3.125Gbps FIR EQUALIZER IMPLEMENTATION IN 65nm CMOS TECHNOLOGY

by

Hande AKIN KURNAZ

B.S., Microelectronics Engineering, Sabancı University, 2004

Submitted to the Institute for Graduate Studies in

Science and Engineering in partial fulfillment of

the requirements for the degree of

Master of Science

Graduate Program in Electrics-Electronics Engineering

Boğaziçi University

2008

3.125Gbps FIR EQUALIZER IMPLEMENTATION IN 65nm CMOS TECHNOLOGY

APPROVED BY:

Prof. Günhan Dündar   . . . .
(Thesis Supervisor)

Prof. Cem Ersoy   . . . .

Assist. Prof. Şenol Mutlu   . . . .

DATE OF APPROVAL: 20.06.2008

ACKNOWLEDGEMENTS

ABSTRACT

3.125Gbps FIR EQUALIZER IMPLEMENTATION
IN 65nm CMOS TECHNOLOGY

This thesis describes channel degradation in a basic telecommunication system with its sources (crosstalk and metallic channel loss) and results (inter-symbol interference). Compensation of this channel degradation via methodology called equalization is focused on. Adaptive equalization techniques such as zero forcing, least mean squares (LMS), recursive least squares (RLS) and constant modulus algorithm (CMA) are theoretically explained and LMS and RLS are supported with regarding MATLAB Simulink simulations using 30inch PCB trace model as the channel model. Comparison of adaptation algorithms, equalization cost functions and tap spacing of tapped delay line in FIR equalizer in Simulink are also held for this thesis. Coefficients obtained from Simulink environment are used to verify performance of FIR equalizer designed in STMicroelectronics CMOS065 (65nm) technology for 3.125Gbps data rate. Building blocks of FIR equalizer are analyzed in detail and design limitations are summarized. Simulations showed that closed eye at the receiver after 30inch PCB channel, can be cleaned up to data eye with 28ps jitter by means of 4-tap FIR equalizer with T/8 tap spacing operating at 1.2V power supply, 3.125Gbps data rate and at the expense of only 13mA of current consumption.

ÖZET

3.125Gbps HIZINDA 65nm CMOS TEKNOLOJİSİNDE
FIR EŞİTLEYİCİ UYGULAMASI

Bu tez telekominikasyon sistemlerinde kanala bağlı bilgi bozulmasının sebeplerini (çapraz karışma ve metalik kanal kayıpları) ve sonuçlarını (simgeler arası karışma-ISI) açıklamaktadır. Tezin devamında bu bozulmanın eşitleyici sayesinde telafi edilmesi üzerinde yoğunlaşılmıştır. Telafi metodunun adaptif olanları (RLS, LMS, CMA, ZF) teorik olarak incelendikten sonra, MATLAB Simulink adındaki simulasyon ortamında simulasyonlarla desteklenmiştir. Simulasyonlar esnasında 30inç PCB kanal modeli kullanılmıştır. Adaptasyon algoritmaları, eşitleyicilerin maliyet fonksiyonları, çıkma aralıklarının etkisi karşılaştırmalı olarak incelenmiştir. Simulink ortamından elde edilen katsayılar sonlu dürtü yanıtı (FIR) eşitleyicinin transistor seviyesindeki tasarımın performansını, STMicroelectronics CMOS065 (65nm) teknolojisini kullanarak, 3.125Gbps hızında verifikasyonu için kullanılmıştır. FIR eşitleyicinin ana blokları detaylı olarak incelenmiş ve uygulama limitasyonları özetlenmiştir. Simulasyonlar kanal modelinden sonraki kapalı göz biçimindeki iletilmiş bilginin, eşitleyici kullanıldıktan sonra 28ps et kalınlığına kadar temizlendiğini göstermektedir. Bunun için T/8 aralıklı gecikmelerden oluşan 4-çıkmalı sonlu dürtü yanıtı eşitleyici, 1.2V güç kaynağı kullanarak, 3.125Gbps veri hızında ve sadece 13mA akım harcayarak gerçeklenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

CMFB	Common Mode Feedback
CMA	Constant Modulus Algorithm
E()	Expectation Operator
FSE	Fractionally Spaced Equalizer
∇	Gradient
()*	Hermetian Transpose
LMS	Least Mean Squares
LA	Limiting Amplifier
SSE	Symbol Spaced Equalizer
RLS	Recursive Least Squares
RMS	Root Mean Square
$()^{T}$	Transpose
UI	Unit Interval
ZF	Zero Forcing

1. INTRODUCTION

1.1. Equalization for High Speed Communication

The simplest communication system (Figure 1.1) has three components, the transmitter to send the data converting the digital bits into electrical or optical data, the receiver to convert the analog data back into binary data and in-between a channel (copper wire, coaxial cable, optical fiber etc.) which makes this data transmission possible.
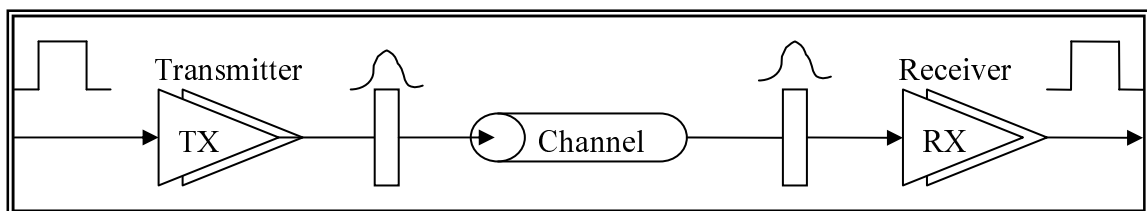
Figure 1.1. Block diagram of a typical high-speed digital data transceiver

Every material in nature shows filtering behavior. Since the transmission channel is not an all-pass filter, it may cause some deterioration on the transmitted data by behaving different frequency components in a different way. A transmission channel is usually a low-pass filter, and different frequency components lose different amplitude of power (frequency dependent loss) and experience different amount of phase distortion during propagation through the channel. These two most dominant effects contribute to the corruption of the original signal, which is the well-known inter-symbol interference (ISI).

Depending on channel characteristics and receiver specifications, data might be required to be recovered to become understandable by the receiver side. Especially with the increasing data rate requirements, to take some responsibility over the shoulders of basic components of communication system for high quality transmission, additional block called equalizer started to be used. The mission of this additional block is simply recovering the data deteriorated due to non-idealistic channel characteristics, and helping

the receiver to sample the arrived data correctly. It achieves this mission by removing or reducing the ISI on the data. An equalizer provides an inverse channel response such that the overall (i.e. combination of channel and equalizer) frequency response is flat over the bandwidth of interest. Since most common channel behavior is low pass filter, a high pass filter compensating the channel transfer function can be placed to the transmission path to obtain flat transfer function before data sampling. This is shown in Figure 1.2.
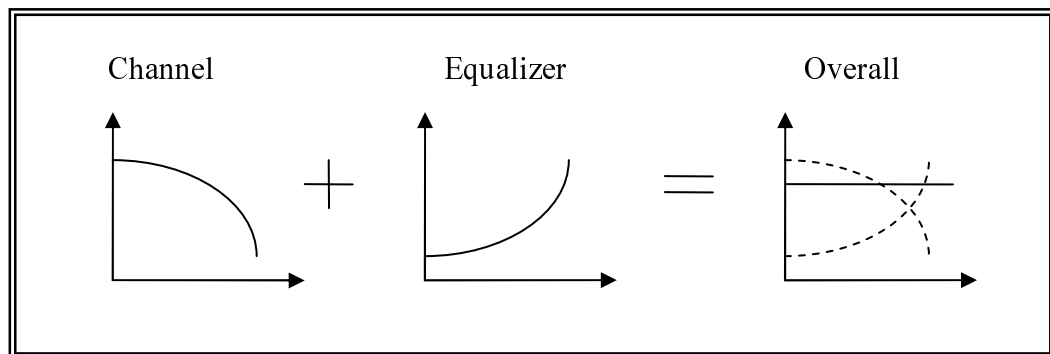
Figure 1.2. Effect of channel equalization

There are many ways to implement an equalizer. Equalizer can be placed either on the transmitter side or receiver side. Equalization on transmitter side is named as pre-emphasis. This method is used to amplify the high frequency components of the transmitted data or decrease the strength of the low frequency components. This way to equalize the transfer function of the path till receiver for all the frequency ingredients of the data stream is aimed. However, for this method to be applicable, the channel characteristics should be known a priori to be compensated before the transmission takes place. Equalization on receive side is also possible. Receiver equalizers can be composed of only passive components (passive equalizers). They can also be implemented as linear transversal filters and called linear transversal equalizers. The latter is advantageous due to its less area occupation.

Since the channel is before the equalization in RX equalization, in case the channel characteristics are not known a priori, they can be estimated by the help of some adaptation algorithms. The most common way to achieve this is to train the equalizer by comparing the equalized data with the ideal stream and trying to minimize the error in between. This

way the channel characteristics might be estimated and compensated accordingly. This method is called adaptive equalization.

In this thesis, different examples to adaptive equalization at the receiver side are implemented using powerful tool, MATLAB Simulink. Different adaptation algorithms, cost functions and equalizer properties of linear transversal equalizers (such as number of taps, tap spacing etc.) are investigated. In the end, basic building blocks of a linear transversal equalizer is implemented in Cadence environment using STMicroelectronics CMOS 65nm technology and simulated using Spectre simulation tool. These building blocks are used to implement an equalizer operating at 3.125Gbps and compensating TYCO 30inch channel characteristics, with the coefficients obtained from its MATLAB counterpart. The improvement, managed by the equalizer is observed comparing the eye diagrams of the data at the input of the receiver (output of the channel), and the output of the equalizer. The simulation results show that, the eye opening at the output of the equalizer is highly improved when compared to the almost closed eye at the output of the channel.

1.2. Literature Survey

When we go back in the history of equalization concept, the best summary that would come across belongs to Robert W. Lucky, inventor of adaptive equalization concept, who published a very detailed summary of equalizers designed between the years of 1968-1973 [1]. As the time passes, the market for high-bandwidth communications continues to drive the demand for higher speed transceivers and therefore, equalizers. In the wireline arena, 10Gb/s data transmission over copper channels (backplane PCB, UTP cable, etc.) for enterprise networking and data-center applications is reaching maturity and commercialization using low-cost CMOS transceivers. At the same time, up to 40Gb/s data transmission is being demonstrated over short channels [2].

The latest CMOS technology along with advances in design techniques has enabled us to continue to push the speed envelope. In addition to integrating more powerful signal-

processing functionality, such as FFE and DFE, on chip, innovative coding and circuit techniques help to overcome bandwidth limitations, adapt to different channels, and tolerate PVT variations while lowering power consumption and cost [2].

At relatively low bit rates, most adaptive equalizers have been implemented using a digital approach [3], [4]. Design of a digital equalizer at the receiver side involves a delay element and a decision circuit that requires a recovered clock. The extraction of the clock depends on the input data of the clock and data recovery (CDR) circuit, which increases the system complexity and could lead to problems with CDR locking. On the other hand, an analog approach is often preferred for higher speeds for its low power consumption and simplicity. A number of papers have been reported on analog cable equalization at bit rates on the order of 100 Mb/s [5], [6]. Some papers reported cable equalizers with bit-rates up to 3.5 Gb/s [7], [8]. In another paper, an analog adaptive equalizer running at 10 Gb/s using a BiCMOS fabrication process is represented [9]. More recently, a CMOS equalizer operating at 10 Gb/s was presented [10], [11] where more design effort was needed to overcome the gain limitations of CMOS. An analog FIR approach to 10 Gb/s equalization was presented in [12]. 20Gbps adaptive equalizer is implemented in 0.13um CMOS technology [5]. Using SiGe BiCMOS technology, even 49Gbps equalization using 7-tap transversal filter is achieved [13], [9].

For this specific thesis, a very nice paper written by Jin Liu et al. [17] is taken as a reference and implemented in 65nm, whereas the reference paper refers to 90nm CMOS technology design.

Some equalizer implementations are summarized in Table 1.1, according to their technology, input data rate, performance, chip area, power consumption and power supply values.

Table 1.1. Literature survey on equalizers

REF.	TECH.	INPUT DATA RATE	MAX. COMPENSATED LOSS	MAX. P-P JITTER	CHIP AREA	POWER	SUPPLY
[15]	0.35-µm CMOS	200Mb/s	1dB at 100MHz	N/A	1.3 mm2	19.5 mW	2.3V
[16]	0.25-µm BiCMOS	20Gb/s	20dB at 10GHz	~10ps	N/A	32 mW	2.5V
[17]	0.25-µm CMOS	2.5-3.5Gb/s	21dB at 1.25GHz	100ps	0.095 mm^2	95 mW	2.5V
[18]	0.18-µm CMOS	125Mb/s	N/A	2.61ns	27738 µm2	3.7 mW	1.6V
[9]	0.18-µm CMOS	10Gb/s	16.7dB at 5GHz	27.11ps	0.86×1.28 mm^2	34.2 mW	1.8V
[19]	0.18-µm CMOS	11.8Gb/s	12dB at 5.875GHz	47ps	1175×1135 µm^2	201 mW	1.8V
[20]	0.13-µm CMOS	10Gb/s	20dB at 5GHz	15.67ps	0.94x0.65 mm^2	133 mW	1.6V
[5]	0.13-µm CMOS	20Gb/s	15~20dB at 10GHz	14ps	0.8×0.25 mm^2	60 mW	1.5V
[20]	0.11-µm CMOS	10Gb/s	20dB at 5GHz	27.8ps	47×85 µm^2	13.2 mW	1.2V
[21]	90-nm CMOS	10Gb/s	20-30dB at 5GHz	N/A	N/A	130mW (w. PLL)	1V
[22]	90-nm CMOS	10Gb/s	23dB at 5GHz	N/A	270x200 µm^2	22 mW	1.2V

1.3. Thesis Organization

In this thesis, in Chapter 2, basic communication system components will be mentioned. It will be followed by channel non-idealities such as metallic channel loss, crosstalk noise and their effects on data transmission. ISI will also be focused on with the conditions leading to it, and its effects on the reliability of the data transmission. The next subject in this chapter is removal of ISI using equalization techniques.

Chapter 3 focuses on adaptive equalization techniques, that are used when the channel characteristics are not known a priori or when channel transfer function is time-variant. Introduction to adaptation algorithms such as Zero Forcing Algorithm, Least Mean Squares and Recursive Least Squares Algorithm and Constant Modulus will be provided.

Introducing the theoretical aspects of communication systems and adaptation methodologies, empirical phase of the study will be presented in the following chapters. Chapter 4 is a summary of equalizer implementation using MATLAB Simulink. In this chapter, comparison of different adaptation algorithms, cost functions, and comparison of linear transversal equalizers with different characteristics will take place. The coefficients obtained from MATLAB simulations will be used in CMOS implementation of the equalizer as well.

In Chapter 5, transistor level implementation of a linear transversal equalizer is discussed. Simulations with Spectre simulator are used to evaluate the performance of the equalizer by observing the eye diagrams of the data at the input and output of the equalizer. The performance of each building block used to implement the top level equalizer is also investigated.

Finally, conclusions are drawn in Chapter 6, and the directions of future work are discussed briefly.

2. COMMUNICATION SYSTEMS

Communication systems are systems that are responsible for information transfer between transmitting and receiving parts through a channel. At its simplest, the system contains modulator, transmitter, transmission channel, receiver and demodulator (Figure 2.1).

A modulator takes the source signal and transforms it so that it is physically suitable for the transmission channel. A transmitter actually introduces the modulated signal into the channel, usually amplifying the signal. A transmission channel that is the physical link between the communicating parties. A receiver detects the transmitted signal on the channel and usually amplifies it (as it will have been attenuated by its journey through the channel). A demodulator receives the original source signal from the received signal and passes it to the sink.
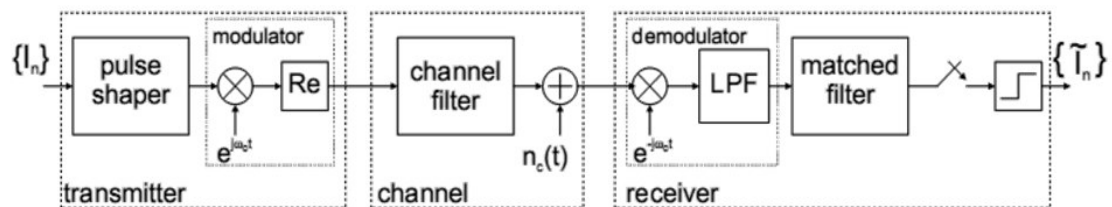
Figure 2.1. Basic communication system [35]

As well as the correct functionality of transmitter and receiver, the quality of the channel also determines accuracy in data transmission, which can be quantified via BER, jitter etc. measurements. Channel quality requirement became more visible in the picture with the higher speed demand in data transmissions. Non-ideal channel characteristics, such as channel bandwidth and crosstalk noise, often deteriorate the signal quality of the received signal and causes error in data recovery. This means, it is no longer sufficient to solely increase the speed of the ICs to achieve higher data rates and that's why new research areas emerged to improve bandwidth of transmission channels for less deterioration [21].

High bandwidth need leads industry standards to be developed. These standards define the channel characteristics and I/O electrical specifications of short reach (4-in, on-board) and long reach (30-in+, inter-card) serial links operating at data rates in excess of 10 Gb/s. While serial link transceivers in the 6-Gb/s range are often intended to extend the bandwidth of "legacy" backplane channels, reliable operation above 10 Gb/s will require in many cases improved channel characteristics. Therefore, the standards above 10 Gb/s are primarily aimed at new optical backplane designs benefiting from improvements in board, connector, and chip-level package technologies.

Even with improved backplane designs, however, the need to remain price-competitive, complexity of implementation and lower degree of integration will discourage adoption of the most exotic (and expensive) board, connector, and package technologies. As in the recent past, advanced equalization capabilities in the I/O circuitry will be employed to compensate for the signal distortions of lower cost interconnect technologies such as PCBs. Optimizing cost tradeoffs at the system level requires knowledge of how much equalization is needed for a specific combination of board, connector, and package technologies.

Before concentrating on equalization techniques, understanding of channel distortion and its effects on data deterioration is necessary.

2.1. Channel Non-idealities

There are various communication channels with distinct channel characteristics. They can be classified as on chip, chip to chip, board to board, box to box and system to system interconnects.

Interconnects less than 1cm lengths are defined as on chip interconnects. They can be manufactured from Al, Cu or CMOS compatible materials such as Si, SiO2. Chip to chip interconnects are between 1cm and 10 cm length ranges. The density of Cu traces on FR4 is constrained by EMI and crosstalk problems. On chip and chip to chip connections are

very promising interconnect candidates for the future. Board to board connection is the class for interconnects between 1m to 10m length. Due to higher bandwidth, they are potential applications for optical interconnect. Their materials are mainly polymer WG, fiber ribbon with VCSELs etc. Box to box interconnect is usually optical fiber channel which is longer than 10m.

Due to limited bandwidth of PCB trace material, optical channel has been considered for beyond 10Gbps data transmission for above chip-to-chip applications. Currently, as mentioned before, the main barriers are high cost, lower degree of integration, and complexity of implementation. For optical channels polarization mode dispersion (PMD), chromatic dispersion (CD) are the main physical effects degrading the signal integrity at very high speeds.

A typical backplane/line card application is shown in Figure 2.2.a. A long (30-in or more) transmission line on the backplane is used to transfer data from a processor or ASIC on one line card to a processor or ASIC on another line card. Several physical effects degrade signal integrity at data rates above a few gigabits per second. Skin effect and dielectric losses of the transmission lines become severe at these data rates. Via stubs on the circuit boards and other impedance discontinuities associated with the chip packages and connectors cause reflections easily observed in the channel impulse response (Figure 2.2.b) In the frequency domain, these reflections cause notches which further degrade the channel frequency response (Figure 2.2.c).
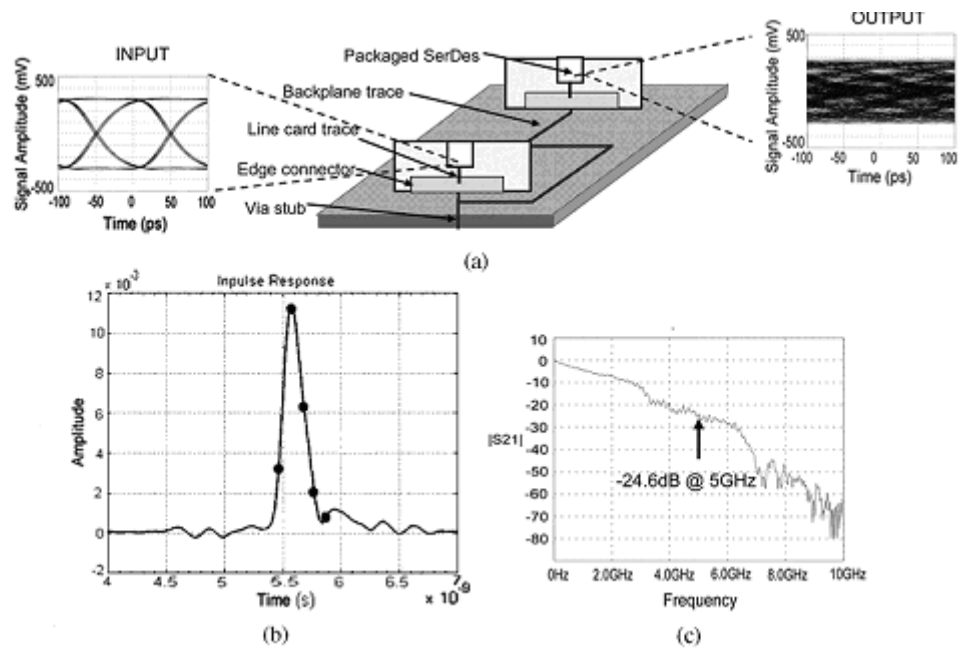
Figure 2.2. Backplane channel characteristics [21]

a) Backplane/line card application, b) Channel impulse response, (c) Channel

frequency response

Since the transmitted signal is attenuated by loss, it is easily corrupted by crosstalk from other channels. Even for greenfield backplanes with improved board technology, the loss at 5 GHz (Nyquist frequency for 10-Gb/s data) may be 20–30 dB. With the channel adding so much loss and distortion to the signal, the data eye at the far end of the link (Figure 2.2.a) is completely closed, and advanced equalization is required to recover the transmitted bits.

At this high data rate, the channel non-idealities result in signal loss and reflections as well as significant high-frequency crosstalk. Figure 2.2.b illustrates measurements of a typical legacy backplane channel with approximately 20 dB of loss at 3.125 GHz and crosstalk energy that actually exceeds the signal energy at slightly higher frequencies [21].

PCB interconnects are very popular in today's communication channels due to their low cost and less complexity and from now on this interconnection will be focused on. ISI, its effects and equalization to compensate ISI up to a certain extent will be discussed after focusing on the non-idealities - metallic channel loss, crosstalk noise - of the PCB channel.

2.1.1. Crosstalk

There are various noise sources that cause errors in data and clock recovery, for example, the crosstalk noise, the power supply noise, and reflection noise.

Power supply noise is induced by switching large currents in short duration across the parasitic inductance in power distribution network; it increases with the switching frequency of I/O driver, output signal swing, and number of switching drivers at the same time. Reflections are due to impedance discontinuities; common reflection noise for backplane applications includes card-to-board connectors, cable-to-card connectors, long vias with their respective end pads, wire bonds or flip-chip solder balls and orthogonal wiring [23]. Crosstalk is caused by the electromagnetic coupling between signal lines through mutual capacitance and mutual inductance. Among these various noise factors, the dominant one for backplane is the crosstalk noise, specially the near end cross talk (NEXT) at the connectors.
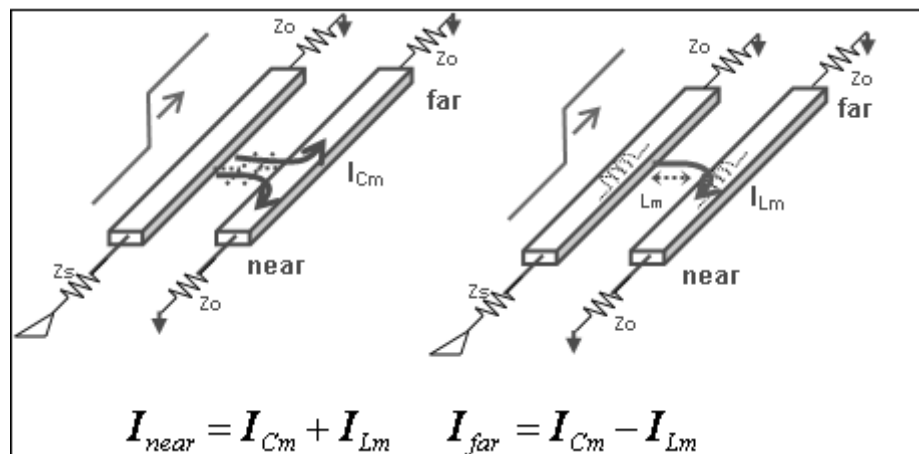
Figure 2.3. Near End & Far End crosstalk theory [24]

The mutual inductance will induce current on the victim line opposite of the driving current. (Lenz Law) The mutual capacitance will pass current through the mutual capacitance that flows in both directions on the victim line [24]. The near and far end victim line currents sum to produce the near and the far end crosstalk noise as represented in Figure 2.3. Since the current flowing to the near node is sum of both inductive and

capacitive currents, NEXT is always positive. FEXT is only positive when I_{LM} is less than I_{CM}, which is not very usual.

Figure 2.4 illustrates the near end crosstalk (NEXT) and the far end crosstalk (FEXT) in a representative communication system.
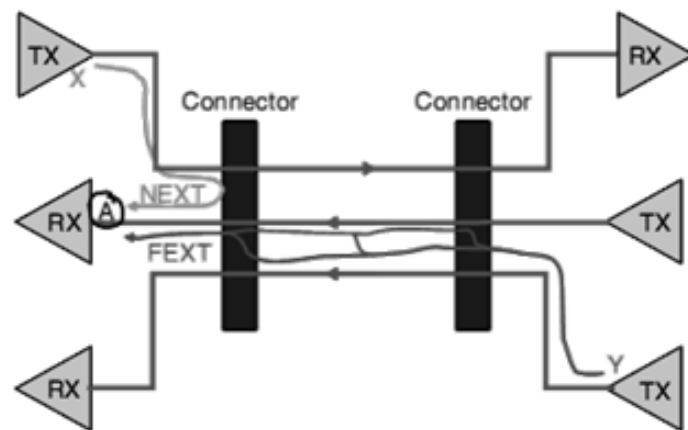
Figure 2.4. Near End & Far End crosstalk representation in a communication system [23]

For the receiver at point A in Figure 2.4, the crosstalk generated from nearby transmitter at point X is called NEXT. In this figure, only the major coupling path through the connector is illustrated. In reality, the PCB traces between the two connectors also contribute for NEXT. FEXT is generated from transmitter at the other end, point Y in the figure. FEXT also has multiple paths. Since the FEXT transfer channel has much longer distance than that of NEXT, the FEXT transfer function has much more severe attenuation than the NEXT transfer function. Therefore, NEXT is more critical to correct data recovery of weak received signal in receiver end. The NEXT transfer function increases with frequency [23]. For high-speed data transmissions, effective equalization method to mitigate NEXT has become necessary.

2.1.2. Metallic Channel Loss

For PCB channel in chip-to-chip communications, there are mainly two non-ideal characteristics that limit the data transmission rate and distance. The first was cross talk; the latter is the limited bandwidth due to frequency dependent channel loss. For all metallic media, including PCB traces and metallic cables like unshielded twist pair (UTP) cables, shielded twisted pair (STP) cables and coaxial cables, the channel losses at higher frequencies are mainly caused by skin effect and dielectric loss [23]. Other loss scheme like radiation loss is negligible even when the signal frequency is up to 10 GHz.

2.1.2.1. Sources of Metallic Channel Loss. When a DC signal propagates through a PCB trace, or through any wire for that matter, the current flows so that it's evenly distributed through the conductor cross section (Figure 2.5). As the signal frequency rises, the magnetic field pushes out of the conductor while the current crowds out toward the wall of the conductor. At even higher frequencies, the current is flowing mainly within the thin layer under the conductor surface, or under the "skin" of the conductor. In the absence of other conductors in proximity, the current distributes itself evenly along the perimeter of the conductor cross-section. This in turn leads to frequency dependence of the inductance and resistance. This phenomenon is called skin effect [25].
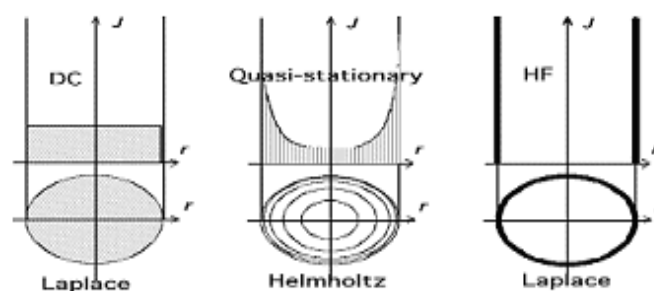
Figure 2.5. Current density J as a function of frequency. Current spread out towards the conductor surface as the frequency rises [25]

Dielectric loss is the loss of electromagnetic power due to the non-ideal characteristics of the dielectric material such as isolator around the conducting media during electromagnetic wave propagation. There are two main mechanisms responsible for non-ideal characteristics of the dielectric material. First, there is some, although very small, amount of direct current leakage through the dielectric even at zero frequency. Second, there is a polarization loss, which can be easily understood if the molecules of FR4, or other PCB materials, are viewed as dipoles. Water, for example, is a very prevalent molecule in FR4. When put in an alternating electric field, H_2O molecules tend to change their orientation following the electric field, just like a weathercock in a wind constantly changing its direction. It is the signal driver that causes the current to flow in the trace, and therefore causes the high frequency field to move down the trace along with the signal. The field, in turn, forces the molecules of the surrounding dielectric to oscillate. While doing so the field loses some energy. Since the signal and the field "around it" are inseparable, the signal loses some of its strength too. The dipole molecules of the dielectric react differently to the external field at different frequencies. From DC to some high frequencies of about several hundred MHz, these molecules easily follow the external alternating field. However, at even higher frequencies, they find it harder and harder to respond to the external force. As the field frequency increases, the dipole molecules of the dielectric, first, lag behind the field, and then totally ignore it at very high frequencies. Via this mechanism the dielectric molecules drain the energy from the field – but drain it to various degrees at various frequencies. Since a digital signal comprises a bunch of various frequency components, each such signal component gets attenuated differently from others. With various frequency components being attenuated to a different degree, the signal shape changes in a rather convoluted manner [25].

