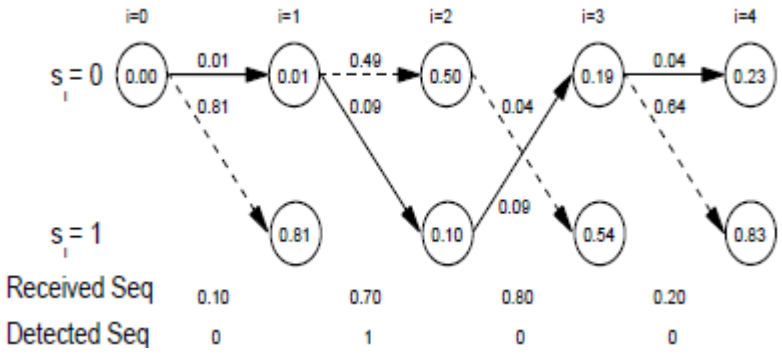


Figure 2.9: Graphical illustration of the Viterbi's algorithm(reprinted from [1]

In the diagram the dotted arrows represent the reception of a bit 0 and the other the reception of a bit 1.

The problem of Viterbi's algorithm is that it requires knowledge of the channel transfer function for the calculation of the weight of each path. The number of operations in the method grows linearly with the size of the sequence l, but the problem is the computation required to store and compute all the paths for the received sequence as the number of states increases with the size of the channel impulse response.

2.2.3 Equalization with Filters

In filter equalization the main idea is to compensate the channel impulse response $Hc(f)$ by introducing a filter $He(f)$ whose impulse response is the inverse of the channel. In this method the equalizer compensates the distorted pulses by reducing the effects of Inter Symbol Interference. Equalization with filters can be made in continuous or in discrete time where the compensation is done using samples of the received waveform. Equalization with filters can also be divided by the nature of their operation equalizing filters can be pre-set or adaptive. Pre-set means that the setup of the filter coefficients is only made at the beginning of the operation, adaptive requires the filter coefficients to be updated as operation takes place.

2.2.3.1 TX FIR equalization

To perform pre equalization of the channel is usually place an FIR filter that pre-distorts the waveform enhancing the high frequency content of the transmitting waveform.This enhancement can be done using *Pre-emphasis* or *De-emphasis*. Using Pre-emphasis we increase the high frequency content of a signal relative to the low frequency content, de-emphasis decreases the low frequency content relative to the high frequency content. The amount of emphasis is usually specified in the value of the coefficients of the TX FIR filter. The following example shows the impact of the emphasis filter on the transmitted waveform.

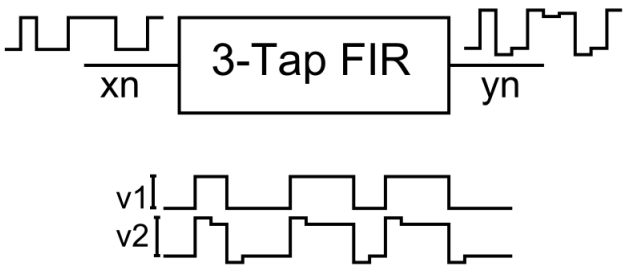


Figure 2.10: Impact of emphasis on the Tx Waveform

The increase in voltage can be expressed as:

$$dbIncrease = 20\log(\frac{V2}{V1})$$

The amount of voltage swings(difference between the voltages in the two differential lines) in the waveform is a function of the number of the taps of the equalizer

$$\text{Number of voltages} = 2^{\text{number of FIR Taps}-1}$$

The number of taps in the Tx equalizer allows us to better compensate the overall frequency response,however the use of a big number of taps is not advised due to power restrictions in the equalizer.

Disadvantages of emphasis The emphasis process increase the signal edge rate which increases the cross-talk on the neighboring channels. Meanwhile, because pre-emphasis emphasizes the transition bits and de-emphasizes the remaining bits, if there is any discontinuity along the channel, the reflection at the discontinuity is more complicated to deal when using emphasis.

2.2.3.2 Continuous time linear Equalizers

In Continuous time linear equalizers (CTLE) equalizers the compensation is done by compensating the channel attenuation at high frequency by introducing a high frequency boost, thus increasing the effective bandwidth of the channel.

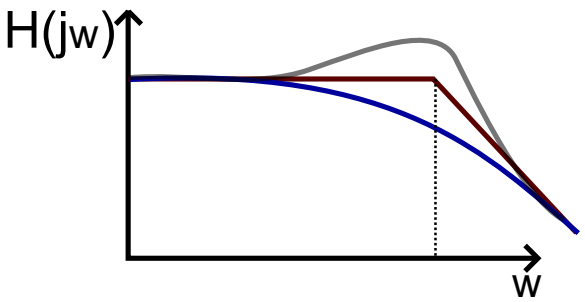


Figure 2.11: Compensation Scheme

By increasing the effective bandwidth ISI becomes less significant and it reduces the probability of error. The increase of bandwidth is achieve with the introduction of a zero near the cut-off frequency of the channel :

$$He(s) = K\frac{s+\omega_z}{s+\omega_p} \quad \omega_z < \omega_p$$

Compensating with higher order systems with more than one zero can also be achieved but generally with worse results, because phase linearity becomes an issue with higher order filters.

2.2.3.3 Circuit realization of CTLE

As we have seen earlier the desired response of the equalizer is a response with a zero and a pole.This is not possible due to the introduction of high frequency poles imposed by the electronic components. Equalizers are physically implement with a differential pair as seen in 2.12:

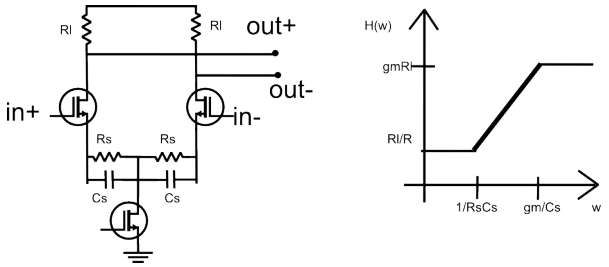


Figure 2.12: Simple Equalizer schematic

Using simple circuit analysis we can observe that for low frequencies the circuit behaves as common source with source resistance, providing a DC gain equal to R_l/R_s. The high frequency gain equals $g_m R_l$ as the circuit behaves as common source amplifier. The position of the zero is controlled by the capacity Cs and the value of the trans-conductance of the transistors of the differential pair. The actual response does not stabilize at *gmRl* due to high frequency poles introduced by the parasitic capacitances of the transistors.

Other circuits derived from the previous are also used:

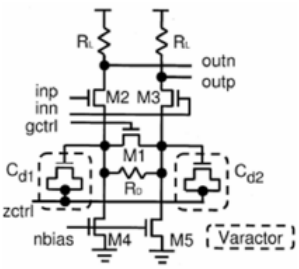


Figure 2.13: Equalizer Schematic reprinted from [9]

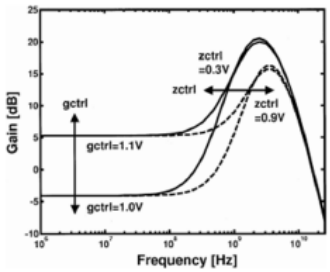


Figure 2.14: Equalizer Transfer Function reprinted from [9]

This circuit was presented for equalization at $3.5 Gbits/s$ in [9] and the parameters of the equalizer transfer function can be tuned by providing two control voltages, $zctrl$ controls the frequency of the zero and $Gctrl$ controls the initial Dc gain.

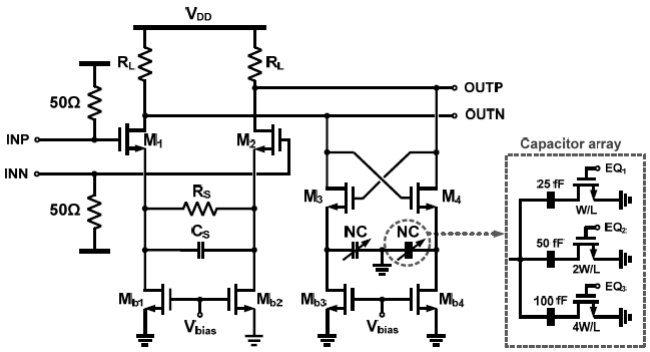


Figure 2.15: Equalizer schematic reprinted from [9]

This circuit, presents very low power consumption $2.46 mW$ during normal function. The tuning of the equalizer is based in capacitive source degeneration and configures the CTLE gain with a variety of 8 different gain stages separated with a $1.5 dBs$ increment.
The next circuit also allows the tuning of different high frequency boosts controlled with a control voltage $Vctrl$ reprinted from [11]

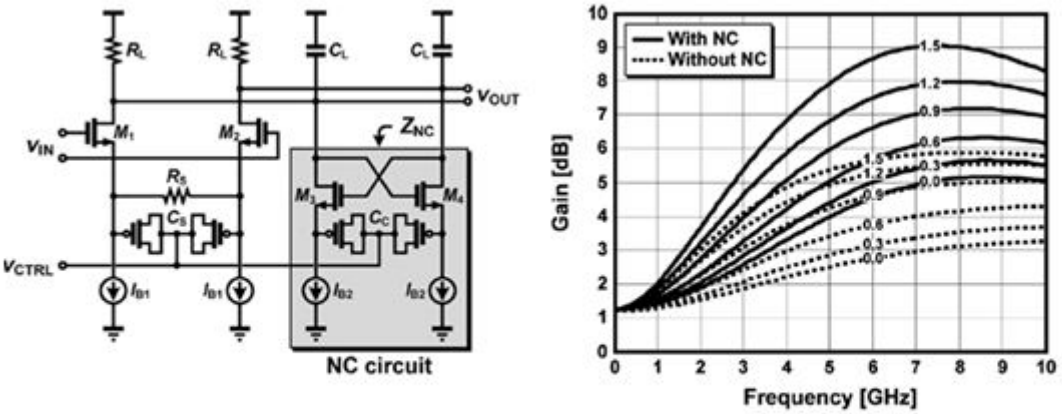


Figure 2.16: Equalizer schematic and transfer function

2.2.4 Adaptation techniques for CTLE

In this section some adaptation techniques are studied for the tuning of the CTLE's operation. As we discussed earlier the problem with CTLE's is finding the equalizer transfer function that better compensates the high frequency loss presented in the channel so that:

$$H_c(jw)H_e(jw) = K,\ \forall\, w(\text{bit Rate}/2)$$

So the adaptation algorithms must be able to tune the CTLE to better compensate the channel.

2.2.4.1 Asynchronous Under Sampling Histogram

Is based on the assumption that by under sampling the waveform after the CTLE and constructing a histogram based on the amplitude of the samples, we are able to evaluate the impact of the CTLE on the eye opening. The voltage histogram with lowest variance δ^2 represents the better eye opening. Notice that the lower the variance the higher the peaking factor in the histogram.

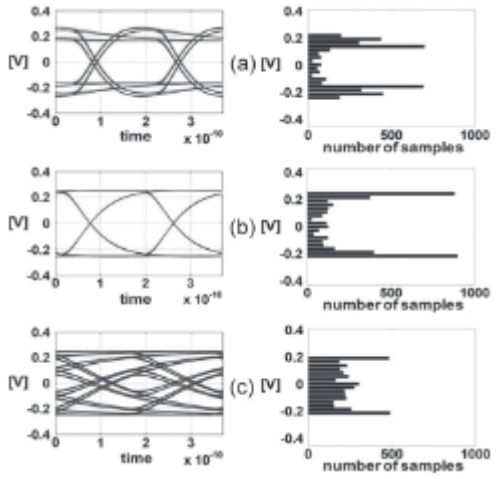


Figure 2.17: Relation between eye opening and Under sampling Histogram (reprinted from [**?**])

In a channel with no noise and no ISI the histogram would only present two peeks representing the two voltages the "0" or the "1" bit. Adaptation following this method is theoretically simple, due to the existence of a small set of predefined coefficients for the equalizer. While in the adaptation period all the CTLE coefficients are tested and a histogram of the samples is made, the coefficients are selected so that the histogram with largest peak is selected. The problem with this approach is that the adaptation process usually takes some time, because we need to transmit a large sequence and analyze it to make the histogram a valid measure of the equalizer performance. In spite of this, this adaptation technique is one of the most widely spread for tuning CTLE coefficients due to its simplicity and low power circuit implementation.

2.2.4.2 Power Sensing

Adaptation through power sensing follows a different approach from the previous one. In this process the idea is to adjust the equalizer based on difference between the power of the high frequency and low frequency components of the received signal.

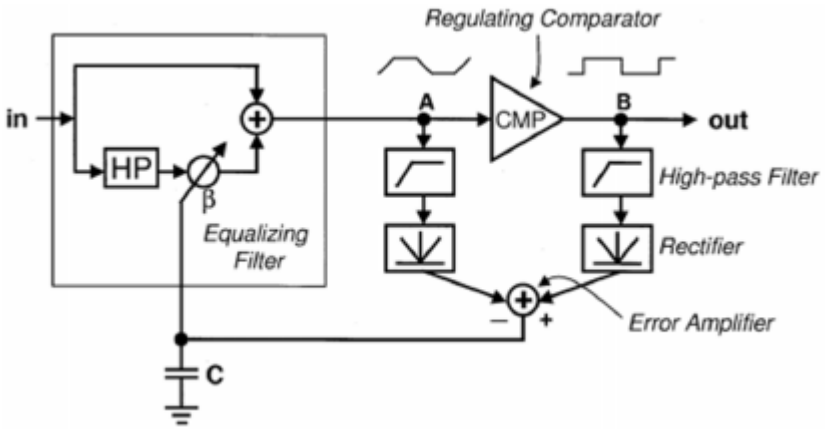


Figure 2.18: Adaptation using power sensing as described in [2]

In this scheme there are two different paths in the receiver the the unity gain path, and the high frequency boosting gain path, the boost gain control is controlled by the difference of powers between the received signal and the recovered signal after a regulating comparator. The power is detected with a filter followed by a rectifier and the difference of powers controls the position of the zero in the equalizer. Another more complete version of the adaptive CTLE is also presented in the same article.The new version also controls the gain at low frequencies 2.19 .

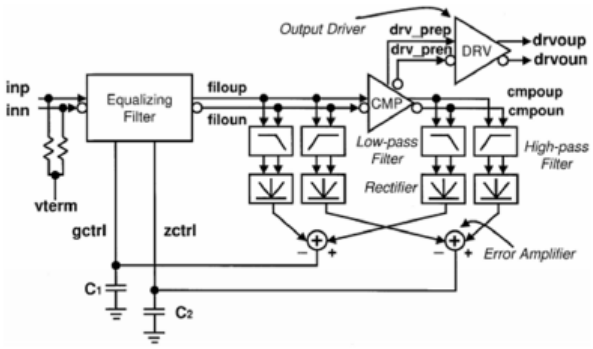


Figure 2.19: Adaptation using power sensing as described in [2]

2.2.5 Discrete Time Equalizers

In this section the compensation of ISI is accomplish using discrete filtering, we will discuss techniques of adaptation using training sequence and blind adaptation methods. In Discrete Time equalizer the compensation is based on the samples of the received sequence, as we seen earlier. If the impulse response of the channel is known the optimal receiver uses the MLSE method, but this method is not practical in very high speed communication due to the computation complexity of the algorithm.

Consider the following simplification of the discrete channel:

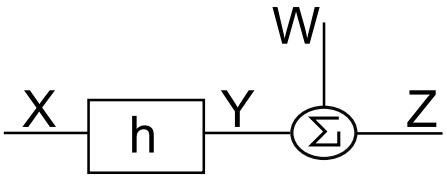


Figure 2.20: Discrete channel model

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h_c(n-k) \text{ and } z = w + y$$

The idea is to introduce a filter in series with the channel so that:

$$h_c(n) * h_e(n) = \delta(n)$$

Taking the z transform results in:

$$H_e(z)H_c(z) = 1$$

Graphically the problem with discrete equalization can be resumed in figure 2.21:

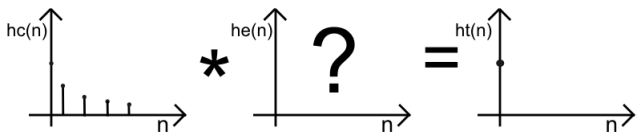


Figure 2.21: Discrete equalization

2.2.5.1 Transversal equalizers

By the properties of the linear convolution we can see that for exact cancellation we need an IIR filter equalizer to make the perfect cancellation of ISI.

Example Consider the channel impulse response with a single post cursor ISI we want to find the

coefficients of a three tap equalizer that reduces the ISI.

By making the linear convolution, flipping and sliding the equalizer response over the channel response :

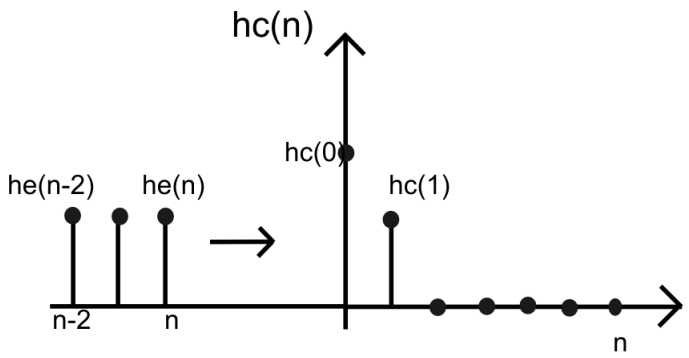


Figure 2.22: Convolution between equalizer and channel

We obtain the following equations for the overall response:

$$\begin{cases} he(0)hc(0) = 1 \\ hc(1)he(0) + he(1)hc(0) = 0 \\ he(2)hc(0) + he(1)hc(1) = 0 \\ he(2)hc(1) \neq 0 \end{cases}$$

If we consider he(0)=1 we get a value for the remaining ISI equal to

$$hc(1)\frac{h(1)^2}{h(0)^2}$$

For the case of a N tap equalizer and a two tap channel response we get the expression for the last tap of the overall response equal to:

$$ht(2+N-1) = he(1)\frac{he(1)}{he(0)}^{N-1}$$

The transversal filter is one of the most popular form of an easy adjustable equalizing filter and the impulse response of the equalizer is the same as the filter coefficients

$$z(n) = \sum_{k=-N}^{+N} c_k x(n-k)$$

Figure 2.23 presents the basic structure of a linear equalizer.

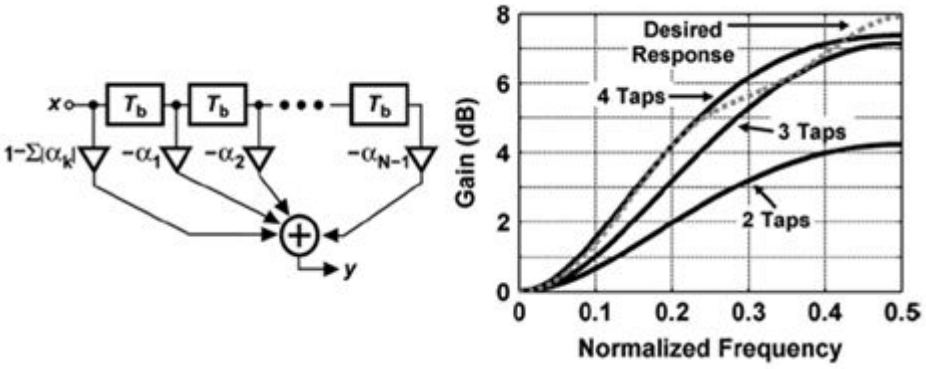


Figure 2.23: Linear equalizer

The transversal equalizer consists in a delay line with 2N T-second taps (T=symbol duration). The output of the equalizer is calculated through the weighted sum of the received samples by the filter coefficients, being the central the main contribution to the value of the output. The central tap corresponds to the current symbol to be calculated as other taps produce echoes to cancel the ISI in the current symbol.The basic limitation of the linear equalizer are that it performs poorly on channels having spectral nulls and the propagation of noise through the filter.

2.2.5.2 Zero Forcing equalizers

The Zero forcing equalizer makes the equalizer transfer function equal to the inverse of the channel transfer function in a limited number of points equal to the equalizer length:

$$He(z) = \frac{1}{Hc(z)}$$

In the zero forcing solution the coefficients are chosen so that:

$$z(k) = \begin{cases} 1 \; for \; k = 0 \\ 0 \; for \; k = \pm 1, \pm 2, ..., \pm N \end{cases}$$

The number of taps of the equalizer equals 2N. In the zero tap equalizer one can only guarantee zero Inter Symbol Interference to the 2N adjacent bits of the sequence in relation to the current sampled bit.

Let us consider the following array definition:

$$Z = \begin{bmatrix} z(-N) \\ . \\ . \\ z(0) \\ . \\ . \\ .z(N) \end{bmatrix} \quad C = \begin{bmatrix} c_{-N}) \\ . \\ . \\ c_0 \\ . \\ . \\ .c_N \end{bmatrix} \quad X = \begin{bmatrix} x(0) & x(-1) & . & . & x(-N) \\ x(1) & x(0) & . & . & .x(-N+1) \\ . & . & . & . & . \\ x(N) & x(N-1) & . & . & x(0) \end{bmatrix}$$

Z represents a vector with the received samples to be presented to the element making the symbol decision.**C** is the array with the equalizer coefficients. **X** represents the samples present in the equalizer, X is a Toeplitz matrix. We can write the following equation:

$$Z = XC \rightarrow C = X^{-1}Z$$

By solving this equation we can make the ISI equal to zero in the 2N side lobes.
The length of the filter who performs zero forcing equalization is a function of the ISI introduced by the channel.
With a finite number of taps the ISI is minimized and the solution is optimal in terms of reduction of ISI at the sampling points. The ZF solution requires initial eye opening to perform equalization. The process uses a an estimate of the impulse response of the channel thus making it inappropriate for equalizing time variant channels.

2.2.5.3 Decision Feedback equalizers

One of main problem with the equalization using transversal filters was the impossibility of obtaining an IIR equalizer response, decision feedback equalizers can realize an IIR response because it uses both a forward and a feedback filtering paths. The idea behind the DFE is that if the values of the past decisions are known (decisions are assumed to be correct) then the ISI introduced by these symbols is subtracted to make the current decision. In the DFE the feedback filter uses previous quantized samples and because of this the output of the feedback filter is free of noise.