Channel loss due to these two factors can be expressed by the following equation [23]:

$$C(f) = e^{-[h_s(1+j)\sqrt{f}+h_d f]l]}$$

(2.1)

where h_s is skin-effect loss coefficient, h_d is dielectric loss coefficient, l is length of the media, and f is frequency.

Figure 2.6 shows the measurement of the channel loss; specifically it is the S21 parameter of a single-ended 180-inch PCB micro strip with two SMT connectors. The trace width affects the loss characteristics; a wider trace introduces smaller attenuation.
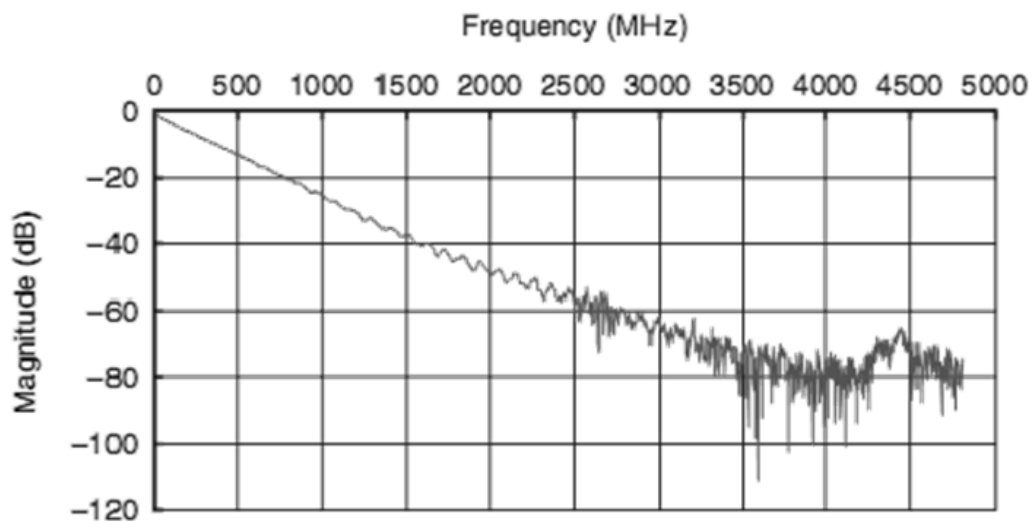
Figure 2.6. Measured S21 parameter for a PCB trace [23]

Both the equation and the plot show that the channel loss increases with frequency, specifically, the attenuation due to skin effect increases exponentially with the square root of frequency and the attenuation due to dielectric loss increases exponentially with frequency. Thus, the metallic transmission media have limited bandwidth which limits the data transmission rate. Also shown in the equation is that the channel loss increases exponentially with media length, as a result, the data transmission has limited transmission distance. Indeed, the equation shows that there is a trade-off between the transmission rate and the transmission distance. For the same media, if the transmission distance is shorter, it is possible to transmit at higher data rate. [23]

Each different media has its own unique skin effect coefficient and the dielectric loss coefficient. For example, the UTP cables widely used for building wiring have larger attenuation coefficients than those of the coaxial cables.

Figure 2.6 shows that at 1 GHz, the channel loss for this trace is about 25 dB. The signals shown in Figure 2.7 and Figure 2.8 are for data transmission through this channel at data rate 1 Gbps. When the channel loss is the only non-ideal factor being considered, the loss characteristics can be identified by the channel loss at symbol rate frequency; for 1 Gbps data rate, the corresponding symbol rate frequency is 1 GHz. When the channel attenuation is about 25 dB at symbol rate frequency, the channel loss is quite severe and it causes the received signal to have closed eyes.
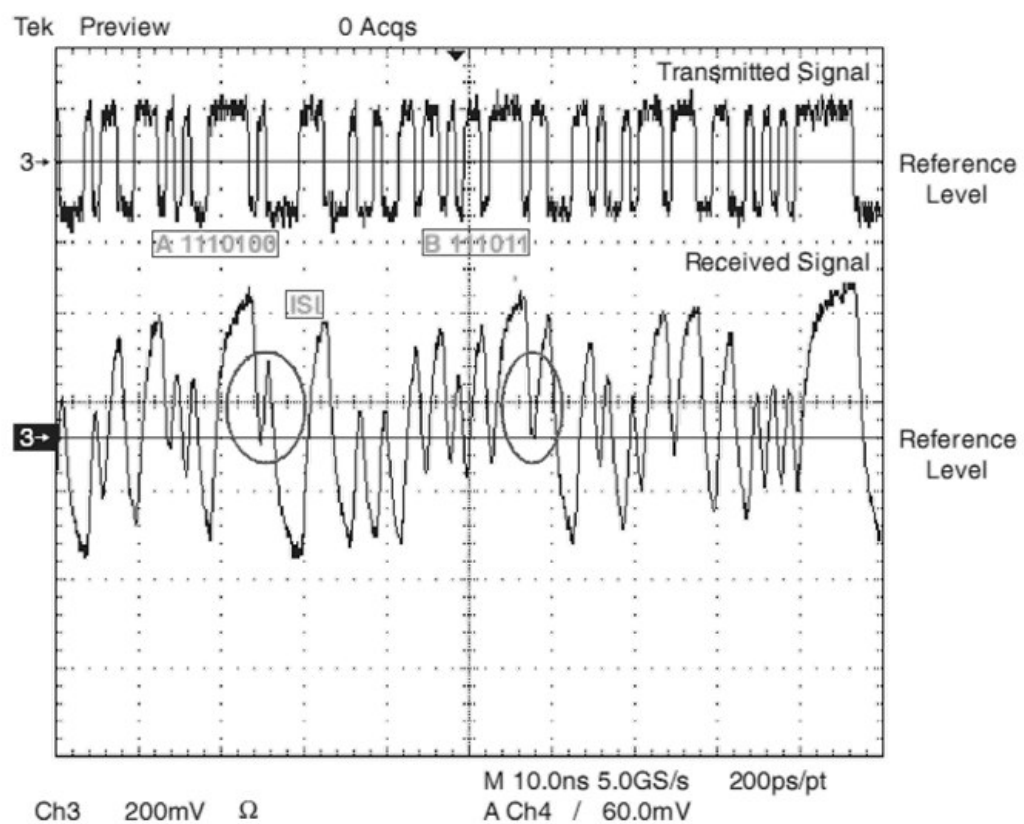
Figure 2.7. Illustration of ISI-the transmitted signal (top) and the received signal (bottom) through a band limited channel [23]
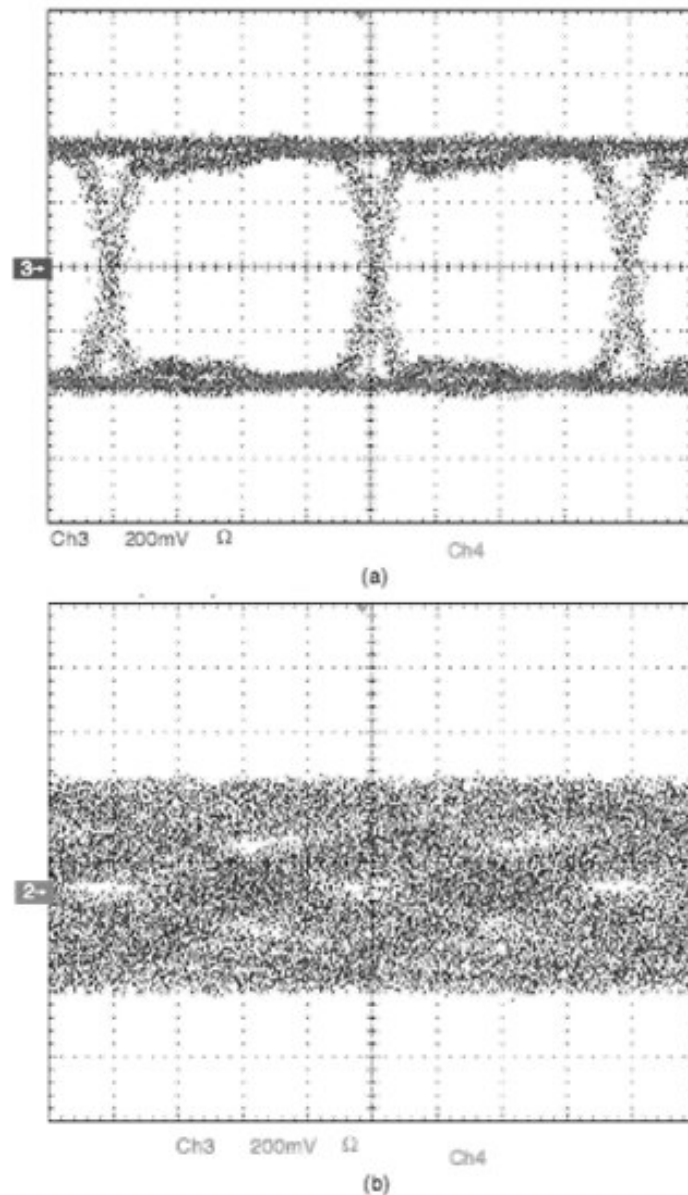
Figure 2.8. Eye diagram of (a) the transmitted signal and (b) the received signal [23]

2.1.2.2. Result of Channel Loss: Inter-Symbol Interference (ISI). The channel transfer function in the previous sections shows that different frequency suffers different degrees of attenuation and phase delay. A transmitted square wave contains many frequency components, after transmission through the channel; the frequency components suffer dispersion due to different degrees of magnitude attenuation and phase delay. Due to similar dispersion effect on light, we see the appearance of rainbow. The term, inter-symbol interference, describes the dispersion effect in discrete time domain, where the

transmitted data are treated as digital symbols with pulse modulation. Figure 2.9 shows the communication channel with transmitter pulse modulation and receiver filter. For binary data, which is also known as two-level pulse amplitude modulation (2-PAM) shown in Figure 2.7, the discrete information-bearing symbol $\{I_n\}$ is either "1" or "0" and the modulation pulse is a square pulse as shown.
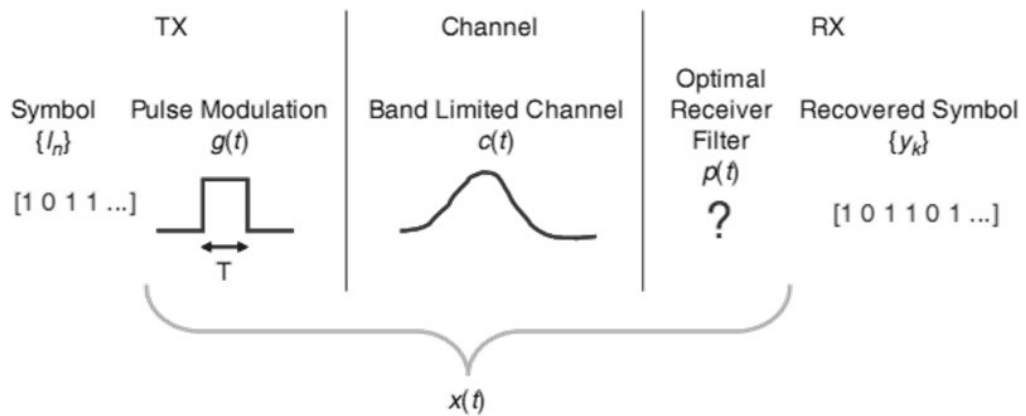
Figure 2.9. Communication channel with transmitter pulse modulation and received filter [23]

For several types of digital modulation techniques the received signal after the receiver filter, without considering channel noise, can be expressed as [23].

$$y(t) = \sum_{n=0}^{\infty} l_n x(t - nT) \qquad (2.2)$$

Where x(t) is the overall response including the transmitter modulation, channel function, and receiver filter. To obtain the recovered symbol, y(t) is sampled at times t = $kT + \tau_0$, k = 0,1, ... , where τ_0 is the transmission delay. We then have:

$$y(kT + \tau_0) \equiv y_k = \sum_{n=0}^{\infty} l_n x(kT + \tau_0 - nT) \qquad (2.3)$$

or equivalently,

$$y_k = \sum_{n=0}^{\infty} l_n x_{k-n} = x_0 I_k + \sum_{\substack{n=0 \\ n \neq k}}^{\infty} l_n x_{k-n} \tag{2.4}$$

The term $x_0 I_k$ represents the desired information symbol at the kth sampling point and the last term, which contains information from the previous and future samples, represents ISI.

If the channel has infinite bandwidth, the channel impulse response is an impulse, $\delta(t)$. Because of bandwidth limitation, the channel impulse response, c(t), is a spread pulse as shown in Figure 2.9. Convolving the modulation pulse, g(t), with c(t) results an impulse response whose pulse width is wider than T, the pulse width of g(t). The receiver filter is usually designed as match filter. The overall impulse response x(t) will have wider pulse width than T, as illustrated in the top plot in Figure 2.10. The bottom plot in the same figure shows that consecutive symbols of "1"s are transmitted. Looking at the sampling point KT, the recovered symbol is the sum of the desired symbol value labeled by point 0; it equals x_0 on curve C, plus ISI from neighboring symbols, namely point 1 from curve B ($=x_1$), point 2 from curve A ($=x_2$) and point 3 from curve D ($=x_{-1}$). In summary, ISI occurs when the overall impulse response, x(t), has wider spread than the symbol period T.
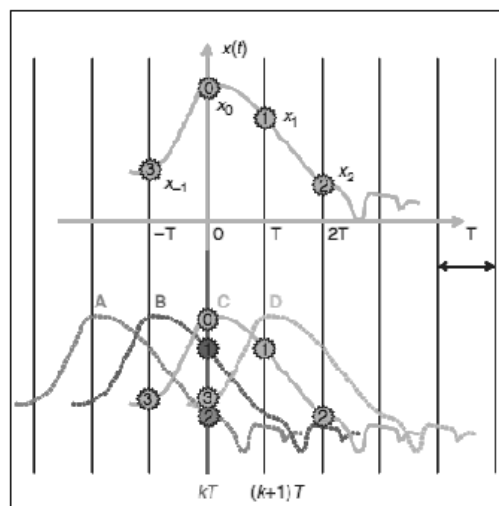
Figure 2.10. Illustration of ISI [23]

The best way to qualitatively measure effects of ISI in signal integrity is the eye diagrams as shown in Figure 2.8. The effect of ISI and other noises can be observed on an oscilloscope displaying the output of the matched filter on the vertical input with horizontal sweep rate set at multiples of 1/T. Such a display is called an eye diagram.

The effect of ISI is to cause a reduction in the eye opening by reducing the peak as well as causing ambiguity in the timing information. Clock and data recovery is impossible and equalization is mandatory to restore the timing information in this case. In summary equalization is used to remove side effects of channel loss to improve the received signal quality for correct clock and data recovery, so that the system achieves lower bit error rate for the goal of error-proof data communications.

There are two types of equalization schemes: one is at the transmitter side; the other is at the receiver side. The communication channel characteristics vary and adaptive equalization is generally required.

2.2. Channel Equalization

There are two types of equalization: transmitter pre-emphasis and receiver equalization. Both seek to either emphasize the high-frequency components or to deemphasize the low frequency components of the transmitted or received signal, in order to compensate the effect that the high-frequency components are attenuated more than the low-frequency components through the channel. Using both the transmitter and receiver equalization allows the best system performance in terms of BER (bit error rate).

The transfer function of both types of equalizer is a high-pass filter; though in practice, it is a band-pass filter. One reason is that the bandwidth limitation of semiconductor devices cannot achieve infinite bandwidth; the other is to avoid noise amplification. Though the spectrum of transmitted signal is infinite, the main slope within the symbol rate frequency contains most of the information, as shown in Figure 2.10. With additive white Gaussian noise (AWGN) and crosstalk noise, there is significant amount of

noise beyond the symbol rate frequency bandwidth. If the equalizer still has significant gain after this bandwidth, the high-frequency noise will be amplified and it deteriorates the signal quality [23].

In its broad sense, the term "equalizer" applies to any signal processing device designed to deal with ISI. Below is the classification of some of these equalization techniques:
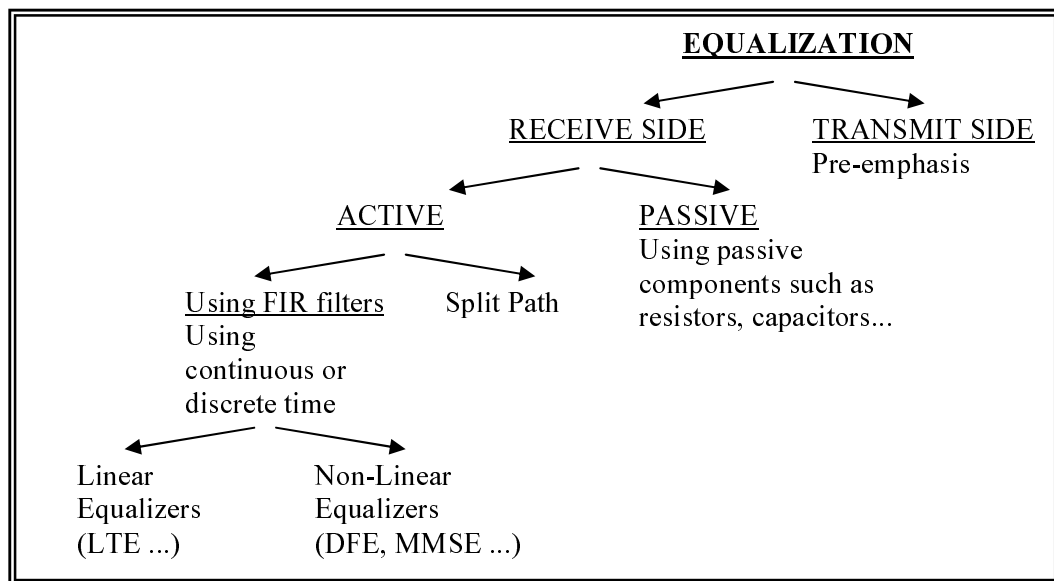
Figure 2.11. Classification of equalization techniques

Equalization eliminates the problem of frequency-dependent attenuation by filtering the transmitted or received waveform so the concatenation of the equalizing filter and the transmission line gives a flat frequency response [26].

2.2.1. Transmitter Equalizer (Pre-Emphasis)

Pre-emphasis is realized at the transmitter side. In some cases it increases the high-frequency components, which can cause EMI and more severe crosstalk problems. In other cases it reduces the power of low frequency components, known as de-emphasis. FIR

filters are generally used for transmitter pre-emphasis. A simplified approach is to use two differential amplifiers, with the first one controlled by the original code, and the second by emphasis code (produced by inverting the original code and delay one symbol period). In some cases, the FIR filter was approximated by a transition filter implemented with a look-up table.

Below figure represents an example for pre-emphasis implemented by 4-tap FIR filter. The input signal comes from either data serializer if available, or tap delay line realized with simple digital delay unit. The first tap, with the input data $b[n]$, is the main tap and the other three taps, whose inputs are one symbol to three symbols delay separately, compensate for the post cursor ISI. MDACs are used to multiply input data with the tap coefficients to build up the FIR filter required. The current outputs of MDACs are converted to voltage through off-chip or on-chip 50Ohm resistors.
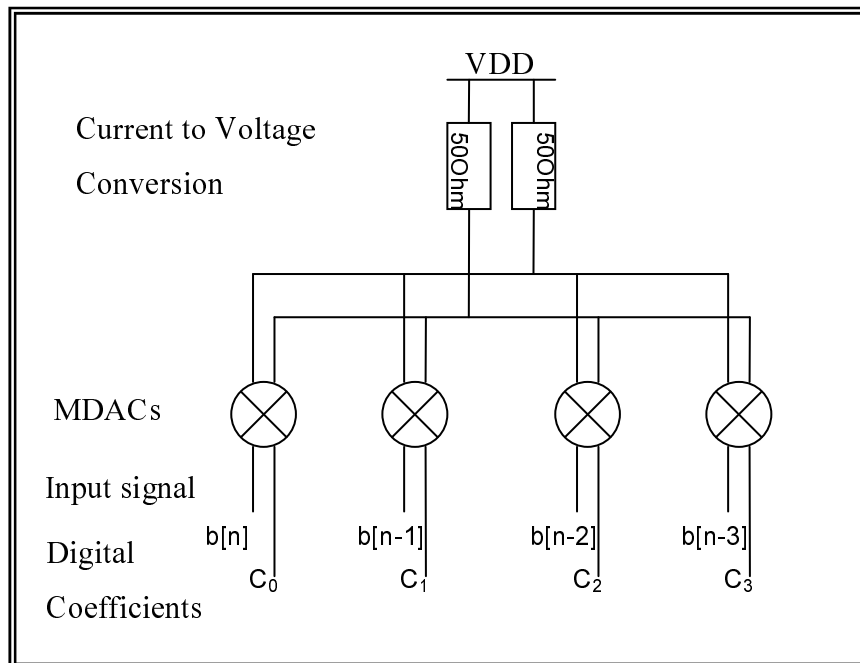
Figure 2.12. Pre-emphasis with FIR filter

Transmitter equalizer can also be a de-emphasis filter which reduces the power of low frequency component in advance. The simplest way to implement is increasing the signal amplitude at each transition edge and reducing the signal amplitude when there is no transition. The de-emphasis equalizer uses the inverted signal of previous bit as emphasis

signal. During '0' to '1' transition edge, signal amplitude is increased; in '1' to '0' transition edge, signal amplitude is further increased to negative direction. In other periods when there is no transition, the emphasis signal is opposite to the current bit and signal amplitude is reduced. To better control the strength of pulse, the auxiliary 3-bit DACs can be used to set the emphasis level [27]. Figure 2.13 shows the block diagram of de-emphasis equalizer. This de-emphasis is actually a 2-tap FIR filter with a high-pass frequency response.
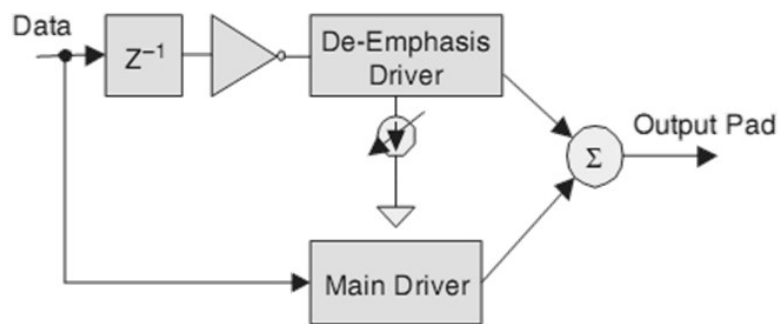
Figure 2.13. Block diagram of de-emphasis equalizer [23]
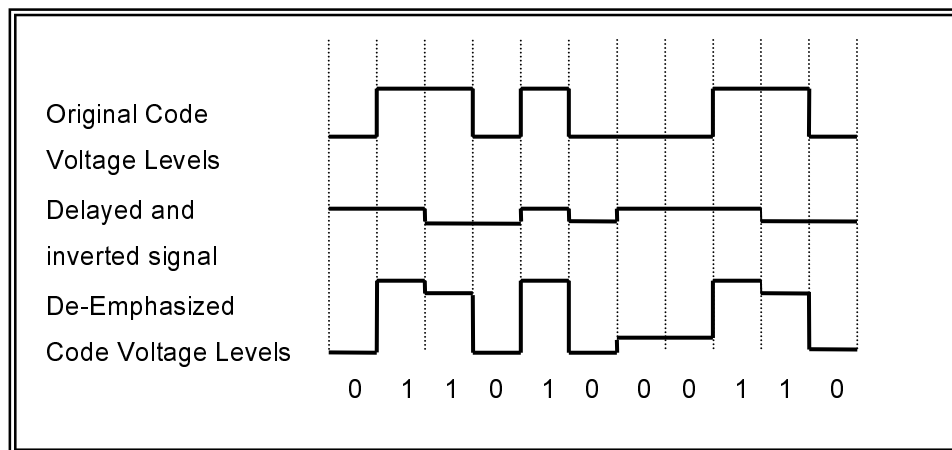
Figure 2.14. Illustration of de-emphasis equalizer functionality

2.2.2. Receiver Equalizer

Receiver equalization is a function applied at the receiver that counteracts the data degradation in the long transmission line. The equalizer could be either digital equalizer which is equivalent to applying the pre-emphasis techniques to the receiving end of the channel, or it could be the analog equalizer which employs RC filter to compensate the channel loss.

To be more specific there are generally four categories of receiver equalizers for over Gbps data transmissions: passive-component equalizer, active continuous-time equalizer using split-path amplifier, active equalizer using discrete-time FIR filter and active equalizer using continuous-time FIR filter.

2.2.2.1. Passive Component Equalizers. Passive equalizers consist of only passive components such as capacitor, resistor and inductor. They require no power to operate. Since there is no involvement of active components, there is no noise source in the passive equalizer design. High dynamic range due to absence of power supplies to limit voltage swing is also another advantage of these equalizers. Since the passive components rarely break they are known with their extremely good reliability also. Despite their simplicity and the advantages above, they are disliked for their cost and size due to inductor involvement. Passive equalization is preferred in the case when the received signal has large amplitude and the receiver sensitivity is high. Passive compensation has been relegated to applications that correct at one specific length and bit rate and tend not to be applied to a wide range of conditions.

Below figure represents an implementation of a passive equalizer using bridged T similar network [28]. Passive components in the equalizer define frequency characteristics in different band dependently, which eases the design procedure [23]. For example, R3, R4, R5 and L2 set the characteristics impedance; C2 and R2 set the low frequency compensation; mid-band frequency compensation is set by L2; and L1 and C1 set the high frequency compensation.
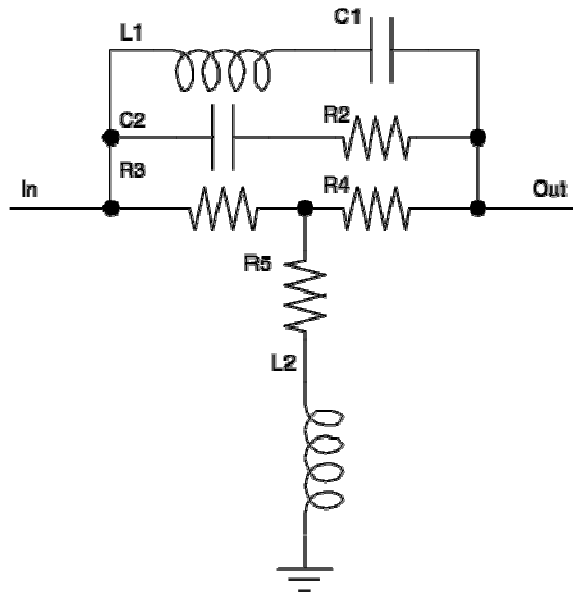
Figure 2.15. Passive T-bridged equalizer [23]

2.2.2.2. Active Continuous Time Equalizer. One way to achieve continuous time equalization with active devices is using split path amplifier: Split-path amplifier divides the signal path into two paths [23]. One path comprises a high pass filter or peak response filter to amplify the high frequency component. Another path is an all pass filter or a low pass filter to match the time delay of first path. Weighted sum of two paths is equivalent to a variant gain high pass filter, whose gain factor can be varied by controlling the weight of those two paths. Figure 2.16 shows a 3.2 Gbps adaptive cable equalizer using a peak response filter which is also a feed-forward amplifier as it is equivalent to add a zero or a feed-ward path [29]. An equalizer control circuit compares the power ratio at two specific frequency points using band-pass filters to set the weighted factor of those two paths.
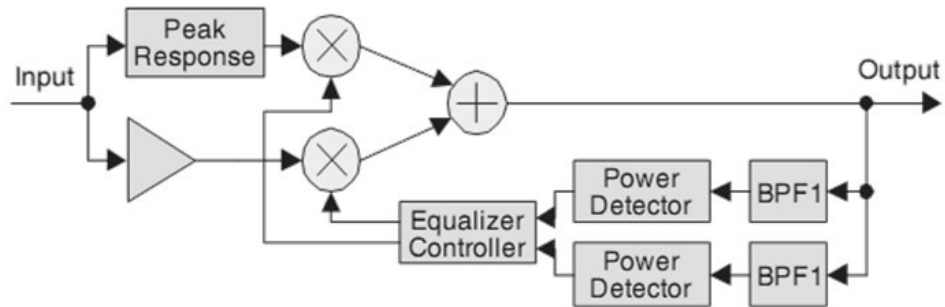
Figure 2.16. Split-path equalizer example [23]

To better match phase delay in split path amplifier, the flat response path must use the same amplifier as in the feed-forward path. Wider gain control range is achieved by jointly adjusting the poles position in both paths. Traditional OPAMP-based amplifier using feedback resistor provides precise gain and low non-linearity [23]. However, negative feedback loop prevents the amplifier working in GHz range.