The sequence produced at the output of the channel equals:

$$z(k) = y(k)hc_0 + \sum_{j \neq k} y_j hc_{k-j}$$

y(k) represents the received symbol at time k

The summation represents the ISI so the basic idea of the decision feedback equalizer is to simply subtract the ISI.

$$z(k) = y(k) - \sum_{j \neq k} y_j hc_{k-j}$$

The transfer function of the DFE equals:

$$H_e(z) = \frac{A(z)}{1+B(z)} = \frac{\sum_{n=0}^{M} a_n z^{-n}}{\sum_{n=0}^{N} b_n z^{-n}}$$

Where b_n and a_n represent the value of the coefficients of the feedback filter and transversal filter, M and N is the size of the transversal and feedback filter.

Again the problem with equalization is to find the coefficients of both filter in order to achieve the best reduction of ISI fed into the decision device. Assuming that the output of the equalizer is given by:

$$z_k = \sum_{j=-M}^{0} y_{k-j} h_{e,j} + \sum_{j=1}^{N} z_{k-j}^{d} h_{e,j}$$

The first summation gives the influence of the transversal filter on the symbol zk and the second the influence of the feedback filter on the output, note that the feedback filter uses already decided symbols z_{k-j}^{d} thus meaning that the feedback samples don't have the influence of noise.
Defining the following vectors:

$$IF = \begin{bmatrix} z_{k-1}^{d} & z_{k-2}^{d} & z_{k+N-1}^{d} & . & . & z_{k-N}^{d} \end{bmatrix}^T$$

$$IB = \begin{bmatrix} y_{k-1} & y_{k+N-1} & y_{k+N-1} & . & . & y_k \end{bmatrix}^T$$

$$hE,B = \begin{bmatrix} h_{E,-M} & h_{E,-M+1} & . & . & h_{E,0} \end{bmatrix}^T$$

$$hE,F = \begin{bmatrix} h_{E,1} & h_{E,2} & . & . & h_{E,N} \end{bmatrix}^T$$

Where:

- IF represents the vector with the previous N decisions
- IB represents the vector with the M samples at the entrance of the equalizer
- hE B represents the transversal filter coefficients
- hE F represents the Feedback filter coefficients

The filter coefficients are chosen to minimize the square error function $E[Zk - Xk]^2$. Resulting in the optimal filter coefficients:

$$hE,B = (E(IBIB^T) - E[IBIF^T]E[IFIB^T])^{-1} E(IFIK)$$

$$hE,F = -E[IFIB^T] hE,B$$

Where E denotes expectation. If we not know the value of the expectation values á priori we can estimate them using alternate process as we will see next.

2.2.5.4 Adaptation of discrete equalizers

As operation takes place the channel characteristics change. The optimum equalization parameters that were calculated at the beginning of the transmission might not be optimum later on. One solution would be to update the equalizer coefficients with periodically, but the majority of adaptation algorithms needs the transmission of a training sequence that would consume much time that could be used transmitting necessary information.
The tap weights of the equalizer can be updated periodically or as operation takes place being referred as decision directed. Decision directed equalization tries to minimize the error between the output and input of the decision element to adjust the feedback filter coefficients. The decision direct technique can have problems with convergence if the eye at the input of the equalizer is initially closed. In this case, the equalizer taps need to be initialized using an alternate process .
Let us consider the following structure of a system with adaptive equalization

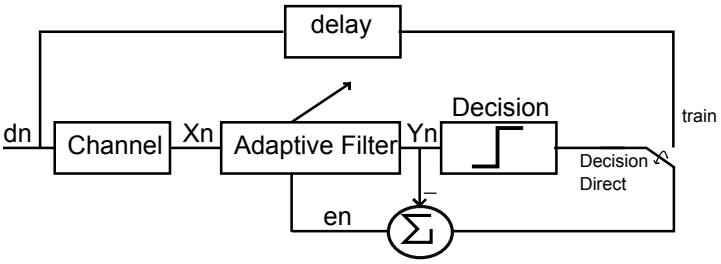


Figure 2.24: Adaptive equalization structure

Figure 2.24 represents the usual adaptation process done in two stages, first calculate the initial coefficients of the equalizer with a training sequence and then switch to decision directed mode.
We see that the adaptive filter is updated using an estimate of the error signal $[d(n) - y(n)]$. The most common method for adaptive equalization is know as the **Least Mean Squares Algorithm (LMS)**.The LMS algorithm consists on the minimization of the quadratic error between the desired response and the signal before the decision device. Let us consider the following model that uses the least mean squares algorithm to adapt a transversal filter to the channel:

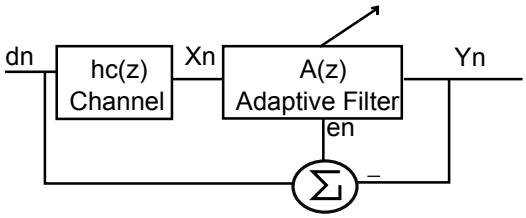


Figure 2.25: Adaptive equalization scheme with transversal filter

The quadratic error is from now on defined as ε.

$$\varepsilon = E[e^2[n]] = E[(d[n] - y[n])^2]$$

By the properties of the expected value we get:

$$\varepsilon = E[d^2[n]] + E[y^2[n]] - 2E[y[n].d[n]]]$$

In the case of no feedback structure the quadratic error becomes:

$$y(n) = \sum_{j=0}^{N} c_j x(n-j) = c^T x(n) \longrightarrow \varepsilon = E[d^2[n]] - 2c^T p + c^T Rc$$

In this equation the quadratic error is a dependence on:

- $E[d^2[n]]$: variance of the desired response
- $2c^T p$: where p represents the cross correlation between the desired response and y(n)
- $c^T Rc$: where R represents the self correlation of y(n)

Notice that the expression makes perfect sense, as the quadratic error increases with the cross correlation between the desired response and the the output of the equalizer, meaning that the equalizer already reached a state where its transfer functions is the inverse of the channel. The expression encountered before defines the dependence of the quadratic error as function of the filter coefficients and its called *performance surface*.

The perfect equalizer is found when the quadratic error is minimum for that coefficients. So we define the gradient vector as:

$$\nabla(\varepsilon) = \frac{\partial E[e^2(n)]}{\partial c} = -2p + 2Rc$$

The local minimum is found when $\nabla(\varepsilon) = 0$

$$c_{opt} = R^-1p; \;\; \varepsilon_min = E[d^2(n)] - p^T c_{opt}$$

Now considering that the coefficients are being updated in time we can define the update dynamics as function of the gradient of the quadratic error:

$$c_{(n+1)} = c_{(n)} - \mu \frac{\partial E[e^2(n)]}{\partial c} \Leftrightarrow c_{(n+1)} = c_{(n)} - 2\mu E[e(n)x(n)]$$

This is know as the **gradient algorithm**, where μ represents the adaptation step. The problem with the gradient algorithm is the calculation of the estimate of $E[e(n)x(n)]$ so the LMS algorithm uses a simplification that consists in replacing the average values of e(n) and x(n) by their instant values witch results in:

$$c_{[n+1]} = c_{[n]} - 2\mu e(n)x(n) \;\; , \;\; 0 < \mu < \frac{1}{NE[x^2(n)]}$$

2.2.5.5 Adaptation of Decision Feedback Equalizers

The adaptation of the DFE can be made using the LMS algorithm seen earlier. The LMS algorithm tries to reduce the quadratic error between the desired response and the response at the entrance of the decision element. The DFE can work in one of two modes, training mode or decision direct mode.

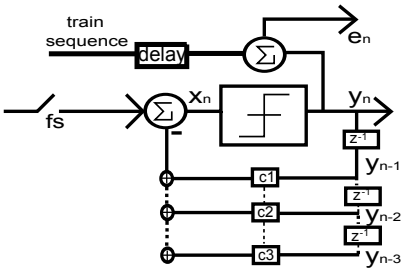


Figure 2.26: Training mode

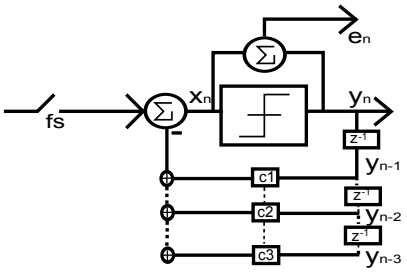


Figure 2.27: Decision Direct mode

The expression for the coefficient adaptation can be expressed as :

$$c_{(n+1)} = c_{(n)} - 2\mu e(n)y(n-1)$$

Lets analyse this equation more carefully as it is the core of LMS adaptation.
If the error is calculated based on the receiver having a replica of the transmitted sequence and simply calculating the difference between the transmitted bit and the value that is at the entrance of the decision element, the equalizer knows the effects of the channel on the transmitted bit and can calculate the coefficient taps that best compensate distortion. If the equalizer on the other hand is working in *decision direct*, if the decision element makes the wrong decision, the equalizer will be working on the assumption that the distortion suffered by the bit is different from the reality, this will cause the equalizer to converge to an improper value. On the other hand the information that is sent is also an important factor, as the equalizer taps will converge to the value that best compensates the distortion caused by the channel in that particular sequence used in the training. So in order to better train the equalizer the training sequence must be as random as possible to better adapt the equalizer to the channel and not to that particular sequence. The simplification used in LMS for the value of the gradient makes that the coefficients do not converge to the optimal solution instead it oscillates around it. This oscillation is called *Gradient noise*. Other algorithms use different approaches for the calculation of the gradient and can make the adaptation step variable, despite all, the LMS is the most used due to its simplicity and easy implementation. To decrease the computation to implement the LMS some simplifications can be made in the original expression for the error calculation, in the next example is shown the coefficient update equation for the sign LMS:

$$c_{(n+1)} = c_{[n]} - 2\mu e(n)y(n-1)$$

$$c_{(n+1)} = c_{[n]} - 2\mu sign(e(n))y(n-1)\ sign(x) = \begin{cases} 1\ for\ x > 0 \\ -1\ for\ x < 0 \end{cases}$$

This simplification implies that the taps of the equalizer will converge always with fixed step μ and will only converge for the value of the post-cursor ISI, if the channel introduces decision errors. As the step size is fixed and the error is binary, the implementation becomes severely simplified at the cost of convergence speed and steady state error.

2.3 Final remarks

In this Chapter we reviewed some important topics about aspects that degrade communication quality in high speed serial interfaces and the need of equalization to reduce the effects of channel distortion and jitter in binary transmission

The focus of the study was on techniques to implement and adapt CTLE and a DFE to the transmitting environment.The method of asynchronous under-sample Histograms showed promising results in the adaptation of a zero pole and two pole linear equalizer.The LMS algorithm also proved to be a possible solution to the adaptation of a decision feedback equalizer. In the next chapters a study of the applicability of these techniques to the implementation of adaptive equalization in a high speed interface will be conducted.

Chapter 3

System Specification

As mobile industry continues to increase, companies in order to succeed in such competitive market need to provide the customers products with the best technologies in the market at the lowest costs possible. The increase in processor computation power brought the increase in complexity of mobile phones, that today offer functionality which are closer to a computer than to a simple phone. The increase in computation power was kept up with the evolution of peripheral such as cameras and displays. This led to the urge of the establishment of a protocol that could connect a high variety of peripherals with the processor in mobile devices. Such protocol would need to be flexible to be easily adapted to multiple interfaces, transmit at a high bit Rate to serve the high speed requirements of some applications and yet maintain a low power consumption required by the mobile industry. This protocol became a standard in mobile devices for the communication in system–on-chips ranging from smartphones, tablets and notebooks, the protocol established standards both in hardware and in software allowing the creation of hardware products that work seamlessly with numerous processors and components from multiple vendors.
Synopsys is one of the major contributors to the diffusion of this protocol in mobile industry, providing IP (controller + physical layer) solutions to its clients. This IP will serve as the basis for the establishment of more complex protocols that benefit from the abstraction provided.
This work is focused on the study and implementation of a architecture of adaptive equalization for the physical layer associated with the referred protocol. The rest of this chapter will provide an additional insight on the physical layer studied denominated from now on as *PHY*.

3.1 The PHY

As said earlier Synopsys offers solutions as a combination of a physical layer denominated as the *PHY* and a controller that interacts directly with the PHY. The PHY is a physical layer optimized for high speed and low power operation. The flexibility of being used in multiple applications is introduced by the ability of the PHY to switch between high speed mode and low power mode in only a few micro seconds.

3.1.1 Architecture of the PHY

The PHY can work both in Mesochronous and Plesiochronous clock mode because of the capability of performing both phase and frequency recovery. The PHY clocking flexibility makes it even more suitable for a larger number of operations, as the receiver and the transmitter do not need to share the same reference clock.

The PHY can have multiple data Links and each link has a TX and a RX which is fixed (one must define the direction data flow).Each link has its own configuration settings. By configuring how many links are transmitting and at what bit Rate, the PHY can be a very physical layers as one can further trade-of easily between link bandwidth and power consumption. The following picture demonstrates a typical mode of operation between two PHY's.

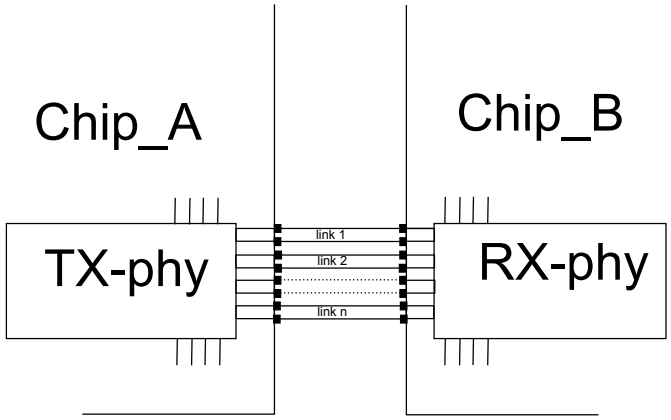


Figure 3.1: Typical PHY mode of operation

The PHY supports two main transmission modes: Low speed mode supporting bit Rates from $3Mbps$ to $0.5Gbps$ and High Speed mode with bit Rates that are expected to reach the $12Gbps$.

The following table shows the bit Rates used in high speed(HS) mode.

Table 3.1: High speed bit Rates

	Type A(Gbps)	Type B(Gbps)
HS Bit Rate 1	1	1.5
HS Bit Rate 2	2.5	3
HS Bit Rate 3	5	6
HS Bit Rate 4	10	12

The transmitted waveform consist in a differential signal with configurable differential voltages, common mode voltage and configurable bit Period. These settings are chosen accordingly to the current mode of operation and physical characteristics of the channel.

3.1.2 Characterization of the PHY

In this section we will study the characteristics of some blocks present in the PHY, both in the TX and in the RX, the study will obviously be focused in blocks that are relevant for the theme of the work. This information is necessary because it is important to know the environment in witch the adaptive equalization will be implemented.

3.1.2.1 Characterization of the TX-phy

In the following image is a highly simplified version of a block diagram of a TX block of the PHY.

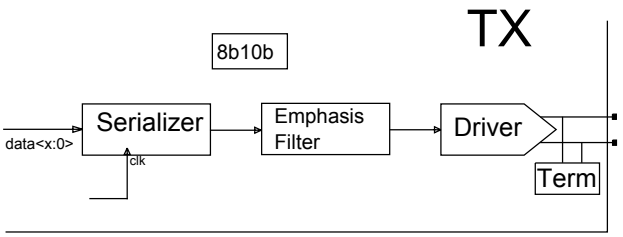


Figure 3.2: Block diagram of the TX PHY

The transmission block consists of:

- *Term:* line termination, adapted to the channel to avoid reflections.
- *Serializer:* To performs parallel to series conversion.
- *PLL:* To Generates the reference clock for transmission.
- *Emphasis Filter:* Performs Tx equalization
- *Driver:* Outputs the sequence for transmission.
- *8b10b:* Performs 8b10b conversion in the bit stream.

The termination of the line is configurable by the PHY. The value of the termination influences the differential voltage of the transmitted signal,a non terminated line requires a higher differential and common mode voltage. The emphasis filter consists in a 3 tap FIR filter designed to pre-distort the bit stream emphasizing the high frequency components of the transmitted sequence.
The 8b10b block performs line coding conversion to achieve DC balance of the transmitted waveform and to provide the receiver with more edges to improve clock and data rate recovery (CDR).

3.1.2.2 Characterization of the RX-PHY

In the following image is a highly simplified version of a block diagram of a RX block of the PHY.

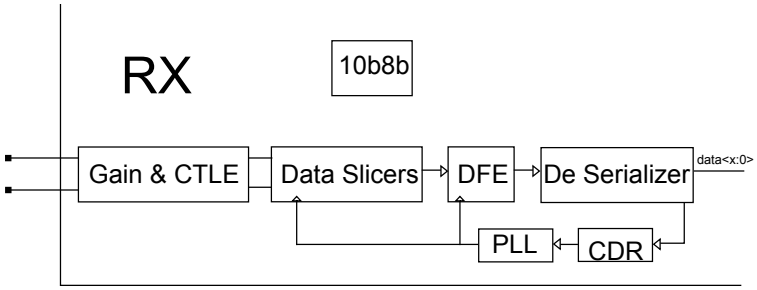


Figure 3.3: Block diagram of the RX PHY

The RX block consists of:

- *Term:* line termination, adapted to the channel to avoid wave reflections.
- *Gain & CTLE:*The CTLE performs analogue RX equalization increasing the high frequency components of the received signal. The Gain increases the signal power for proper detection.
- *Data Slicer:* Sample the Input Waveform at a rate given by the local reference clock.
- *DFE:* Performs RX Digital Equalization of the sampled waveform.
- *CDR:* Performs clock and data recovery based on received waveform.
- *PLL:* Generates a reference clock based on the clock estimation given by the CDR block for sampling the input data.
- *De-Serializer:* performs series-parallel conversion.

The RX CDR performs clock and data recovery block has a circuit capable of generating several clock phases used to perform CDR. The presence of this clock references will help in the task of adapting the CTLE as we will see later on.

3.2 The transmission environment

3.2.1 Transmission environment S-Parameters Simulation

The usual application of the protocol in question is in the transmission between two integrated circuits inside a PCB board.

The losses and distortion that the signal suffer are mainly attributed to the losses in the PCB trace that connects the two chips and their packages.

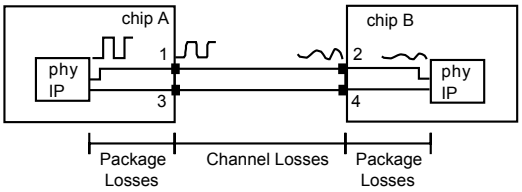
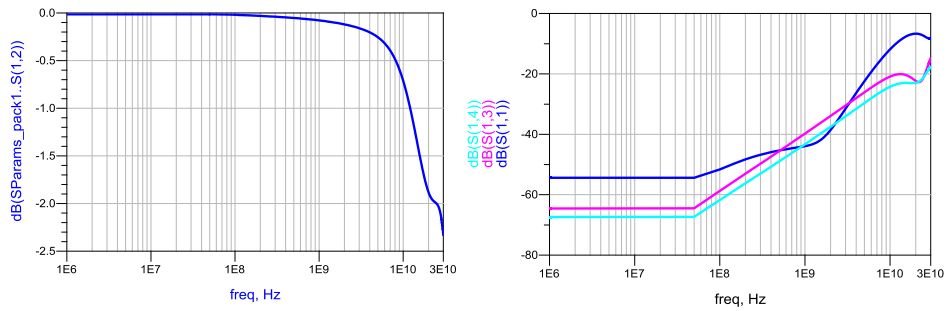


Figure 3.4: Schematic representing the connection between two chips with port identification.

Both the channel and packages responses were described with S-parameters. For the package and channels the models were constructed by inspection in the laboratory using a *Vector Network Analyser(VNA)* during the interface testing. The network analyser works by applying a series of sinusoidal stimuli at each port and measuring the magnitude and phase response at every other port of the network under study.By sweeping the frequencies of the stimula the device can build a sampled description of the network with S-parameters. The process produces a *.snp* file, giving for each point of frequency the magnitude and phase of every element of the S matrix for a n port network. By analysing the content of the .s4p file in ADS, it was observed an expected symmetry in terms of the S matrix, as a result of the intrinsic symmetry of the electrical circuit being analysed. For example the transmission coefficients S_{12} and S_{34} and similarly the reflection parameters show approximately the same frequency behaviour for both input ports. Having this in mind from now we will only analyse the S-parameters for one input Port.
The following figures were obtained in ADS by performing S parameter simulation for the reference package used by the Interface.

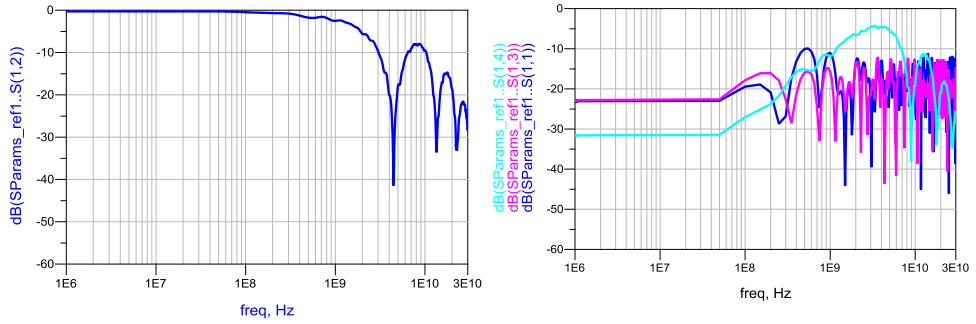


(a) Insertion Loss S_{12}

(b) S_{11} S_{13} S_{14}

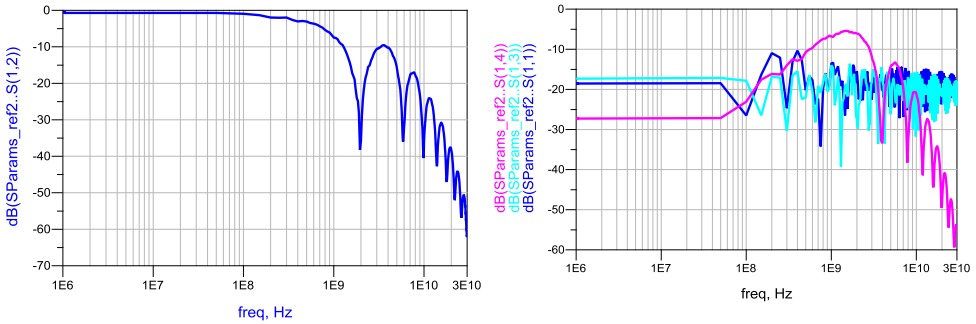
Figure 3.5: Reference package S parameter simulation results

The frequency behaviour of the package insertion loss is quite good presenting a loss of less than $0.5\,dB$ at $10\,GHz$. The reflection and the effects of cross talk increase with frequency. In fact the reflection coefficient reaches the $-10\,db$ at about $12\,GHz$. This aspect does not seems critical since at the Nyquist frequency of the highest bit rate ($6\,GHz$) the packages does not introduce a large amplitude distortion.
Next we will analyse the response for both reference channels following the same methodology. Reference channel models are used to characterize the medium used to connect two chips using the PHY physical layer. Reference channel 1 and 2 set both of the extreme conditions for attenuation and phase response we need to consider for the characterization of the transmission environment. Figure 3.6 3.7 present the results for the S parameter simulation of both reference channels. First we analyse the insertion loss and then we analyse the reflection coefficient and the effects of the cross talk separately.



(a) Insertion Loss S_{12} (b) S_{11} S_{13} S_{14}

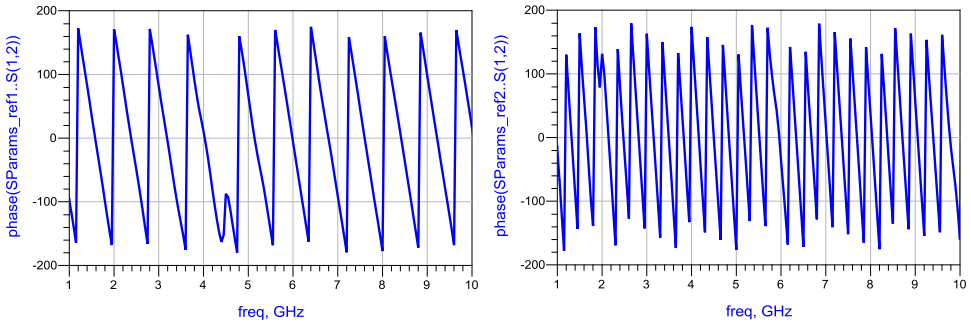
Figure 3.6: Reference channel 1 S-parameter simulation results



(a) Insertion Loss S_{12} (b) S_{11} S_{13} S_{14}

Figure 3.7: Reference channel 2 S-parameter simulation results

If we focus our attention on the insertion loss of both channels we can see that reference channel 1 presents a notch at the frequency of $4.450\,GHz$ where the insertion loss reaches the $-41.449\,dB$. Reference channel 2 also presents a notch but at an earlier frequency of about $1.950\,Ghz$ with the insertion loss reaching $-38\,dB$.
The worst behaviour observed in reference channel 2 corresponds to a PCB trace of longer dimensions than the one of the model of channel 1. Now we will take a closer look at what happens in terms of phase of the Insertion Loss for both channels.



(a) Reference channel 1 S_{12} (b) Reference channel 2 S_{12}

Figure 3.8: Phase of the insertion loss for reference channel 1 and 2

Observing figure 3.8, we notice the effects of the notches of both channels cause an inversion of the phase behaviour,introducing phase distortion. The behaviour near the notch, both channels is almost of linear phase. It is not expected that phase distortion to be a major issue while trying to perform equalization. From the phase of the insertion loss we can estimate the delay caused by each channel to be: $\frac{\partial -\angle(S12)}{\partial f}$

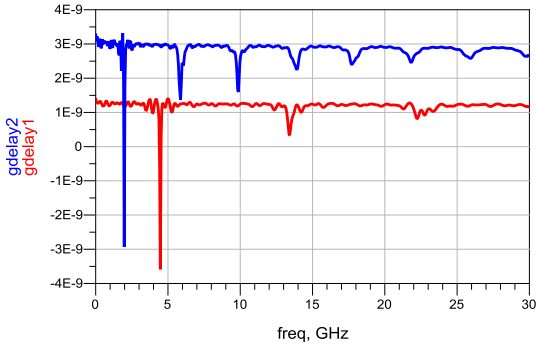


Figure 3.9: Reference channel 1 and 2 group Delay

By inspection we determine that the group delay of channel 1 is about 1.3*ns* and channel 2 is about 2.9*ns*. The group delay of the package is negligible relatively to the channel.
As we can see the insertion loss for reference channel 2 is worse than the one presented by channel 1 and its expected that the eye diagrams to be more closed in reference channel 2 for the same bit rate.

3.2.2 Eye diagrams for the different bit Rates

In order to analyse the integrity of the link and quality of the signal received, the system was simulated in ADS using its powerful capabilities. Eye diagrams were produced as a final result for each channel at the different bit rates used in normal operation of the interface. The parameters used in the channel simulation were chosen to match closely the real operating conditions of the interface.
The simulation setup is presented in figure 3.10:

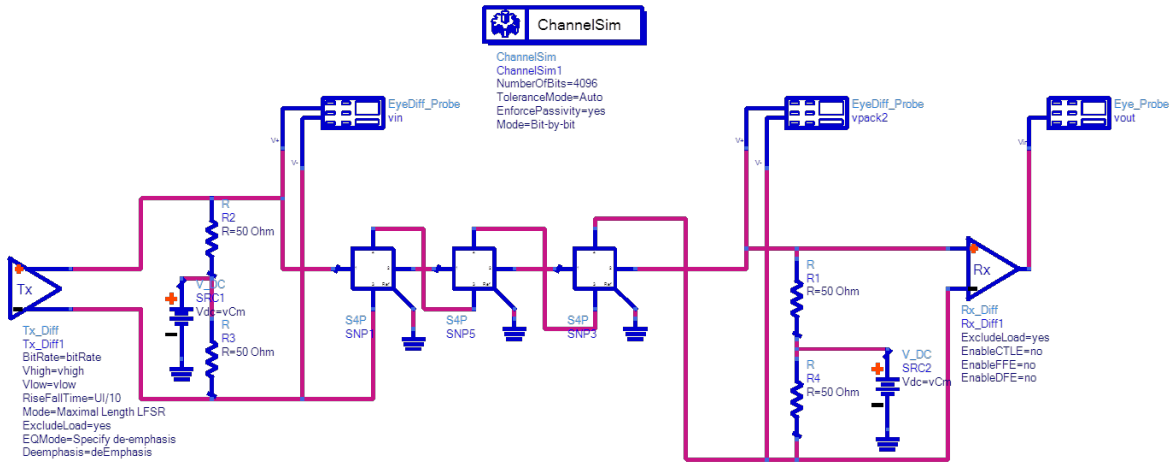


Figure 3.10: Channel Simulation setup in ADS

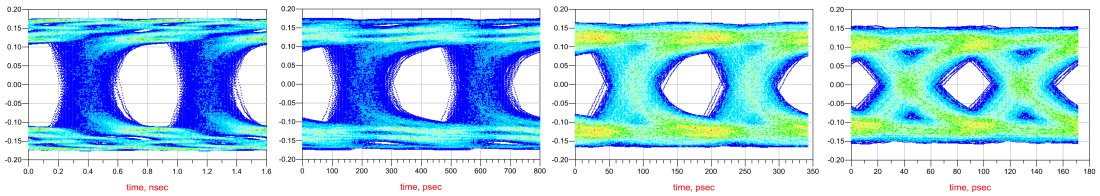
The parameters of the simulation were:

- ADS Channel simulator in bit by bit mode. Transmitting 4096 bits
- Packages and channel model represented by its *.s4p* file (the simulation includes 2 models of the chip packages and a model for the channel)
- Transmitter source
  - Maximum length PRBS source with 8 taps.
  - 160*mV* differential voltage.

  - $0.1 * UI$ of rise and fall time.
  - 8b10b line coding.
  - Periodic jitter Amplitude =5% and frequency= $26.5MHz$.
    Random jitter RMS value= $0.1UI$.

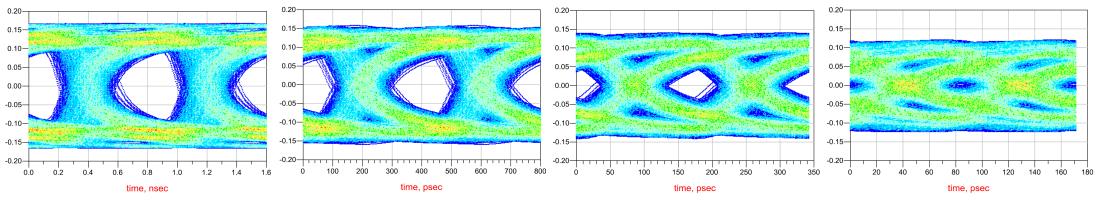
- Eye differential probes to plot the Eye diagram
- Lines terminated with matched load of 50Ω and common mode voltage of $80mV$

Figures 3.11 and 3.12 present the eye diagrams at the receiver using the simulation set-up previously described. Using the different bit Rates as required, for both reference channels.



(a) 1.5Gbps (b) 3Gbps (c) 6Gbps (d) 12Gbps

Figure 3.11: Reference Channel 1 eye diagrams



(a) 1.5Gbps (b) 3Gbps (c) 6Gbps (d) 12Gbps

Figure 3.12: Reference Channel 2 eye diagrams

The following tables present some important information extracted from the eye diagrams:

Table 3.2: Results for the Eye diagram simulation channel 1

Bit Rate	Height(mV)	Width(ps)	JitterPP(ps)	JitterRms(ps)	Q factor
HS Bit Rate 1	197	536.9	264.4	66.28	6.893
HS Bit Rate 2	161	288.5	112.2	27.74	7.607
HS Bit Rate 3	137	103.8	67.75	14.61	6.862
HS Bit Rate 4	108	44.60	37.74	7.755	5.975

Table 3.3: Results for the Eye diagram simulation channel 1

Bit Rate	Height(mV)	Width(ps)	JitterPP(ps)	JitterRms(ps)	Q factor
HS Bit Rate 1(1.5 Gbps)	139	532.3	268.4	66.31	6.854
HS Bit Rate 2(3 Gbps)	110	230.4	170.3	36.00	5.953
HS Bit Rate 3(6 Gbps)	37	80.62	90.91	18.27	3.742
HS Bit Rate 4(1.5 Gbps)	0	13.29	60.92	13.84	2.323

We can see from the results of the simulation that as the bit rate increases, the effects of the band-limited channel become more severe, increasing the amount of Inter-Symbol Interference. This effect is a evidence of the Nyquist criteria, as a higher bit Rate spreads the signal spectrum to a zone where the magnitude of the channel insertion loss is not flat.
Channel 2 presents worst performance compared to channel 1, when simulated under the same conditions. This is due to the poorer behaviour of the channel's 2 insertion loss.
The need for equalization becomes essential at 12 *Gbps*, since transmission at this bit rate violates the eye mask specified for this interface, even when the source is not simulated with the maximum allowed jitter at the transmitter.

3.3 Proposed equalization structure

In the previous section we noticed the need for equalization, so that the interface can transmit at a bit Rate of 12*Gbps* using both reference channels. So to improve this situation the following equalization architecture is proposed:

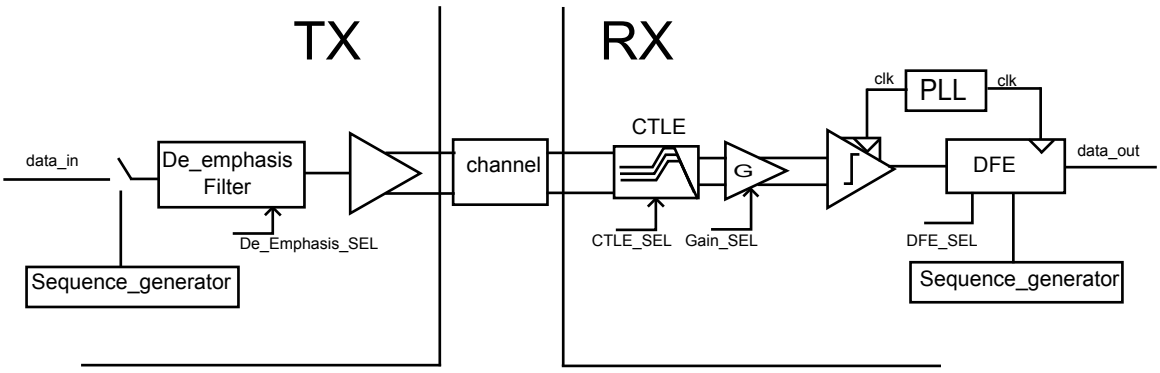


Figure 3.13: Equalization Architecture

In the figure 3.13 can see 4 main blocks that are used in the equalization of the channel.

- The de-emphasis filter consists in a FIR filter with 2 or 3 taps that is used to pre-distort the transmitting waveform. The filter emphasizes the high frequency content of the signal, placing a boost at the Nyquist frequency.

- The continuous time linear equalizer (CTLE) provides a series of configurable settings to boost high frequency content.
- The gain serves as a compensation for the signal attenuation introduced by the channel and possibly by the CTLE.
- The DFE works as a sampled feedback filter designed to eliminate the post cursor ISI.
- The PLL generates the clock for the DFE and data Slicers.

In this scheme the emphasis filter is chosen to implement the best boost for the designed application of the PHY, meaning that at least it should be known à priori that the channel in question has characteristics similar to reference channel or reference channel 2 as such the emphasis filter has its settings fixed since the beginning of its operation. The DFE and CTLE will be adapted at the beginning of transmission and every time the controller specifies a retraining of their settings. The DFE contains an input that defines how the adaptation takes place either choosing adaptation using a training sequence, or adaptation using blind mode. Adaptation of the DFE on training mode requires the transmission of a training sequence, and the local generation of the same sequence in the RX PHY. The comparison of the received sequence and the sequence generated locally generates the error signal required to perform adaptation using the LMS algorithm. On the other hand if the adaptation is done using blind mode, the DFE will perform adaptation without the need of a training sequences, calculating the error signal based on the difference between the signal at the output and input of the decision element. The process of adaptation of these two filters will be approached with more detail in chapter 5.

3.4 Final remarks

In this chapter we characterized the PHY interface in which is intended to implement adaptive equalization, the architecture of both the transmitter and the receiver and the modes of transmission contemplated by the interface were briefly presented.
The model of the channels used in the interface simulation were analysed by means of S-parameters. Results for two reference channels were discussed based on the transfer function of the channels (S12) and eye diagram plots, to assess the channel and package performance at different bit rates. The analysis showed the need for channel equalization After we proposed an equalization architecture intended to compensate for the channel ISI that requires few alteration to the current structure of the Interface.In the following chapters we will discuss in detail each of the equalization blocks present in the proposed architecture.

Chapter 4

Equalization architecture

In this chapter we will study the influence of the different blocks of equalization proposed in the last chapter for transmission at $12Gbps$ using the PHY interface. The equalization structure consists of a transmitting FIR filter with fixed tap values, a continuous time linear equalizer with a set of configurable boosts and an adaptive DFE.
The de-emphasis filter introduces a boost at the Nyquist frequency of the transmission bit Rate with the objective to perform pre-equalization of the channel. The inclusion of TX equalization loosens the demands of the equalizers at the RX PHY. The de-emphasis filter has fixed tap values since the beginning of operation of the interface, so in this chapter we choose the optimum value of de-emphasis that better compensates each reference channel. The CTLE consists of an analogue filter that introduces a high pass transfer function to compensate the channel high frequency loss. In this chapter we will design a CTLE with different boosts and choose the setting that better compensates each reference channel. Furthermore we will see the impact of this equalizer in the overall system impulse response. In the final section we will study the effects of the incorporation of a DFE in the equalization structure, evaluating the impact of this equalizer in the overall eye opening and discover the optimum values for the tap values.
In the end of the chapter we will make a summary of the improvements brought by the proposed equalization structure when performing transmission at $12Gbps$.

4.1 De-Emphasis Study

By placing an emphasis FIR filter in the transmitter we can choose the best amount of high frequency boost that better compensates the effects caused by the channel. The emphasis filter increases the amplitude of the transmitted waveform near the signal transitions, this is achieved by considering two or more replicas of the signal shifted in time. This way the filter has knowledge of when he signal changes.

The next figure shows the influence of the emphasis process in the transmitted impulses.

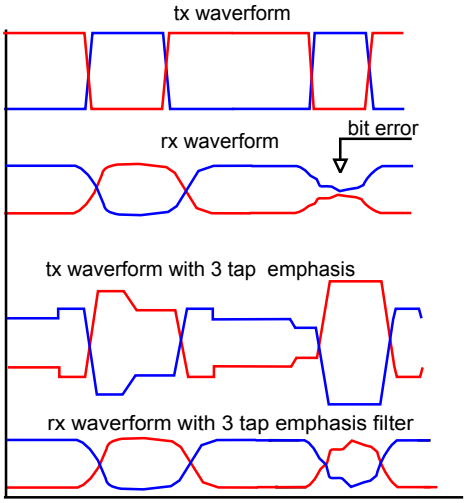


Figure 4.1: Emphasis error correction

We can calculate the tap values from the amount of emphasis expressed in dB using the following procedure. Lets take the example of a a 2 tap FIR filter and a de-emphasis of 2dB :
Consider that the FIR filter consists in a delay line with only one post cursor tap as represented in 4.2

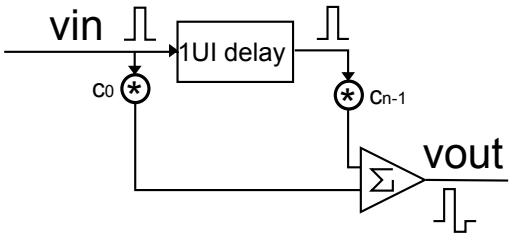


Figure 4.2: 2 Tap emphasis filter

$$C_0 + C_{n-1} = 1$$
$$C_0 - C_{n-1} = 1 * 10^{-2dB/20}$$
$$\Leftrightarrow C_0 = 0.89432; C_{-1} = -0.1034$$

The same process could be used to emphasis filters with more taps but more equations would be required to find the value of the taps.
For the study of the De-emphasis we used the ADS channel simulator with a simulation setup described in 3.10 in chapter 3. The simulation used a two tap FIR filter to introduce de-emphasis in the transmitted waveform.

For this simulation we introduced different amounts of de-emphasis in the transmitter and visualized its impact in the eye diagram at the receiver. Figure 4.3 shows the transmitted waveform for different values of de-emphasis used in the simulation.

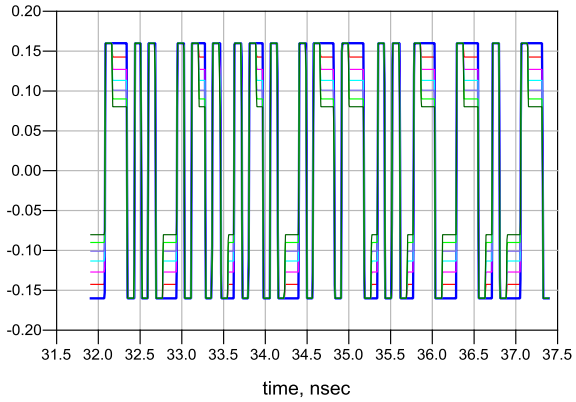
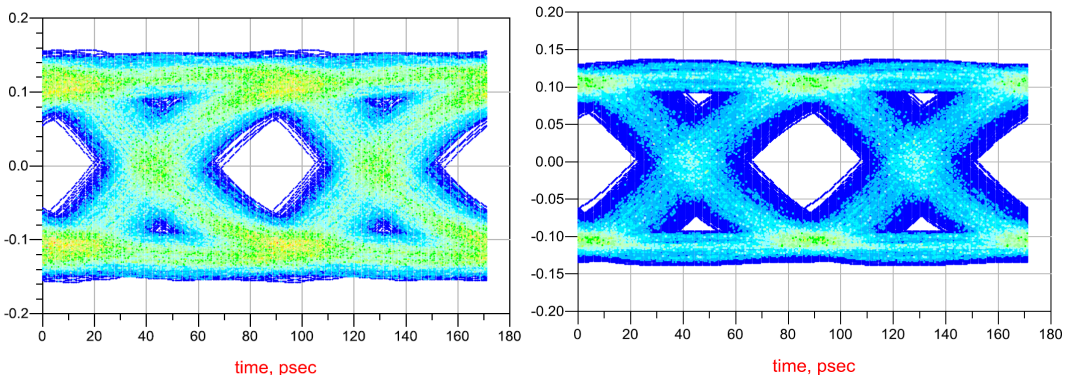


Figure 4.3: Transmitted waveforms for 0, 1, 2 to 6dB of de-emphasis

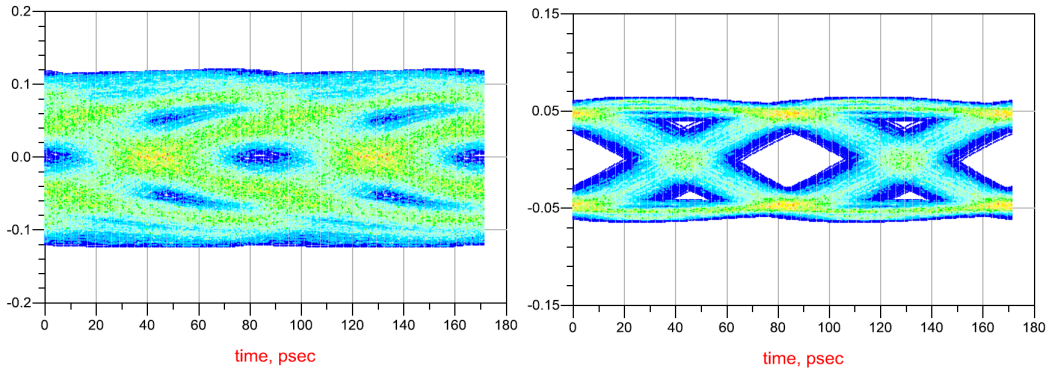
In figure 4.4 and 4.5 is demonstrated the impact of the emphasis process in the eye diagrams at the receiver for both reference channels. Although simulation was made from $0\,dB$ of emphasis to $10\,dB$ only the optimum amount of emphasis is represented for comparison.



(a) 0dB emphasis (b) 2dB emphasis

Figure 4.4: Eye diagrams at the RX PHY using reference channel 1

By visual inspection we can see the improvement brought by the emphasis filter in the simulations with reference channel 1 and 2. In order to proper describe its impact in the eye diagram, the following tables show eye diagram measurements for the experiment.