Phase mismatch between feedback loop and input signal also limit using a feedback loop amplifier in high frequency range. Figure 2.17 shows a wide band split-path amplifier without feedback loop; the amplifier gain is controlled through the load resistor instead of using feedback resistor.
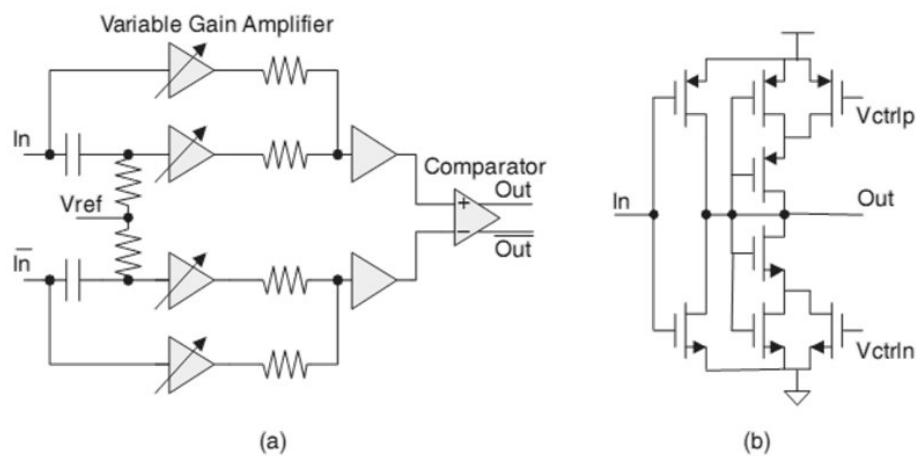
Figure 2.17. Wideband split-path amplifier (a) without feedback loop and (b) variable gain amplifier [23]

It is known that the transconductance of a source degeneration transconductor is close to the conductance, 1/R of the degeneration resistor. If the degeneration comprises a resistor and a capacitor, the resistor corresponds to an all path loop while the capacitor corresponds to a high pass path. Therefore, such a transconductor cell serves as a compact split-path amplifier. To tune the high frequency gain and low frequency gain, the capacitor and resistor are implemented with varactor and a linear MOS transistor. Figure 2.18 shows the schematic of the source degeneration transconductor. Varying the controlling voltage of varactor and MOS transistor, Vctrl and gmctrl, will change the high frequency boosting and low frequency gain.
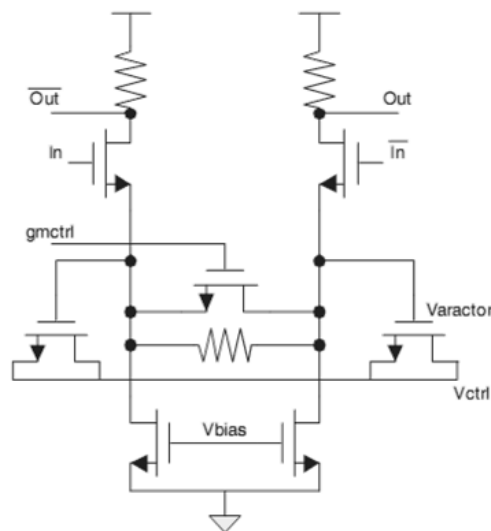
Figure 2.18. Source degeneration transconductor filter [23]

More common way to achieve continuous time equalization with active devices is using FIR filters. FIR filters can be either discrete time or continuous time. Traditional discrete-time transversal FIR filters have been widely used in hard disk read channel equalization and in broadband modems equalizer [23].

Depending on the circuit realization of tap delay line and multiplier, the discrete-time FIR filters can be grouped into following four categories:

- Fully digital realization [9] [10]
- Digital tap delay line + multiplying digital to analog converter (MDAC) [32]
- Serial sampling analog tap delay line + analog multiplier
- Parallel sampling analog tap delay line + analog multiplier

Structures of the first two types require high speed ADC to convert received analog signal into digital bits, which is hard to realize with CMOS technology at Gbps data rate. Tap delay line of the third type has been realized with unity gain sample-and-hold (S&H) cell. Analog input signal passes through the delay line directly and there is no need for high-speed ADC. The disadvantage of this structure is that each S&H cell introduces distortion and attenuation to the delayed signal. All distortion and attenuation due to nonlinearity, clock feed through and limited bandwidth of S&H cell will accumulate along the line [33]. Another main drawback of serial sampling delay is that each delay unit must settle down in one symbol period which requires high frequency clock and wide bandwidth S&H. To avoid error accumulation, parallel sampling units of the fourth type sample input signal in sequence and switch them to the corresponding multipliers through rotating switch matrix [33].

Below is one example of realization of switched capacitor FIR filter for channel equalization for the fourth type.
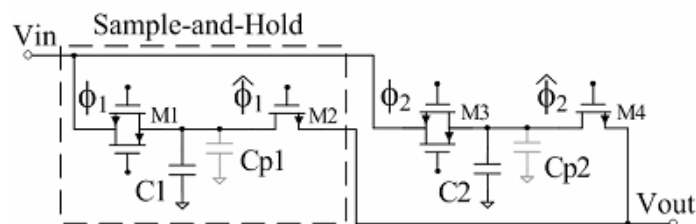
Figure 2.19. Example channel filter with top plate S/H cells [33]

Figure 2.19 shows an example, passive filter channel with two S/H cells that use bottom-plate sampling. At first, ignore the parasitic capacitors and assume the transistors act as ideal switches. The odd numbered transistors are switched so that Vin is sampled onto C_1 and C_2 during consecutive sampling periods. Then the even numbered switches are

closed, connecting C_1 and C_2 in parallel between ground and the output node. Charge sharing gives an output that is a weighted sum of two consecutive input samples, as desired. The following equation describes the ideal transfer function, including a signal inversion [34].

$$\frac{Vout}{Vin}(z) = -K\left[\frac{C_1 z^{-2} + C_2 z^{-1}}{(C_1 + C_2)}\right] \tag{2.5}$$

The capacitors in the equalizer are chosen to give this transfer function. The scale factor K depends on the output parasitic capacitances.

Instead of using sample and hold cells for one period delay in FIR filter, delay of one period can also be implemented using continuous time transport delay cells. This method is known as continuous time FIR equalization. This way clock and data synchronization complexity can be exterminated. One method to obtain transport delay is to use source followers [23]. However, due to bandwidth limitation of CMOS circuit, the frequency range of filter will be limited. Artificial transmission line implementation can be used to overcome this bandwidth limitation in expense of area due to long line requirements.

2.3. Receiver Equalization Using FIR Filters

From now on receiver equalization with FIR filters will be focused on. These are discrete time equalizers that are designed to overcome data distortions due to ISI and/or noise at the receiver end.

2.3.1. Communication System Summary

Before defining the channel and equalizer characteristics in discrete time domain, it is useful to revisit the basic communication system again as follows [35]:
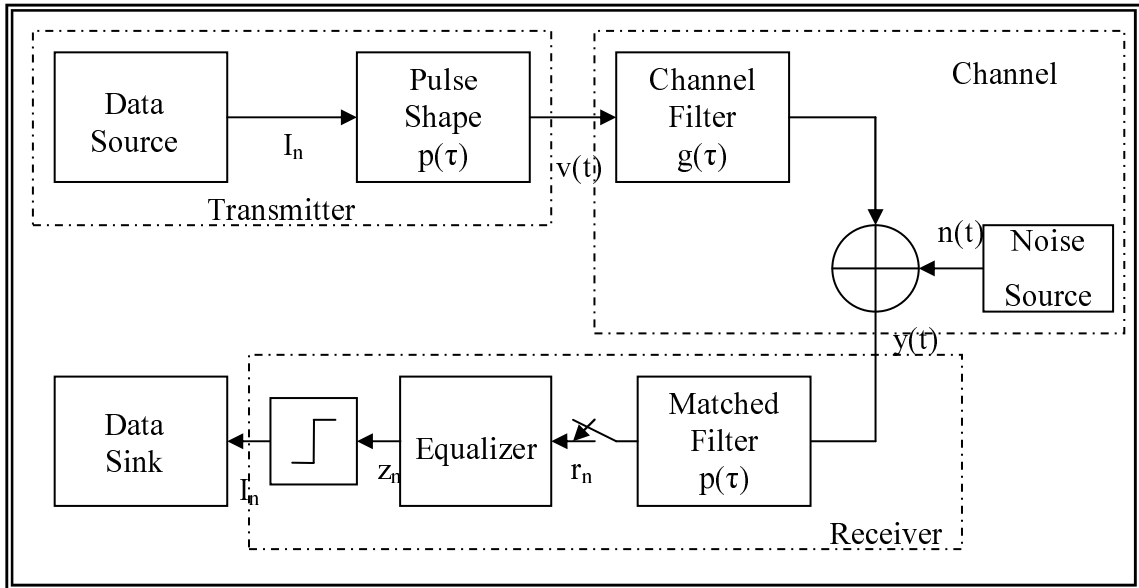
Figure 2.20. Basic communication system revisited

I_n is the input data to the pulse shape circuitry (modulator) and the output of pulse shape cell is the convolution of input signal with the pulse shape transfer function and can be represented as follows:

$$v(t) = \sum_n I_n p(t - nT) \tag{2.6}$$

The signal then passes through the channel with transfer function g(τ) and reaches the receiver end as signal y(t) after addition of noise n(t). Transmit filter and the channel can be combined to one filter with transfer function h(τ) due to their linearity. The frequency response of this combined filter is as follows:

$$H(f) = P(f)G(f) \tag{2.7}$$

The receiver input after combined filter and addition of noise will be represented as:

$$y(t) = \sum_n I_n h(t - nT) + n(t) \tag{2.8}$$

Receiver then takes this signal and passes it through the matched-filter to form the signal r(t):

$$r(t) = \int y(t)p(t - \tau)d\tau \qquad (2.9)$$

This signal r(t) is sampled to form the discrete time signal r_n=r(nT). If the above two equations are combined to one equation, the relation between the input symbols and received signals becomes more obvious:

$$r_n = \sum_k I_k q(nT - kT) + \eta_n \qquad (2.10)$$

where q is convolved transmit and channel responses with the receive filter;

$$q(\tau) = h(\tau) * p(\tau) \qquad (2.11)$$

and noise component of the received symbol is found by passing the noise through the receive filter:

$$\eta_n = \int n(t)p(nT - t)dt \qquad (2.12)$$

2.3.2. Equalization Using FIR Filters

The equalization techniques that can be implemented with the use of FIR filters can be summarized in the following table:
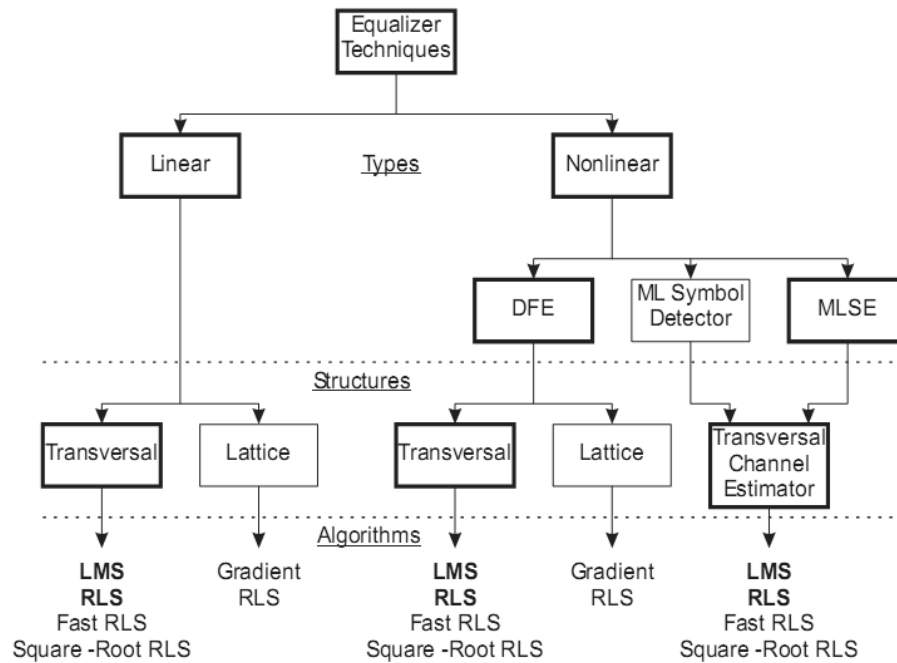
Figure 2.21. Equalizer types, structures, and algorithms [36]

If the channel is time invariant, there is no need for adaptive equalization technique involvement. However, in some applications the channel characteristic may vary with the time, and the equalizer should adapt its coefficients for data recovery according to the varying channel response. Some examples to these adaptive algorithms are LMS and RLS.

Non linear equalizers are beyond the scope of this thesis and also ZF, LMS, RLS and CMA algorithms will be shortly described in the following chapter.

2.3.2.1. Linear Equalizers. If an equalizer uses only received signal samples in its calculations and do not need any previously detected symbols, it is referred as linear equalizer. There are two commonly used types of linear equalizers: the linear transversal equalizer and the fractionally spaced equalizer.

Linear transversal equalizer is one of the most common equalizer in the literature due to its implementation simplicity. As represented in Figure 2.21, it consists of tapped-delay line with tap spacing equal to the symbol rate. Its input is the sampled output of the matched filter at the receiver side. As mentioned in the previous chapter, these delay cells can be either continuous or discrete. If the tap delays are discrete, the clock should be synchronized to these sample-hold cells first. Each delay cell can be considered as a register and shift register behavior results after cascading these delay cells. The contents of these registers are shifted at each symbol period, and they are added after being multiplied by specific coefficients. This sum of multiplications makes up the estimate of the current symbol. This operation can be simplified by the following equation:

$$z_n = \sum_{k=N_1-1}^{N_2-1} c_k r_{n-k} \qquad (2.13)$$

where z_n is the estimated output of the filter, r_n is the input sequence to the equalizer, c_k is the coefficient of kth tap, N_1 is the number of non-causal equalizer taps (future taps) and N_2 is the number of causal equalizer taps (previous taps) and z^{-1} is tap delay which is equal to symbol period.
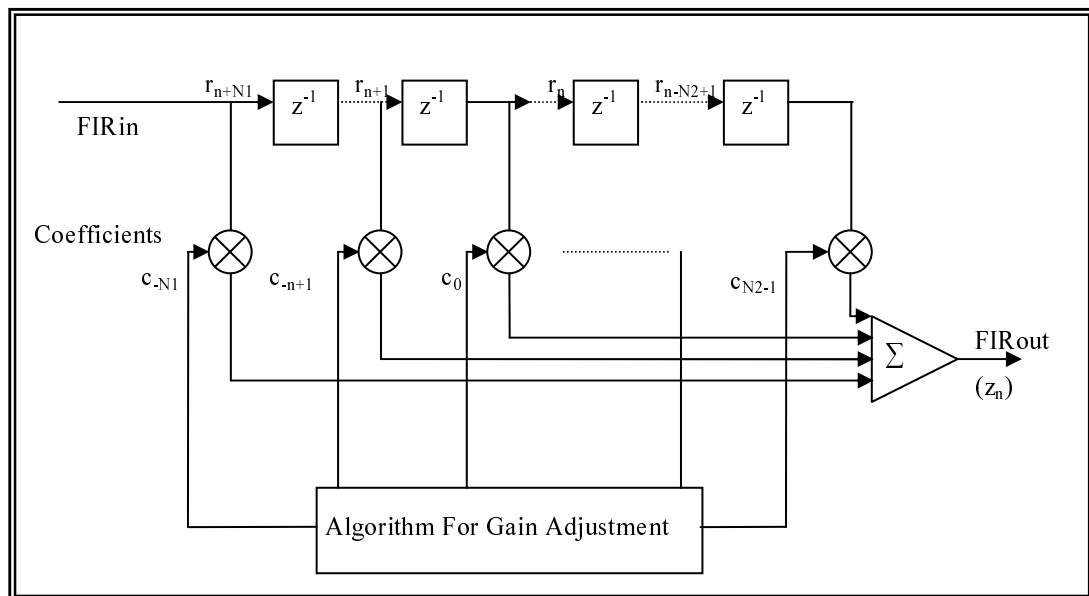
Figure 2.22. Symbol spaced equalization using FIR filter

The causal part of the linear transversal equalizer is used to remove ISI due to the symbol interference with the previous symbols, whereas the non-causal part represents the interference with the future symbol tails. Non-causal taps are only needed when the current received signal is not the strongest element, otherwise N_1 can be assumed to be set to 0.

In the receiver, decisions are taken at the symbol rate f_b to retrieve the data. So far, it has been assumed that all the functions in the receiver are carried out at that rate, including the equalization. However, it is known that the signal spectrum exceeds the symbol rate by the amount of the roll-off the Nyquist filter.

Therefore sampling at the symbol rate generates aliasing and the image of the base band spectrum occurs around the frequency f_b. According to the sampling theory, the phase of this image is linked to the sampling times. A shift in timing produces a rotation of the phase, and the base-band spectrum and the image no longer add up in phase, in the filter transition band Δf. Therefore the equalizer is sensitive to the sampling times, and equalization may become impossible for frequencies in the vicinity of half the symbol rate. As is well known in multi-rate filtering, the solution to the problem is to increase the sampling rate sufficiently to avoid aliasing, which leads to the so-called fractionally spaced equalizer, which is yet another type of linear equalizers [37].

A fractionally spaced equalizer is a linear equalizer that is similar to a symbol-spaced linear equalizer. By contrast, however, a fractionally spaced equalizer receives K input samples before it produces one output sample and updates the weights, where K is an integer. In many applications, K is 2. The output sample rate is expressed as 1/T, while the input sample rate is K/T. The weight-updating occurs at the output rate, which is the slower rate. Below is a schematic of a fractionally spaced equalizer [38].
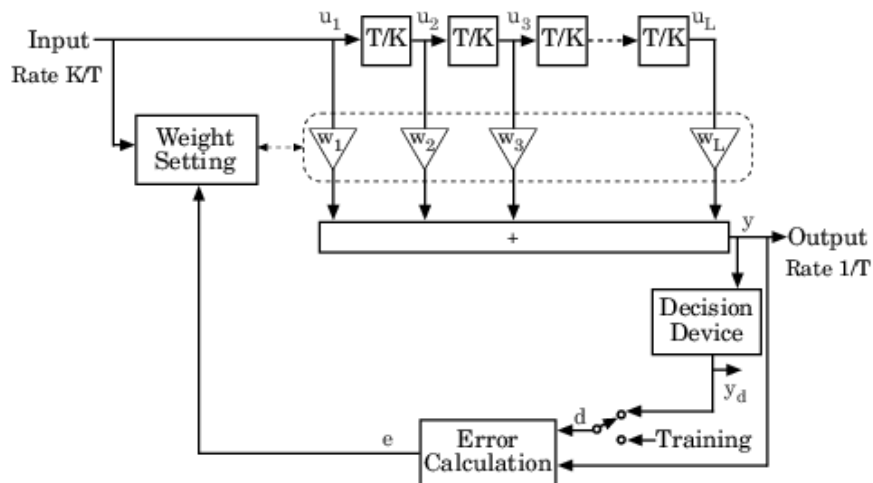
Figure 2.23. Fractionally spaced equalization using FIR filter [38]

Another expression of an equalizer with double sampling is shown in below figure. The input signal sample sequence is split into two sequences which are fed to two separate equalizers $H_1(z)$ and $H_2(z)$ operating at the symbol rate.
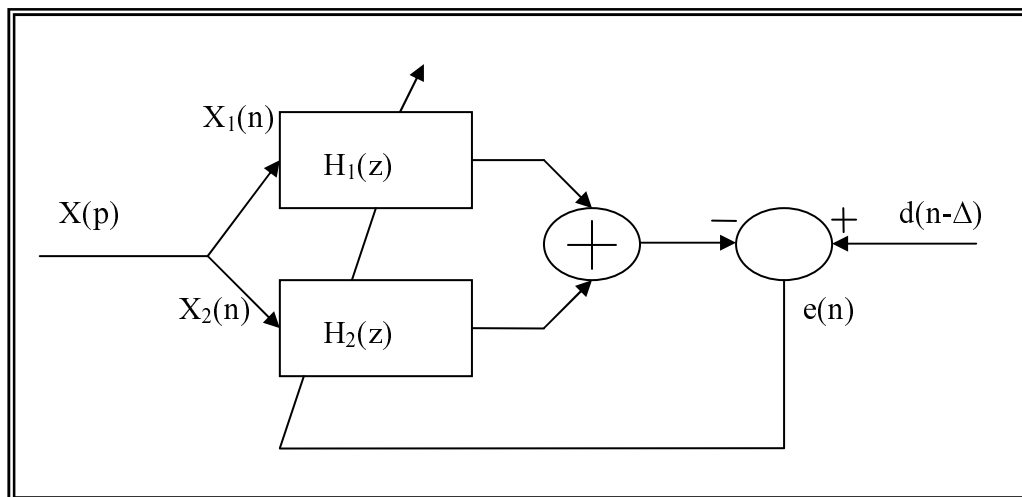
Figure 2.24. Fractionally spaced equalization using FIR filter

The output error in that case is [37]:

$$e(n) = d(n - \Delta) - \left(H_1^{'}X_1(n) + H_2^{'}X_2(n) \right) \qquad (2.14)$$

The optimal coefficient vectors are given by:

$$\begin{bmatrix} H_{1opt} \\ H_{2opt} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}^{-1} \begin{bmatrix} r_{d1} \\ r_{d2} \end{bmatrix} \qquad (2.15)$$

for i=1,2 and j=1,2:

$$R_{ij} = E[X_i X_j], r_{di} = E[d(n - \Delta)X_i(n)]$$

The updating of the coefficients is carried out at the symbol rate. It is worth pointing out that the input signal spectrum, except for the noise, goes to zero in the vicinity of the symbol frequency f_b.

As in any transversal equalization, the fractionally spaced equalizers also suffer from noise amplification. The best solution is to complete FSE by a feedback section, to make the so-called fractionally spaced DFE. This combination is generally recognized to be the most efficient approach to adaptive equalization [37].

2.3.2.2. Equalization Criteria. Considerable research has been performed on the criterion for optimizing the filter coefficients $\{c_k\}$. Since the most meaningful measure of performance for a digital communications system is the average probability of error, it is desirable to choose the coefficients to minimize this performance index. However, the probability of error is a highly nonlinear function of $\{c_k\}$.Consequently, the probability of error as a performance index for optimizing the tap weight coefficients of the equalizer is impractical. Two criteria have found widespread use in optimizing the equalizer coefficients [36]. One is the peak distortion criterion and the other is the mean square error criterion.

The worst case ISI that occurs at the output of an equalizer is defined as the peak distortion. Minimization of peak distortion by optimizing the filter coefficients is called the peak distortion criterion. The ideal ISI removal could be achieved if equalizer with infinite taps could be realizable. In real case, peak distortion criterion stands for ISI removal over the range of the filter taps.

Assuming infinite non-causal and causal filter taps, the overall response of channel and equalizer could be represented as follows:

$$q_n = \sum_{j=-\infty}^{\infty} c_j f_{n-j} \tag{2.16}$$

where $\{f_n\}$ is the impulse response of the discrete time linear channel model, $\{c_n\}$ is the equalizer impulse response and q_n is the cascade of these filters.

If the input is represented as I_k, the output of the receiver output sampled at k^{th} instant is represented as follows:

$$\hat{I}_k = q_0 I_k + \sum_{n \neq k} I_n q_{k-n} + \sum_{j=-\infty}^{\infty} c_j \eta_{k-j} \tag{2.17}$$

The first term is desired symbol scaled by the channel and equalizer transfer functions. The second term represents ISI and the final term is the noise. The peak value of the second term is shown as follows [36]:

$$\mathcal{D}(c) = \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} |q_n|$$

$$= \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \left| \sum_{j-\infty}^{\infty} |c_j f_{n-j}| \right| \tag{2.18}$$

In order to obtain the desired signal back at the output of equalizer D(c) should be forced to 0. The tap values that forces D(c) to 0 can be obtained from the solution of the following equation:

$$q_n = \sum_{j=-\infty}^{\infty} c_j f_{n-j} = \begin{cases} 1 & (n = 0) \\ 0 & (n \neq 0) \end{cases} \tag{2.19}$$

In z-domain, this condition can be summarized as

$$Q(z) = C(z)F(z) = 1 \qquad (2.20)$$

or simply:

$$C(z) = \frac{1}{F(z)} \qquad (2.21)$$

From above equations, it is understood that minimization of ISI is the same as having the inverse of channel transfer function as the equalizer transfer function. In other words, complete elimination of ISI requires the use of an inverse filter of F(z).

Assuming L non-causal and L+1 causal filter taps, the overall response of channel and equalizer could be represented as follows:

$$b_k = \sum_{l=-L}^{L} c_l q_{k-l} = \begin{cases} 1 & k = 0 \\ 0 & k = \mp1, \mp2, \dots, \mp L \end{cases} \qquad (2.22)$$

where $\{q_n\}$ is the impulse response of the discrete time linear channel model, $\{c_n\}$ is the equalizer impulse response and b_k is the cascade of these filters.

This convolution can be visualized using a matrix representation as follows [35]:

$$
\underbrace{\begin{bmatrix}
q_0 & \cdots & q_{-L+1} & q_{-L} & q_{-L-1} & \cdots & q_{-2L} \\
\vdots & & \vdots & \vdots & \vdots & & \vdots \\
q_{L-1} & \cdots & q_0 & q_{-1} & q_{-2} & \cdots & q_{-L-1} \\
q_L & \cdots & q_1 & q_0 & q_{-1} & \cdots & q_{-L} \\
q_{L+1} & \cdots & q_2 & q_1 & q_0 & \cdots & q_{-L+1} \\
\vdots & & \vdots & \vdots & \vdots & & \vdots \\
q_{2L} & \cdots & q_{L+1} & q_L & q_{L-1} & \cdots & q_0
\end{bmatrix}}_{Q}
\begin{bmatrix}
c_{-L} \\ \vdots \\ c_{-1} \\ c_0 \\ c_1 \\ \vdots \\ c_L
\end{bmatrix}_{C}
=
\begin{bmatrix}
0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0
\end{bmatrix}_{\delta}
\qquad (2.23)
$$

$C_{pd}=Q^{-1}\delta$, where C_{pd} stands for the optimum coefficients to minimize peak distortion.

With the assumption of finite number of equalizer taps, ISI outside the range of the equalizer taps cannot be covered. However, if the equalizer is long enough, this excess ISI can be ignored and should not cause obvious performance degradation.

Since peak distortion method does not consider the noise at the receive path, it is only useful to use this method for systems where ISI is the dominant factor. If ISI is dominant over noise at the receive path, any adaptive algorithm will converge to zero forcing equalizer.

Mean square error criterion is yet another most common equalization criterion. If the equalizer tap coefficients are optimized to minimize the difference between the desired signal and output of the equalizer, this equalizer works under mean square error criterion.

Error between the information signal transmitted in k^{th} sampling interval, I_k, and estimate of this symbol at the output of equalizer, $\hat{I}_k$, is represented as:

$$
\varepsilon_k = I_k - \hat{I}_k \qquad (2.24)
$$

When the information symbols $\{ I_k \}$ are complex-valued, the performance index for MSE criterion, denoted by J, is defined as [36]:

$$
J = E|\varepsilon_k| = \left| I_k - \hat{I}_k \right|^2 \qquad (2.25)
$$

where E is the expectation operator.

When the information symbols are real-valued, MSE is simply the square of the real part of ε_k.

First let's assume we have infinite number of taps in our equalizer to minimize J. In this case the estimate $\hat{I}_k$ can be expressed as

$$\hat{I}_k = \sum_{j=-\infty}^{\infty} c_j v_{k-j}$$

where: (2.26)

$$v_k = \sum_{n=0}^{L} f_n I_{k-n} + \eta_k$$

If this is substituted into the expression of J (equation 2.25) and expansion of the result yields a quadratic function of the coefficients { c_k }. If orthogonality principle is invoked in mean square estimation, coefficients can be selected to render the error ε_k orthogonal to the signal sequence v_k for $-\infty < l < \infty$:

$$E(\varepsilon_k v_{k-l}^*) = 0, \quad -\infty < l < \infty$$ (2.27)

Substitution for ε_k yields:

$$E\left[\left(I_k - \sum_{j=-\infty}^{\infty} c_j v_{k-j}\right) v_{k-l}^*\right] = 0$$ (2.28)

or equivalently,

$$\sum_{j=-\infty}^{\infty} c_j E\left(v_{k-j} v_{k-l}^*\right) = E(I_k v_{k-l}^*), \quad -\infty < l < \infty$$ (2.29)

Simplification can be done as follows:

$$E\left(v_{k-j} v_{k-l}^*\right) = \sum_{n=0}^{L} f_n^* f_{n+l-j} + N_0 \delta_{lj} \tag{2.30}$$

$$= \begin{cases} x_{l-j} + N_0 \delta_{lj} & (|l - j| \leq L) \\ 0 & (otherwise) \end{cases}$$

and

$$E(I_k v_{k-l}^*) = \begin{cases} f_{-l}^* & (-L \leq l \leq 0) \\ 0 & (otherwise) \end{cases} \tag{2.31}$$

Now if we substitute the last two equations to 3.29 and take z transform of both sides of the resulting equation we obtain:

$$C(z)\left[F(z)F^*(z^{-1}) + N_0\right] = F^*(z^{-1}) \tag{2.32}$$

Therefore, the transfer function of the equalizer based on the MSE criterion is

$$C(z) = \frac{F^*(z^{-1})}{[F(z)F^*(z^{-1}) + N_0]} \tag{2.33}$$

When the noise whitening filter is incorporated into C(z), we obtain an equivalent equalizer having the TF:

$$C'(z) = \frac{1}{[F(z)F^*(z^{-1}) + N_0]}$$

$$\tag{2.34}$$

$$= \frac{1}{[X(z) + N_0]}$$

It is obvious that if N_0 is very small compared to the signal and can be ignored, the coefficients that minimize the peak distortion are approximately equal to the coefficients that minimize the MSE performance index. This means that when $N_0=0$, minimization of MSE results in complete elimination of the ISI, otherwise it eliminates both ISI and additive noise [36].

As it is not possible to realize an equalizer with infinite number of taps, equalizer with finite number of taps working under MSE criterion will be focused on.

The output of the equalizer in the kth signaling interval is:

$$\hat{I}_k = \sum_{j=-K}^{K} c_j v_{k-j} \qquad (2.35)$$

The MSE for the equalizer having 2K+1 taps, denoted by J(K) is:

$$J(k) = E\left|I_k - \hat{I}_k\right|^2 = E\left|I_k - \sum_{j=-K}^{K} c_j v_{k-j}\right|^2 \qquad (2.36)$$

Minimization of J(K) with respect to the tap weights {c_j} or, equivalently forcing the error $\varepsilon_k = I_k - \hat{I}_k$ to be orthogonal to the signal samples $v^*_{j\text{-}l}$, | l | $\leq$ K, yields the following set of simultaneous equations:

$$\sum_{j=-K}^{K} c_j \Gamma_{lj} = \xi_l, \qquad l = -K, \dots, -1, 0, 1, \dots, K \qquad (2.37)$$

where

$$\Gamma_{lj} = \begin{cases} x_{l-j} + N_0 \delta_{lj} & (|l - j| \leq L) \\ 0 & (otherwise) \end{cases} \qquad (2.38)$$

and

$$\xi_l = \begin{cases} f^*_{-l} & (-L \leq l \leq 0) \\ 0 & (otherwise) \end{cases} \qquad (2.39)$$

It is convenient to express the set of linear equations in matrix form. Thus,

$$\Gamma C = \xi \qquad (2.40)$$

where C denotes the column vector of 2K+1 tap weight coefficients, Γ denotes (2K+1)x(2K+1) Hermitian covariance matrix with elements Γ_{ij}, and ξ is a (2K+1)-dimensional column vector with elements ξ_l. The solution of last equation is:

$$C_{opt} = \Gamma^{-1}\xi \qquad (2.41)$$

3. ADAPTIVE EQUALIZATION

Until now, it was assumed that the channel characteristic was known by the receiver and was time-invariant. However, in most communication systems that employ equalizers, the channel characteristics are unknown a priori and, in many cases, the channel response is time-variant. In such a case, the equalizers are designed to be adjustable to the channel response and, for time-variant channels, to be adaptive to the time variations in the channel response.

In the case of linear equalizer, there were two wildly used criteria for determining the values of optimum equalizer coefficients. One criterion was based on minimization of peak distortion at the output of the equalizer. The other was based on minimization of the mean-square error at the output of the equalizer. Four algorithms for performing the optimization process automatically and adaptively will be presented: zero forcing, least mean squares, recursive least squares and constant modulus algorithms.

3.1. Zero Forcing Algorithm

In the peak distortion criterion, the peak distortion $D(c)$, given by equation 2.18 is minimized by selecting the equalizer coefficients $\{c_k\}$. When $D_0 > 1$, the distortion $D(c)$ at the output of the equalizer is minimized by forcing the equalizer response $q_n = 0$, for $1 \leq |n| \leq K$ and $q_n = 1$. Only in this condition, there is a simple computational algorithm, called the zero-forcing algorithm that achieves these conditions [36].

The zero-forcing solution is achieved by forcing the cross-correlation between the error sequence $\varepsilon_k = I_k - \hat{I}_k$ and the desired information sequence $\{I_k\}$ to be zero for the shifts in range $0 \leq |n| \leq K$.

In other words, the current error between the estimate and the desired symbol should be independent of the previous or future symbols. This can be demonstrated as follows:

$$E\left(\varepsilon_k I_{k-j}^*\right) = E\left[\left(I_k - \hat{I}_k\right)I_{k-j}^*\right]$$

$$= E\left(I_k I_{k-j}^*\right) - E\left(\hat{I}_k I_{k-j}^*\right), \quad j = -K, \dots \dots, K \tag{3.1}$$

Assuming that information symbols are uncorrelated, i.e., $E(I_k \hat{I}_j^*)=\delta_{kj}$, and that the information sequence $\{\hat{I}_k\}$ is uncorrelated with the additive noise sequence $\{\eta_k\}$. For $\hat{I}_k$, following expression can be used:

$$\hat{I}_k = q_0 I_k + \sum_{n \neq k} I_n q_{k-n} + \sum_{n \neq k} c_j \eta_{k-j} \tag{3.2}$$

After taking the expected values in equation 3.1 as follows:

$$E\left(\varepsilon_k I_{k-j}^*\right) = \delta(j) - q_j, \quad j = -K, \dots \dots, K \tag{3.3}$$

Therefore, the conditions

$$E\left(\varepsilon_k I_{k-j}^*\right) = 0, \quad j = -K, \dots \dots, K \tag{3.4}$$

are fulfilled only when $q_0=1$ and $q_n=0$, for $1 \leq |n| \leq K$.

Simple recursive algorithm for adjusting the equalizer coefficients is:

$$c_j^{(k+1)} = c_j^{(k)} + \Delta \varepsilon_k I_{k-j}^*, \quad j = -K, \dots, -1, 0, 1, \dots, K \tag{3.5}$$

where $c_j^{(k)}$ is the jth coefficient value at k^{th} sampling time, $\varepsilon_k = I_k - \hat{I}_k$ is the error at that time and delta is a scale factor that controls the time step of adjustment. The term $\varepsilon_k I_{k-j}^*$ is an estimate of the cross correlation (ensemble average). The averaging operation of the cross-correlation is accomplished by means of the recursive first order difference equation, which represents a simple discrete-time integrator [36].

3.2. LMS Algorithm

In LMS algorithm, equalizer coefficients are iteratively updated to minimize the mean square error cost function. This algorithm attempts to find the desired tap coefficient vector which will produce the minimum mean square error, where each coefficient vector has its own MS error.

It is best to visualize the dependence of the mean-squared error on the elements of the tap-weight vector as a bowl shaped surface with a unique minimum. This surface is referred as the error-performance surface of the adaptive filter [39]. The adaptive process has the task of continually seeking the bottom or minimum point of this surface. At the minimum point of the error performance surface, the tap weight vector takes the optimum value, which is defined by Wiener-Hopf equations, reproduced here for convenience:

$$C_{opt} = \Gamma^{-1}\xi \qquad (3.6)$$

where C_{opt} denotes the optimum column vector of 2K+1 tap weight coefficients, Γ denotes (2K+1)x(2K+1) covariance matrix of signal samples $\{v_k\}$, and ξ is a (2K+1) dimensional column vector of channel filter coefficients. The optimum equalizer coefficients vector C_{opt} can be determined by inverting covariance vector Γ. This is also known as Wiener-Hopf equation. Instead of direct matrix inversion, with high implementation complexity, it is also possible to determine this optimum vector using iterative procedure. One of the simplest iterative procedures is known to be steepest descent algorithm.

3.2.1. Steepest Descent Algorithm

3.2.1.1. Qualitative Definition. To find the minimum value of the mean-square error, J_{min}, by the steepest descent algorithm, the procedure is summarized as follows [39]:

(i)   An initial value for tap weight vector, w(0), is determined to provide initial guess as to where the minimum point of error-performance surface may be located. Unless some prior knowledge is available, it is commonly set to null vector.

(ii)   Using this initial or present guess, gradient vector is computed, the real and imaginary parts of which are defined as the derivative of the mean squared error J(n), evaluated with respect to the real and imaginary parts of the tap-weight vector w(n) at time n (i.e. the nth iteration).

(iii) Next guess at tap-weight vector is computed by making a change in the initial or present guess in a direction opposite to that of the gradient vector.

(iv)   The process is repeated going back to step 2.

It is intuitively reasonable that successive corrections to the tap-weight vector in the direction of the negative of the gradient vector should eventually lead to the minimum mean-squared error, J_{min}, at which point the tap weight vector assumes its optimum value w_o.

3.2.1.2. Quantitative Definition. Consider a transversal Wiener filter as shown in figure below [40]:
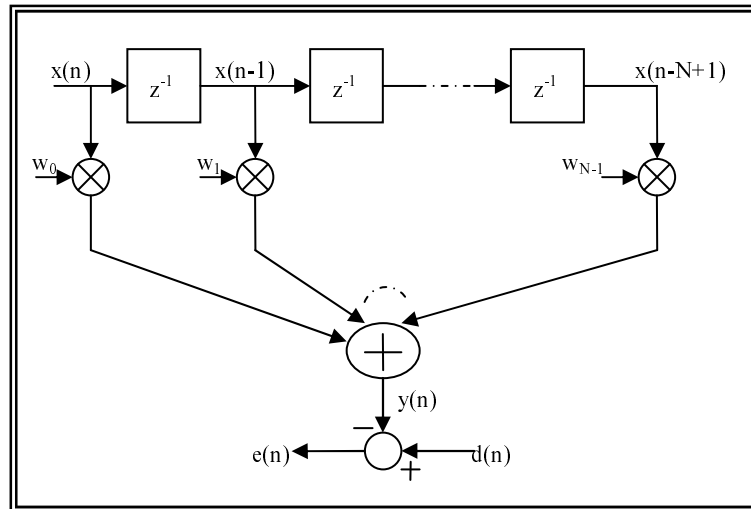
Figure 3.1. Transversal wiener filter representation

The filter input, $\mathbf{x}(n)$, and its desired output, $d(n)$, are assumed to be real-values stationary processes. The filter tap weights, w_0, w_1, …, w_{N-1} are also assumed to be real values. The filter input and tap-weight vectors are defined, respectively, as the column vectors:

$$\boldsymbol{w} = [w_0 \, w_1 \, ... \, w_{N-1}]^T \tag{3.7}$$

and

$$\boldsymbol{x}(n) = [x(n) \, x(n-1) \, ... \, x(n-N+1)]^T \tag{3.8}$$

where T in the superscript stands for transpose.

The filter output is

$$y(n) = \sum_{i=0}^{N-1} w_i x(n-i) = \boldsymbol{w}^T \boldsymbol{x}(n) \tag{3.9}$$

Since $\mathbf{w}^T\mathbf{x}(n)$ is a scalar and thus it is equal to its transpose, i.e. $\mathbf{w}^T\mathbf{x}(n) = (\mathbf{w}^T\mathbf{x}(n))^T = \mathbf{x}^T(n)\mathbf{w}$ and $y(n)$ can also be written as:

$$y(n) = \mathbf{x}^T(n)\mathbf{w} \qquad (3.10)$$

Error function is defined as:

$$e(n) = d(n) - y(n)$$

$$= d(n) - \mathbf{w}^T\mathbf{x}(n) \qquad (3.11)$$

$$= d(n) - \mathbf{x}^T(n)\mathbf{w}$$

In Wiener functions the performance function is defined to be

$$\xi = E[|e(n)|^2] \qquad (3.12)$$

When e(n) is replaced with Eq. 3.11:

$$\xi = E[e^2(n)] = E[(d(n) - \mathbf{w}^T\mathbf{x}(n))(d(n) - \mathbf{x}^T(n)\mathbf{w})] \qquad (3.13)$$

Expanding the right hand side of the above equation and since **w** is not a statistical variable, shifting it out of the expectation operator, E[.], the following equation is obtained:

$$\xi = E[d^2(n)] - \mathbf{w}^T E[\mathbf{x}(n)d(n)] - E[d(n)\mathbf{x}^T(n)]\mathbf{w} + \mathbf{w}^T E[\mathbf{x}(n)\mathbf{x}^T(n)]\mathbf{w} \qquad (3.14)$$

Defining (Nx1) cross correlation vector between **x**(n) and d(n) as:

$$\mathbf{p} = E[\mathbf{x}(n)d(n)] = [p_0\ p_1\ \dots\ p_{N-1}]^T \qquad (3.15)$$

and (NxN) autocorrelation matrix of the filter input as:

$$\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)] \qquad (3.16)$$

It is obtained:

$$\xi = E[d^2(n)] - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w} \qquad (3.17)$$

The final function is a quadratic function of the filter tap-weight vector **w**. It has a single global minimum obtained by solving Wiener-Hopf equation:

$$\boldsymbol{R}\boldsymbol{w}_0 = \boldsymbol{p} \qquad (3.18)$$

This is achieved by the following iteration process called steepest descent algorithm if R and p are not available:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \mu \nabla_k \xi \qquad (3.19)$$

where

$$\nabla \xi = 2\boldsymbol{R}\boldsymbol{w} - 2\boldsymbol{p} \qquad (3.20)$$

and gradient operator is defined as the column vector:

$$\nabla = \left[\frac{\partial}{\partial w_0} \ \frac{\partial}{\partial w_1} \ \cdots \ \frac{\partial}{\partial w_{N-1}} \right]^T \qquad (3.21)$$

Going back to LMS algorithm, it adapts the filter tap weights so that e(n) is minimized in the mean square sense. When the processes x(n) & d(n) are jointly stationary, this algorithm converges to a set of tap-weights which, on average, are equal to Wiener-Hopf solution. LMS algorithm is a practical algorithm for realizing Wiener filters, without explicitly solving Wiener-Hopf equation. The conventional LMS algorithm is a stochastic implementation of the steepest descent algorithm. It simply replaces the cost function $\xi = E[e^2(n)]$ by its instantaneous coarse estimate $\xi' = e^2(n)$. Substituting $\xi' = e^2(n)$ for ξ in the steepest descent recursion, following statement is obtained:

$$\overline{W}(n+1) = \overline{W}(n) - \mu \nabla e^2(n) \qquad (3.22)$$

where

$$\overline{W}(n) = \left[w_0(n) \ w_1(n) \ \cdots \ w_{(N-1)}(n) \right]^T \qquad (3.23)$$

Note that the i-th element of the gradient vector is:

$$\frac{\partial e^2(n)}{\partial w_i} = 2e(n)\frac{\partial e(n)}{\partial w_i}$$

$$= -2e(n)\frac{\partial y(n)}{\partial w_i} \qquad (3.24)$$

$$= -2e(n)x(n-i)$$

Then:

$$\nabla e^2(n) = -2e(n)\bar{x}(n) \qquad (3.25)$$

where

$$\bar{x}(n) = [x(n) \; x(n-1) \; ... \; x(n-N+1)]^T \qquad (3.26)$$

Final step is substituting 3.25 into 3.22 to finally obtain LMS recursion as follows:

$$\overline{W}(n+1) = \overline{W}(n) + 2\mu e(n)\bar{x}(n) \qquad (3.27)$$

LMS algorithm has many advantages over other adaptive algorithms such as RLS. It is simple in implementation as it requires only (2N+1) operations per iteration. Variations of the algorithm that use only sign of the error value or the signs of data symbols or both have also proven to be effective but converges much slowly than the original algorithm. It also has stable and robust performance against different signal conditions. However, it has the disadvantage of slow convergence due to eigenvalue spread.

3.3.   RLS Algorithm

The LMS algorithm for adaptively adjusting the tap coefficients of a linear equalizer or a DFE is basically a steepest descent algorithm in which the true gradient vector is approximated by an estimate obtained directly from the data [36].

The major advantage of the steepest descent algorithm lies in its computational simplicity. However, the price paid for the simplicity is slow convergence, especially when channel characteristics result in large eigenvalue spread in the autocorrelation matrix. In other words, the gradient algorithm has only a single adjustable parameter for controlling the convergence rate. Consequently the slow convergence is due to this fundamental limitation.

In order to obtain faster convergence, it is necessary to devise more complex algorithms involving additional parameters. In deriving faster converging algorithms, least squares approach should be adopted. Thus, received data should be directly dealt with in minimizing the quadratic performance index, whereas previously the expected value of the squared error was to be minimized. This simply means that performance index is expressed in terms of a time average instead of a statistical average [36].

In recursive implementations of the method of least squares, the computation starts with known initial conditions and the information contained in new data samples are used to update the old estimates. Therefore the length of observable data is variable. Accordingly, the cost function to be minimized as $\xi(n)$, where n is the variable length of the observable data [39]. Also, it is customary to introduce a weighting factor into the definition of the cost function $\xi(n)$:

$$\xi(n) = \sum_{i=1}^{n} \beta(n,i)|e(i)|^2 \qquad (3.28)$$

where e(i) is the difference between the desired response d(i) and the output y(i) produced by the transversal filter whose tap inputs (at time i) equal to u(i), u(i-1), …, u(i-M+1). That is, e(i) is defined by:

$$e(i) = d(i) - y(i)$$

$$= d(i) - \boldsymbol{w}^H(n)\boldsymbol{u}(i)$$

(3.29)

where **u**(i) is the tap-input vector at time i, defined by:

$$\mathbf{u}(i) = [u(i)\ u(i-1)\ ...\ u(i-M+1)]^{\mathrm{T}} \tag{3.30}$$

and $\mathbf{w}(n)$ is the tap-weight vector at time n, defined by:

$$\boldsymbol{w}(n) = [w_0(n)\ w_1(n)\ ...\ w_{M-1}(n)]^T \tag{3.31}$$

The weighting factor $\beta(n,i)$ in first equation has the property that

$$0 < \beta(n,i) \leq 1, \ \ i = 1,2,...,n \tag{3.32}$$

The use of the weighting factor is intended to ensure that data in the distant past are forgotten. A special form of weighting that is commonly used is the exponential weighting factor or forgetting factor defined by:

$$\beta(n,i) = \lambda^{n-i}, i = 1,2,...,n \tag{3.33}$$

The inverse of (1-λ) is a measure of the memory of the algorithm. When λ=1, it stands for infinite memory. In the method of exponentially weighted least squares, cost function to be minimized becomes:

$$\xi(n) = \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2 \tag{3.34}$$

At time n, the best estimates of the autocorrelation and cross correlation matrixes are respectively:

$$\boldsymbol{\Phi}(n) = \sum_{i=1}^{n} \lambda^{n-i}\boldsymbol{u}(i)\boldsymbol{u}^H(i) = \lambda\boldsymbol{\Phi}(n-1) + \boldsymbol{u}(n)\,\boldsymbol{u}^H(n)$$

$$\boldsymbol{z}(n) = \sum_{i=1}^{n} \lambda^{n-i}\boldsymbol{u}(i)\boldsymbol{d}^*(i) = \lambda\boldsymbol{z}(n-1) + \boldsymbol{u}(n)\,\boldsymbol{d}^*(n) \tag{3.35}$$

The best estimate of the optimum weight vector is given by Wiener Hopf solution:

$$w(n) = \boldsymbol{\Phi}^{-1}(n)\boldsymbol{z}(n) = \boldsymbol{P}(n)\boldsymbol{z}(n) \tag{3.36}$$

Applying the matrix inversion lemma,

$$\boldsymbol{\Phi}^{-1}(n) = \boldsymbol{P}(n) = \lambda^{-1}\boldsymbol{P}(n-1) - \left(\frac{\lambda^{-2}\boldsymbol{P}(n-1)\boldsymbol{u}(n)\boldsymbol{u}^H(n)\boldsymbol{P}(n-1)}{1+\lambda^{-1}\boldsymbol{u}^H(n)\boldsymbol{P}(n-1)\boldsymbol{u}(n)}\right)$$

(3.37)

$$= \lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}\boldsymbol{K}(n)\boldsymbol{u}^H(n)\boldsymbol{P}(n-1)$$

where **K**(n) is the gain vector defined by:

$$\boldsymbol{K}(n) = \boldsymbol{G}(n)\boldsymbol{u}(n)$$

(3.38)

and

$$\boldsymbol{G}(n) = \frac{\lambda^{-1}\boldsymbol{P}(n-1)}{1+\lambda^{-1}\boldsymbol{u}^H(n)\boldsymbol{P}(n-1)\boldsymbol{u}(n)}$$

(3.39)

After some mathematical manipulations, weight updating formula becomes:

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \boldsymbol{K}(n)\alpha(n)$$

$$= \boldsymbol{w}(n-1) + \boldsymbol{G}(n)\alpha(n)\boldsymbol{u}(n)$$

(3.40)

where α is the difference between desired and estimated symbol:

$$\alpha(n) = d(n) - \boldsymbol{w}^H(n-1)\boldsymbol{u}(n)$$

(3.41)

There is a formal resemblance with LMS. The weight vector is updated proportionally to the product of the current input **u**(n) and some error signal α(n). The error in RLS is defined a priori in the sense that it is based on the old weight vector **w**(n-1), whereas in LMS the error e(n) = d(n) - **w**^{H}(n)**u**(n) is computed a posteriori, that is, based on the current weight vector. A more important difference though is that the constant learning rate μ of LMS is replaced by a matrix **G**(n) [=μ(n)**P**(n-1)]that depends on the data and on the iteration. In that respect, RLS can be thought of a kind of LMS algorithm having a matrix controlled optimal learning rate. The weight update formula could also be rewritten as:

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \mu(n)\boldsymbol{P}(n-1)\alpha(n)\boldsymbol{u}(n)$$

(3.42)

where μ(n) is a time and data dependent scalar learning rate equal to:

$$\mu(n) = \frac{\lambda^{-1}}{1 + \lambda^{-1} \boldsymbol{u}^H(n) \boldsymbol{P}(n-1) \boldsymbol{u}(n)} \qquad (3.43)$$

3.4. Constant Modulus Algorithm (CMA)

The CMA algorithm is a member of the family, so-called blind algorithms, which means that no reference training sequence is available for adaptation sequence, which has to rely on a priori known characteristics of the signal [37].

This algorithm is commonly used in digital phase locked loops, and in channel equalizer when appropriate. Since there is no need to transmit an initial learning sequence, this approach is particularly useful in circumstances where it is impossible or impractical to use a learning sequence, as in broadcasting for example.

The cost function J_{CM} is defined by:

$$J_{CM} = 0.25 E[A^2 - |\tilde{y}(n)|^2]^2 \qquad (3.44)$$

The instantaneous gradient is the derivative, with respect to the coefficients, of the quantity in brackets. For real signals, one gets:

$$Grad(n) = -E[A^2 - |\tilde{y}(n)|^2]^2 \tilde{y}(n) X(n) \qquad (3.45)$$

The filter coefficients are updated by:

$$W(n+1) = W(n) + \delta[A^2 - |\tilde{y}(n+1)|^2]^2 \tilde{y}(n+1) X(n+1) \qquad (3.46)$$

A critical issue with the above algorithm is convergence. Since the cost function is not quadratic, and the algorithm includes nonlinear operations, local minima exist. In addition, the performance, particularly the convergence time, cannot be predicted.

4. MATLAB REALIZATION

Adaptive filters can be integrated in systems with different functionalities, being prediction, system identification, equalization and interference canceling examples of such. In the present case, one such filter has been incorporated in a typical adaptive equalizer configuration. By dynamically adjusting the finite impulse response (FIR) filter coefficients, as dictated by an adaptation algorithm, one can compensate the channel effects. One of the algorithms chosen for the simulations is the Recursive Least Squares (RLS), which is one of the most efficient algorithms for implementation of adaptive filters. The second one is the Least Mean Squares (LMS) algorithm, which still dominates most of the applications in this area. Despite requiring more iteration to converge, it is possible to do all the processing between each iteration step with much less computational effort in the LMS algorithm. RLS and LMS algorithms are simulated as symbol spaced equalizer under mean square error criterion. LMS is simulated as symbol spaced and fractionally spaced equalizer under MSE criterion. And finally LMS is simulated as symbol spaced equalizer under MSE and CMA criteria. The summary of simulations done is shown in the below figure:
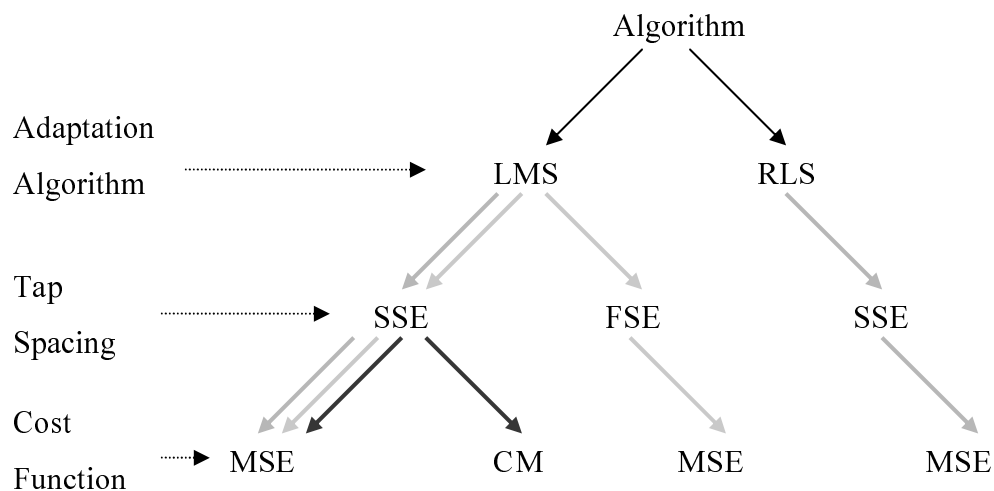

Figure 4.1. Matlab simulation summary (3 comparison sets are represented as 3 different array colors)

In Figure 4.1, green array represents the comparison of LMS and RLS algorithms with symbol spaced equalizer under mean square error criterion. Similarly, blue array stands for the comparison path between the symbol spaced and fractionally spaced equalizer with the same time coverage, coefficients obtained from LMS algorithm depending on mean square cost function. Finally, the comparison of cost functions mean square error and constant modulus is represented by the red array with the coefficients obtained from LMS algorithm for a fractionally spaced equalizer. We can rename the comparisons according to their comparison subjects as adaptation algorithm based, tap spacing based and cost function based comparisons respectively.

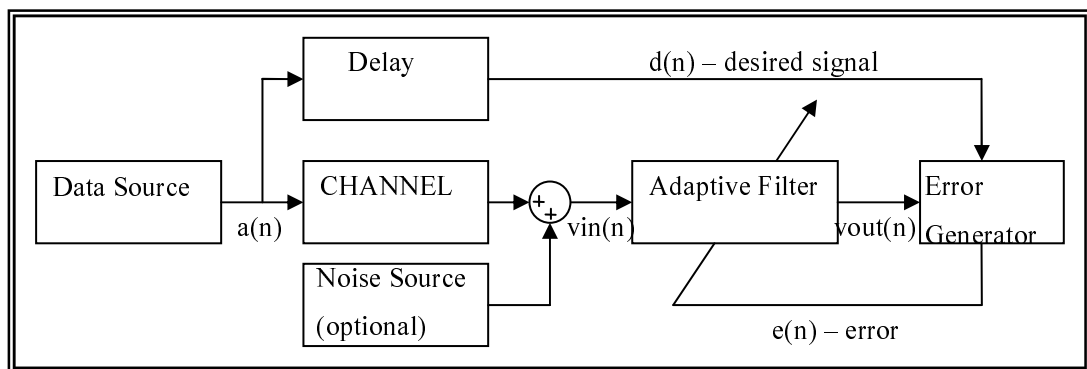The block diagram of simulated systems is shown in Figure 4.2.