(a) 0dB emphasis (b) 8dB emphasis

Figure 4.5: Eye diagrams at the RX PHY using reference channel 2

Table 4.1: Eye diagram measurements at the Rx-PHY while using reference channel 1

de-emphasis(dB)	Height(mV)	Width(ps)	JitterPP(ps)	JitterRms(ps)	Q factor
0dB	109	42.02	42.92	8.589	6.039
2dB	134	47.17	38.59	8.681	9.19

Table 4.2: Eye diagram measurements at the Rx-PHY while using reference channel 2

de-emphasis(dB)	Height(mV)	Width(ps)	JitterPP(ps)	JitterRms(ps)	Q factor
0dB	0	13.29	69.22	13.84	2.32
8dB	53	46.74	39.02	8.238	7.22

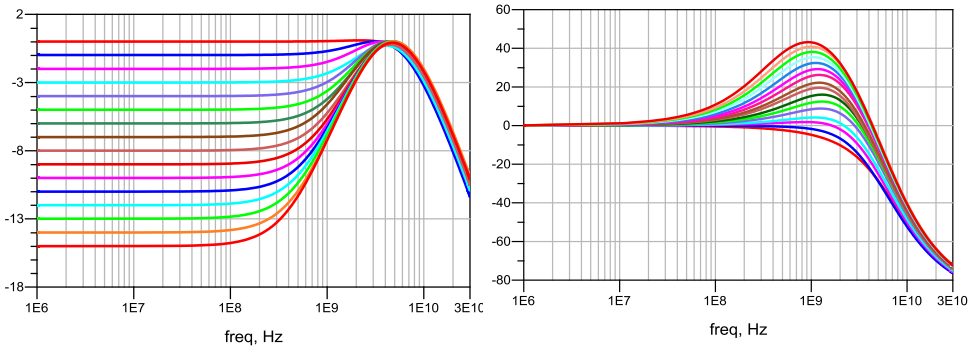
Observing the results for the eye diagrams present in the tables we can analyse the influence of the emphasis filter in performing pre-equalization of both reference channels. For reference channel 1 the optimum amount of de-emphasis is about $2\,dB$, this means that a transition bit has a $2\,dB$ higher differential voltage than a bit locate inside a sequence of 1's or 0's. The increase in differential voltage near the bit transitions opposes the low pass characteristic of the channel reducing the ISI. For reference channel 2, the introduction of an emphasis filter with $8\,dB$ of boost is capable of opening a completely closed eye diagram. Observe that the emphasis filter is able to compensate the ISI induced jitter, reducing the peak to peak jitter almost by half in reference channel 2.

4.2 CTLE design

For the design of the CTLE some considerations were made. For example the CTLE would have to be a passive filter, meaning that the maximum allowed gain would be 0dB. This consideration was made to describe a situation with a lower power consumption because there is no need to supply an active component. Other consideration was that the CTLE would be capable to compensate the effects of the ISI even when transmitting at $12\,Gbps$ this implies that the boost would be placed near the Nyquist frequency at $12/2 = 6GHz$. The CTLE consists in a single zero two pole transfer function with the following transfer function.

$$H(s) = \frac{s+z}{(s+p1)(s+p2)} \text{ with } s = j\omega$$
$$H(j\omega) = \text{DC gain}\frac{1+j\omega/z}{(1+j\omega/p1)(1+j\omega/p2)}$$
$$\text{By taking the magnitude and phase we get :}$$
$$|H(j\omega)| = \text{DC gain}\frac{\sqrt{1+(\omega/z)^2}}{\sqrt{(1+(\omega/p1)^2)*(1+(\omega/p2)^2)}}$$
$$\angle H(j\omega) = \arctan(\omega/z) - \arctan(\omega/p1) - \arctan(\omega/p2)$$

For the calculation of the different settings of the CTLE it was proposed the creation of 16 transfer curves separated by 1 dB at low frequencies, furthermore to simplify we considered the 2nd pole of the transfer function to be fixed at $6Ghz$. The location of the zero and the 1st pole for each of the different boosts was calculated using the ADS optimizer. In order to reduce the load impose to the optimizer we made some restrictions to the location of the zero and the pole. For example the pole and zero could not be separated by more than one decade, because the zero makes the transfer function rise at a rate of $20db/dec$, and the maximum allowed boost is $15dB$. The first pole also should not pass the $4.5Ghz$ barrier because its desired that the CTLE would have some 'bandwidth' at the boosting frequency of at least $1Ghz$. The results from the optimization are the following set of CTLE settings depicted in magnitude and phase:



(a) Magnitude (dB) (b) Phase(degrees)

Figure 4.6: CTLE settings

As we observe from figure 4.6 the amount of phase distortion imposed by the filter is proportional to the amount of boost that it provides. Notice that the phase for the CTLE never reaches 45^o, this is caused by the restriction of the zero and the pole not to be separated by more than one decade. The following table gives the frequency for the zero and the first pole of the curves presented in figure 4.6, consider that second pole is located at $6GHz$:

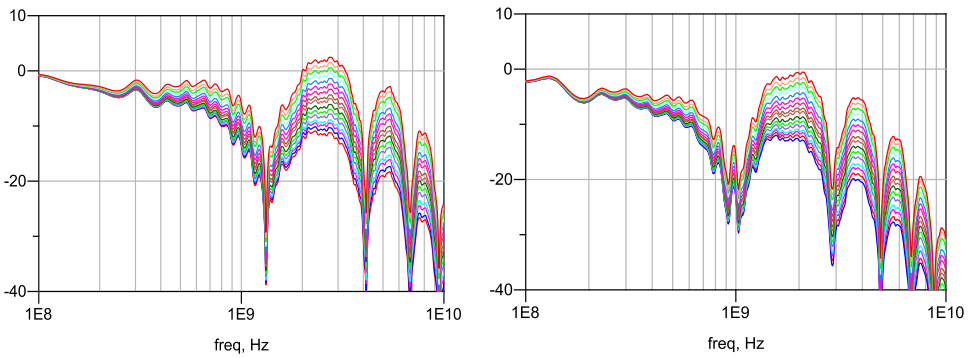
Table 4.3: Location of the 1st pole and zero for different CTLE boosts

boost	zero(Gbps)	pole(Gbps)	boost	zero(Gbps)	pole(Gbps)
0dB	3.810	5.820	8dB	0.964	3.828
1dB	2.389	3.808	9dB	0.864	4.078
2dB	2.059	3.898	10dB	0.764	4.008
3dB	1.894	4.028	11dB	0.674	3.998
4dB	1.614	4.128	12dB	0.594	3.938
5dB	1.414	4.098	13dB	0.534	3.998
6dB	1.244	4.078	14dB	0.484	4.128
7dB	1.044	3.738	15dB	0.424	3.948

By carefully analysing the table we can see that the distance between the zero and the pole determines the amount of high frequency boost introduced by the equalizer. A higher separation between the zero and the first pole allow the zero to introduce a higher gain.

4.2.1 CTLE adaptation

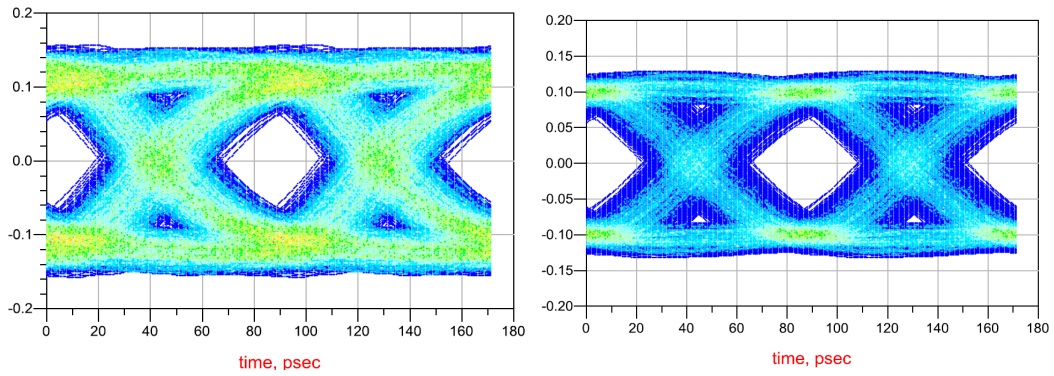
In figure 4.7 we present the transfer function of channel plus equalizer for each of the two reference channels. For visualization purposes the transfer characteristic of the equalizer was brought to 0dB at DC value by a correction factor equal to $10^{dBboost/20}$



(a) Reference channel 1 (b) Reference channel 2

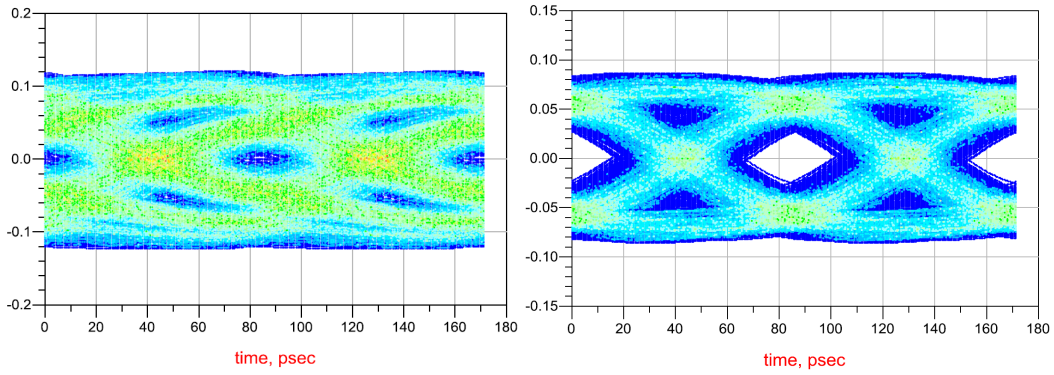
Figure 4.7: Insertion loss of channel plus CTLE

The CTLE presents a large variety of transfer characteristics for the equalization of a wide range of channels. From the figure we can see that when applying the CTLE in series with both reference channels we can achieve situations between under equalization to over equalization. The next figure presents the effects of the CTLE on the received eye diagram for both reference channels, In the simulation we considered 0dB of emphasis at the transmitter and selected the boost that provides the best Quality factor for the receiver eye diagram.



(a) 0dB CTLE boost (b) 3dB CTLE boost

Figure 4.8: Reference Channel 1 CTLE equalization



(a) 0dB CTLE boost (b) 10dB CTLE boost

Figure 4.9: Reference Channel 2 CTLE equalization

Table 4.4: Results for CTLE eye opening channel 1:no CTLE vs Optimum setting

Boost	Height(mV)	Width(ps)	JitterPP(ps)	JitterRms(ps)	Q factor
0dB	94	47.60	38.16	8.125	5.412
3dB	127	46.31	39.45	7.931	8.614

Table 4.5: Results for CTLE eye opening channel 2:no CTLE vs Optimum setting

Boost	Height(mV)	Width(ps)	JitterPP(ps)	JitterRms(ps)	Q factor
0dB	0	4.005	72.94	14.49	2.22
10dB	44	45.45	40.31	8.347	5.044

The results for the CTLE adaptation allows us to draw some conclusions. From the results presented earlier we can notice that the setting that provides the best eye diagram at the receiver, provides almost the same boost as the simulations for the emphasis process in the last section. This means that the equalization effort can be divided between these two filters. For example for reference channel two the high frequency boost required is about 8 to 10 dB, this means that for example the emphasis filter could provide $3dB$ of boost and the CTLE could introduce the remaining 5 to $7dB$. To proper observe the cooperation between the emphasis filter and the CTLE in the equalization process we performed a simulation where is inserted a higher loss channel and we perform CTLE adaptation considering $8dB$ of de-emphasis at the TX-PHY.

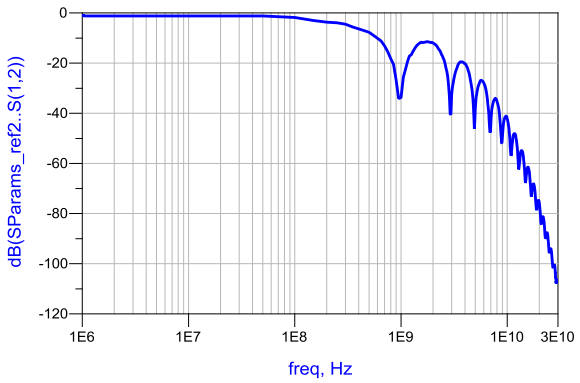
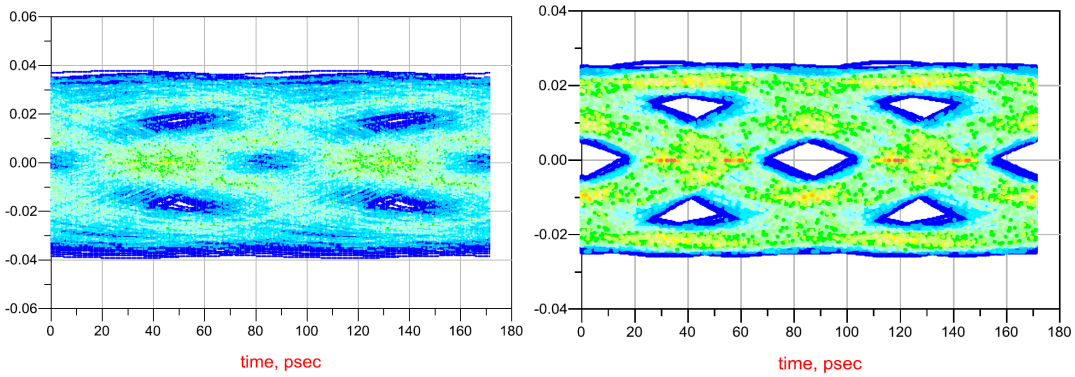


Figure 4.10: Insertion loss of 2 Channels in a series

The eye diagrams at the RX-PHY before and after CTLE adaptation are the following.



(a) 0dB CTLE boost (b) 8dB CTLE boost

Figure 4.11: Eye diagram with 2 reference channel 2 in series emphasis of 8dB

Table 4.6: Eye diagram after CTLE compensation with 2 channels in series -De-emphasis 8dB

Boost	Height(mV)	Width(ps)	JitterPP(ps)	JitterRms(ps)	Q factor
0dB	0	0	85.76	18.94	1.80
8dB	5	33.02	57.89	13.07	2.77

As we can see the insertion of a higher loss channel increases the amount of ISI completely closing the eye diagram even with $8\,dB$ of de-emphasis. From this simulation we can see the benefit of having both equalization at the transmitter and at the receiver, because even by using $8\,dB$ of De-emphasis the receiver still presents a completely closed eye. The high frequency boost introduced by the CTLE further equalizes the channel opening the eye diagram.

4.2.2 CTLE simulation in Spice

To visualize the effect of the CTLE on the channels impulse response we made a simulation environment switch to Hspice. The impulse response of the channel plus equalizer could be easily simulated in Matlab but it is easier to simulate the effects of cross-talk between the networks using Hspice. Another advantage of using the Hspice simulator came from the freedom that was introduced by the creation of the verilog A models for the components used in the simulation. Verilog A models replicate electrical components by describing them by their fundamental mathematical equations. Verilog A follows the same philosophy of Verilog defining portions of the circuit by modules that perform operations in parallel.

The following example describes the differential receiver used in the simulations in spice.

```
module Receiver(inP, inN, out);
    electrical inP, inN, out;
    input inP, inN;
    output out;
    real vout, vdiff;
    parameter real td=0.1p, tr=0.1p, tf=0.1p;

analog begin //marks the beginning of the analog circuit definition
      vdiff=V(inP)-V(inN)
      V(out)<+ transition(vdiff/2, td, tr, tf);
end
endmodule
```

The component parameters described as *parameter* can be altered during the simulation making each module very versatile, for example in this block we can alter the value of the delay introduced by the receiver. The line *analog begin* is a keyword to mark the beginning of an analogue description of the block, the module will do the operation described inside that block at every time step defined by the simulator. The instruction *V(out)<+ transition(vdiff/2, td, tr, tf);* defines a transition filter to introduce slew rate in the transitions when outputting a variable value to an output pin of the module.

The following code instantiates the Receiver module in Hspice simulator:

```
Xre n6 n7 n8 Receiver td=td tr=tr tf=tf
```

The CTLE was described in Verilog A by using the *laplace_zp*() function that implements a m pole n zero transfer function inside a Verilog A module. The CTLE was designed following the architecture depicted in 4.12, in the figure the *CTLE_SEL* is a control signal that chooses the amount of high frequency boost introduced by the CTLE.

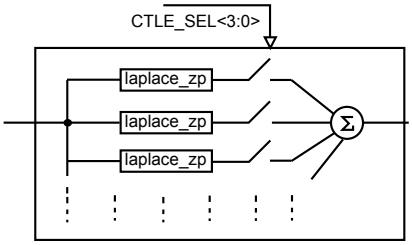


Figure 4.12: Verilog A CTLE Model

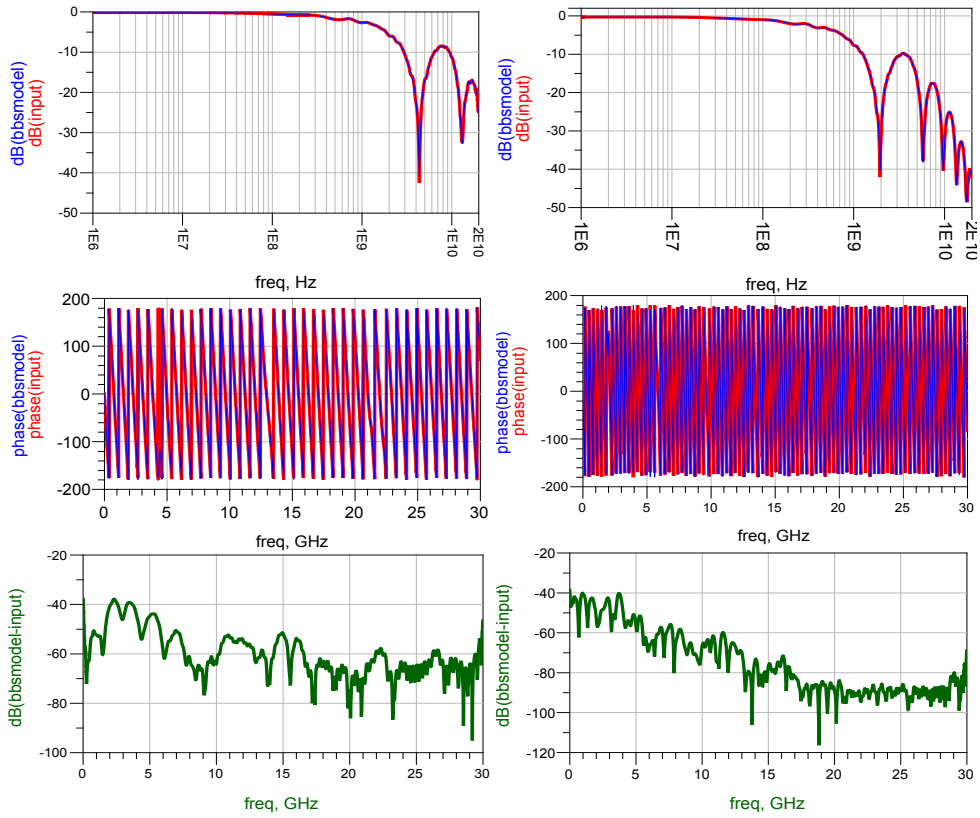
To simulate the channel impulse response using Hspice it was required to convert the channel S parameter description into a spice sub-circuit.This task was achieved using Ads Tool "Broad

Band SPICE Generator", a tool that converts a S parameter network description into a rational model in a form of a series expansion in partial fraction:

$$\sum_n^N \frac{r_n}{s-p_n}$$

The number of poles used to describe the circuit varies with the dynamics of the network described by the S parameter file and passivity is required for the description.

Figures 4.13(a) and 4.13(b) represent the model conversion from S-parameters to a spice circuit using the Broad Band SPICE model Generator



(a) channel 1 (b) channel 2

Figure 4.13: Magnitude, phase and overall Deviation between Hspice and S parameter models

In the graphs the red Line represents the insertion loss expressed in S-Parameter and the blue line expresses the Insertion loss for the Spice model. As we can see the generated spice model is an accurate description of both the channels.The magnitude of deviation for the insertion loss between S-parameter and Spice models is always inferior to $-40dB$.

To verify the impulse response the following simulation was made:

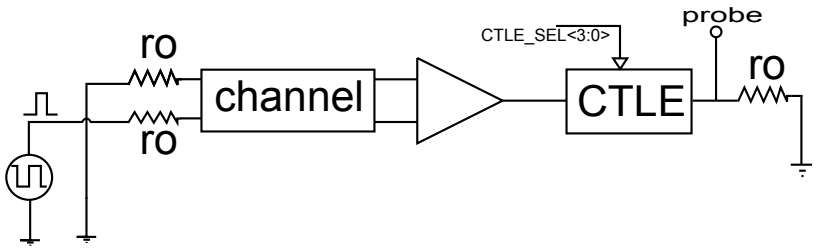
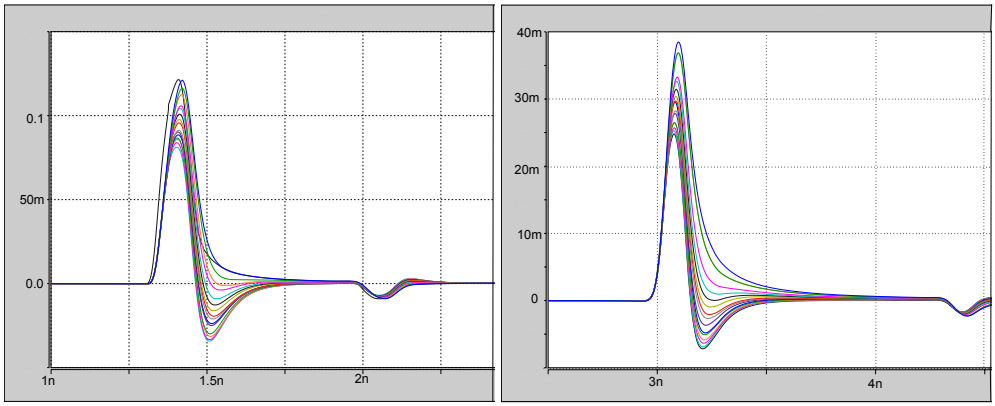


Figure 4.14: Verilog A CTLE Model

In this simulation it was transmitted a pulse with $160mV$ of amplitude and with pulse width equal to $1UI = 1/12\,Gbps$ through the port of the network while connecting the other port to ground. The results for the simulation of the impulse response for both channels, applying different boost settings to the CTLE, are depicted in figure 4.15.



(a) channel 1 (b) channel 2

Figure 4.15: Reference channel impulse responses

We can see that the values for the group delays are consistent with the ones presented in the ADS simulation on chapter 3 From figure 4.15 we can see that the second channel introduces more ISI than the channel 1. The post cursor ISI is calculated by sampling the impulse response after the main peak giving us h_1, with sample time equal to the bit time $T_s = 1/12GHz$ and dividing by the amplitude of the main cursor h_0.

The following table give us the value of the first post cursor ISI for the simulation described in 4.14.

Table 4.7: ISI variation for different CTLE settings for reference channel 1

CTLE boost	main Tap(mv)	Channel 1 1st post cursor ISI
0	117	0.231
1	117	0.183
2	113	0.112
3	106	0.0585
4	**104**	**0.01112**
5	101	-0.0723
6	97.7	-0.1212
7	95.7	-0.15384
.	.	.
.	.	.

Table 4.8: ISI variation for different CTLE settings for reference channel 2

CTLE boost	main Tap(mv)	Channel 1 1st post cursor ISI
.	.	.
.	.	.
5	29	0.2044
6	27.4	0.148
7	26.7	0.091
8	28.2	0.0446
9	**25**	**0.0001**
10	24.2	-0.0618
11	23.8	-0.1100
12	23.1	- 0.145
.	.	.
.	.	.

The values displayed in the tables 4.7 and 4.8 were obtained in spice using the *Measure* command.The process to obtain this values was the following: First we measured the value of the peak of the impulse response, then we measured the value of the impulse response $1UI$ after to measure the amount of 1st post cursor ISI for each CTLE setting. The value obtained was then normalized by the peak value to provide more conclusive measures for the ISI. Notice the variation of the Post cursor ISI variation when applying different amounts of boosting, the situations with positive ISI gives us a situation of under equalization as situations with negative post cursor gives us situations of over-equalization meaning that the introduction of high frequency boosting was beyond the required to compensate for the channel losses. Notice that the CTLE settings that provide best 1st post Cursor ISI reduction is almost the same as the one that provided best eye opening diagram for both channels (see 4.4 and 4.5).

This little difference can be explained by the fact that the distortion caused in the eye diagram is not only caused by the 1st post cursor but by all ISI cursors and the CTLE setting that best compensates the 1st post cursor might not be the one that best reduces the overall ISI.

4.3 DFE Study

The final block of the equalization architecture is the DFE. This non-linear feedback filter reduces the amount of ISI by subtracting an estimate of the post cursor ISI to the input signal. In the following figure is represented the structure of a 3 tap DFE.

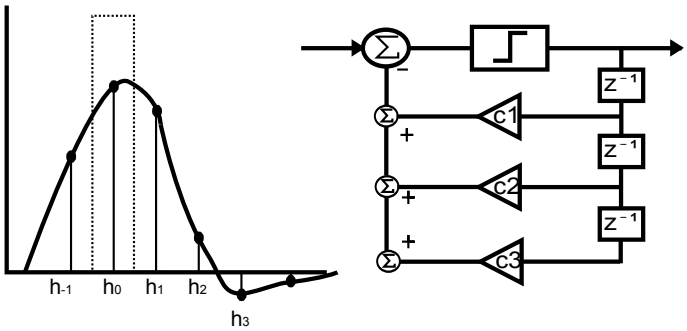


Figure 4.16: Decision feedback equalization filter structure

For optimum adaptation the values of c1 c2 and c3 should be as close as possible to $c_1 = h_1/h_0$ $c_2 = h_2/h_0$ and $c_3 = h_3/h_0$, so that the effects of the first three post Tap ISI become negligible. The insertion of a decision feedback equalizer in the equalization structure must be studied carefully because the DFE presents a much higher power consumption relatively to the CTLE.

4.3.1 DFE adaptation

In order to study the impact of the DFE in the overall equalization scheme we performed channel simulation in ADS. The insertion of the DFE in the data path was studied for both reference channels. Simulations were made to demonstrate how the DFE opens the eye diagram and how the number of taps affects the performance of this filter. In the following simulations we introduce a DFE with optimized tap values, using the final values resulting from the LMS adaptation algorithm. For the following simulations we considered also the inclusion of a CTLE with the optimum boost selected for each channel(see tables 4.4 and 4.5). In these simulations we considered $0dB$ for the emphasis filter.

The following table gives the improvement of the eye diagram in function of the number of taps for both reference channels.

Table 4.9: Channel 1 Eye diagram stats for the inclusion of the DFE

# taps	Height(mV)	Width(ps)	Q factor	tap values(10^{-3})
0	127	46.30	8.634	[]
1	123	45.45	7.961	[-7.453]
2	124	51.46	8.038	[-7.16 3.23]
3	124	50.17	8.015	[-7.15 3.31 0.528]
4	124	46.31	7.980	[-6.85 3.12 0.663 1.081]
5	124	46.74	8.006	[-6.723 3.137 -0.581 1.015 -0.035]

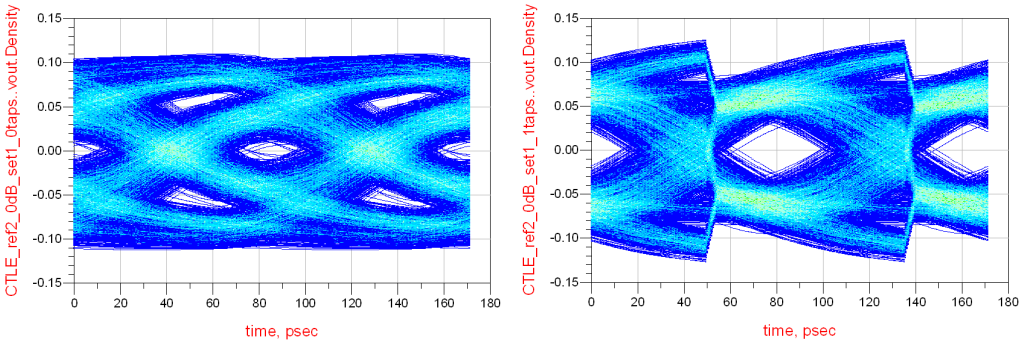
Table 4.10: Channel 2 Eye diagram stats for the inclusion of the DFE

# taps	Height(mV)	Width(ps)	Q factor	tap values(10^{-3})
0	44	45.45	5.004	[]
1	45	40.45	4.987	[-5.334]
2	55	41.60	5.906	[-4.444 7.631]
3	57	44.17	6.226	[-3.832 7.692 1.762]
4	58	42.02	6.306	[-3.288 8.480 2.375 0.812]
5	58	41.17	6.317	[-3.530 8.581 2.623 1.227 0.967]

We can see from table 4.9 that the inclusion of a DFE in the equalization structure for reference channel 1 did not had a big effect on the eye opening, even when using a 5 Tap filter, this means that the CTLE is sufficient for the compensation of the channel. When using reference channel 2 the situation is a bit different as the channel is not totally compensated and the need for further compensation is achieved by the DFE. When examining the tables 4.9 and 4.10 we can observe that the value of the taps of the DFE converge to almost the same value in both channels. This is an interesting conclusion as we can see that the DFE has a higher impact when equalizing reference channel 2.This difference can be explained by the fact that even though the taps converge to the same value, their influence must be normalize relatively to the peak value of the received waveform.

We will now analyse a situation where the impact of the DFE o is more clear. In this simulation we consider the same situation as before with reference channel 2, but using the wrong CTLE setting calculated for the channel, for example by selecting a CTLE boost of only $1dB$ instead of the optimum $10dB$.

Figure 5.5 and table 4.11 demonstrates the impact of a single tap DFE on the Eye diagram. Providing result before and after the inclusion of a one Tap DFE.



(a) 0 taps (b) 1 tap

Figure 4.17: Eye diagram after the inclusion of a 1 tap DFE with optimized tap value(c1=-0.025,and 1dB of boost from the CTLE)

The following table give statistics for the inclusion of a DFE with a diferent number of taps in the simulation setup presented before.

Table 4.11: Channel 1 Eye diagram stats for the inclusion of the DFE

# taps	Height(mV)	Width(ps)	Q factor	tap values(10^{-3})
0	0	20.58	2.492	[]
1	39	40.31	4.349	[-27]
2	54	47.60	5.255	[-26.14 -6.012]
3	55	49.31	5.346	[-27.33 -6.123 -4.124]
4	56	51.03	5.391	[-24.12 -5.461 -2.456 -0.837]
5	56	47.60	5.391	[-25.548 -7.054 -3.758 -1.60 -0.181]

The discontinuity observed in the eye diagram is caused by the subtraction of the value of the estimate of the post cursor Inter-symbol interference, the subtraction is made near the symbols transitions to not disturb the sampling point of the equalizer. In the first experiment the DFE didn't had much influence in the eye opening because the CTLE performed most of the task of removing the ISI that's why the values of the taps were so small relatively to the signal amplitude. In the second experiment because the choice for the CTLE boost was wrong, there was still a large amount of ISI to correct and the final value for the DFE tap was $5x$ times larger than the one in the first experiment. The insertion of the DFE with only a single tap was capable of mitigating the major part of the ISI present.

4.3.2 DFE in Hspice

To better understand the impact and the operation of the DFE it was constructed a verilog A model for the DFE. The following verilog A code describes a one tap decision feedback equalizer with fixed tap values:

```
module OneTapDfe(in, clk, out);
    input in, clk;
    output out;
    electrical in, clk, out;
    parameter real Vth=0.5, vhigh=1, vlow=-1, dataVth=0;
    parameter real td=0.01n, tr=0.01n, tf=0.01n;
    parameter real iniTap=0.025;
    real sample, cTap;
    real x, y, yold;
    analog begin
        @(initial_step) begin
           sample=0;
           cTap=iniTap;
           x=0;
           sele=0;
           y=0;
           yold=0;
        end
        @(cross(V(clk)-Vth, +1)) begin
           sample=V(in);
           x=sample-cTap*y;
           yold=y;
           if(x>dataVth) begin//decision
                y=vhigh;
           end
           else begin
                y=vlow;
           end
        end
        V(out)<+ transition(y, 0, tr, tf);
   end
endmodule
```

The module definition is again self explanatory but a few comments might be important: The line *x=sample-cTap*y;* makes the subtraction of the post cursor ISI and utilizes the last decision *y* to calculate the sign of the ISI that needs to be cancelled by the DFE. The line *@(cross(V(clk)-Vth, +1))* shows the sampled nature for the operation of the DFE working in time steps given by the clock transition.

Because in the verilog A model the subtraction of the post cursor ISI is instantaneous there is no need to make the subtraction of the ISI near a transition and we can perform all the operations associated with the DFE at the rising edge of the clock.

To demonstrate the error correction capabilities of the Decision feedback equalizer the following simulation set-up was considered:

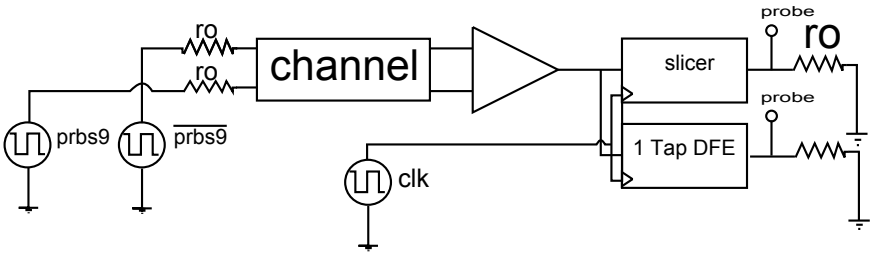


Figure 4.18: Decision feedback equalization simulation

The channel simulated was reference channel 2. This channel with a 12$Gbps$ bit Rate presents decision errors when it receives a long sequence of *0's* or *1's* (larger than 4 bits in a row) and when the bit changes the waveform does not passes the decision threshold needed to detect the bit correctly, introducing a decision error.

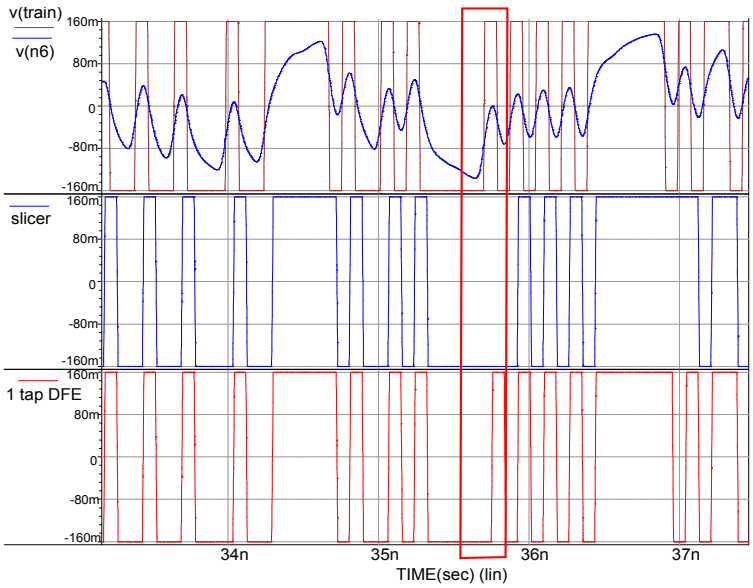


Figure 4.19: Decision feedback equalization simulation

In the figure 4.19 is the result of the spice simulation for the verilog A model of the DFE. The first two signals represent the transmitted sequence with appropriate delay and the received waveform that is presented to the slicer and to the one tap DFE.

The following two waveforms present the recovered sequence for a simple slicer and a one tap DFE with appropriate tap value. As we can see in the figure only in the third case the receiver is able to recover the correct sequence.In this case the the incorporation of a single tap DFE with the optimized value for the feedback tap was able to correctly estimate the transmitted sequence by subtracting the channel induced ISI.

4.4 Final remarks

In this chapter we analysed the impact of each one of the components proposed in the equalization architecture in chapter 3 to perform channel equalization at $12Gbps$. For the study of the impact of the emphasis filter in the integrity of the received data, it was performed eye opening analysis on the data at the receiver when applying a 2 tap FIR filter to the transmitter.By optimizing the values of the filter taps we were able to select the amount of de-emphasis that best equalizes each reference channel.
For the CTLE we proposed the creation of 16 equalizer boost settings to make the filter capable of being adapted to a series of different channels. After the creation of these settings we performed AC simulation to see the impact of the incorporation of the CTLE in the insertion loss of the two reference channels used by the interface.
After the AC simulations it was again performed eye opening analysis on the data at the receiver when applying the different boost settings to the CTLE.It was concluded that the setting that provided better AC compensation was the one that provided best eye opening of the data at the receiver.
We visualized the impact of the insertion of the DFE in the equalization chain, and notice that if we choose the best CTLE setting for the channel, the DFE does not introduce a big improvement to the eye opening of the received data.
So in order to prove the equalization capabilities of the DFE it was performed another simulation where it was chosen the wrong CTLE boost setting for the equalization of the channel.In this simulation was observed that the DFE was able to open a completely closed eye diagram with only a single tap.
After these simulations it was created a Verilog A model of the DFE and used this model to simulate the error correction capabilities of a receiver with a simple slicer versus a receiver with the incorporation of a single tap DFE.

Chapter 5

Techniques for adaptation for the CTLE and DFE

In the last chapter we saw the impact of the CTLE and the DFE in the equalization structure of a high speed interface operating at $12Gbps$.
In this chapter we focus our attention in describing techniques to adapt the CTLE and the DFE to best compensate for channel losses. In this chapter are presented some algorithms to adapt the DFE and the CTLE to the transmission environment. To narrow down the focus of our studies, some techniques presented in chapter 2 were discarded for being too complicated or just inappropriate to implement in the studied interface.
The focus for the adaptation of the CTLE is based on the construction of histograms of the waveform at the output of the equalizer. Although this approach is conceptually very simple it can prove to very effective as we will see along the chapter.
The process of adaptation is sequential, first we adapt the CTLE and after we adapt the DFE. This can not be done in parallel because the choice of the CTLE setting will have influence on the amount of ISI that reaches the DFE. For the adaptation of the DFE we will try to apply the same techniques used for the CTLE so in that way make the process of adaptation of both filters use the same circuitry, and thus making the process of adaptation simpler. For the DFE we also describe the possibility to apply the Least Mean Squares algorithm to perform adaptation using training and blind mode.

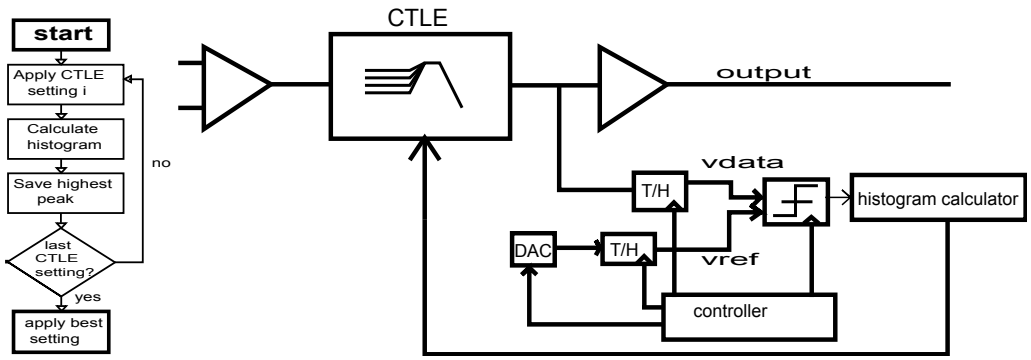
5.1 CTLE adaptation

In this section the main objective is to describe techniques that allow for the interface to tune the CTLE to the receiver in such a the way to best equalize for the channel distortion. Some architectures presented for the adaptation for CTLE's described in chapter 2 were discarded because they required to perform a lot of changes in the interface.For example the system demonstrated in 2.18 required an analog circuit to sense the power of the signal at different bands which would be too complicated to implement and would be very power inefficient.
The adaptation technique described in 2.19 requires the creation of a clock asynchronous with all the bit Rates present in the interface, this would require the creation of an new circuit for clock generation which is inappropriate once the interface already possesses a PLL. The main objective behind having an asynchronous clocks is to be able to sample the received waveform in all the phases present in a one bit time interval to estimate the eye opening. Although this is not possible, in the interface there is a circuit that is able to generate several different clocks that allows to sample the bit at different phases during one UI. We will study a technique derived from the asynchronous under sampling histograms algorithm, that uses the clock phases provided by the interface to sample the received waveform. The voltage histogram algorithm makes an histogram of the sampled voltages so in that way can estimate the opening of the eye diagram.
Other algorithm discussed for adaptation of the CTLE is the edge histogram algorithm that is based on making an histogram of the zero crossings of the waveform at the output of the CTLE. The construction of the edge histogram also uses the different clock phases provided by the interface PLL to sample the waveform after the CTLE to be able to detect in 1UI time where the data transition occurred. By building an histogram of the transitions per zero during an UI time, we can estimate the total amount of transition jitter present, this measure can be used to infer the eye width of the eye diagram. The simulations conducted for the testing of this algorithms will be performed using the reference channels characterized in chapter 3 and the CTLE settings proposed in the chapter 5.

5.1.1 Adaptation with voltage Histograms

This adaptation technique is based on the asynchronous under-sample histograms algorithm that says which the CTLE setting that produces the voltage histogram with highest peak, is the one that provides the best eye opening. The main difficulty in applying this algorithm to the CTLE present in the PHY interface is the limited number of clock phases for sampling the received waveform.

The process of adaptation follows the algorithm in 5.1(a) and can be implemented using the architecture represented in 5.1(b).



(a) histogram algorithm

(b) histogram algorithm implementation

Figure 5.1: Voltage histogram algorithm

The circuit in 5.1(b) samples the waveform and then trough a successive approximation ADC, calculates the digital word for the sampled voltage. Then the histogram calculator simply increases the bin corresponding to the digital word .
To demonstrate the algorithm the following simulation was made using the channel simulator in ADS:

- PRBS9 source transmitting 4096 bits with 2dB of de-emphasis.
- Two reference channel 2 in series with Insertion loss represented in 4.10
- Apply the 16 settings to the CTLE and plot the eye diagrams.

The simulation showed that the CTLE settings that better equalized the channel were the ones present in table 5.1.

Table 5.1: Eye diagram statistics

boost	Height(mV)	Width(ps)	JitterPP(ps)	JitterRms(ps)	Q factor
4dB	4	31.17	54.03	12.82	2.536
5dB	6	35.16	50.16	12.81	2.680
6dB	6	32.59	53.17	12.88	2.787
7dB	4	30.45	55.32	12.23	2.783

For the construction of the histograms were considered 16 samples phases and a total of 64 histogram bins, notice that the histogram bins are normalized by the total number of samples.

The next figure represents the voltage histograms after the CTLE, when applying all the 16 boost settings to the CTLE. The colour code presented in the figure will be used for the figures.

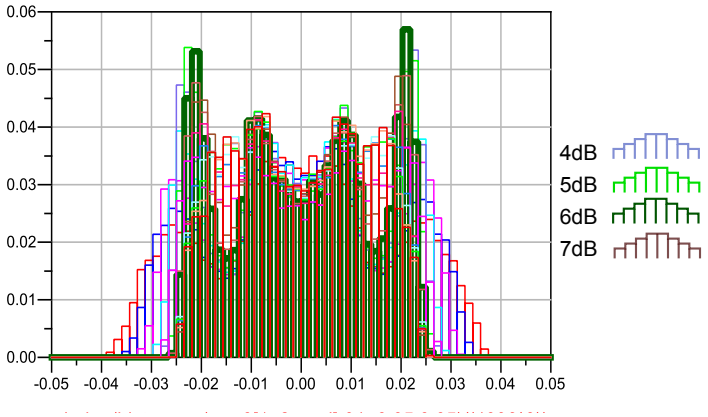
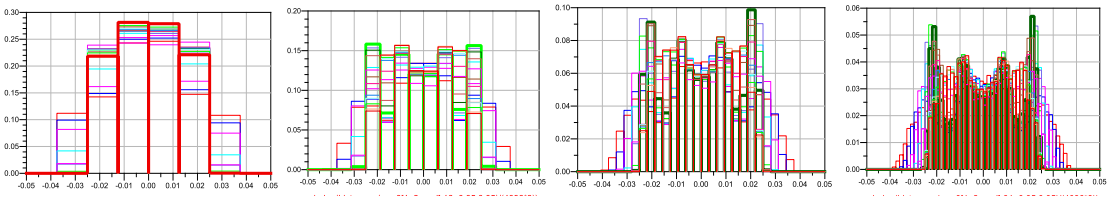


Figure 5.2: Voltage histograms after the CTLE when applied the 16 boost settings

As we see in figure 5.2 the CTLE settings that provided the the best eye opening were the ones that produced the largest peak in the histograms of the received data, even when only considering 16 samples phases.
The creation of the voltage histogram takes a lot of computation power because every sample after the CTLE must be converted into a digital word through a successive approximation ADC. Also the thresholds used in the ADC can be unknown, as the interface does not know the amount of attenuation suffered by the channel. Other thing that is important knowing is how this method for adaptation works when some conditions change, for example the number of bins or the number of Sample Phases.