Figure 4.2. Basic communication system

4.1. Comparison of Adaptation Algorithms

The first simulation setup aims to compare RLS and LMS algorithms with the simulation setups shown below:
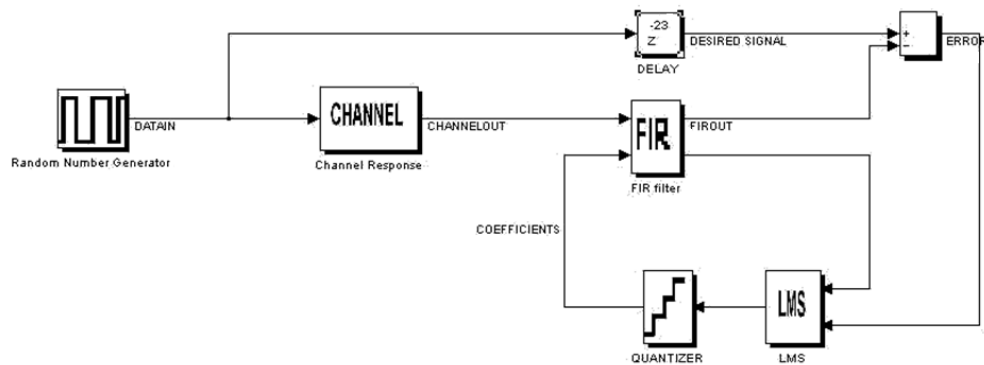
Figure 4.3. Simulation setup for symbol spaced equalizer under mean square error criterion with LMS adaptation algorithm
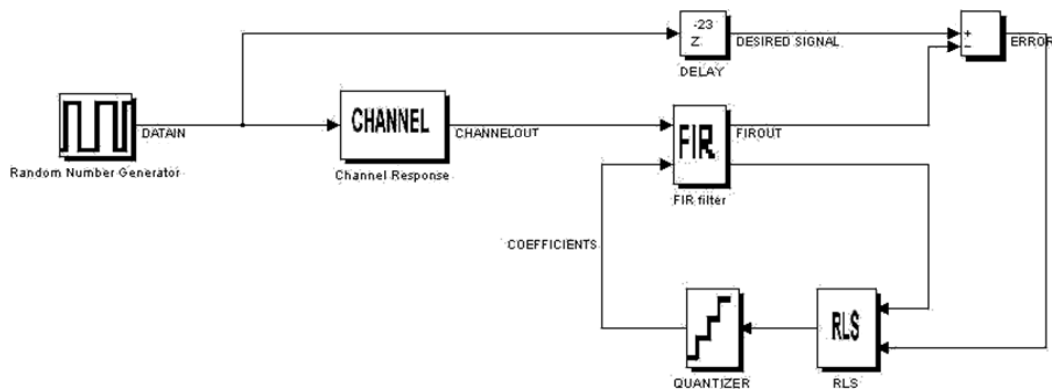

Figure 4.4. Simulation setup for symbol spaced equalizer under mean square error criterion with RLS adaptation algorithm

The only difference between two simulation setups is the adaptation algorithm. Before focusing on the simulations, the sub-blocks will be investigated [41].

4.1.1. Subblocks

4.1.1.1. Random Number Generator. This block is used to generate the random at 3.125Gbps required for the simulation. It is simply composed of a random number generator, whose seed can be controlled changing the parameter "seed" and data rate can be controlled by modifying the parameter "samplet" of the subsystem as shown in figure.
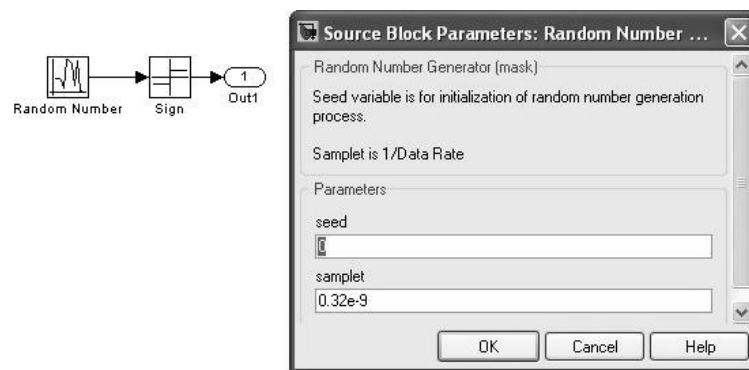

Figure 4.5. Data source – Random number generator

4.1.1.2. Channel Response. Channel model is a 4-port, differential input-differential output, s-parameter model (s4p) and it is converted to continuous time transfer function using MATLAB utilities. It is a model of 30inch PCB channel used for chip to chip communication.

4.1.1.3. FIR Filter. FIR filter is composed of tapped delay line and multiplier cells. For this specific representation 4-tap FIR filter is selected and shown in below figure. Number of taps can be increased by cascading new transport delay cells and summing their outputs after multiplying with the regarding coefficients. The delay cells at the output filter input samples and filter output is used to avoid zero delay loops.
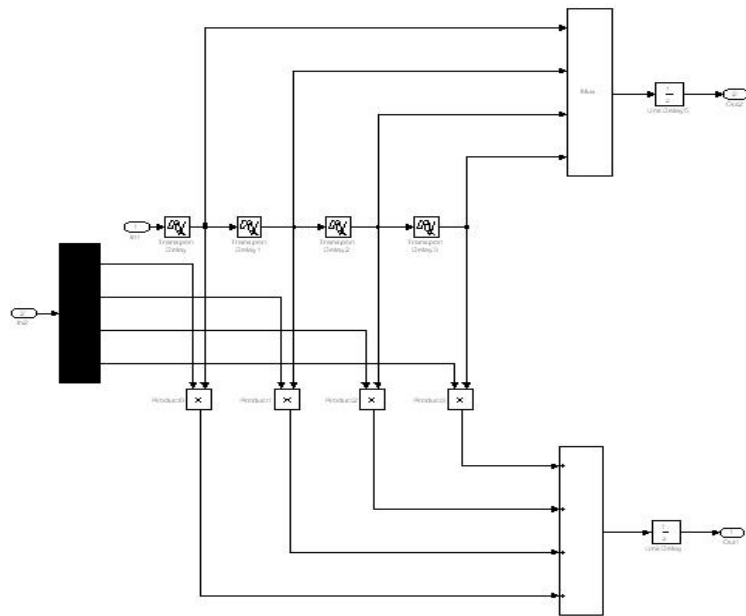
Figure 4.6. FIR equalizer with 4 taps

4.1.1.4. Quantizer. This block is used to determine the number of bits required to convert analog coefficients to digital accurately enough. It is simply a lookup table with the option "round to nearest output". Increasing the resolution of the lookup table, hence the number of bits is better for accurate analog to digital conversion. However, due to transistor implementation limitations, number of bits should be limited to its possible minimum value.
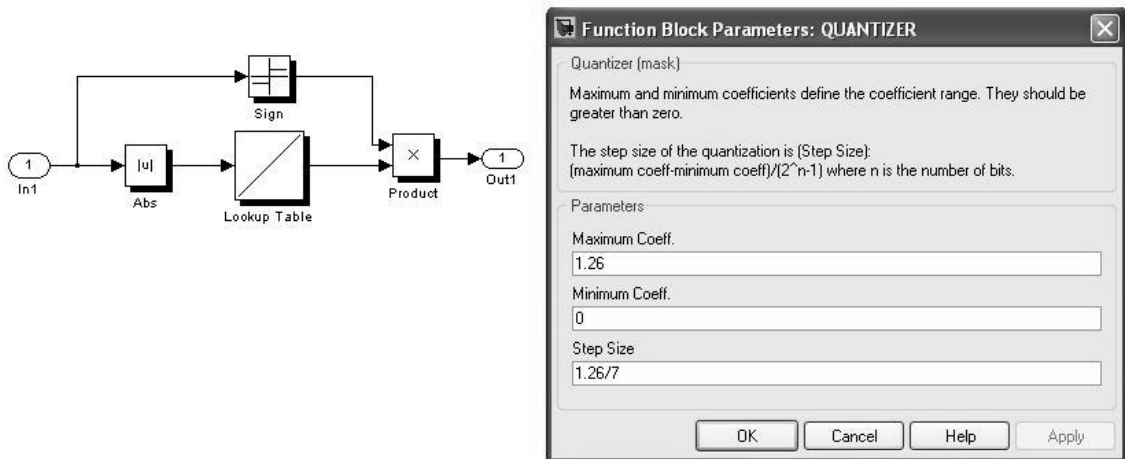
Figure 4.7. Quantization block for FIR filter coefficients

Minimum of the lookup table values is zero and the maximum is optional. The conversion resolution," Step Size", is calculated as follows:

$$Step\ Size = \frac{maximum\ coefficient - minimum\ coefficient}{2^n - 1} \tag{4.1}$$

where n is an integer greater than 1 representing the number of bits used to convert coefficients from analog to digital counterparts. The sign information is carried to the output by a sign and product block to separate the sign bit from the conversion bits.

4.1.1.5. Delay.   The output of the random number generator block is delayed to compensate for the channel and equalizer delay. It becomes the desired signal and the equalizer output is subtracted from this desired signal to generate the error to be used in the coefficient adaptation algorithm.

4.1.2. Comparative Simulations

4.1.2.1. RLS vs. LMS. The coefficient update mechanism in RLS algorithm is as follows:

$$\boldsymbol{C}(n) = \boldsymbol{C}(n-1) + \boldsymbol{k}(n)\varepsilon(n) \tag{4.2}$$

where **C**(n) is the vector with the filter's coefficients; **k**(n) is the Kalman's gain vector and ε(n) is the difference between the desired signal and the output of the filter in the nth iteration. This formula is implemented in Matlab as follows:
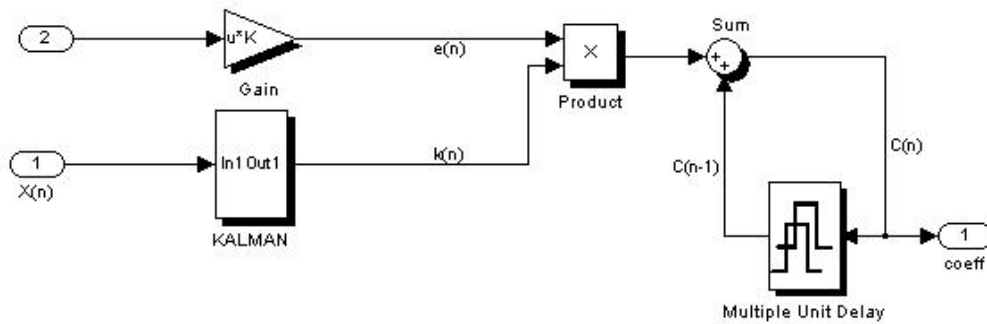

Figure 4.8. RLS adaptation algorithm block

Kalman's gain vector is given by [41]:

$$k(n) = \frac{P(n-1)a(n)}{\alpha + a^T(n)P(n-1)a(n)} \tag{4.3}$$

where a(n) is the vector that contains the filter's coefficients; P(n) is the inverse of the autocorrelation matrix estimation of the signal received by the filter.

P(n) is a square matrix with the same dimension as the number of coefficients of the filter and may be recursively defined as in the following equation:

$$P(n) = \frac{1}{\alpha}\left[\frac{P(n-1) - P(n-1)a(n)a^T(n)P(n-1)}{\alpha + a^T(n)P(n-1)a(n)}\right] \tag{4.4}$$

where α is a forgetting factor of the initial samples. With this recursive expression one has to initialize P(0) (usually P(0) = 100I/α^2(0) ,where I is the identity matrix) and use the P(n−1) matrix that resulted from the previous iteration.

The above last two functions are implemented in Matlab as Kalman filter with the following code:

```
function k=rlsalgo4(a,pp)
% RLS Processor
p=zeros(4,4);
p=[];
pp=pp';
for i=1:4:16, p=[p; pp(i:i+3)]; end
p1=(p-(p*a*a'*p)./(0.99+a'*p*a))./0.99;
k=(p*a)./(0.99+a'*p*a);
k=k';
for i=1:4, k=[k p1(i,:)]; end
k=k';
```

Figure 4.9. Kalman filter matlab code

p1 and k are outputs of the algorithm and p1 is fed back to the code as pp after delayed for one sample time as in the following figure:
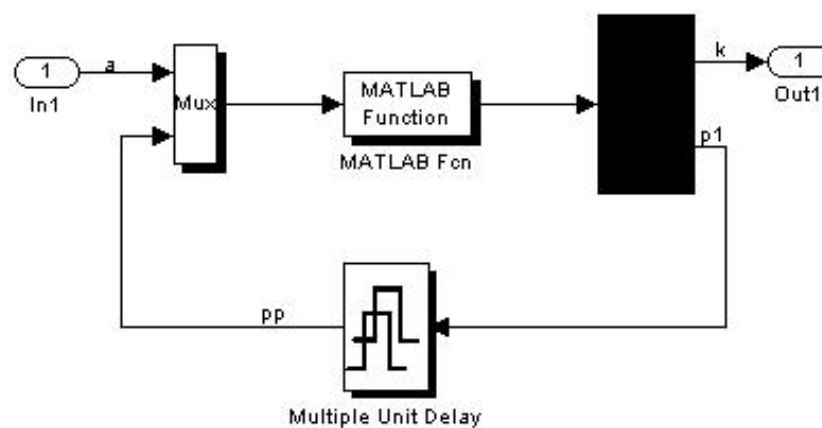

Figure 4.10. Kalman filter block top level

The coefficient update mechanism in LMS algorithm is as follows:

$$C(n) = C(n-1) + \mu\varepsilon(n)X(n) \tag{4.5}$$

where $C(n)$ is the vector with the filter's coefficients; $X(n)$ is the vector that contains the tap values and $\varepsilon(n)$ is the difference between the desired signal and the output of the filter in the nth iteration. This formula is implemented using Matlab Simulink as follows:
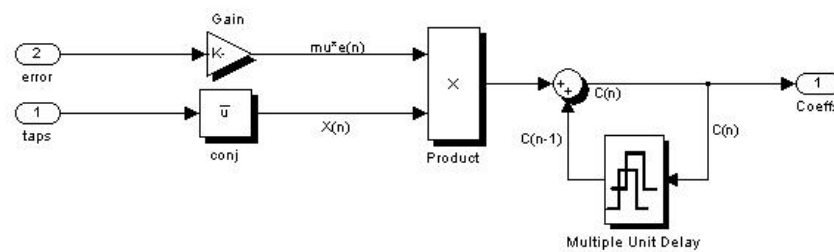

Figure 4.11. RLS adaptation algorithm block

A four tap symbol spaced equalizer is used for the comparison. Coefficients obtained by two different algorithms, LMS and RLS versus time, are represented below:
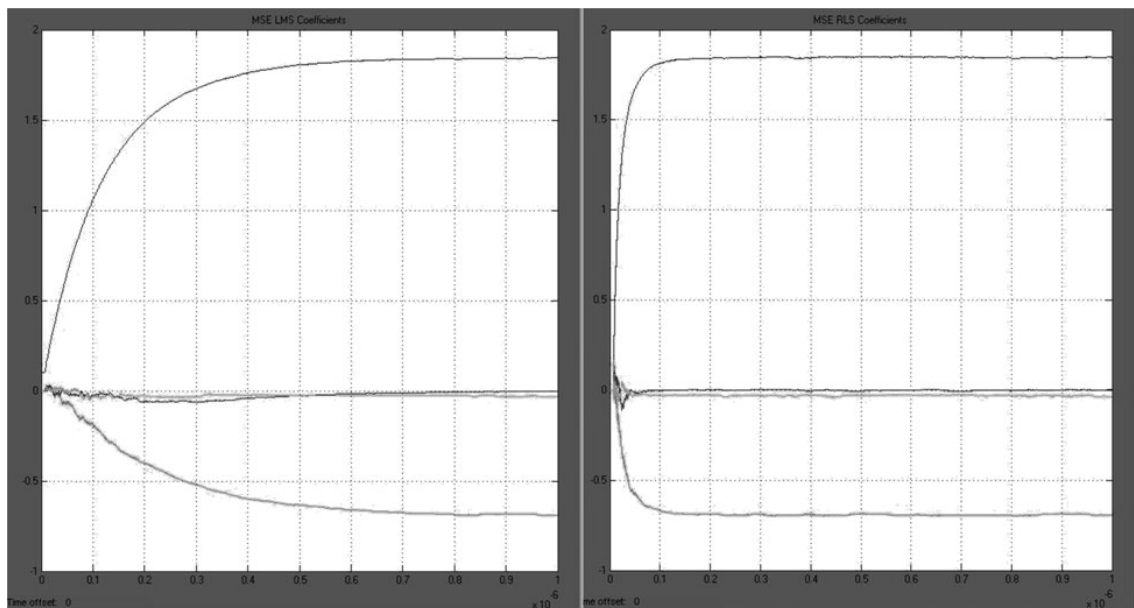

Figure 4.12. Coefficient convergence of LMS (left plot) and RLS (right plot)

They converged to the same coefficients [1.846 -0.691 0.004 -0.034], where the first coefficient is for the first tap, to end up with 0.023 RMS error. RMS error is observed using "Running RMS" block from Simulink with the input of difference between the desired signal and the estimated output. The comparison figure shows that, RLS is faster in terms of convergence rate.

4.1.2.2. SSE vs. FSE. In this comparison, equalizers under mean square error criterion covering the same time span with different tap spacing and number of taps. For both equalizers LMS algorithm is used to adapt the tap coefficients.

Time range of 2T is set as equalizer coverage. So for the symbol spaced equalizer, 4 taps with T tap spacing is selected and for fractionally spaced counterpart, 4 taps with T/2 tap spacing is used. And the root mean square of the error between the reference signal and estimated signal is nearly the same and around 0.029 for both types of equalizers. More detailed analysis can be found in Table 4.1 and Figure 4.13. The table is simply the summary of RMS errors, with the column representing the number of taps and row representing the tap spacing as a ratio of T, data period.

Table 4.1. RMS values of error versus tap spacing & number of taps

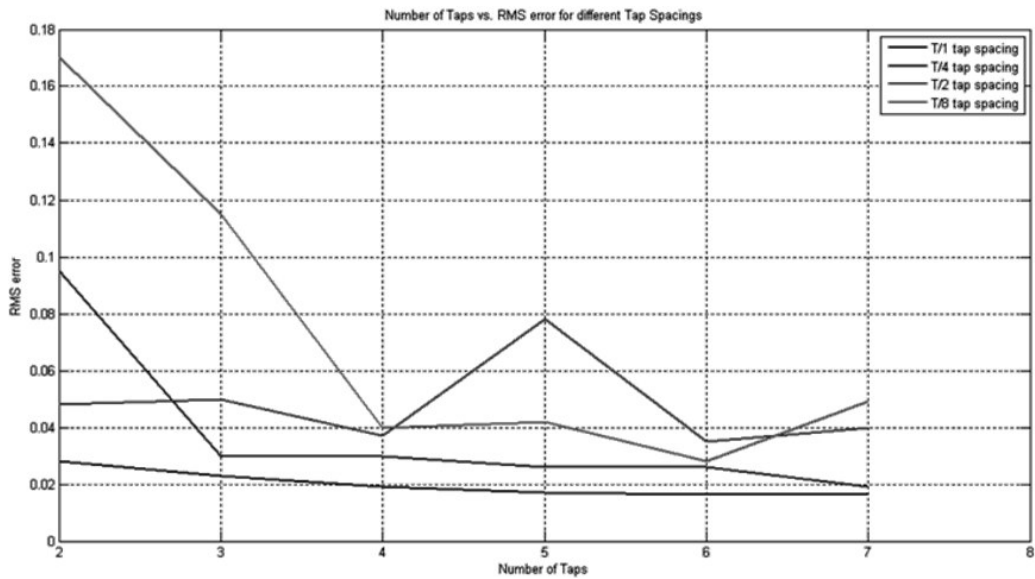|	T/1	T/2	T/4	T/8
2 taps	0.029	0.35	0.107	0.402
3 taps	0.028	0.095	0.048	0.170
4 taps	0.023	0.030	0.050	0.115
5 taps	0.019	0.030	0.037	0.040
6 taps	0.017	0.026	0.078	0.042
7 taps	0.0165	0.026	0.035	0.028
8 taps	0.0165	0.019	0.040	0.049

Figure 4.13. RMS values of error versus tap spacing & number of taps

4.1.2.3. MSE vs. CMA. Until now, the cost function of mean square error was used in calculations. This cost function will be compared with another cost function, called Constant Modulus. LMS is used as adaptation algorithm. Under constant modulus criterion, LMS algorithm does not use the difference of the reference signal and estimated signal as the error. This phenomenon is called blind equalization. Instead, error is defined as follows:

$$\varepsilon(n) = y(n)[1 - |y(n)|^2] \qquad (4.6)$$

where y(n) is the output of the equalizer and update mechanism works as usual:

$$\boldsymbol{C}(n) = \boldsymbol{C}(n-1) + \mu\varepsilon(n)\boldsymbol{X}(n) \qquad (4.7)$$
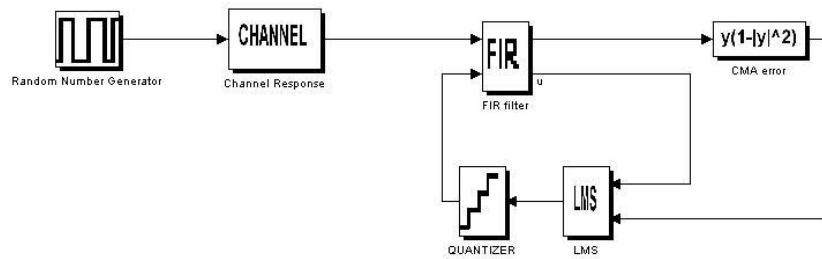
Below figure is the simulation setup in MATLAB:

Figure 4.14. Constant modulus algorithm (CMA) simulation setup

The CMA error block above is to implement the error to be used by LMS and it is as follows:
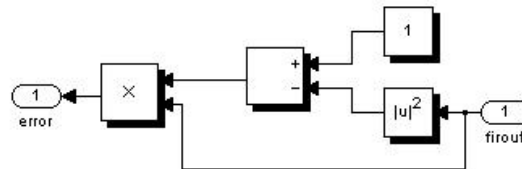

Figure 4.15. Error implementation in CMA

Coefficients obtained by two different cost functions, MSE and CM, versus time, are represented below:
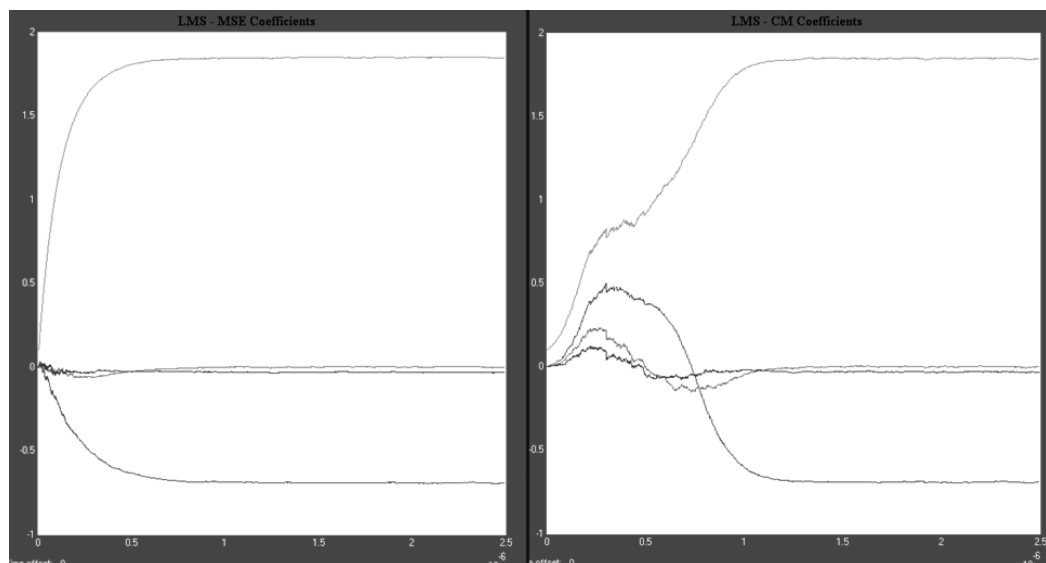

Figure 4.16. MSE vs. CM

4.1.2.4. Coefficient Resolution and Number of Taps. First of all, for the circuit to be easily realizable with transistors, minimum number of taps and quantization bits that will still be adequate to open the received data eye should be determined. The decision will be based on the root mean square of the error value.

This will be done by using LMS setup and once this minimum is determined, the error output will be compared with the RLS counterpart.

RMS error values for different number of taps and number of quantization bits are summarized in the below table:

Table 4.2. RMS values of error versus number of bits & number of taps

LMS	2TAPS	3TAPS	4TAPS	5TAPS	6TAPS	7TAPS	8TAPS
2bits	0.054	0.054	0.052	0.052	0.054	0.054	0.054
3bits	0.063	0.063	0.062	0.062	0.062	0.062	0.062
4bits	0.038	0.038	0.038	0.038	0.038	0.038	0.038
5bits	0.030	0.030	0.030	0.030	0.030	0.030	0.030
6bits	0.024	0.024	0.024	0.021	0.019	0.019	0.019
infinite	0.029	0.028	0.023	0.019	0.017	0.0165	0.0165

4.2. Top Level Simulation

Assuming that channel is time invariant, the adaptation algorithm is implemented in MATLAB Simulink to obtain coefficients for using in Cadence simulations.

Equalizer with 2 taps and T/4 symbol spacing is simulated in addition to 4 taps and T/8 symbol spacing. LMS algorithm under MSE criterion is employed. The simulation results are summarized in the below figures:
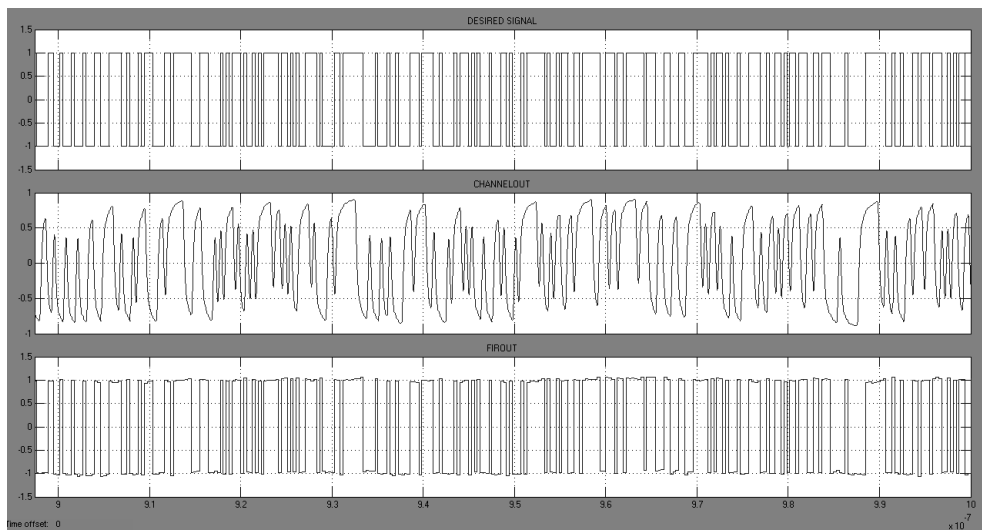
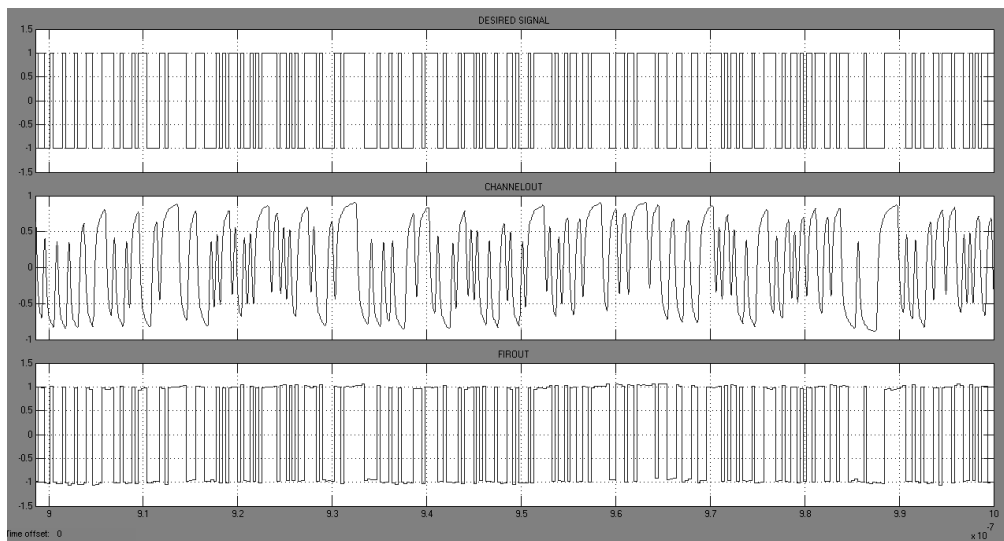Figure 4.17. 4 taps with T/8 tap FIR equalizer MATLAB simulation output.

Figure 4.18. 2 taps with T/4 tap FIR equalizer MATLAB simulation output.

5. ANALOG CMOS IMPLEMENTATION OF FIR EQUALIZER

The building blocks required in design of FIR filter is going to be presented in this chapter. The blocks are MDAC (multiplying digital to analog converter) [17], CMFB (common mode feedback) required for MDAC, unit delay cell (transport delay) [17], limiting amplifier for amplifying FIR output [17] and ADC (analog to digital converter) [42] to convert tap values for digital circuitry to be able to use them in coefficient calculations. STMicroelectronics CMOS065 technology is used for design. Simulation results belong to schematic designs that are designed to work at 3.125Gbps data rate (1.5625GHz for clock) in schematic design phase.