5.1.1.1 Influence of the number of Bins

The influence of number of bin is important because it is related with the number of comparisons that must be made for each sample when constructing the histogram.The higher the number of bins the more complex is the algorithm. To study the influence of the number of bins in the histogram method we repeated the construction of the histogram with 64, 32, 16 and 8 bins, the results are shown in figure 5.3:



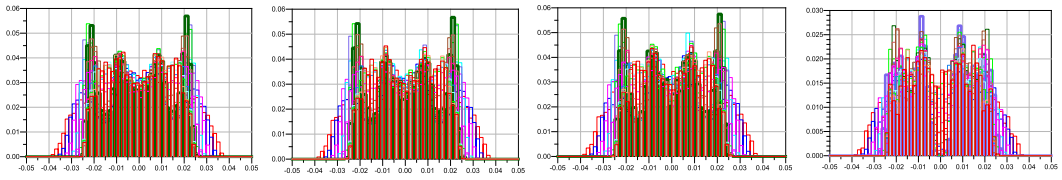
(a) 8 bins (b) 16 bins (c) 32 bins (d) 64 bins

Figure 5.3: Sensitivity of the histogram algorithm to the number of histogram bins

As we can see from the previous figure only the histograms with more than 32 bins make the proper decision of the best CTLE setting for the compensation of the channel. Although the 16 bins selected one of the best CTLE settings in some trials the algorithm failed by selecting others than the ones presented in table 5.1. So from this simulation we can see that there is a need of at least a 32 bins histogram to implement the algorithm. This implies that each sample must be compared with at least $\log_2(32)$ thresholds when using a successive approximations ADC.

5.1.1.2 Influence of the number of sample Phases

For the determination of the sensitivity of the algorithm to the number of sample phases we perform the same experiment as before, but sampling the waveform with 8, 4, 2, 1 sample phases during an UI and plotted the voltage histograms of the output of the CTLE with 64 bins.
The results are the following:



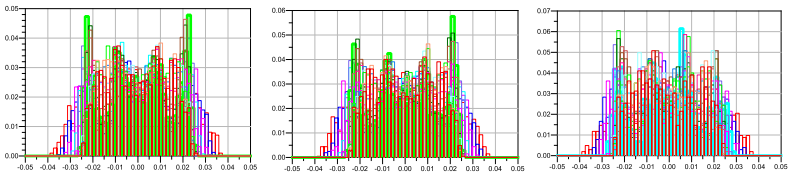
(a) 8 sample phases (b) 4 sample phases (c) 2 sample phases (d) 1 sample phase

Figure 5.4: Voltage histogram algorithm sensitivity to the number of sample phases

In figure 5.4 we can visualize that even with only with one sample phase the histogram produces a peak that provides the proper selection of one of the optimum settings. Sampling in more than one sample phase does not bring too much complexity has the PLL already provides several sample phases to sample the waveform after the CTLE.

5.1.1.3 Influence of the number of samples

As the voltage histogram is a statistical algorithm the higher the number of samples the bigger is the accuracy of the results. In the next experiment we apply the voltage histogram algorithm sampling the waveform with 4 phases and using 4096, 2048, 1024 and 512 samples to build the histogram. The results are the following.



(a) 4096 samples (b) 2048 samples (c) 1024 samples

Figure 5.5: Histogram algorithm sensitivity to the number of total samples

As we can see when the algorithm takes at least 2048 samples to correctly select one of the best boost settings for the CTLE.

5.1.1.4 Voltage histogram summary

If we consider a 4096 sample / 64 bin histogram and that the ADC is able to calculate a digital word for the sample at every 10 received bits we can estimate the following time required for adaptation of the CTLE working at $12Gbps$.

$$time = 15settings * 4096samples * log_2(64) * comparations * 10UI/sample \approx 250\mu s$$

As we can see the algorithm showed promising results in the conducted experiments selecting the correct values even in the presence of several limitations, but the algorithm possesses some limitations

1. Requires additional analogue circuitry to work, the successive approximation ADC is required for the creation of the histogram.
2. Requires the creation of a digital controller for the process.
3. Requires knowledge of the input signal level for the creation of the threshold for the bins
4. Takes a while to choose a CTLE setting, in the order of the μs
5. The algorithm is incapable of detecting the amount of jitter present in the Received waveform.

The jitter limitation is an important restriction to the use of this algorithm.Notice that the channel may introduce 0 distortion, presenting a voltage histogram with only two peaks and still present a completely closed eye if the amount of peak to peak jitter is more than $0.5UI$.So the simple measure of the behaviour of the voltage distribution may not be enough to choose the proper CTLE setting.

5.1.2 Adaptation Using DC balance

This algorithm is based on the DC balance of the transmitted signal when considering a sufficient high number of bits. In the case of the studied interface, the DC balance is even higher because of the use of 8b10b line coding for transmission. The following figure shows the voltage histogram for the transmitted sequence.

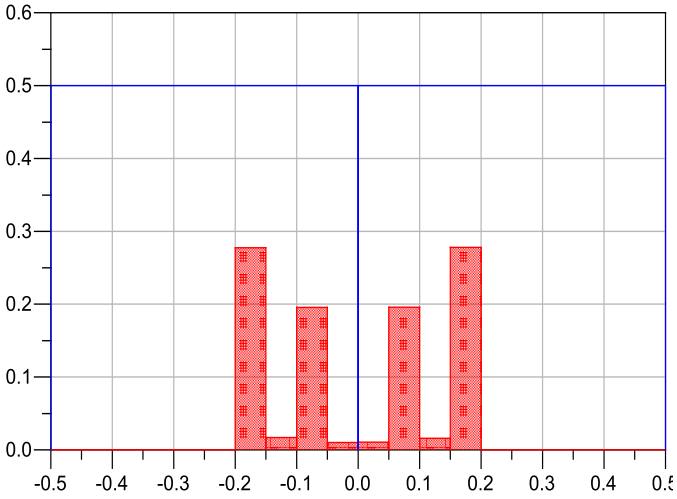


Figure 5.6: Voltage distribution of transmitted data

The red bars show the histogram of the transmitted voltage when using 20 bins. Notice the influence of the $2dB$ of emphasis in the distribution of the voltages of the transmitted data.The blue bar show the histogram of the number of 1's and 0's transmitted.
From the figure we can observe the balance between the number of 0's and 1's in the transmitted sequence showing equal probability of transmission over the 4096 bits.
This algorithm is a simplification of the voltage histogram presented earlier, once it only considers if the received bit is detected as a 0 or a 1.The algorithm assumes that the CTLE boost that produces the best balance between the number of detected 0's and 1's, is the best setting for the CTLE. So in order to demonstrate the algorithm, we ran another analysis on the experiment data provided by the voltage histogram simulation. To test this algorithm we exported the waveform that was sampled $16x/bit$ and analysed it using a script in Matlab. The script sampled the waveform at different phases and calculated the number of bits that were 1 and the number of bits that were 0 acting like an ideal slicer. The following table represents the difference between the number of detected 0's and 1's when applying each boost of the CTLE and sampling the waveform after the

CTLE at a different clock phase.

Table 5.2: mismatches between 0's and 1's in function of the CTLE boost and sampling phase

boost(dB)	0	1	2	3	**4**	**5**	**6**	**7**	8	9	10	11	12	13	14	15
UI/16	48	16	16	-2	-2	-2	7	-4	-4	-2	-2-	-2	-4	-3	-2	-4
2UI/16	-4	18	-6	-2	-2	-2	-4	-4	-4	-2	-2	-2	-4	-6	-12	-24
3UI/16	12	-2	-4	-2	-2	-2	-3	-4	-4	-2	-2	-8	-18	-16	-12	-2
4UI/16	12	-2	-4	-2	-2	-2	-3	-4	-4	-12	-12	-8	-12	-12	-34	-42
5UI/16	-6	-8	-4	-2	-2	-2	-3	-6	-12	-8	-24	-20	-30	-27	-26	-32
6UI/16	-8	-20	-6	-2	–2	-4	-10	-14	-24	-4	-26	-28	-16	-14	30	36
7UI/16	-20	-24	-22	-12	-21	-22	-16	-18	-34	-52	-6	20	28	64	50	62
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

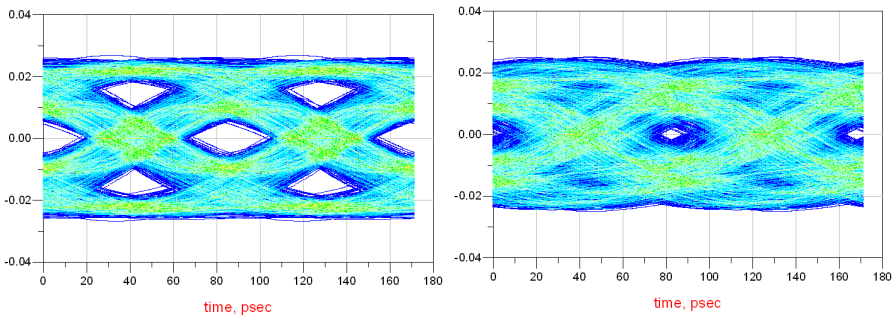
The optimum sampling phase is $UI/3$, sampling the waveform in the maximum opening of the eye diagram. This phase produces the lowest total number of mismatches for all CTLE settings. The algorithm shows that the settings 4, 5, 6, 7 produce the lowest mismatches between 0 's and 1's when comparing to the other boost settings. But this algorithms finds too many local minimums even when using settings that completely close the eye diagram for example $15dB$ of boost and using the sample phase $3UI/16$.
The algorithm using DC balance gives an idea of the eye opening for each CTLE setting but is not precise enough to make a proper decision on the ideal boost that best compensates the channel.

5.1.3 Adaptation with Edge Histograms

This section presents a new type of algorithm for the adaptation of the CTLE based on making the histogram of the transitions trough 0 of the waveform at the output of the CTLE. The algorithm assumes that the best CTLE boost minimizes the total amount of transition jitter in the waveform after the CTLE. For the demonstration of this algorithm it was used the same simulation from the two previous algorithms,that consists in:

- PRBS9 source transmitting 4096 bits with 2dB of de-emphasis.
- Two channels in series with Insertion loss as in 4.10
- Apply the 16 settings to the CTLE and plot the eye diagrams.

The following figure demonstrates the correlation between the eye diagram opening and the amount of peak to peak jitter for two boost settings used used in the simulation.



(a) 4dB boost | jitterPP=45ps (b) 14dB boost | jitterPP=75ps

Figure 5.7: Relation between eye opening and peak to peak Jitter

The edge histogram with the more bins with the transition count equal to zero is the setting that minimizes the total jitter in the eye diagram.We can see that the setting with 4dB of boost reduces the amount of peak to peak jitter maximizing the eye opening relatively to the 14dB CTLE setting. The creation of the edge histograms followed the procedure describe in figure 5.13.

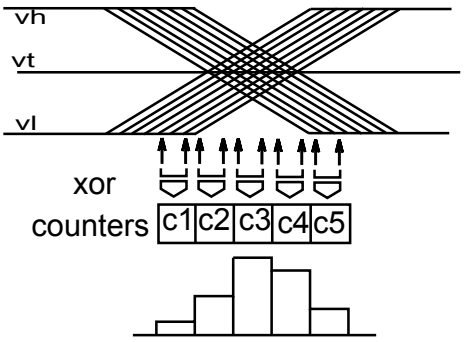


Figure 5.8: Edge histogram calculation

To create the edge histogram the waveform after the CTLE is oversampled using different sample phases given from the PLL and for each sample make a decision. The result of each decision is passed to an XOR gate with the adjacent clock phase. The result of the XOR allows the detection of a transition. The information relatively to where the transition occurred allows the creation of an histogram that gives direct information of the eye width of the received waveform. The CTLE boost that produces the edge histogram with the biggest number of bins with no transitions, is the optimum CTLE boost. To demonstrate this algorithm a matlab script was created, that for each sample decision (sampled at $UI/16$) calculated the XOR with both the adjacent samples and constructed the histogram.

The results are the following.

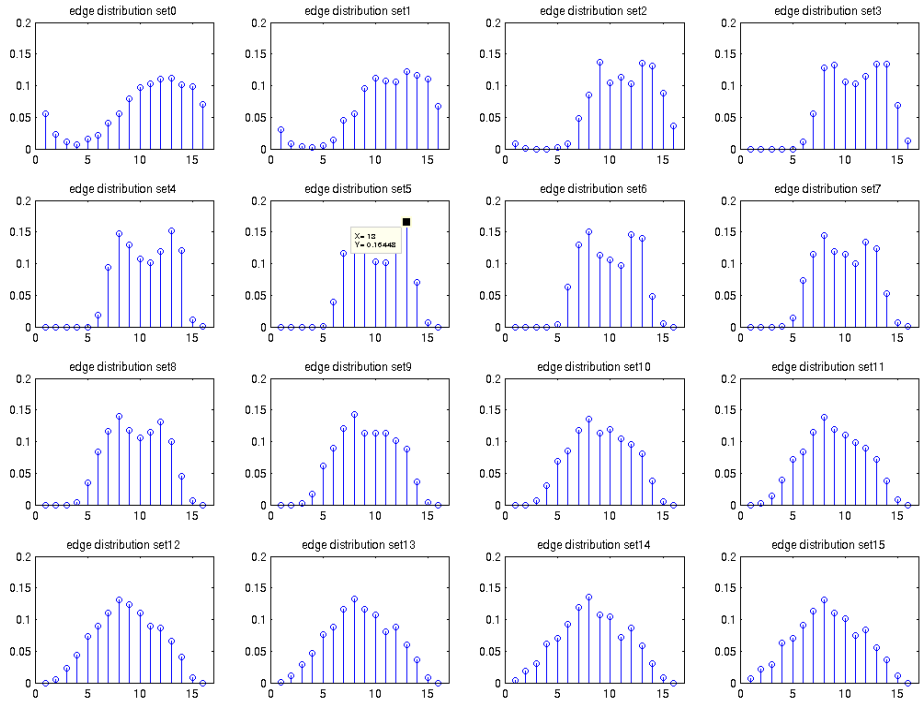


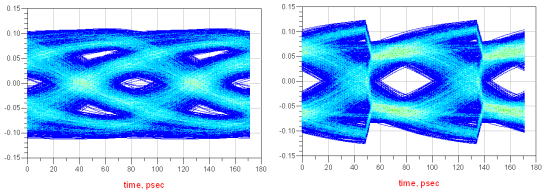
Figure 5.9: Edge histogram results for the adaptation of the CTLE

The results show the edge histogram for each one of the 16 CTLE settings used for the equalization of the channel. As we can see the histogram corresponding to the settings that better equalize the channel corresponds to the histogram with the biggest number of bins equal to 0.
The eye width can be easily calculated using the number of bins of the edge histogram equal to zero. For example for the setting with $5dB$ boost we get 6 bins equal to zero to $UI/16$ $30ps$ of eye width, this method presents data similar to the one present in table 5.1.
One of the main advantages of this algorithm over the ones presented earlier is that it gives a metric of the eye opening in percentages of the UI for each CTLE boost, becoming easy for the interface to have knowledge of possible eye mask violations. Other advantage is that this process doesn't need to know the optimum sampling time for the waveform.With this algorithm we can even perform CDR for the configuration of the receiver, as the optimum sampling point will be given by the clock phase that is positioned in the center of the histogram with the transition count equal to zero. One of the main disadvantages of this process is the requirement of many asynchronous operations during an UI (the sampling and the XOR of all the samples) and the synchronization of all the operation can be hard to achieve specially with the interface running at $12Gbps$. The process also suffers severely from jitter in the sample clocks and some transitions can be undetected if the jitter is bigger than the difference between the sample phases.

5.2 DFE adaptation

5.2.1 Adaptation with voltage Histograms

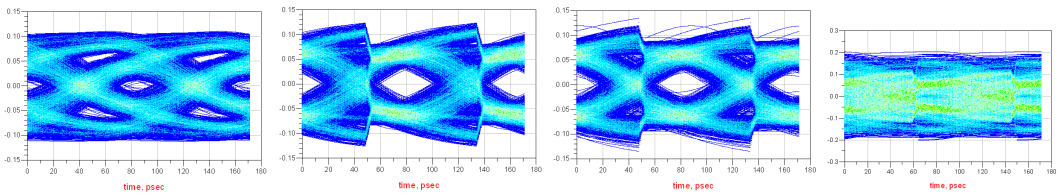
The method for equalization using voltage histograms follows the same methodology as describe for the adaptation for the CTLE. Making the histogram of the voltages of the waveform at the output of the DFE. In this experiment is observed the process of adaptation of a 1 tap DFE using voltage histograms, the channel used for adaptation is reference channel 2. The transmitter is set to transmit 4096 bits of PRBS sequence with $0dB$ of de-emphasis and a $1dB$ of boost for the CTLE. Before applying the voltage histogram algorithm we observe the eye diagram before and after the inclusion of a 1 tap DFE with the tap value optimized in ADS.



(a) 0m tap (b) -0.025 tap

Figure 5.10: Eye diagram for a 1 tap DFE:No tap vs optimum tap value

We can see from 5.10 that the inclusion of a single tap DFE brought significant improvement to the eye diagram after the DFE. cNotice that the optimum value for the tap is negative this means that the the channel is under-equalized and some ISI still needs to be subtracted by the DFE. In the following experiment we sweep the value of the tap for the DFE from 0 to -0.090 with a step of 0.030 and observe the effect on the opening of the eye diagram. The results for the eye diagrams after the DFE are the following:



(a) 0m tap (b) -0.030 tap (c) -0.060 tap (d) -0.090 tap

Figure 5.11: Eye diagram for 1 tap DFE with negative tap values

Table 5.3: Eye diagram statistics for the inclusion of 1 tap DFE

tap value	Height(mV)	Width(ps)	jitterPP	Q factor
0	0	20.58	58.31	2.492
-0.030	31	27.44	48.49	4.005
-0.060	37	32.21	50.21	4.316
-0.090	0	0	85.7	1.663

The following figure presents the voltage histograms for a 64 bins histogram for the tap values values presented in table 5.3.

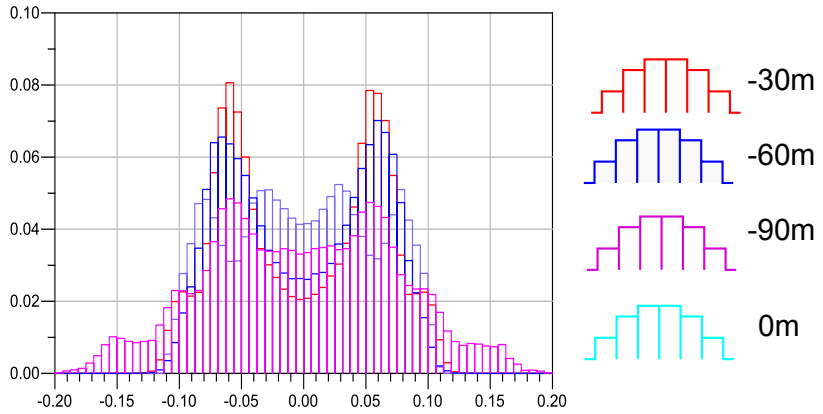


Figure 5.12: Received voltage histogram

We can see that the voltage histogram algorithm correctly selects the optimum tap value for a large number of bins. However using this method for the adaptation for the DFE is not feasible. For example if we consider the adaptation of a 5 tap DFE, considering 20 tap values for each tap, the algorithm needs the construction of 100 voltage histograms witch makes the process impractical if we consider a successive approximation ADC for the construction of the histogram.

5.2.2 Adaptation with DC balance

In this experiment we try to adapt a 1 tap DFE using DC balance with the simulation setup presented in adaptation of the DFE with voltage histograms.
The following table presents the difference between the number of detected 0's and 1's, in function of the tap value of the DFE and the sample phase used for sampling the waveform.

Table 5.4: Mismatches between 0's and 1's in function of the DFE tap value and sampling phase

DFE tap value	0	-30m	-60m	-90m	DFE tap value	0	-30m	-60m	-90m
UI/16	-12	-12	-4	60	9UI/16	8	30	-2	24
2UI/16	18	-14	-10	24	10UI/16	6	0	0	8
3UI/16	20	10	-44	121	11UI/16	-2	0	0	16
4UI/16	52	28	-2	76	12UI/16	0	0	0	12
5UI/16	32	-4	8	104	13UI/16	0	0	0	22
6UI/16	68	38	38	28	14UI/16	0	0	0	94
7I/16	46	2	16	4	15UI/16	0	0	0	50
8UI/16	20	-2	-6	12	16UI/16	-6	2	-2	64

The process of adaptation using DC balance for the DFE presents the same limitations as it presented for the adaptation of the CTLE, detecting local minimums in tap values that present completely closed eye diagrams.

5.2.3 Adaptation with edge histograms

This method works the same way as presented for the adaptation for the CTLE presented earlier, and consists in making the histogram of the zero crossings of the waveform after the DFE. The experiment setup used for the simulation of the edge algorithm is the same as the one used in the voltage histograms presented in the previous section. In this process, we apply different values for the value of the tap of the DFE and then plot the edge histogram of the output. The tap value that creates the histogram that contains more bins without transitions is the best value for the DFE tap. The results for the sweep of a single tap DFE with a step of $30m$ is the following:

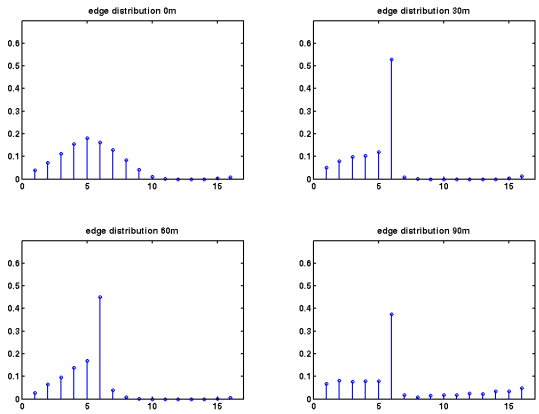


Figure 5.13: Edge histogram results

From the figure we can see that the tap values equal to 0.030 and 0.060 are the ones that present the edge histogram with more bins equal to zero. Despite these two settings being similar a closer looks reveals that the -0.030 tap value has a edge histogram with one more bin equal to

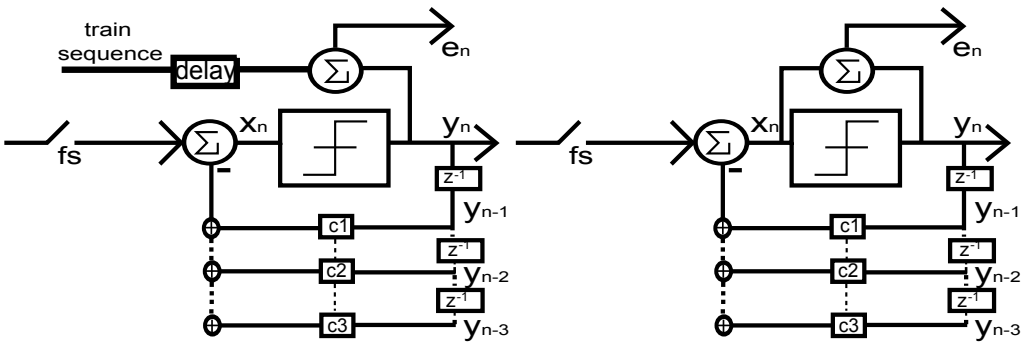
zero in a total of 6 bins leading to a eye width of $UI/16*6 \approx 30ps$ a information that is consistent with the data provided in table 5.3. The process of building the edge histogram is far more simpler and presents better results than the voltage histogram algorithm for the adaptation of the DFE. The time required for adaptation depends on the number of taps of the DFE, the precision chosen for the value of the taps and the number of bits analysed for the construction of the histogram. So for a 2 tap DFE with a number of tap values equal to 20 and considering 4096 bits for the construction of the histogram, the time required for adaptation can be calculated as follows:

$$Time = \#taps * \#DFESteps * \#Bits * 1UI$$
$$Time = 2 * 20 * 4096 * 1/12G \approx 140\mu s$$

In this calculation we estimated that the algorithm is capable of detecting the edge and increase the histogram bins in less than $1UI$, if this is not possible than the the time required for adaptation will be multiplied by the number of UI's required to perform this operation.

5.2.4 Adaptation with LMS algorithm

This section is dedicated to the simulation for the adaptation of the decision Feedback equalizer using the Least Mean Square algorithm. The LMS calculates the filter taps that produce the minimum square error between the transmitted sequence and the sequence presented to the decision device. The DFE can perform adaptation to the channel in two different modes of operation, training mode or decision direct mode as depicted in figure 5.14.



(a) Training mode (b) Decision direct mode

Figure 5.14: Modes of operation for the DFE

The difference between the two modes is the expression of the error e_n used in the equation for the update of the filter coefficients.

$$c_{(n+1)} = c_{(n)} - 2\mu e(n)y(n-1)$$

The insertion of the LMS algorithm in the DFE verilog A model demonstrated in chapter 5 was made by simply including in the model a few lines of code for the error calculation and for the filter coefficients update. We can see that if the decision device estimates the symbol correctly than both the expression lead to the same equation for the evolution of the DFE taps.
The following code segment shows how to implement the LMS algorithm to the DFE Verilog A model for the DFE:

```
if(mode==1)begin\\training
        ....
        trainSample=V(train);//sample the training sequence
        ....
        cTap=cTap-miuT*(trainSample-x)*yold;//update coefficient
end
else begin \\decision direct
        ....
        cTap=cTap-miuD*(y-x)*yold;
        ....
end
```

If the DFE is working in training mode the equalizer needs knowledge of the transmitted sequence with appropriate delay for error calculation. When the DFE is operating in Training mode it can perform adaptation using sign LMS or with traditional LMS as shown in the following figure:

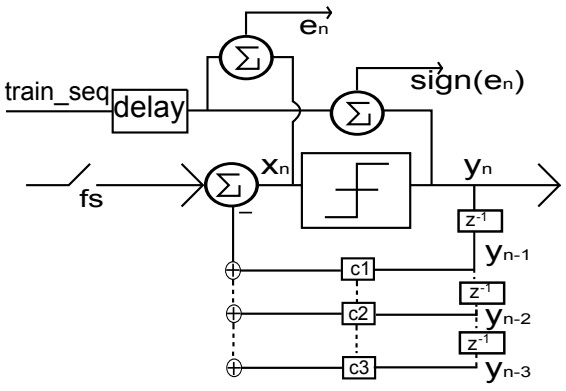


Figure 5.15: Error calculation with sign and traditional LMS in DFE

When using the sign LMS algorithm we use only the sign of the error signal in the coefficient update, the $sign(e_n)$ can only have 3 values or 0, -1 or 1.

If the error signal is equal to 0 the coefficients maintain their value, with +1 or -1 the coefficient will increase or decrease by a fixed quantity μ.
The use of the sign LMS simplifies the coefficient update equation making it more easy to implement in hardware but it has some drawbacks, for example the algorithm takes more time to converge because the error signal is only different from 0 when the decision device makes the incorrect decision. In cases where the ISI does not introduce decision errors during the training period then the FIR taps will remain with their initial values. If instead we use the traditional LMS, with the error being calculated based on the sample at the entrance of the decision device the algorithm will converge independently of the existence of decision errors updating the tap value at each iteration of the DFE clock. In Decision direct mode the DFE does not have knowledge of the transmitted sequence and estimates the error introduce in the channel by calculating the difference between the voltage level at the output and input of the decision element.
In decision direct mode is harder to ensure convergence of the method because if the DFE makes an error estimating the received symbol the error estimation will have the incorrect signal and will cause the value of the taps to drift from the correct value.
To study the implementation of the LMS algorithm on the Decision feedback equalizer the following simulation set-up was used:

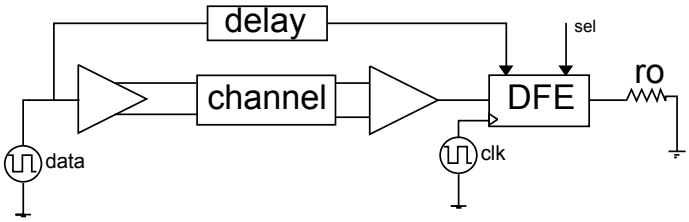


Figure 5.16: Adaptive DFE simulation setup

In the simulation the signal *sel* chooses the mode of adaptation of the equalizer, training or decision direct mode.The *clk* is the reference clock to the DFE required to be set in the middle of the bit for optimum performance.
The channels used for the test are the reference channels described in chapter 3. It was considered the following optimum values for the equalizer taps calculated from an impulse response simulation, where it was transmitted a differential pulse of 2V with pulse width equal to $1UI(1/12Gbps)$ through the channel. Notice that no CTLE was placed in the data path.

The results for both channels impulse response are the following:

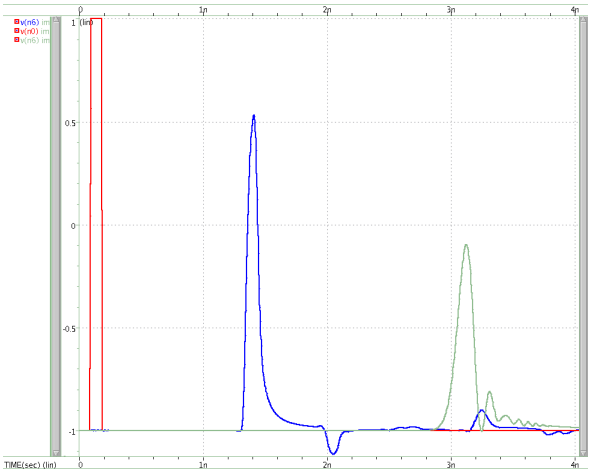


Figure 5.17: Reference channel 1 and 2 impulse response

Using Hspice measure command, the following values for the impulse response and for the optimum values of the FIR feedback filter taps were obtained.

Table 5.5: Impulse response statistics for DFE adaptation

	main tap attenuation	optimum 1st tap	optimum 2nd tap
Channel 1	$\approx 25\%(1.550/2)$	$0.22/1.5 \approx 0.14$	$0.1/1.5 \approx 0.066$
Channel 2	$\approx 50\%(0.965/2)$	$0.21/0.965 \approx 0.21$	$0.14/0.965 \approx 0.14$

In the following sections present simulation results over a set of parameters that affect the performance and convergence speed of the LMS algorithm applied to the adaptation of the Decision feedback equalizer.

5.2.4.1 LMS in Matlab

To study the applicability of the LMS algorithm to the adaptation of the DFE we performed matlab simulation for the implementation of the LMS algorithm to a 3 tap DFE for the compensation of a well behaved channel represented by its impulse response.

The following matlab code demonstrates the principle of operation of a LMS DFE working in decision direct mode.

```
clear all;
close all;
display('LMS DFE');
```

```
N=10000; %length of the sequence
h=[0 0.001 0.2 1  0.3 0.2 0.1 0.05 0];  % channel impulse response
miu=0.0005;%adaptation step
size_feed=3;

d= (randn(N, 1)>0) *2-1;  % Random bipolar (-1, 1) sequence;
ah=conv(h, d);
v= sqrt(0.001)*randn(N, 1);  % Gaussian noise with variance 0.001;
u=ah(1:N)+v;

wf=zeros(size_feed, 1);
sample = zeros(1, size_feed);
for i=1:N
     x(i)=sample*wf;
     w(i)=u(i)-x(i);
     z(i)= sign(w(i));
     e(i)= w(i)-z(i);
     e(i)=z(i)-w(i);
     wf=wf+miu*e(i)*sample';
     sample = [ z(i) sample(1:end-1)];
end
```

In the code we create a random bit sequence passing it through a channel with dominant post Cursor ISI and then apply the LMS algorithm to a 3 Tap DFE to compensate the channel's post cursor ISI. The objective of the simulation is to observe the evolution of the mean square error.

Figure 5.18 demonstrates the results for the adaptation of the DFE:

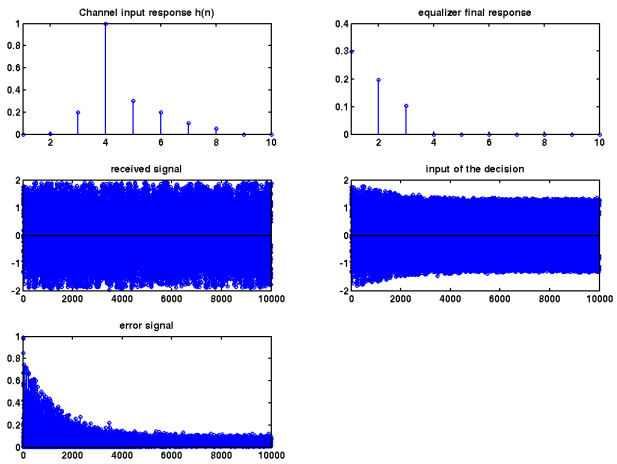


Figure 5.18: LMS algorithm applied to a 3 tap DFE

From the figure we can observe that the final value of the taps of the FIR feedback filter of the DFE evolve to the optimum values for the cancellation of the post cursor ISI. The convergence of the LMS algorithm causes the error signal to decrease to minimum value imposed by the pre cursor ISI and the 4th post cursor ISI that is not compensated by the inclusion of the DFE. Notice that the insertion of noise at the entrance of the equalizer does not affect the convergence of the method due to the small noise variance in comparison with the signal level.

5.2.4.2 Training sequences

In this section we test the influence of the transmitted sequence in the evolution of the taps of the equalizer during LMS adaptation, the test is made with reference channel 2 and using traditional LMS.The sequences tested were PRBS9, CJTPAT and CRPAT. PRBS9 consists on a maximum length sequence generated by a register with 9 memory positions, and its considered to be the most random sequence possible, this pattern is used to describe the random nature of the transmitted data.
CJTPAT(compliance jitter test pattern) consists of a long run of low-density pattern, followed by a long run of high transition density pattern, followed by another short run of low-density pattern, this pattern is destined to stress the receiver CDR to large phase drifts.
CRPAT(compliance random pattern) is designed to be have a relatively flat frequency spectrum used in jitter tests. The results for the evolution of a one tap DFE working in training mode with PRBS9, CJTPAT and CRPAT are represented in figure 5.23:

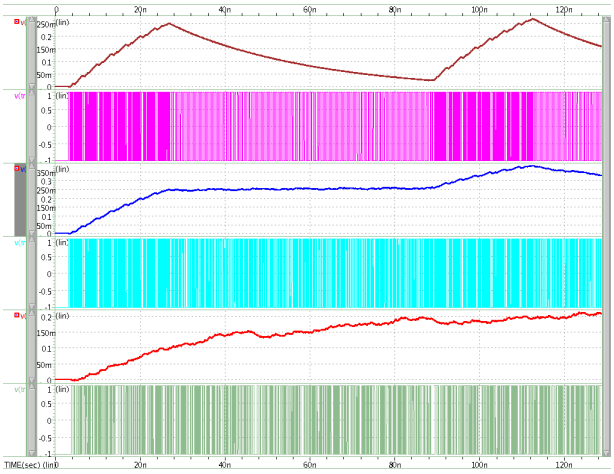


Figure 5.19: Tap evolution with (from top to bottom) CJTPAT, CRJPAT,PRBS9

As we can see only the Prbs made the LMS algorithm converge to the correct tap value, this is because the PRBS9 is the most random sequence exposing the equalizer to an almost flat signal spectrum. The CJTPAT makes the equalizer converge to two different tap values, at first when the equalizer is exposed to the high transition density pattern the tap value increases. This is happens because the equalizer is exposed to a sequence with a lot of ISI, since the signal is dominated by

its high frequency components. When the sequence is followed by a low density transition pattern the equalizer sees a small quantity of ISI. This causes the value that was present in the tap to be too high and the error inverts its polarity making the algorithm to lower the equalizer tap.
When using the CRPAT the equalizer seems to evolve to the correct value but the sequence presents an area with a high density transition patterns and the tap value increases to the incorrect value.
From this simulation we can conclude that only a PRBS sequence can be used during the training period of the equalizer.

5.2.4.3 DFE working in decision vs Training mode

In this section we study the evolution of the taps of the equalizer when working in training mode and in decision direct mode.For the DFE was considered an adaptation step of 0.0025.

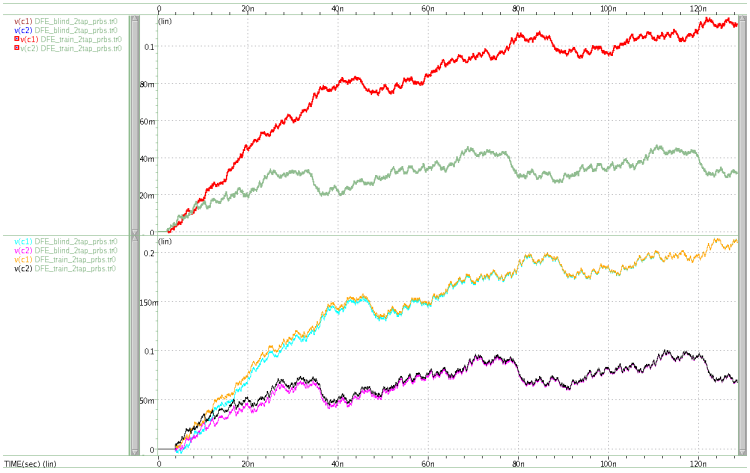


Figure 5.20: Taps evolution for training and decision direct for reference channel 1 and 2

In the first graph we see the evolution of the filter taps when transmitting using reference channel 1. As we can observe the tap evolution was exactly the same when using training mode or when using Decision direct mode, this is because when using reference channel 1 the equalizer does not commit decision errors which leads to the same dynamics for the taps in both modes of operation. When using the reference channel 2 the situation is a little different as sometimes the receiver commits decision errors when its transmitted a chain of more than 4 bits of the same value(1's or 0's). We can see that the evolution of the taps its also very similar, although in the beginning the decision direct takes a little more time to converge, in result of decision errors that cause the coefficient update to decrease the taps values instead of increasing them. We can see that for both the reference channels the Least mean square algorithm presents good behaviour in converging to the correct value in less than $1\mu s$. Although this is a promising result in practice is not possible to realize all the operations involved in one iteration of the LMS algorithm in 1UI when working at 12 Gbps. In the following sections we will test some simplifications to the original algorithm that reduce the complexity of the operations required at each iteration of the LMS algorithm.

5.2.4.4 Sign LMS vs LMS

This section is tested the convergence difference of the tap values for the DFE, when considering the sign LMS algorithm. The figure shows the evolution of the coefficients of a two tap DFE when considering sign LMS vs traditional LMS while performing transmission using reference channel 2.

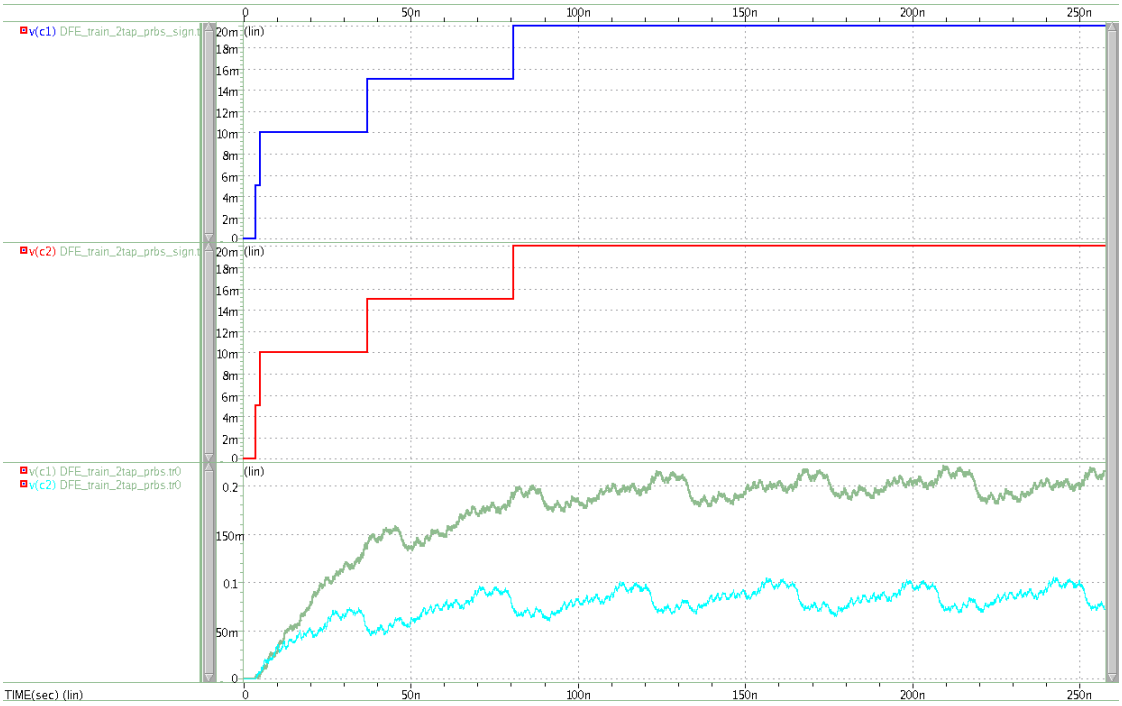


Figure 5.21: Taps evolution using traditional LMS and sign(e) LMS

In the figure we can see the step behaviour for the tap values of the DFE when considering sign LMS. This happens because the taps evolve by increasing or subtracting 2μ to the current tap value.

In the evolution of the taps for the sign LMS algorithm we can notice that the final value for the taps is not the same as the optimum value calculated through the channel impulse response. This happens because as the value of the tap increases the DFE is able to correct the errors induced by the channel making the taps to stabilize their value. The implementation using sign LMS is a major simplification when considering the traditional LMS algorithm as it only considers the signal of the error in the tap evolution expression.

5.2.4.5 Adaptation step influence

As we saw in chapter 2 the choice of the adaptation step is of great importance because it controls the trade off between convergence speed and gradient error. In this simulation we show the evolution of the values of the tap filer in function of the variation of the step size.

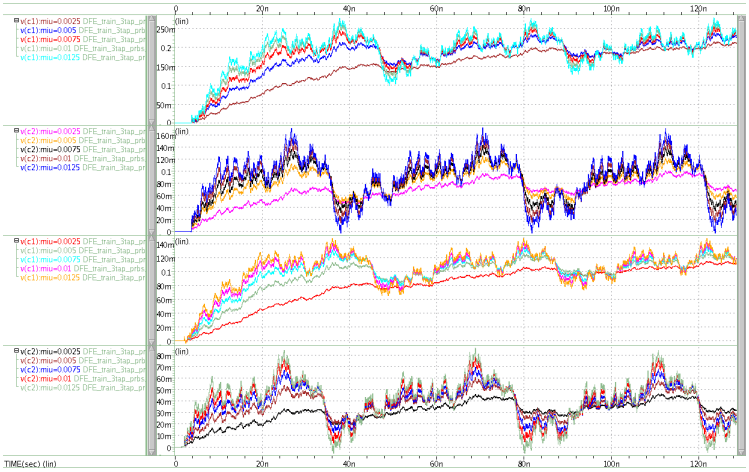


Figure 5.22: Tap evolution with $\mu = 0.0025, 0.005, 0.0075, 0.01, 0.0125$

In this figure is demonstrated the trade of between convergence speed and gradient error for the LMS algorithm applied to the DFE. In the first two graphs is shown the evolution of the two taps of the DFE when using both reference channels. Notice that all the curves converge because the steps sizes used are smaller than the theoretical value of $\frac{1}{NE[y^2(n)]}$. Although being smaller than the required for convergence the step size of 0.0125 already causes inadmissible gradient error for the adaptation process.