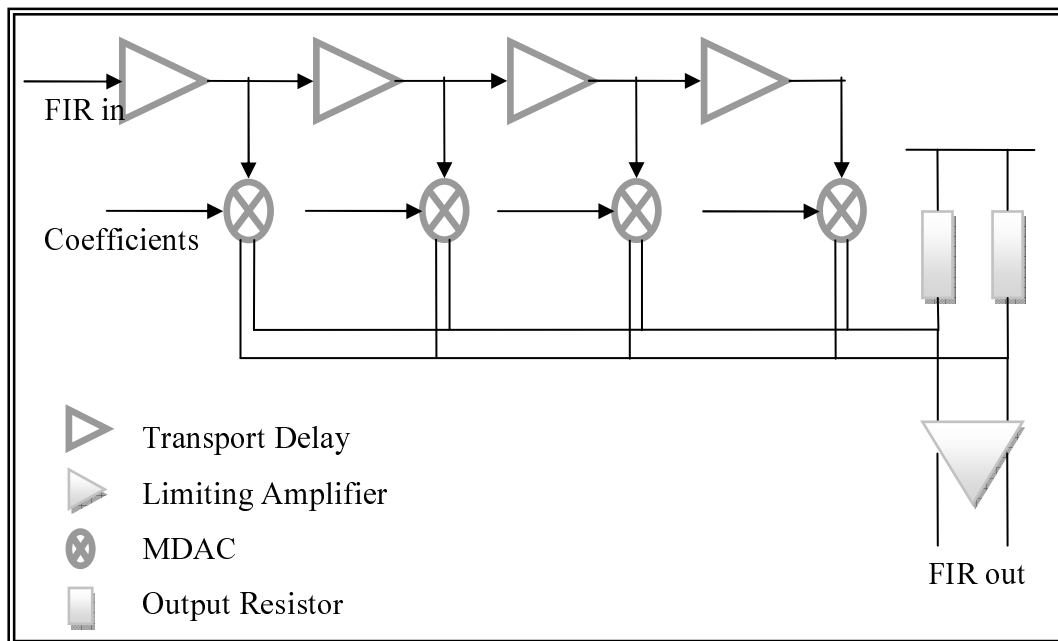At its simplest, top level to be designed is shown in below figure:

Figure 5.1. Top level analog implementation

5.1. Block Level Design

5.1.1. MDAC Design

Multiplying DAC is used to multiply the analog tap signals with digital tap coefficients of FIR filter. The theoretical representation is as follows:
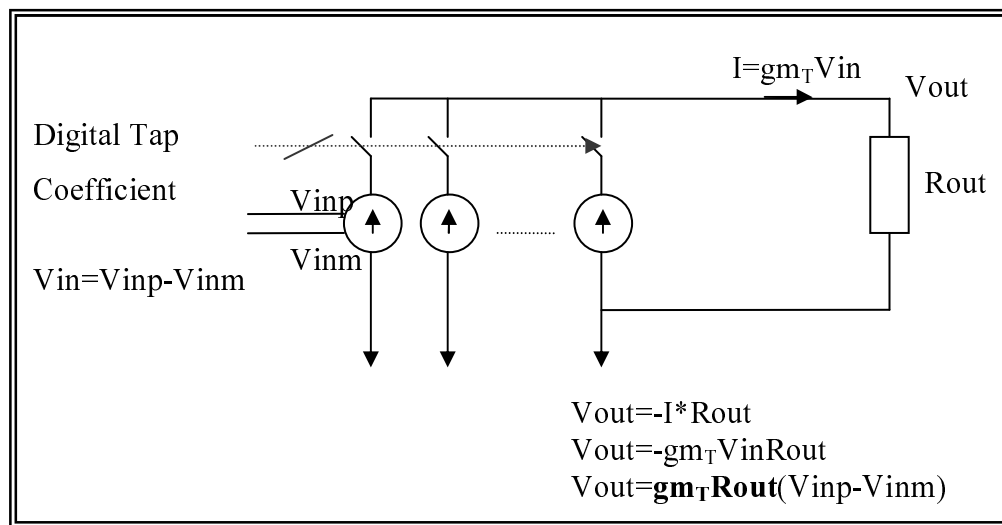

Figure 5.2. MDAC theoretical representation

As obvious from the representation, the block output is linear with its input with some scaling factor of $gm_T Rout$. The transconductance, gm_T, depends on the digital tap coefficient, which is either obtained from MATLAB simulations or adaptation algorithm on IC.

Transistor level implementation of MDAC used is represented in Figure 5.3. One of them being the sign bit, 5 bits are used to represent each tap coefficient. That's why there are 2^n ($2^4=16$) gain values are present for each sign possibility. One of the gains is 0 (zero), so 15 switches are enough to implement 15 different gain values. Decoding the coefficients bits are presented in Table 5.2.

MDAC topology consists of 30 switches to get the differential input data and 60 switches for 15 bits of coefficient control word where only 30 of the switches are active at once depending on the coefficient sign. When both of its inputs are at common mode value, each unit cell, MDACP and MDACN, consumes 50uA to conclude with approximately 1.5mA (50uA x 2 x 15) in total when all control switches are on. The gain range is programmable between 0.075V/V and 1.15V/V linearly with 0.075V/V steps (Table 5.1).

Table 5.1. MDAC gain values vs. coefficient input

COEFFICIENT	GAIN	UNIT
0	0	V/V
1	76.8m	V/V
2	153.7m	V/V
3	230.5m	V/V
4	307.3m	V/V
5	384.2m	V/V
6	461.1m	V/V
7	537.8m	V/V
8	614.6m	V/V
9	691.4m	V/V
10	768.3m	V/V
11	845m	V/V
12	921.8m	V/V
13	998.6m	V/V
14	1.075	V/V
15	1.152	V/V

Table 5.2. Coefficient decoder

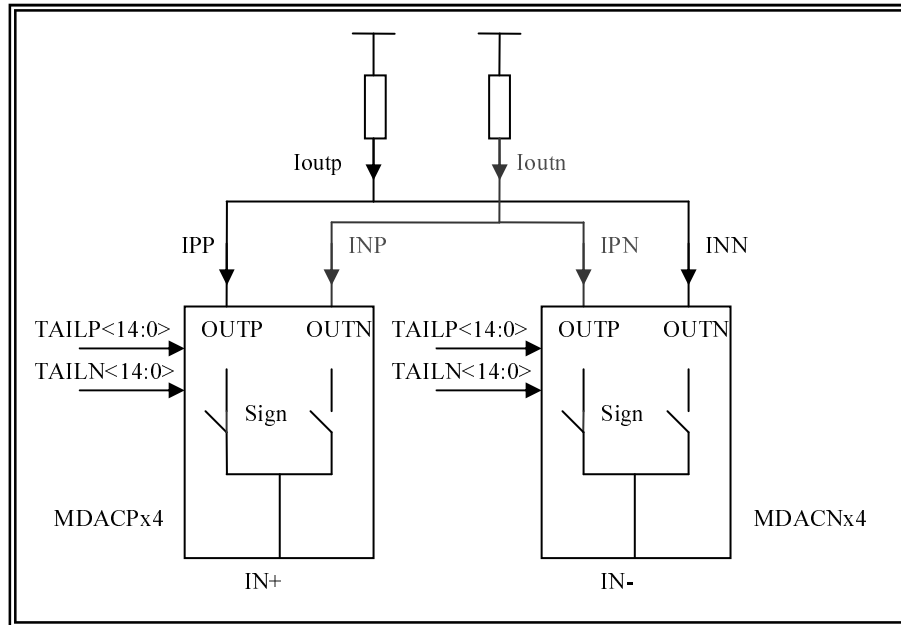| COEFFICIENT	DIGITAL COEFFICIENT <3:0>	DECODED DIGITAL COEFFICIENT <14:0>
0	0000	000000000000000
1	0001	000000000000001
2	0010	000000000000011
..	..	...
14	1110	011111111111111
15	1111	111111111111111

Mentioned topology is represented in Figure 5.3.

Figure 5.3. Top level MDAC cell for 4 taps

When TAILP is active, TAILN<14:0> is all zeros resulting in Ioutp-Ioutn=IPP-IPN.
When TAILN is active, TAILP<14:0> is all zeros resulting in Ioutp-Ioutn=INN-INP.

This way multiplexing due to sign of coefficient is done at output rather than done at input. Since either TAILP or TAILN can be active at one time, the output capacitance is not doubled, even though the transistor amount is doubled.
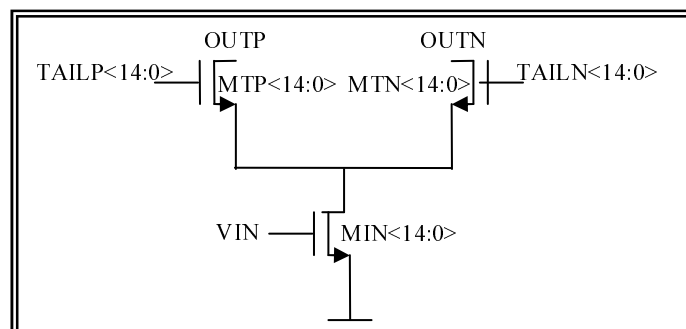
Below is shown one unit cell of MDAC [17].

Figure 5.4. One MDAC unit cell

MIN<14:0> is responsible for creating path to ground for transistors TAILP<14:0> and TAILM<14:0>. This way the transistors used to sense the input data are used to bias unit multiplying cells as well. Depending on the sign bit of multiplier; either switches TAILP<14:0> or TAILN<14:0> are active at one time [17]. The decoding scheme for the sign bit is shown in below figure:
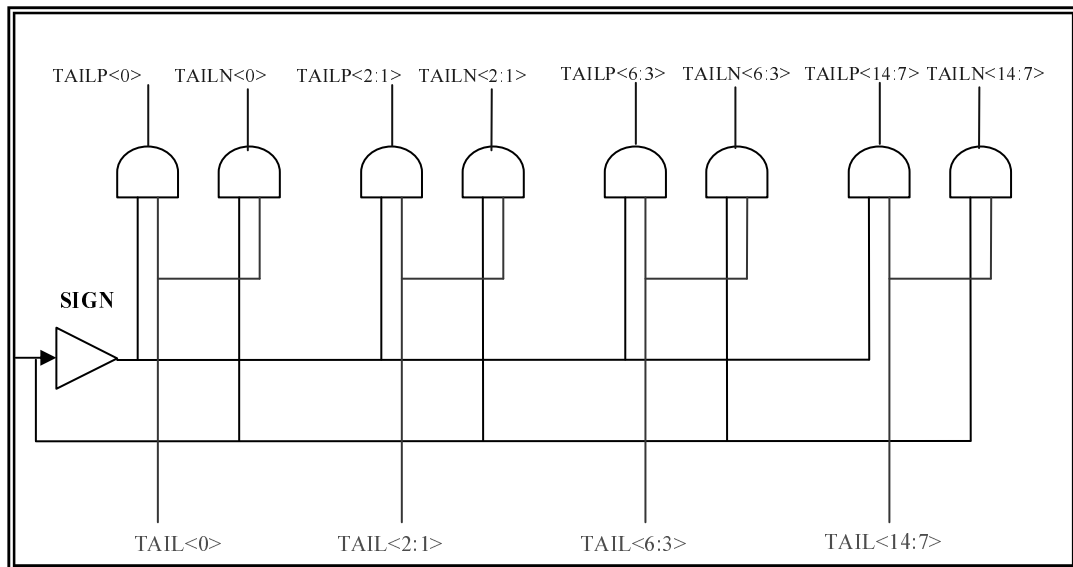

Figure 5.5. Input sign multiplexer for MDAC

When coefficient bits, TAIL<14:0>, are 0 (zero), all of the input switches, MTN<14:0> or MTP<14:0>, will be OFF and oppositely will be ON when all are coefficient bits are 1 (one). The intermediate TAIL<14:0> values result in different conditions by selectively closing and opening the switches that are responsible for sinking current from output resistors.

DC simulation results that show the linearity for input range and coefficient change are summarized in Figure 5.6. Graph is obtained by sweeping input differential voltage for different MDAC coefficients in DC, and plotting differential output over differential input. Another visualization technique of linearity of output according to input for different coefficients, is represented in Figure 5.7. It is simply the plot of differential output versus differential input for different coefficients.
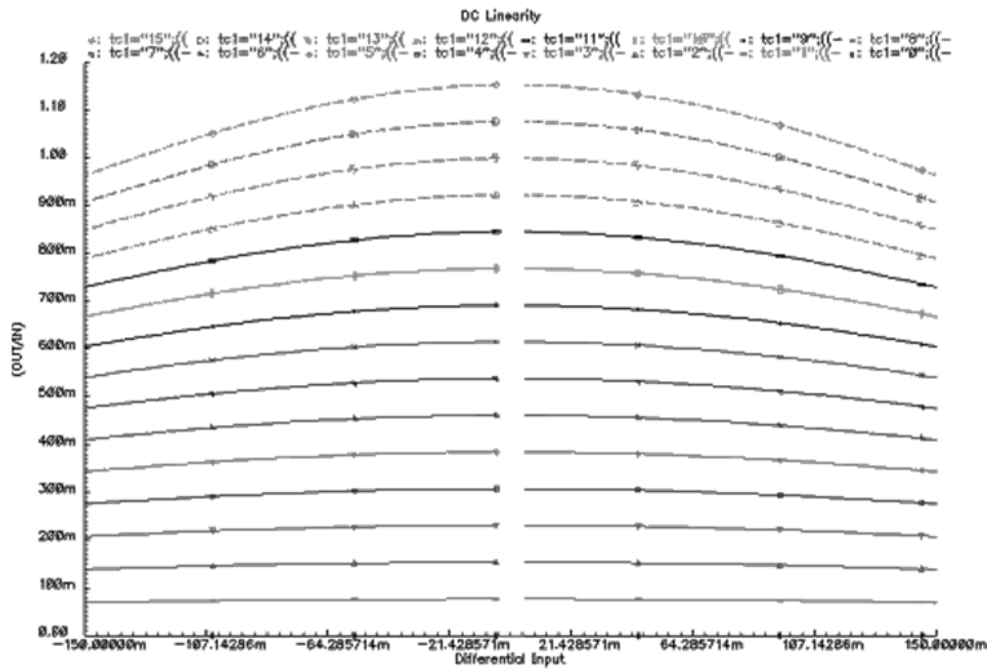
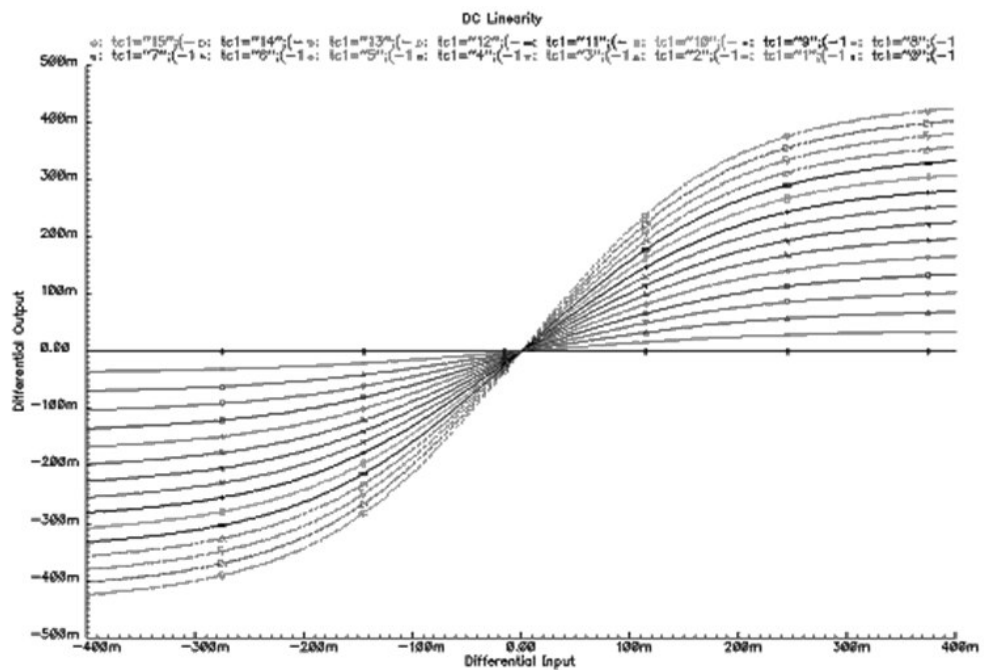Figure 5.6. One MDAC delay cell gain for different control bits over ±150mV input range

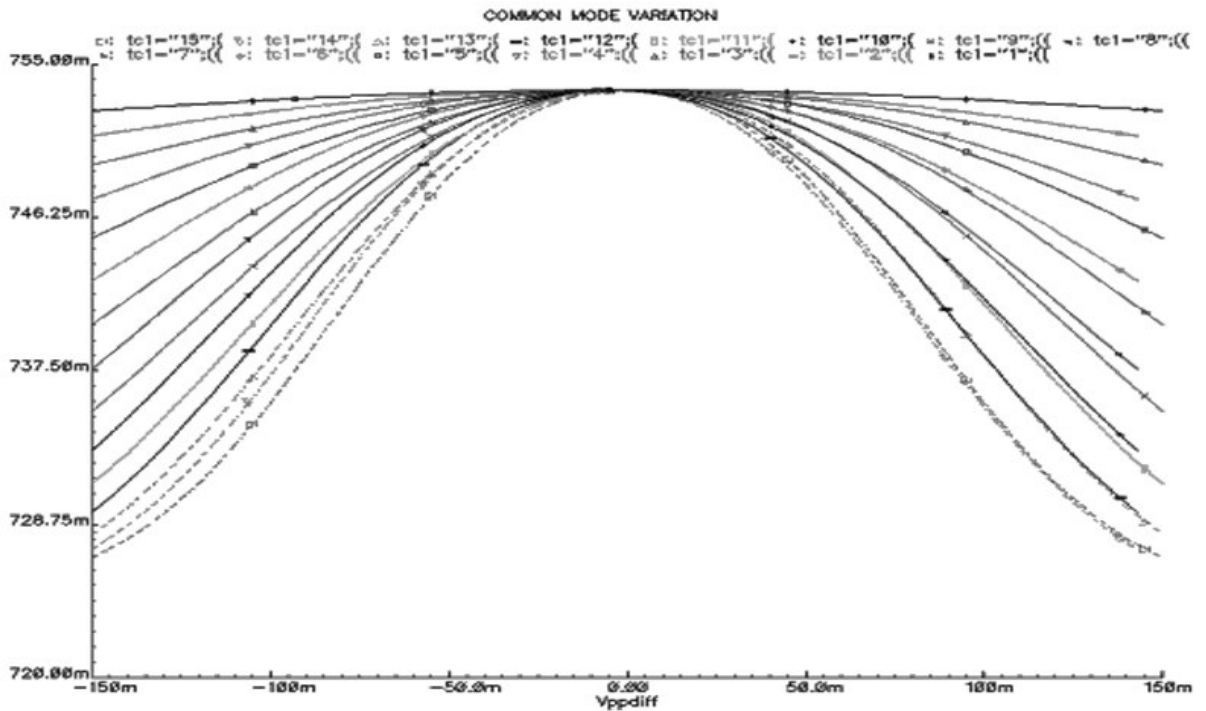Figure 5.7. Input-output linearity with the different control bits over ±400mV input range

Figure 5.8. Output common mode variation with the different control selections over the
input range of ±400mV

Figure 5.8 is the representation of output common mode variation with the changing
coefficients and differential input. The common mode feedback, which will be visited later
in the chapter, aims to fix output common mode voltage to 750mV. There is a maximum of
32mV divergence from the aimed common mode value, when the input amplitude is high.

As input switch, MIN<14:0>, determine the bias currents, linearity for input range
depends on threshold of these input switch transistors. These switch transistors turn off
when input is below the threshold voltage. Input common mode is selected as 0.6V and
MDAC can operate down to 0.4V of input voltage. Input range of 0.4V-0.8V can be
defined as linear operating region of MDAC.

Simulation that illustrates the transient response of MDAC to the coefficient update
mechanism operating at 1.5625GHz (3.125Gbps data rate) is shown in Figure 5.9.
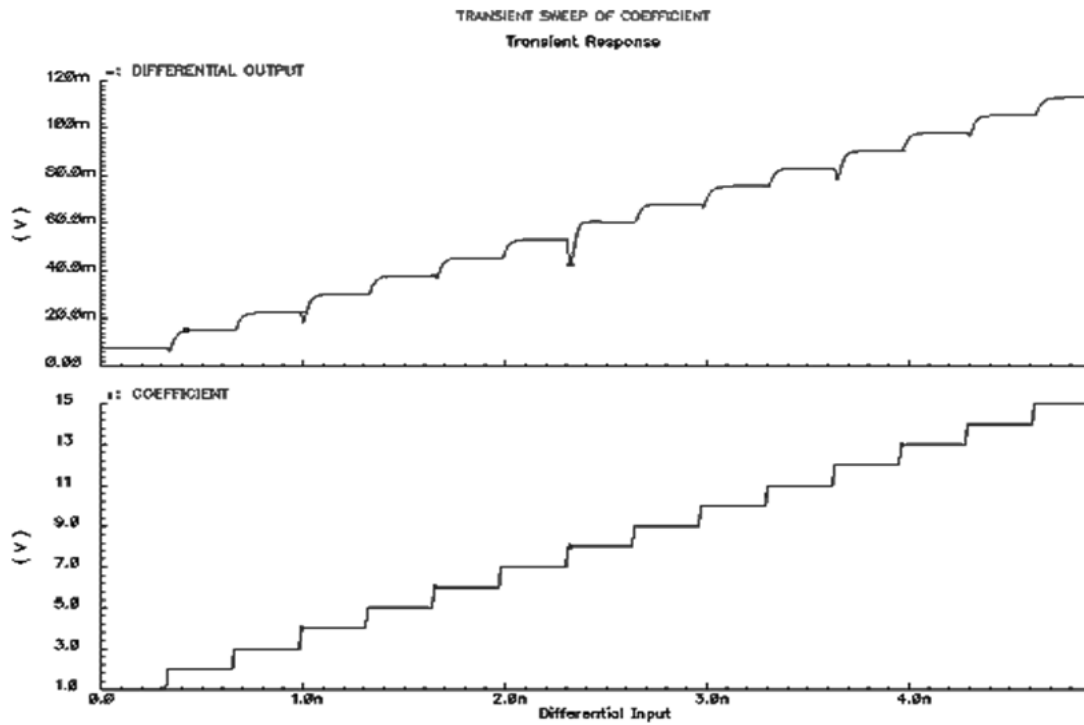
Figure 5.9. MDAC delay cell differential output change with the changing coefficient in transient

To minimize the possible glitches during coefficient transition, the thermometer encoding for coefficient update should be preferred. This way if there is not a huge difference between consecutive coefficients, number of transistors switching will be limited as well.

5.1.2. Common Mode Feedback

Conventional common mode feedback topology operates sensing the common mode, comparing it to the required common mode and sinking or sourcing regarding current from the output nodes according to comparator output. As DC comparators are not fast enough to compensate common mode variations at 3.125Gbps, DC comparison method is supported by another mechanism.

Below figure shows the representation of additional common mode feedback mechanism:
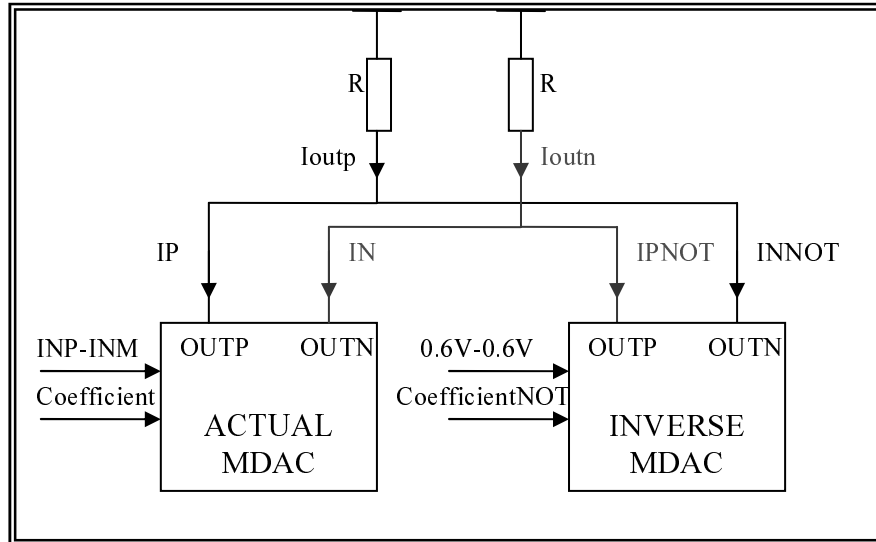

Figure 5.10. Top level common mode feedback mechanism representation

One copy of MDAC cell for each actual MDAC stage is added to the output node, with the input voltages fixed at 0.6V with its coefficients inverted. This way total of $15(Ip+Im)$ is guaranteed on the resistors without distracting the differential voltage as inversed MDAC does not have differential input.

$$2*V_{CM} = [VDD - (15R*I_p)] + [VDD- (15R*I_m)]$$
$$V_{CM} = [(2*VDD) - 15R*(I_p + I_m)] / 2$$

For one MDAC:
$$V_{CM} = [VDD- 7.5R*(I_p + I_m)] = [1.2 - (7.5*300Ohm*100uA)] = 1.2-\Delta V_{CM}$$

As there are four MDAC cells, $\Delta V_{CM}*4$ will be reduced from VDD, which would result in very small common mode voltage.

The aim of the additional MDAC cells is to keep total current flowing through resistors the same for all the coefficient selections. Once VCM is fixed at DC level, conventional common mode feedback mechanism can be applied now to operate at DC.

750mV of common mode voltage is aimed at the output. Therefore, ~1.5mA of current should be stolen from each output resistor to result in required common mode.
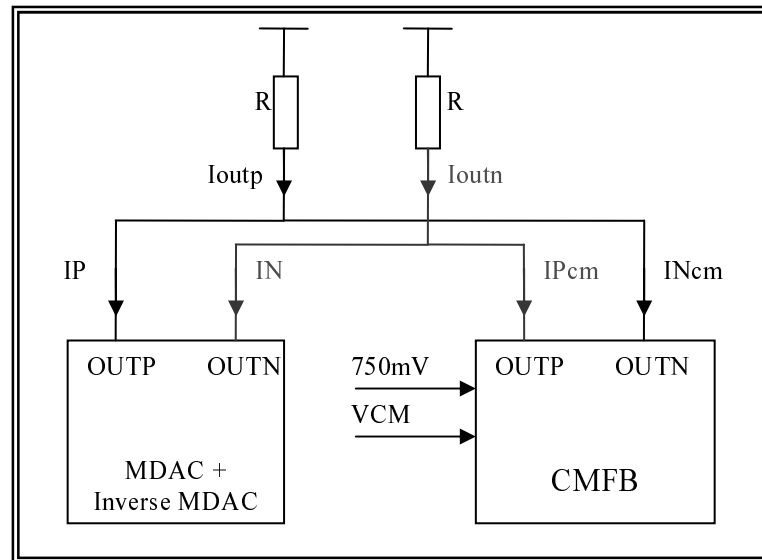

Figure 5.11. Common mode level up shifter mechanism

The common mode feedback mechanism operating at DC consists of comparison of VCM ((Ioutp+Ioutn)/2) with fixed 750mV, and feeding the comparison output to PMOS transistors responsible of sourcing related current to the output node. If VCM is very low, Compout becomes very low, resulting in high current sourced to the output. Knowing that sum of Ioutp and Ioutn is fixed, total current flowing through output resistors will decrease since common mode feedback PMOS transistors will steal some of this fixed current. As the DC current level of output resistors decrease, the common mode voltage will tend to increase [17].
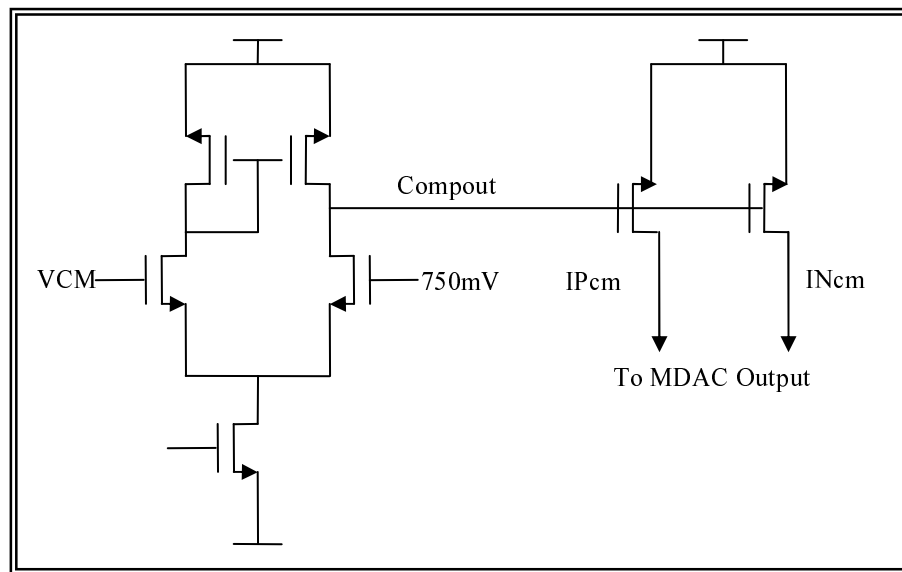
Figure 5.12. Common mode feedback mechanism

5.1.3. Tap Delay Cell

Continuous time transport delay cell with unity gain and high bandwidth is selected to generate tapped delay line. Continuous time implementation is preferred over sample and hold counterpart, in order to avoid complexity due to clock involvement. Examples to possible complexity are clock data synchronization, coupling of clock to the input etc.

Unit delay cell is composed of an inverter with active inductor load. It is a modified inverter to achieve higher bandwidth with the use of two additional transistors which act like an inductive load and to achieve low gain with the use of two more transistors. Transistors MN2 and MP2 are used to form low resistive load to decrease gain, whereas MN3 and MP3 are used as active inductor load to increase bandwidth [17]

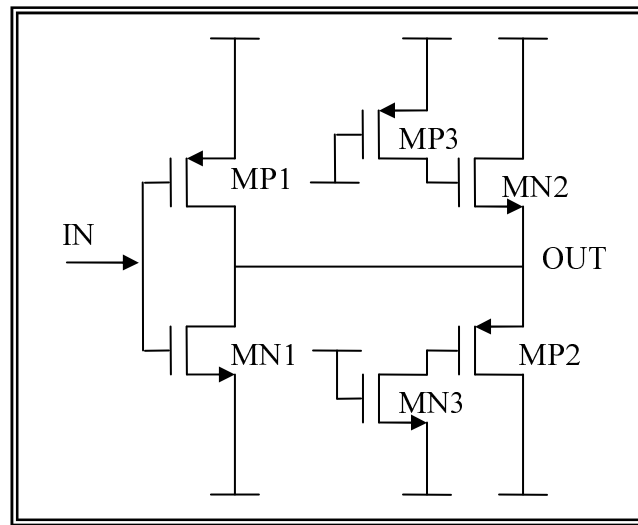INV-AIL delay cell is represented in the below figure:

81

Figure 5.13. Inverter with active inductor representation

AC behavior can be explained by the small signal analysis.
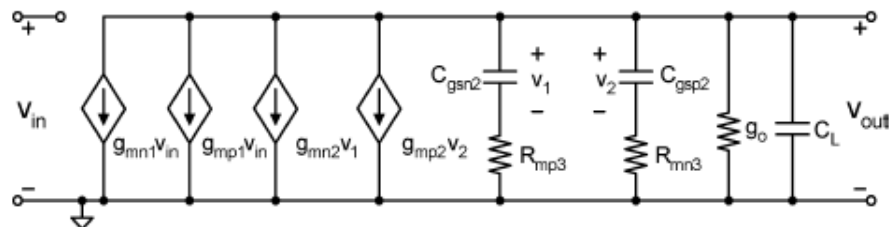
Figure 5.14. Small signal representation of the circuit of Figure 5.13 [17]

g_{mn1}, g_{mp1}, g_{mn2}, g_{mp2} are the transconductances of MN1, MP1, MN2, MP2 respectively. C_{gsn2} and C_{gsp2} are the gate capacitances of MN2 and MP2 respectively.

R_{mn3} and R_{mp3} are on resistances of transistors MN3 and MP3, which operate in triode region.

g_0 is the total channel conductance of MN1, MP1, MN2, MP2 and C_L is the load capacitance.

Small signal transfer function of the system is the following:

$$\frac{v_{out}}{v_{in}} = \frac{g_{mn1} + g_{mp1}}{g_0 + sC_L + \dfrac{g_{mn2} + sC_{gsn2}}{1 + sR_{mp3}C_{gsn2}} + \dfrac{g_{mp2} + sC_{gsp2}}{1 + sR_{mn3}C_{gsp2}}} \tag{5.1}$$

For low frequency, the gain is summarized as:

$$-\frac{g_{mn1} + g_{gmp1}}{g_0 + g_{mn2} + g_{mp2}} \approx -\frac{g_{mn1} + g_{gmp1}}{g_{mn2} + g_{mp2}} \tag{5.2}$$

g_0 can be ignored in the denominator if the length of the transistors are high enough for output resistance to be large enough. $2g_{m1} = 2g_{m2} = g_{mn1} + g_{mp1} = g_{mn2} + g_{mp2}$ where g_{m1} and g_{m2} are defined as averages of transconductances of MN1&MP1 and MN2&MP2 respectively. This condition holds because VGS of MP2 and MP1 and similarly that of MN2 and MN1 are same due to unity gain and input-output common mode voltage of VDD/2. Same transistor sizes are also selected for MP1&MP2 and MN1&MN2 to keep their transconductances same for unity gain. This is clearer in the below picture:
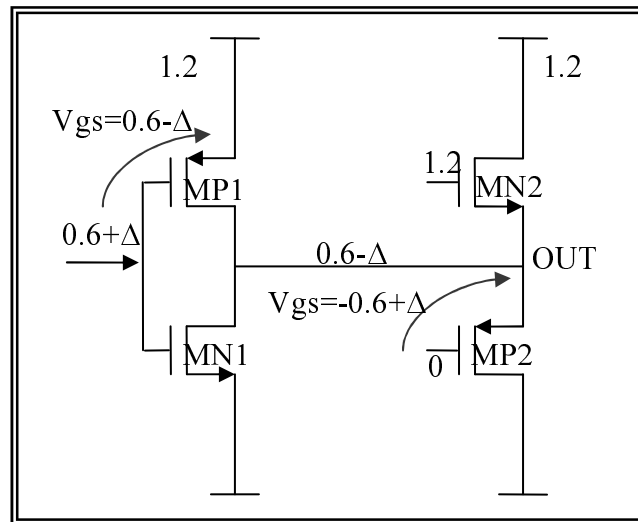

Figure 5.15. Node voltages of PMOS transistors at a random transient time

Wide channel transistors are selected to keep transconductances large. This way high bandwidth and low gain became easier to obtain. Transistors have long channels to

minimize early effect so that the g_0 term in equation 5.2 is negligible and has no effect on the cell gain.

If the condition $R_{mp3}*C_{gsn2}=R_{mn3}*C_{gsp2}=R*C_{gs}$ is satisfied, the transfer function becomes:

$$\frac{v_{out}}{v_{in}} = -\frac{\frac{2g_{m1}}{C_L}\left(s + \frac{1}{RC_{gs}}\right)}{s^2 + \left(\frac{1}{RC_{gs}} + \frac{g_0}{C_L} + \frac{2}{RC_L}\right)s + \frac{2g_{m2} + g_0}{RC_L C_{gs}}} \qquad (5.3)$$

From the above equation, the inductance imitation behavior of MN3 and MP3 can be explored. Absence of these transistors can be modeled with R→∞, where MN2 and MP2 are also in cutoff and transfer function is simplified to:

$$\frac{v_{out}}{v_{in}} = -\frac{2g_{m1}}{sC_L + g_0} \qquad (5.4)$$

The above equation is the transfer function of simple inverter.

When there is a finite resistance added by MP3 and MN3 exist and assumed to be very small i.e. R->0, the transfer function becomes:

$$\frac{v_{out}}{v_{in}} = -\frac{2g_{m1}}{sC_L + 2sC_{gs} + g_0 + g_{m2}} \qquad (5.5)$$

Since usually C_L is total gate capacitance of the next stage, which is:

$$C_{gsmn2} + C_{gsmp2} = C_L \approx 2C_{gs} \qquad (5.6)$$

Combining all assumptions and definitions together results in the following transfer function:

$$\frac{v_{out}}{v_{in}} = -\frac{\frac{g_{m1}}{C_{gs}}\left(s + \frac{1}{RC_{gs}}\right)}{s^2 + \left(\frac{2}{RC_{gs}} + \frac{g_0}{2C_{gs}}\right)s + \frac{g_{m2} + g_0}{RC_{gs}^2}}$$

$$\approx -\frac{\frac{g_{m1}}{C_{gs}}\left(s + \frac{1}{RC_{gs}}\right)}{s^2 + \frac{2}{RC_{gs}}s + \frac{g_{m2}}{RC_{gs}^2}}$$

(5.7)

There are two poles and one zero in the transfer function. This additional zero improves the bandwidth of the cell.

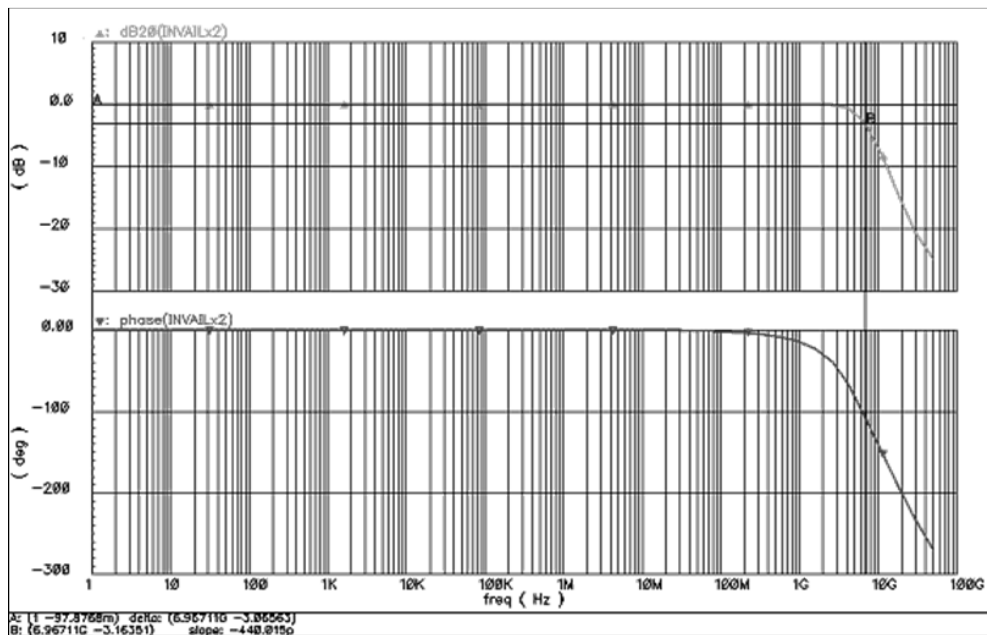AC characteristic of INV-AIL cell is shown in the below figure:

Figure 5.16. AC characteristics of one buffer composed of 2 INV-AIL cells

Delay of two inverters was measured to be around 40ps. Group delay plot of the circuitry is shown below:
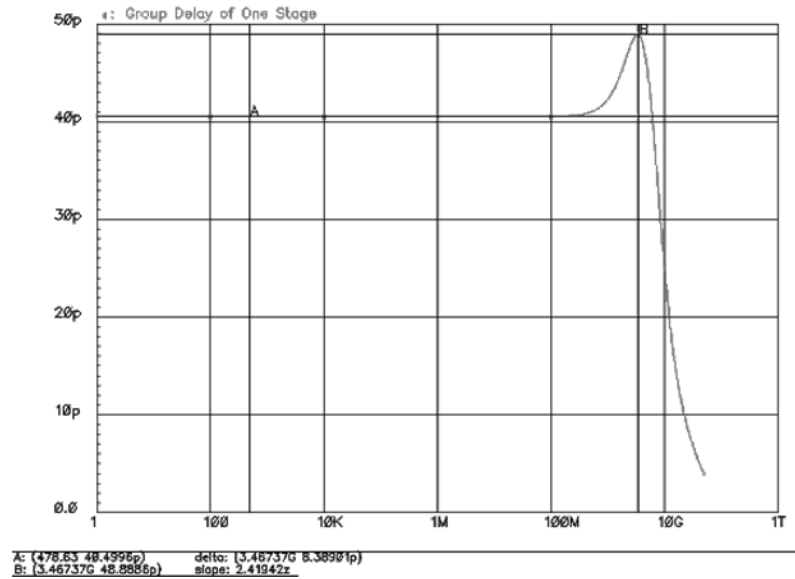
Figure 5.17. Group delay plot of one buffer composed of 2 INV-AIL cells

From the AC plots shown above, it is observed that the bandwidth of one buffer is nearly 7GHz which is far beyond the operating frequency of 1.5625GHz. AC gain is measured as nearly 0.098dB which is very close to 0dB.

As the common mode voltage of the input signal is selected to be 600mV, the transistor sizing is arranged so that the transition threshold for the INV-AIL is at 600mV.

Transient response at 3.125Gbps data rate is shown in the figure below. Square wave input is applied to the first stage and its propagation through each transport delay cell is observed. Reason behind selection of square wave is to observe behavior of cell against a waveform that is rich in terms of various frequency components.
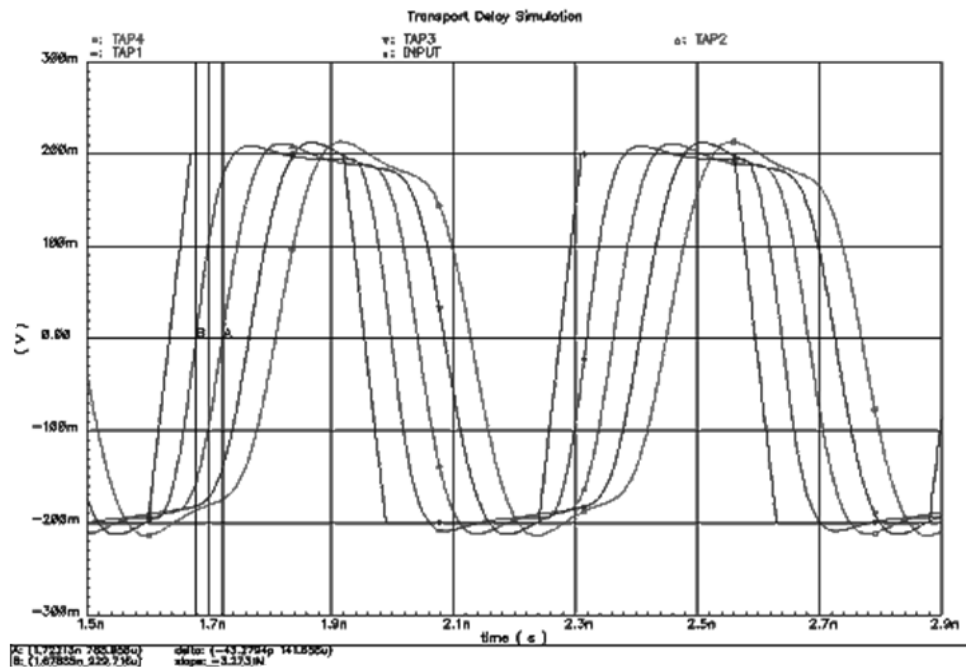
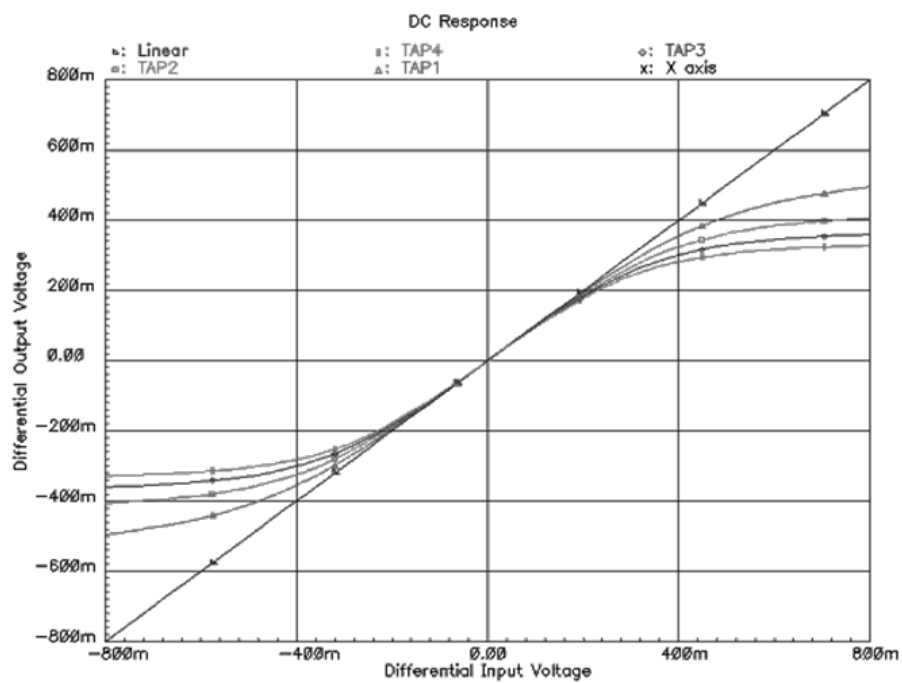Figure 5.18. Transient simulation result of cascaded inverters

Figure 5.19. Linearity of cascaded inverters

The advantage of this topology is achieving high bandwidth and low gain without use of any clock switching. Since there is no need for passive inductor usage, it helps great saving in area. As all tapped delay lines, this topology also suffers from the limitation of number of stages in cascade format. Since the linearity is sacrificed with each additional cascade stage (Figure 5.19), number of taps is limited. Since one buffer stage results in 40ps delay, if symbol spacing is required, the number of buffers should be 8. However, after cascading eight cells, the signal at the end of the last buffer becomes clipped and degraded due to bandwidth constraints. To increase delay without increasing number of stages, using smaller size transistors to obtain higher delay within one buffer could be a solution. The problem with this solution is again loss of bandwidth, since drive strengths of transistors also becomes smaller. Due to these design limitations, four taps with 40ps delay used to implement T/2 time span fractionally spaced equalizer to achieve the best possible performance.

Four taps of tapped delay line with tap spacing of 40ps, has 4.5GHz bandwidth overall and a DC gain of 0.983V/V. Each transport delay stage has bandwidth of 7GHz with 0.99V/V gain.

5.1.4. ADC (Analog to Digital Converter)

4bits (one bit being the sign bit) ADC design requires differential comparison, sample and hold involvement. In order to meet these design requirements, an ADC topology which can compare differential signals and sample and hold the input during comparison is selected. Flash ADC with differential comparator is used for this design.

Below figure shows flash ADC top level representation:
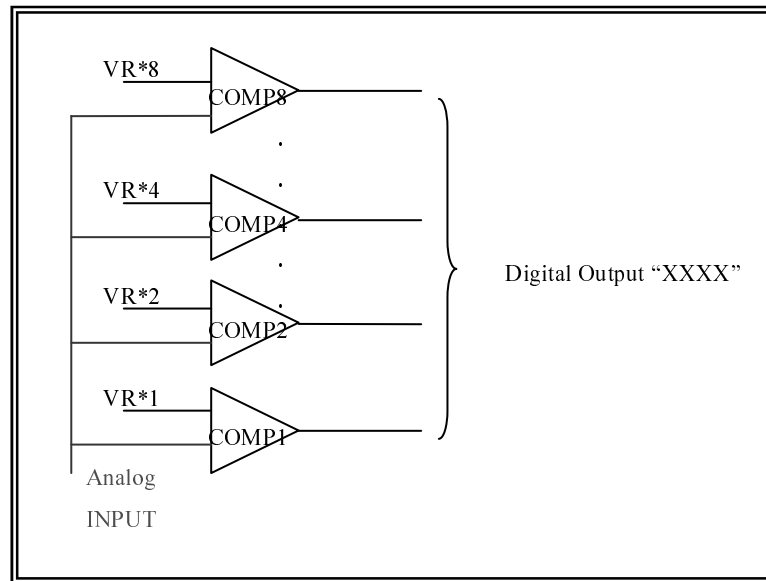
Figure 5.20. ADC top level representation

The topology used is based on the theory of sharing the input sampling, comparing and output sampling responsibilities among three building blocks.

The first block consists of two track & hold circuitries with opposite clocks to sample and hold the analog input. The second block is a simple flash ADC, to compare sampled differential analog input with the differential reference voltage. The delayed version of first stage's clock is applied to the third stage to sample the comparator output and convert it to singled ended CMOS digital signal.
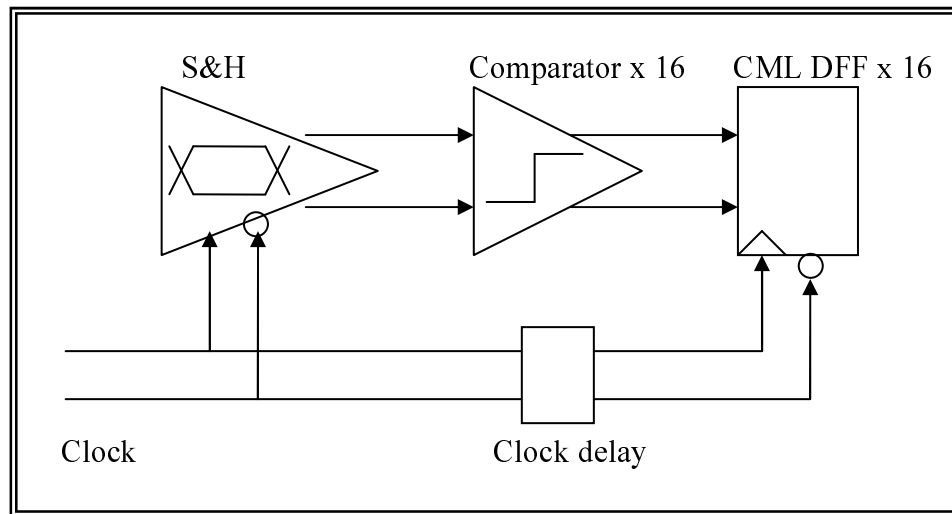
Figure 5.21. ADC schematic top level

5.1.4.1. Sample and Hold. In order to obtain sample and hold circuitry, two track and hold devices operating at non-overlapped, opposite clock signals are cascaded. Track and hold circuitry, in below figure, is a simple differential amplifier with unity gain with transmission gate at its input ports. These transmission gates prevent the signal to pass to the output during hold mode, whereas it allows transaction during track mode. The gate capacitors of the switch transistors are used as hold capacitors.

Figure 5.22. Track and hold circuitry which operates as Sample and Hold circuitry when cascaded with opposite clocks [42]

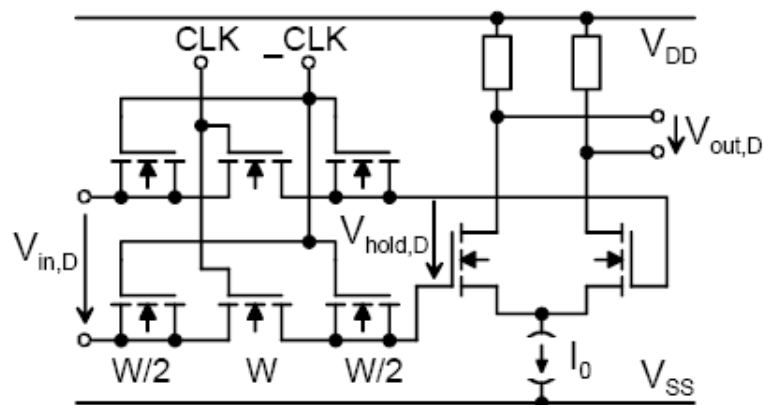Two dummy transistors are added to the drain and source of the n-channel transfer gate so that during switching to hold mode from tracking mode kickback to input can be prevented. They act like storage for drain and source charges, and they have width size of (W/2) which is half of the switching transistor width (W).

Sample&hold circuitry has 0.6V common mode voltage at the input and gives 860mV output common mode voltage at the end. Therefore, comparator following this block should have differential reference voltages with 860mV common mode.

Transient simulation output for the sample&hold circuitry for 3.125Gbps data is shown below:
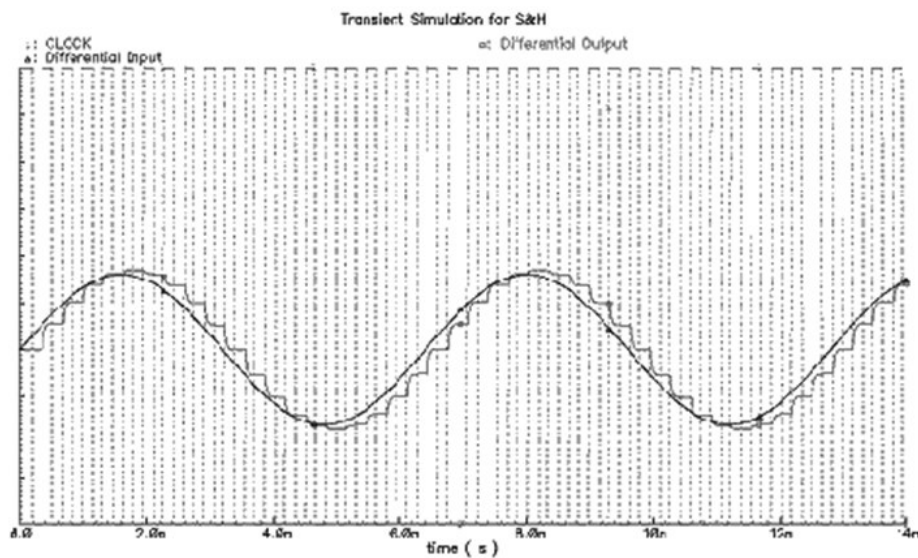

Figure 5.23. Sample and hold transient response

5.1.4.2. Comparator. Differential comparator block following the sample hold circuitry is shown below:
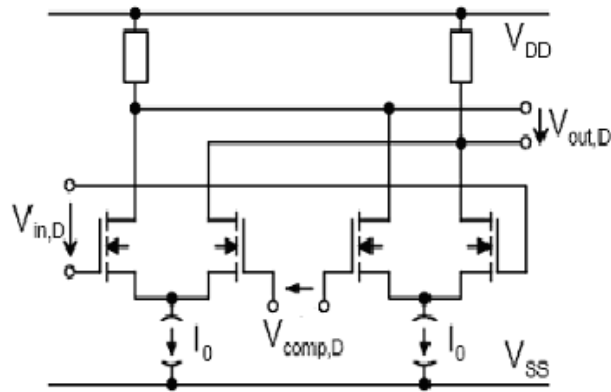
Figure 5.24. Comparator schematic representation [42]

The comparator is a simple Gilbert cell, where two amplifier stages with same gain are added to each other at the output node. $A_v[(VINP-VINM)-(VTHP-VTHN)]$ is the output function of this cell, where A_v is the gain, VINP is the positive input voltage, VINM is the negative input voltage, VTHP is the positive threshold voltage and VTHN is the negative threshold voltage. This way differential comparison can easily be implemented.

The gain and bandwidth features of the cell are summarized in the below table:

Table 5.3. Gain and bandwidth features of the comparator

Gain	Bandwidth
1.032 V/V at DC	10.68GHz -3dB frequency

The DC behavior of the comparator is shown below. Differential input voltage is swept and the voltages where the comparator output trips are observed.
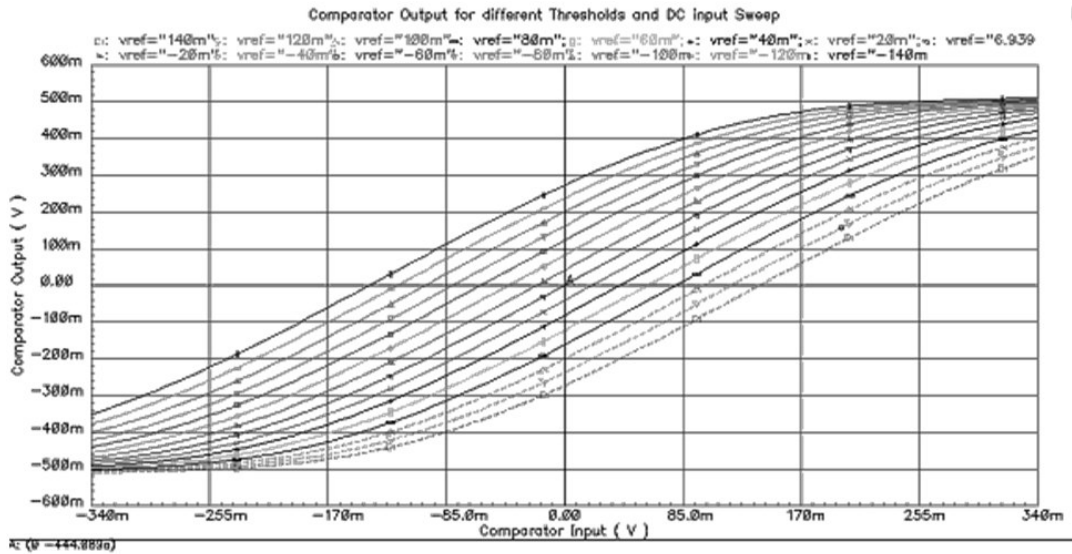
Figure 5.25. Comparator DC response for different reference voltages during input sweep

5.1.4.3. Peak Amplifier. The comparator is followed by limiting amplifier with active peaking to buffer the dc output of the comparator while amplifying the moments when the transition on comparator outputs occurs. The transistors with their gates connected to Vpeak, are capacitors which create path to ground at high frequencies and act like an open circuit during DC. So these transistors do not have any effect at DC, whereas they help cross coupled pair to sink or source current from the ground to the output resistors by creating path to ground at AC.