5.2.4.6 Adaptation in the presence of noise

To observe the behaviour of the taps evolution in the presence of noise it was create a Gaussian noise source modelled in Verilog-A. The code for the verilog A noise source is the following:

```
module whiteNoiseGenerator(in, out);
    electrical in,out;
    parameter real timeStep=5p;
    parameter real mean = 0, variance=0.01;
    parameter real td=0.01n, tr=0.01n, tf=0.01n;
    integer seed;
    real x;

    analog begin
        @ (initial_step) seed=-561;
        @(timer(0, timeStep))
```

```
            begin
                x=$dist_normal(seed, mean, variance)+V(in);
            end
        V(out)<+transition(x, 0, tr, tf);
    end
endmodule
```

The noise source consists in a module that samples the input, a noiseless waveform, and outputs the waveform plus a noise sample.The module inserts noise at a time step defined in the module,the noise sample is taken from a Gaussian distribution $\mathcal{N}(\mu,\delta^2)$ with $\mu = 0$ and δ^2 passed as an argument to the module. The time step chosen for simulation was $5ps$ making the module to insert $UI/(5p) \approx 16 samples/UI$. For the simulation we considered the insertion of noise with variance equal to $0.01, 0.05, 0.1, 0.25, 0.5$ at the entrance of the DFE. The results show the evolution of the Taps when performing equalization of reference channel 2,while working in training mode and decision direct mode.

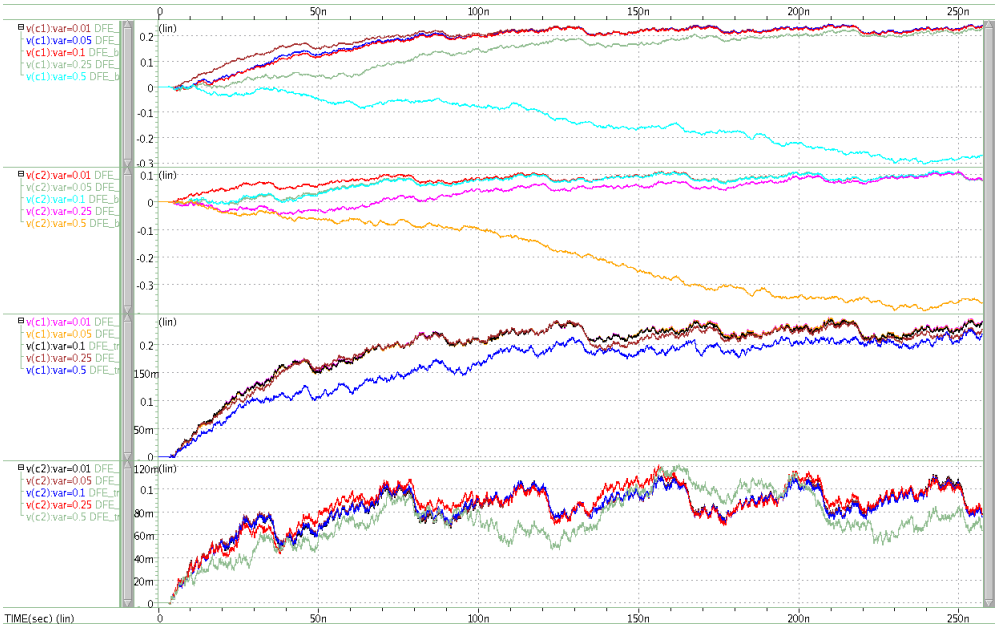


Figure 5.23: Taps evolution in training mode and decision direct mode

As we can see from the simulation only the training mode is sufficiently robust to withstand the presence of such low SNR.The explanation on why the decision direct mode fails is because for high levels of noise the decision device incorporated in the DFE starts to make decision errors.The errors caused by the noise make the DFE taps to diverge from their proper value.

5.3 Final remarks

In this chapter were tested several techniques to adapt the continuous time linear equalizer and the decision feedback equalizer to the transmission environment. For the adaptation of the CTLE were tested three algorithms: the voltage histogram,the edge histogram and the DC balance algorithm. The DC balance algorithm was unable to select the proper CTLE settings that better compensated for the channel ISI. The voltage histogram algorithm was able to select the proper CTLE setting but at the cost of a lot of computation power for the calculation of the histograms. The adaptation process took more than $200\mu s$ to select the best setting for the CTLE. The edge histogram algorithm showed very promising results, selecting the best setting for the CTLE in less time and with a less complex process than the voltage histogram algorithm. For the DFE, the algorithm using DC balance was also unable to properly select the optimum value for the taps of the filter. The voltage histogram proved to be too complex for the adaptation of the DFE.The algorithm would need to calculate a voltage histogram for all the tap values tested, This would make the process of adaptation of the DFE too long for the interface in question. The edge algorithm also showed very interesting results performing adaptation of the DFE to the channel and seems to be a viable solution to the the problem of adaptation of both filters.
In this chapter we also studied the capabilities and limitation of using the LMS to perform channel adaptation to the Decision feedback equalizer.We tested the equalizer working in two different modes, in training mode and in decision Direct mode.We notice that the two modes presented the same tap evolution when considering a channel that does not introduced decision errors.We observed also the difference in convergence of the taps of the equalizer when considering sign LMS versus traditional LMS. In this simulation we were able to see that when only considering the sign of the error signal, the taps of the DFE do not converged to their optimum value.This was caused by the fact that in the middle of the convergence of the taps the DFE was already able to correct the errors introduced by the channel. The noise simulation also allowed to observe the limitation associated with Decision direct mode, in this simulation we observed that the tap values of the DFE started to diverge when the noise power increased. This phenomenon proved to be less important when using training mode as the equalizer was able to converge even when the noise introduced decision errors.

Chapter 6

Conclusions and future Work

6.1 Summary of work developed

This dissertation was focused on the study of adaptive equalization techniques to a high speed interface developed by Synopsys. At first it was made a study on the theme of equalization of high speed interfaces to reduce the effects of inter-symbol interference. An initial research was conducted to understand some concepts relatively to the theme of equalization: the purpose of the equalization and what benefits it brings to a digital communication system. The review was focused on the main branches of equalization. Equalization with filters, is a technique that compensates the high frequency dependent loss of a channel by introducing a high frequency boost. This method of compensation has the objective to make the overall frequency response as flat as possible to reduce the channel induced ISI. Some methods to perform adaptation of these filters were approached and, and in chapter 2 are presented methods to perform adaptation of a continuous time linear equalizer and a decision feedback equalizer.
After this, the actual work began with the characterization of the interface that was intended to host adaptive equalization. The study of the interface was focused on its architecture and its operating modes. The interface has the objective to work at $12Gbps$ and has severe power restrictions because it is dedicated to the mobile industry.
Then the reference channels used by the interface were carefully studied, as the adaptive algorithm is highly dependent of the impairments of the channels. It was concluded that the channels used by the interface presented very similar properties among them. Another important fact is that the protocol restricts the use of the interface in environments that present worse frequency behaviour than the one presented by the reference channels characterized. The frequency response of the reference channels was described and it was made data integrity analysis of the data transmitted by the interface working at $12Gbps$. The need for equalization was justified as the eye diagrams at the receiver violated the eye masks proposed by the protocol. To correct this, it was proposed an equalization structure consisting of an FIR filter(with fixed settings) at the transmitter and two equalizers at the reception, a CTLE with 16 boost settings and an adaptive DFE.

In chapter 4 are presented a a set of CTLE transfer curves created through optimization in ADS. While in this chapter are also presented Verilog A models created to describe the behavior of the DFE and the receiver. Then using both ADS, Matlab and Hspice the equalization structure proposed in chapter 3 was simulated and it was verified that many of the channel induced ISI could be eliminated using the proposed structure. Finally in chapter 5 is made a study of the applicability of the methods described in chapter 2 to adapt the equalizing filters to the interface. It was possible too observe that the methods that consisted in making the voltage histograms of the waveform after the equalizers correctly selected the optimum setting. Although the success in selecting the proper setting for the interface the algorithm presented too much of a a heavy load in terms of computation power required for the interface. So next it was tested a simplification of the algorithm that was based on DC balance of data transmitted by the interface, unfortunately the algorithm was not robust enough in making the correct selection of the settings for the equalizers. Next, it was tested a new algorithm for the adaptation of both filters that consisted in making the histogram of the passing through zero of the waveform after the equalizer.The algorithm states that the setting for the equalizer that minimizes the variance of the edge histogram is the the optimum setting for the equalizer. The algorithm showed promising results, providing the interface with a robust and simple method to perform adaptation of both the DFE and the CTLE. For the adaptation of the DFE was also tested the least mean squares algorithm, this algorithm also was capable of selecting the correct value for the DFE taps during channel adaptation.

6.2 Future Work

Despite the good results presented in simulation, some of the techniques employed might not work in real implementations. This is a possibility that must be considered, as in simulation some factors could have not been taken into account. So a lot of the the possible future work is related to actual implementation of the developed algorithms to a real system on chip containing the interface. The implementation of the adaptive algorithms could be first tested using an FPGA that would implement the algorithm and then send to the interface commands to configure the equalizing filters. Regarding the success of the FPGA implementation, the next step would be ASIC implementation

Appendix A

Simulating CTLE and DFE in Spice

In the following pages is declared how to simulate Verilog A components using Hpsice

A.1 Verilog A components

In this section is presented the code for the Verilog A blocks used in the Spice simulations:

```
//receiver module converts a differential signal into a single ended signal
module Receiver(inP,inN,out);
    electrical inP,inN,out;
    input inP,inN;
    output out;
    real vout,vdiff;
    parameter real thr = 0,td=0.1p,tr=0.1p,tf=0.1p;

    analog begin
        vdiff=V(inP)-V(inN);
        V(out)<+ transition(vdiff/2,td,tr,tf);\\slew rate filter
    end
endmodule
}

//tuned CTLE ,creates a  passive filter with 15 different high frequency boosts.
module tunedCTLE(in,out,sel);
    input in,sel;
    output out;
    electrical in,out,sel;
    parameter real   G=1e9;
    parameter real   thr0=  0,thr1=0.1, thr2=0.2,thr3=0.3,thr4=0.4,
                     thr5=0.5,thr6=0.6,thr7=0.7,thr8=0.8,thr9=0.9,
                     thr10=1.0,thr11=1.1,thr12=1.2,thr13=1.3,thr14=1.4,
                     thr15=1.5,thr16=1.6;
```

```
parameter real   z0=3.810*G,z1=2.389*G,z2=2.059*G,z3=1.894*G,
                 z4=1.614*G,z5=1.414*G,z6=1.244*G,z7=1.044*G,
                 z8=0.964*G,z9=0.864*G,z10=0.764*G,z11=0.674*G,
                 z12=0.594*G,z13=0.534*G,z14=0.484*G,z15=0.424*G;

parameter real   p0=5.820*G,p1=3.808*G,p2=2.898*G,p3=4.028*G,
                 p4=4.128*G,p5=4.098*G,p6=4.078*G,p7=3.738*G,
                 p8=3.828*G,p9=4.078*G,p10=4.008*G,p11=3.998*G,
                 p12=3.938*G,p13=3.998*G,p14=4.128*G,p15=3.948*G;
parameter real   q=5.83*G;
parameter real   Ron =1,Roff=1*G;
electrical out0,out1,out2,out3,out4,out5,out6,out7,out8,out9,out10,
       out11,out12,out13,out14,out15,out16;
integer R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16;
real iout,sele;

analog begin
    @(initial_step) begin
        R0=Roff;R1=Roff;R2=Roff;R3=Roff;R4=Roff;R5=Roff;R6=Roff;R7=Roff;R8=Roff;
        R9=Roff;R10=Roff;R11=Roff;R12=Roff;R13=Roff;R14=Roff;R15=Roff;R16=Roff;
    end
    sele=V(sel);
    case(1)//used to calculate the selected setting
        ((sele>=thr0) & (sele<thr1)):
            begin
                 R0=Ron;
            end
        ((sele>=thr1) & (sele<thr2)):
            begin
                 R1=Ron;
            end
        ((sele>=thr2) & (sele<thr3)):
            begin
                 R2=Ron;
            end
        ((sele>=thr3) & (sele<thr4)):
            begin
                 R3=Ron;
            end
        ((sele>=thr4) & (sele<thr5)):
            begin
                 R4=Ron;
            end
        ((sele>=thr5) & (sele<thr6)):
            begin
```

```
            R5=Ron;
        end



    ((sele>=thr6) & (sele<thr7)):
        begin
            R6=Ron;
        end
    ((sele>=thr7) & (sele<thr8)):
        begin
            R7=Ron;
        end
    ((sele>=thr8) & (sele<thr9)):
        begin
            R8=Ron;
        end
    ((sele>=thr9) & (sele<thr10)):
        begin
            R9=Ron;
        end
    ((sele>=thr10) & (sele<thr11)):
        begin
            R10=Ron;
        end
    ((sele>=thr11) & (sele<thr12)):
        begin
            R11=Ron;
        end
    ((sele>=thr12) & (sele<thr13)):
        begin
            R12=Ron;
        end
    ((sele>=thr13) & (sele<thr14)):
        begin
            R13=Ron;
        end
    ((sele>=thr14) & (sele<thr15)):
        begin
            R14=Ron;
        end
    ((sele>=thr15) & (sele<thr16)):
        begin
            R15=Ron;
    default
```

```
                begin
                     R16=Ron;
                end
          endcas
           V(out0)<+1*laplace_zp(V(in),{-2*3.1415*z0,0},{-2*3.1415*p0,0,-2*3.1415*q,0});
V(out1)<+0.891*laplace_zp(V(in),{-2*3.1415*z1,0},{-2*3.1415*p1,0,-2*3.1415*q,0});
V(out2)<+0.7493*laplace_zp(V(in),{-2*3.1415*z2,0},{-2*3.1415*p2,0,-2*3.1415*q,0});
V(out3)<+0.707*laplace_zp(V(in),{-2*3.1415*z3,0},{-2*3.1415*p3,0,-2*3.1415*q,0});
V(out4)<+0.6309*laplace_zp(V(in),{-2*3.1415*z4,0},{-2*3.1415*p4,0,-2*3.1415*q,0});
V(out5)<+0.5623*laplace_zp(V(in),{-2*3.1415*z5,0},{-2*3.1415*p5,0,-2*3.1415*q,0});
V(out6)<+0.5011*laplace_zp(V(in),{-2*3.1415*z6,0},{-2*3.1415*p6,0,-2*3.1415*q,0});
V(out7)<+0.4466*laplace_zp(V(in),{-2*3.1415*z7,0},{-2*3.1415*p7,0,-2*3.1415*q,0});
V(out8)<+0.3981*laplace_zp(V(in),{-2*3.1415*z8,0},{-2*3.1415*p8,0,-2*3.1415*q,0});
V(out9)<+0.3548*laplace_zp(V(in),{-2*3.1415*z9,0},{-2*3.1415*p9,0,-2*3.1415*q,0});
V(out10)<+0.345*laplace_zp(V(in),{-2*3.1415*z10,0},{-2*3.1415*p10,0,-2*3.1415*q,0});
V(out11)<+0.2813*laplace_zp(V(in),{-2*3.1415*z11,0},{-2*3.1415*p11,0,-2*3.1415*q,0});
V(out12)<+0.2551*laplace_zp(V(in),{-2*3.1415*z12,0},{-2*3.1415*p12,0,-2*3.1415*q,0});
V(out13)<+0.2238*laplace_zp(V(in),{-2*3.1415*z13,0},{-2*3.1415*p13,0,-2*3.1415*q,0});
V(out14)<+0.1999*laplace_zp(V(in),{-2*3.1415*z14,0},{-2*3.1415*p14,0,-2*3.1415*q,0});
V(out15)<+0.1778*laplace_zp(V(in),{-2*3.1415*z15,0},{-2*3.1415*p15,0,-2*3.1415*q,0});
V(out16)<+ V(in);

iout=V(out0,out)/R0 + V(out1,out)/R1 + V(out1,out)/R2+ V(out3,out)/R3 + V(out4,out)/   R4 + V
 V(out9,out)/R9 + V(out10,out)/R10 + V(out11,out)/R11+ V(out12,out)/R12 + V(out13,out)/R13 +

I(out)<+iout;
end
endmodule

module adaptiveTwoTapDfe(in,clk,out,sel,training,tap1,tap2,error);
    input in,clk,sel,training;
    output out,tap1,tap2,error;
    electrical in,clk,out,sel,training,tap,error;
    parameter real Vth=0.5,vhigh=1,vlow=-1,dataVth=0;
    parameter real td=0.01n,tr=0.01n,tf=0.01n;
    parameter real miuT=0.0025,miuD=0.0025,iniTap1=0,iniTap2=0;
    real sample,c1,c2,sele,trainSample;
    real x,y,y1,y2;
    integer i;

    analog begin
        @(initial_step) begin
            sample=0;
            trainSample=0;
            c1=iniTap1;  c2=iniTap2;
            x=0;
```

```
        sele=0;
        c1=0; c2=0;
        y=0;
        y1=0; y2=0; err=0;
    end
    @(cross(V(clk)-Vth,+1)) begin
        i=i+1;
        sele=V(sel);
        if(sele<Vth) begin
            sample=V(in);
            trainSample=V(training);
            x=sample-c1*y-c2*y1;
            y2=y1;
            y1=y;
            if(x>dataVth) begin
                y=vhigh;
            end
            else begin
                y=vlow;
            end
            c1=c1-miuT*(trainSample-x)*y1;
            c2=c2-miuT*(trainSample-x)*y2;
            err=trainSample-x;
        end
        else begin //decision directed
            sample=V(in);
            x=sample-c1*y-c2*y1;
            y2=y1;
            y1=y;
           if(x>dataVth) begin
                y=vhigh;
           end
           else begin
                y=vlow;
           end
           c1=c1-miuD*(y-x)*y1;
           c2=c2-miuD*(y-x)*y2;
           err=y-x;
       end
    end

    V(out)<+ transition(y,0,tr,tf);
    V(tap1)<+ transition(c1,0,tr,tf);
    V(tap2)<+ transition(c2,0,tr,tf);
    V(error)<+transition(err,0,tr,tf);
```

```
    end
endmodule


module whiteNoiseGenerator(in,out);
    electrical out,gnd;
    parameter real timeStep=10p;
    parameter real mean = 0,variance=0.01;
    parameter real td=0.01n,tr=0.01n,tf=0.01n;
    integer seed;
    real x;
    analog begin
        @ (initial_step) seed=-561;
        @(timer(0,timeStep))
        begin
            x=$dist_normal(seed,mean,variance)+V(in);
end
        V(out)<+transition(x,0,tr,tf);
    end
endmodule
```

A.2 Spice simulation of Verilog A Components

The following spice code performs the simulation of a two tap DFE:

```
Teste with prbs
.options accurate nomod
.options list node post
.option post
.temp  70
.param Sel =0    **parameter to select between training mode or decision direct
.param delt=0
.param miu=0.025             **adaptation step
.param delay1 ='1.273n'     **group delay for reference channel 1
.param delay2 = '2.98n-UI/2' **group delay for reference channel 2
 **bit paterns specifications
.param bitRate=12G UI='1/bitRate' vh=1 vl=-1 td='0.1*UI' tr='0.1*UI' tf='0.1*UI'
**receiver parameters
.param thr=0 vh=1 vl=-1
.param tap = 0
.param clkPeriod= 'UI*2.3'
*********************** paterns generation /PRBS*************
vup    n0 0 pat (vh vl td tr tf UI        b00000000011111000010000011101000110........)
vdwn   n1 0 pat (vl vh td tr tf UI        b00000000011111000010000011101000110........)
vtrain train 0 pat(vh vl delay2 tr tf UI  b00000000011111000010000011101000110.......)
********loading hdl components  and channel**************
.hdl "../models/Components.va"
.include "../models/hspiceModelref1.hsp"
.include "../models/hspiceModelref2.hsp"
****** signals ******************
Vsel sele 0 DC='Sel'
**geradoras suport signals
VdfeCLk dfeClk 0 PULSE (0 1 'delay2+UI/2' tr tf 'UI/2' UI)
*** circuit ***
Xch n2 n3 n4 n5 0 bbspice_ref2_subckt    **channel
Xre n4 n5 n6 Receiver thr=0 vhigh=vh vlow=vl td=td tr=tr tf=tf **receiver
Xdfe n6 dfeClk n7 sele train tap err adaptiveTwoTapDfe miuT=0.025 miuD=0.025 iniTap= tap

R1 n0 n2 50
R2 n1 n3 50
R3 tap 0 50
R4 err 0 50
*** analises ***
.tran step=1p stop='100*UI'
.end
```

References

[1] Bernard Sklar. *Digital Communications,Fundamentals and Applications*. Prentice Hall, Ssecond edition, 2001.

[2] Wang-Soo Kim and Woo-Young Choi. A 0.18 micro m cmos 3.5gb/s continuous time adaptive cable equalizer using enhanced low-frequency gain control method. *IEICE Electronics Express*, 2011.

[3] Peffers. Michael A. The benefits of using linear equalization in backplane and cable applications. *Applications report*, June 2013.

[4] Jonathan E. Proesel and Timothy O. Dickson. A 20-gb/s, 0.66-pj/bit serial receiver with 2-stage continuous-time linear equalizer and 1-tap decision feedback equalizer in 45nm soi cmos. *VLSI Circuits (VLSIC), 2011 Symposium*, 2011.

[5] Angus Mclaren. Advanced equalization techniques for pcie 8gbt/s. *Applications report*, 2011.

[6] C. Richard Johnson Rad A. Casas, Zhi Ding Rodney A. Kennedy. Blind adaptation of decision feedback equalizers based on the constant modulus algorithm. 2011.

[7] Sam Palermo. High-speed links circuits and systemsspring 2014, Spring 2014. Texas A&M University, disponível em http://http://www.ece.tamu.edu/~spalermo/ecen689/lecture10_ee720_jitter.pdf.

[8] Wolfgang Freude and René Schmogrow. Quality metrics for optical signals eye diagram, q-factor, osnr, evm and ber. *International Conference on Transparent Optical Networks*, 2012.

[9] Moon-Sang Hwang Jong-Sang Choi. A 0.18 um cmos 3.5-gb/s continuous-time adaptive cable equalizer using enhanced low-frequency gain control method. *IEICE Electronics Express*, 2004.

[10] Wang-Soo Kim and Woo-Young Choi. A 10 gb/s low-pwer adaptive continuous-time linear equalizer using asynchronous under-sampling histogram. *IEICE Electronics Express*, 2013.

[11] Gunhee Han Dongmyung Lee, Yungwon Han. A 8.5-gbit/s fully integrated cmos optoeletronic receiver using slope detection adaptive equalizer. *Solid State Circuits*, 2012.