Figure 5.26. Limiting amplifier with peaking [42]

Limiting amplifier with active peaking consists of unity gain amplifier and cross coupled, capacitively source degenerated differential pair. Using the control voltage, Vpeak, the peaking transistors can be eliminated. If Vpeak is at VDD, the peaking transistors are on and create signal path to ground at high frequencies increasing the gain during switching.
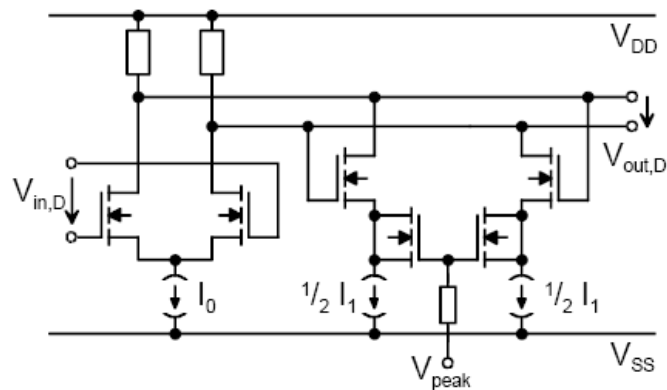
Three stages of limiting amplifier are used in cascade for higher amplification. The performance of limiting amplifier is summarized in table below:

Table 5.4. Gain and bandwidth features of the limiting amplifier with peaking

# OF STAGES	GAIN	BANDWIDTH
1	1.84 V/V at DC	8.5GHz -3dB frequency
3	6.3 V/V at DC	6.83GHz -3dB frequency

Until now comparison is held and comparison output is amplified, however the signals are still not very clean and needs one more sampling before being converted to CMOS logic levels. Sample and hold circuitry was responsible for providing the comparator with a stable analog signal to be compared with reference voltages. Limiting amplifier with peaking is responsible to amplify the transition instants in a sense like a pre-emphasis mechanism. Sampling of comparison result is the last objective to finish ADC design. This way the aim of leaving comparison independent of any clocked operation is achieved.

5.1.4.4. Decision Flip Flop. Finally, the comparator and limiting amplifier is followed by a decision flip-flop to obtain clean and synchronized output. The flip-flop consists of two CML latches with opposite clocks.
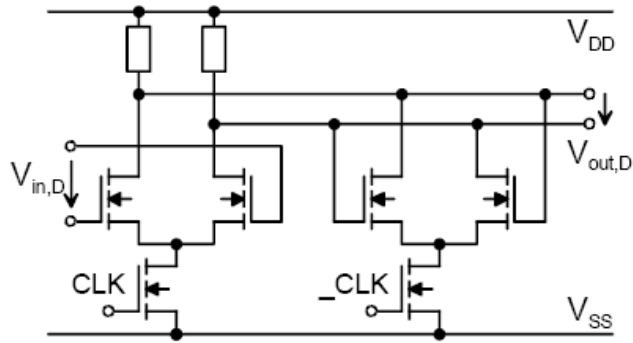
Figure 5.27. CML latch composing CML DFF [42]

Table 5.5. Gain and bandwidth features of differential DFF

Gain	Bandwidth
10.89 V/V at DC	2.32GHz -3dB frequency

5.1.4.5. Top Level ADC. Using the components until now, the worst and best conversion cases are shown below:
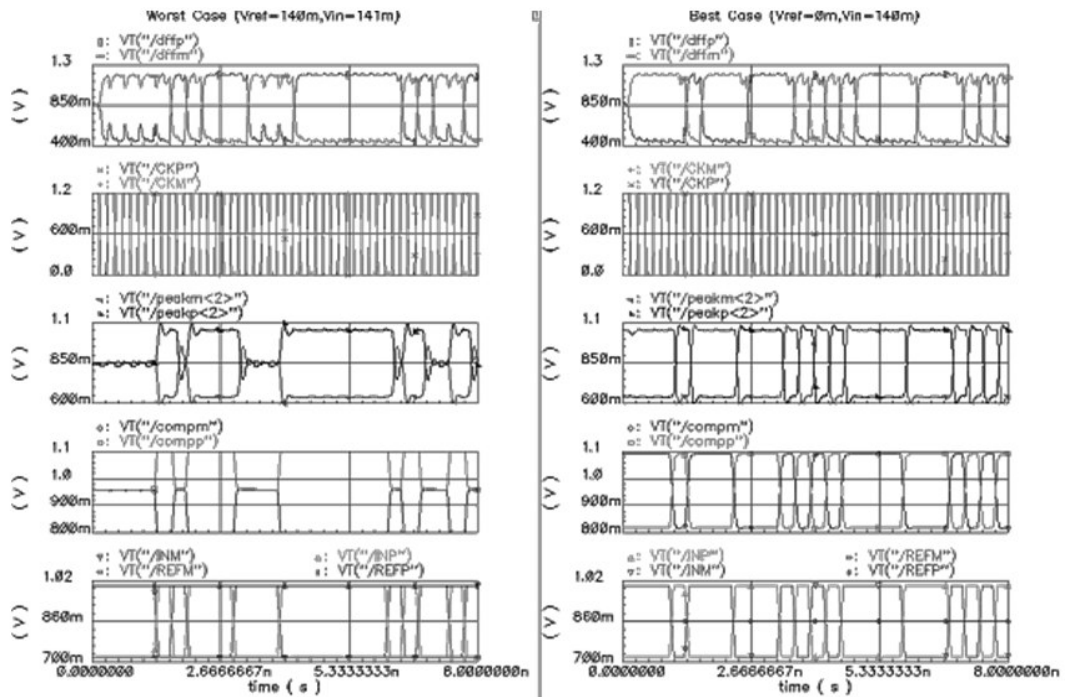

Figure 5.28. Top level ADC simulation for best and worst scenarios

The bottom trace represents the input and comparison threshold waveforms. This is the output of the sample & hold device and input of the comparator. The best case is when the differential threshold is at its minimum (0V), whereas the input differential amplitude is at its maximum (140mV). The worst case on the contrary happens when the differential input (141mV) is just above the highest differential threshold value of 140mV. Second trace is the output of the comparator. Third trace represents the output of the peak amplifier. Fourth trace components are the differential clock applied to the decision flip-flop. And finally, the top trace stands for the decision DFF output.

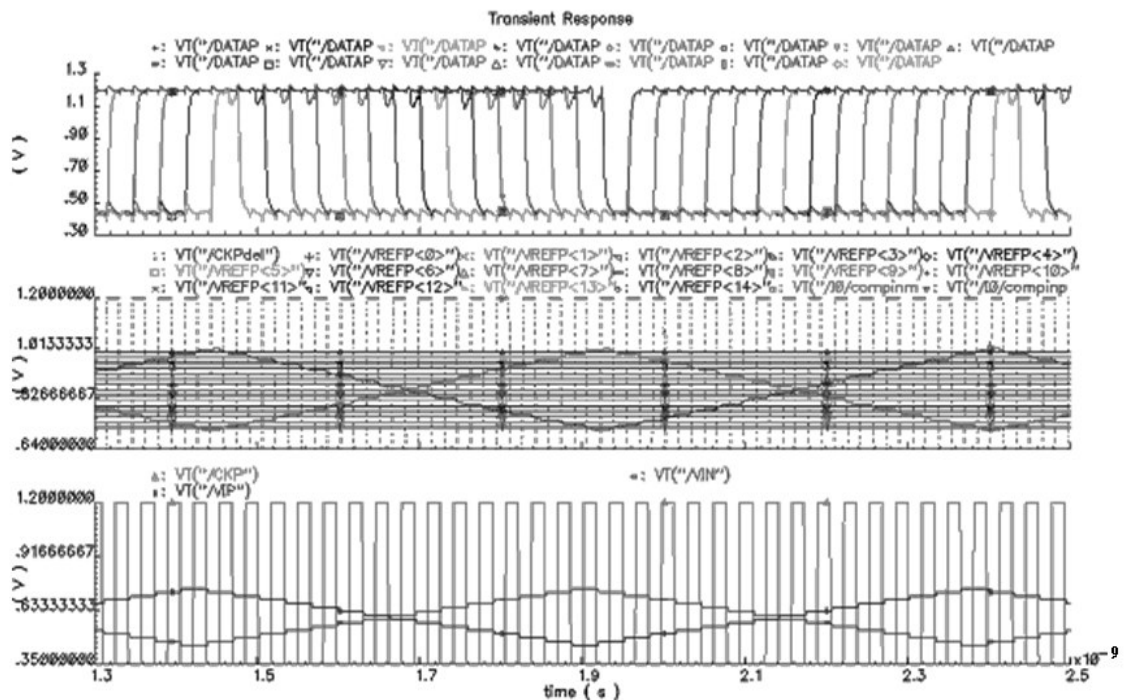Transient simulation of the top level ADC with all the bits is shown below:

Figure 5.29. Converter transient simulation. Traces defined from down to up

1st trace: Input clock vs. Input voltage

2nd trace: Output of the Sample and Hold vs. Comparator thresholds and delayed clock

3rd trace: Sequential settling of comparator outputs due to threshold crossing

5.1.5. Limiting Amplifier

Limiting amplifier is used to open the equalizer output eye in the y axis by amplifying it with constant delay for a wide range of frequency ingredients of its input signal. Its duty is mainly to give constant amplitude independent of the input, when the input is above a certain threshold. This is achieved by cascading number of high gain, high bandwidth amplifiers as in the figure below.
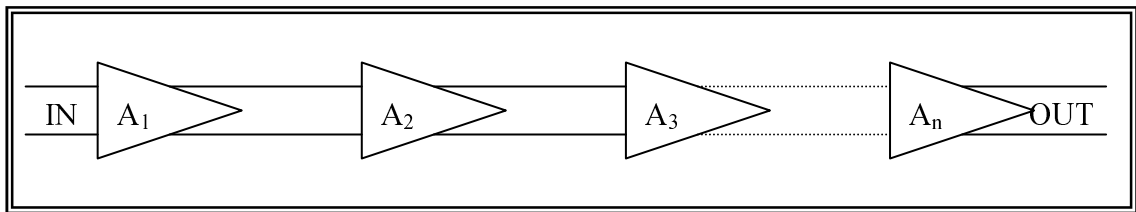
Figure 5.30. Cascading limiting amplifiers

The gain decision for each stage depends on the design requirements. For instance, for high bandwidth but low gain, number of stages can be decreased or sizes of the amplifier stages can be gradually decremented by two. For this design, since LA follows an equalizer, there is no need for high gain, whereas high bandwidth is still a requirement to avoid attenuation of equalizer output. That's why; number of matched amplifier stages is kept at two while maintaining the required gain. The total gain of an LA with matched amplifier stages is given as:

$$Av_{DC} = \prod_{i=1}^{N} Av_{DC_i} = Av_{DC_i}^{N} \tag{5.8}$$

$$BW = f_{ci} \cdot \sqrt{\sqrt[N]{2} - 1} \tag{5.9}$$

Where N is the number of stages and A_n is the gain of one stage. The active inductive peaking topology is used in the design on amplifier stages for high bandwidth. Since

embedded inductors are low quality and usually area consuming, inductive behavior is imitated using active devices as follows:
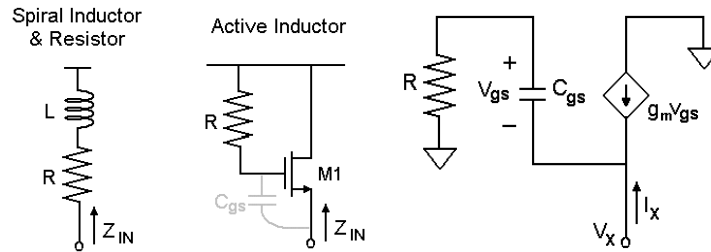
Figure 5.31. Active and passive inductor implementation

Small-signal impedance looking into the source of the active inductor is:

$$z_x = \frac{V_x}{I_x} \approx \frac{1 + sRC_{gs}}{g_m} = \frac{1}{g_m} + s\frac{RC_{gs}}{g_m} \tag{5.10}$$

The resistor is replaced by a PMOS transistor with its gate connected to GND. One stage amplifier is shown in the below figure:
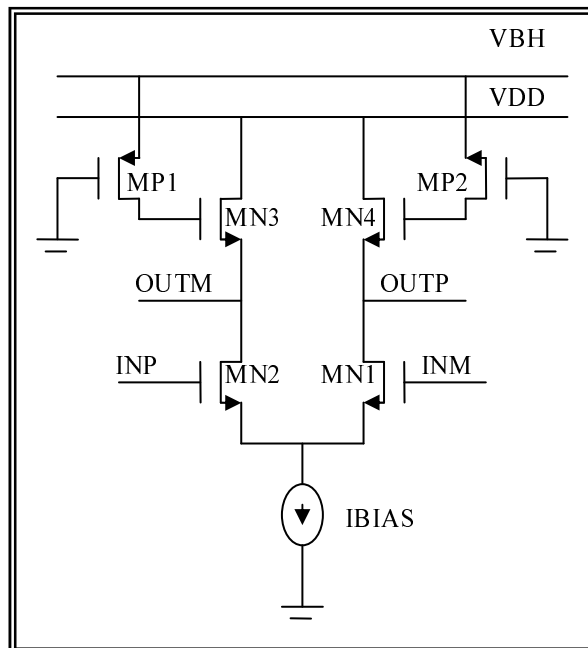
Figure 5.32. Limiting amplifier single stage transistor level implementation

MP1&MP2 are 1.8V transistors and their sources are connected to a higher voltage. This way, gates of MN3 and MN4 are at a higher voltage than 1.2V power supply and it is easier to keep them ON. This approach also provides enough headroom for bias transistor and switch transistors to operate in appropriate regions.

Below is the table for gain, bandwidth values of each amplifier stage and final limiting amplifier.

Table 5.6. Limiting amplifier gain and bandwidth features

	Stage1	Stage2	Total (calculated)	Total (measured)	Unit
Gain	2.62	2.62	6.86	6.87	V/V
3db Freq.	8.8	8.8	5.66	5.5	GHz

In order to obtain the same results using passive devices, 6nH of inductance and 1.38KOhm of resistance would be needed. The below figure shows the comparison of actual circuitry with active inductance and ideal circuitry with passive implementation:
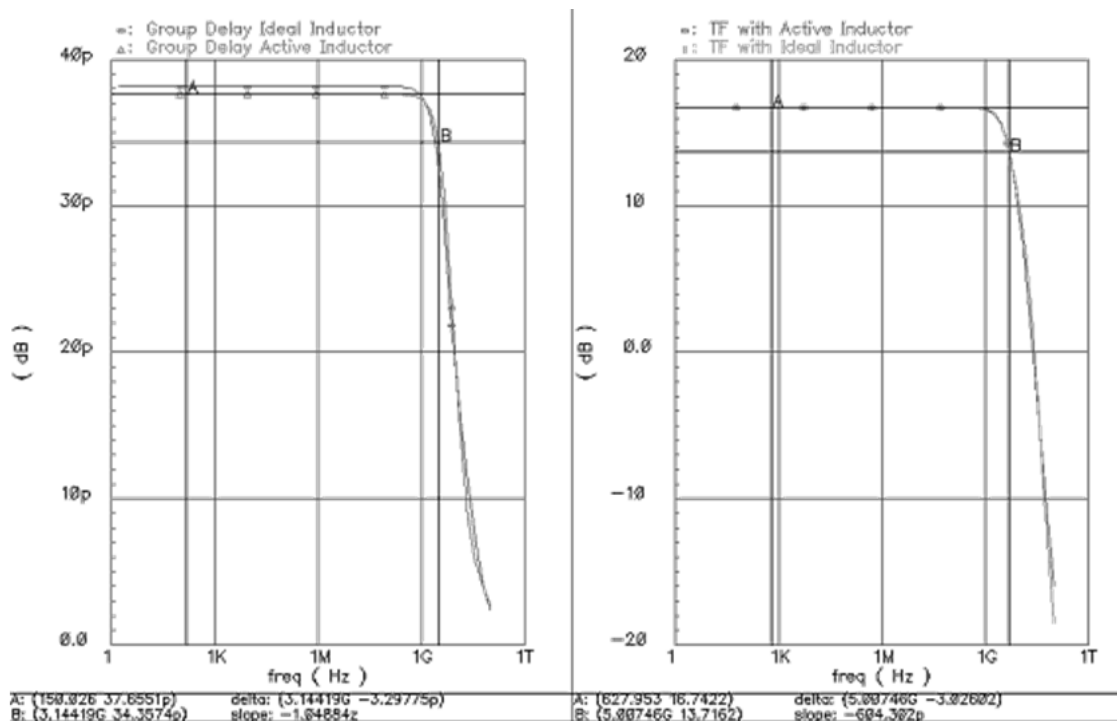
Figure 5.33. Ideal and active inductor transfer function and group delay comparison

From the above figure, it is seen that transfer functions of both circuitries look very similar, but there is a slight difference in their group delay values. This comparison shows that the same performance can be obtained by appropriate transistor sizing instead of using 6nH inductor. This way great amount of layout area and accuracy can be saved.

Each stage consumes 450uA, so the total current required is 1.35mA for the overall limiting amplifier. The common mode voltage at the input and output is 750mV. Input sensitivity of LA is kept in a medium value such as ±120mV, which means that after 120mV of equalizer output, limiting amplifier output will be fixed to ±800mV.
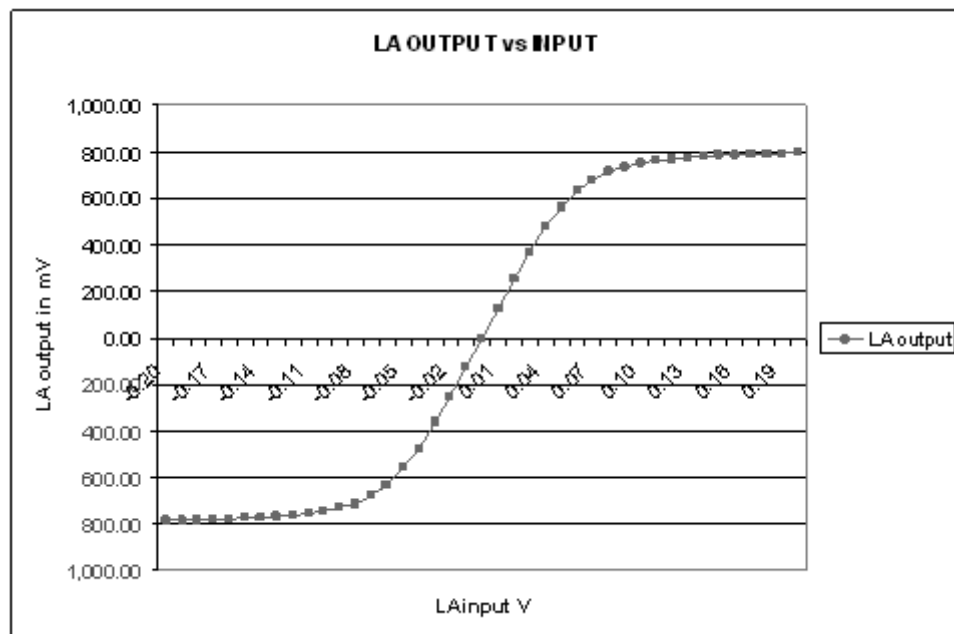
Figure 5.34. Limiting amplifier output saturation graph

The jitter generation of limiting amplifier is another critical issue in LA design. This can be controlled by group delay of the circuitry. If group delay is constant over a wide frequency range, delay spread due to different frequency components in the signal will not cause high jitter generation. The group delay figure is shown below:
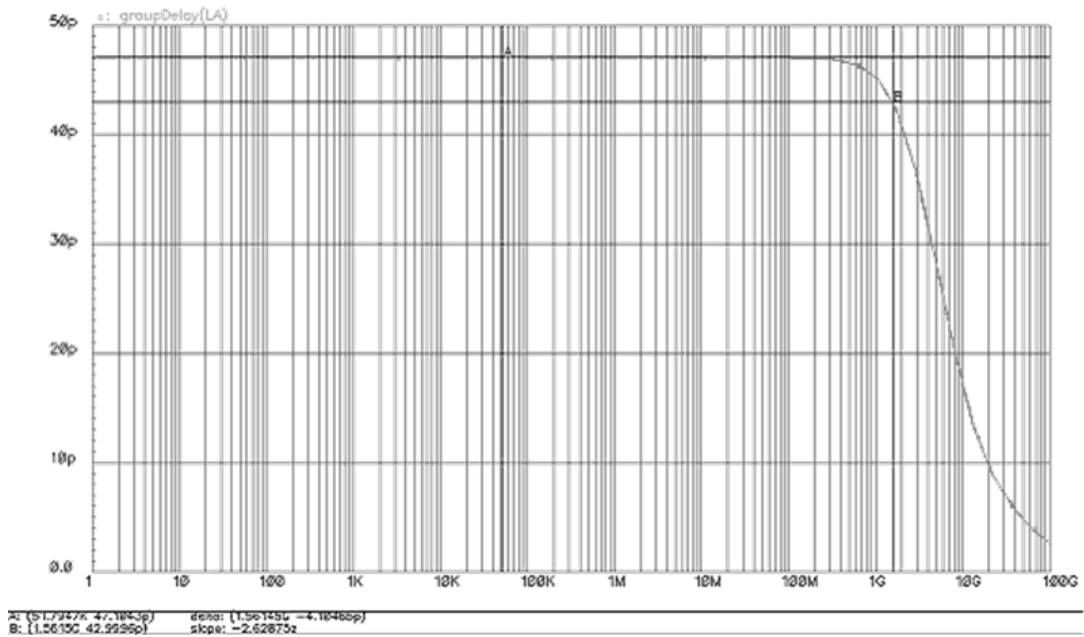
Figure 5.35. Limiting amplifier total group delay

Since a square pulse is rich in terms of different frequencies, square pulse input is used in transient simulations.
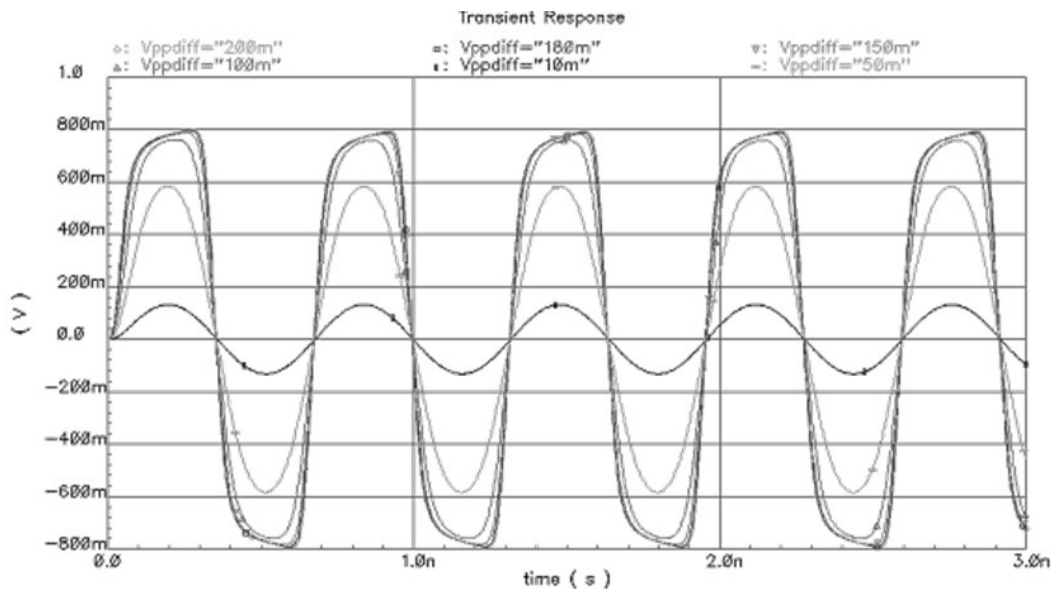
Figure 5.36. Limiting amplifier total transient response with different input amplitudes

5.2. Top Level Simulations

After completing the building blocks design, equalizer performance is observed in various transient simulations at top level. Top level equalizer simulation setup includes a random number generator as data source. Data source is followed by the 30 inch TYCO channel model. To open the data eye which is almost closed after the channel mode, equalizer is instantiated afterwards. Equalizer drives the limiting amplifier for final amplification.

Below is the representation of the top level simulation setup:
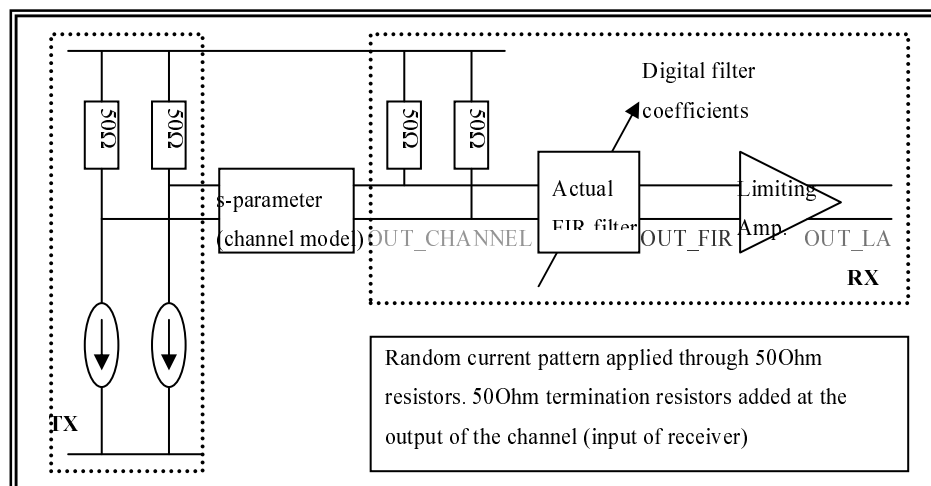
Figure 5.37. Simulation setup for top level system using the actual components

Four tap fractionally spaced equalizer with T/8 tap spacing is chosen as equalizer. The coefficients are obtained from matlab simulations and converted to 5 bit digital word, 5th bit being the sign bit. The coefficients and their digital counterparts are listed below:

Table 5.7. Tap coefficients for 4 tap,T/8 FIR equalizer in analog and digital format

	C1	C2	C3	C4
Analog	15	3	-3	-7
Digital	11111	10011	00011	00111

The signals at the input of equalizer, output of equalizer and output of limiting amplifier are shown in the first sub-window of below figure. The second sub-window is the eye diagrams of the same signals for jitter measurement.
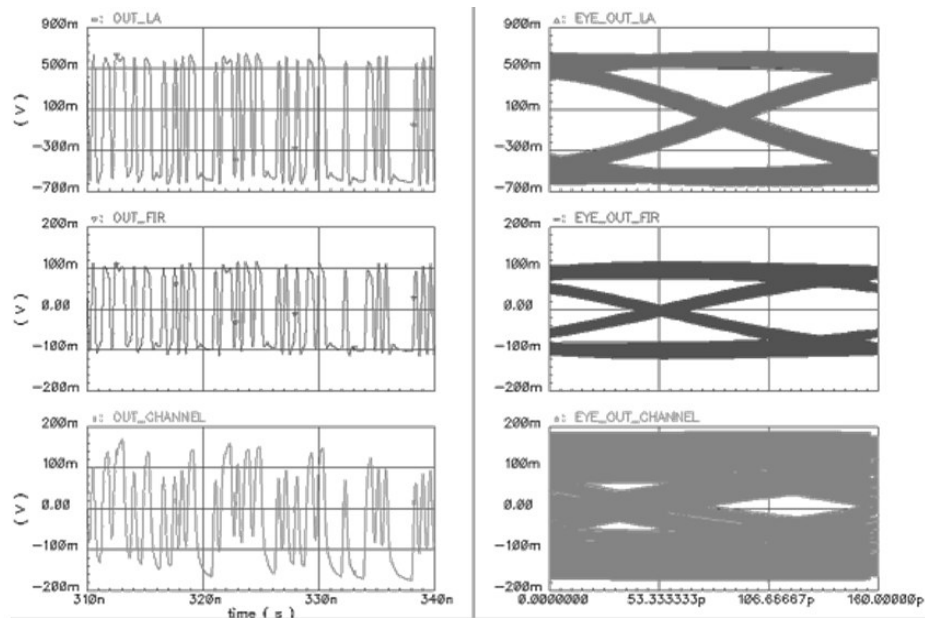
Figure 5.38. Simulation result of setup in Figure 5.36

In order to understand the effects of non-idealities in the circuit design, such as nonlinearity in transport delays, saturation of MDAC gain etc., top level results are compared with ideal simulation setup. Ideal simulation setup containing ideal adders and multiplier with the same input signal is shown below:
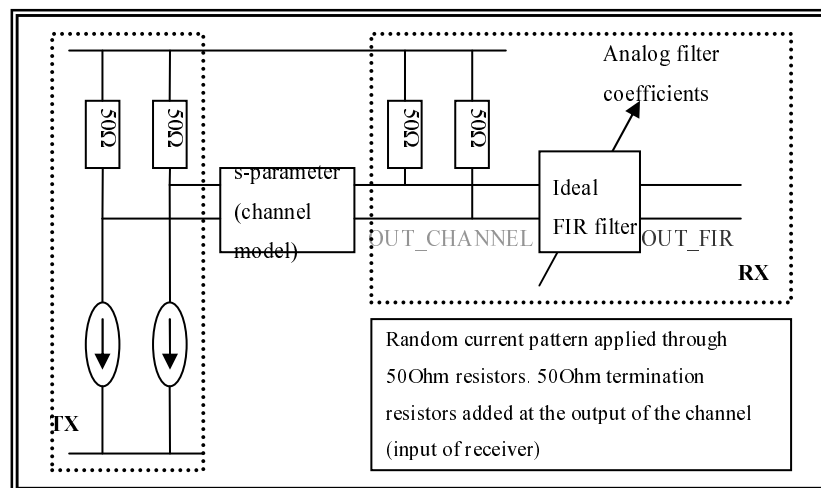
Figure 5.39. Simulation setup for top level system using the ideal components

The comparison of ideal and actual equalizers at node OUT_FIR in terms of eye diagram and the jitter measurement is shown in the below figure:
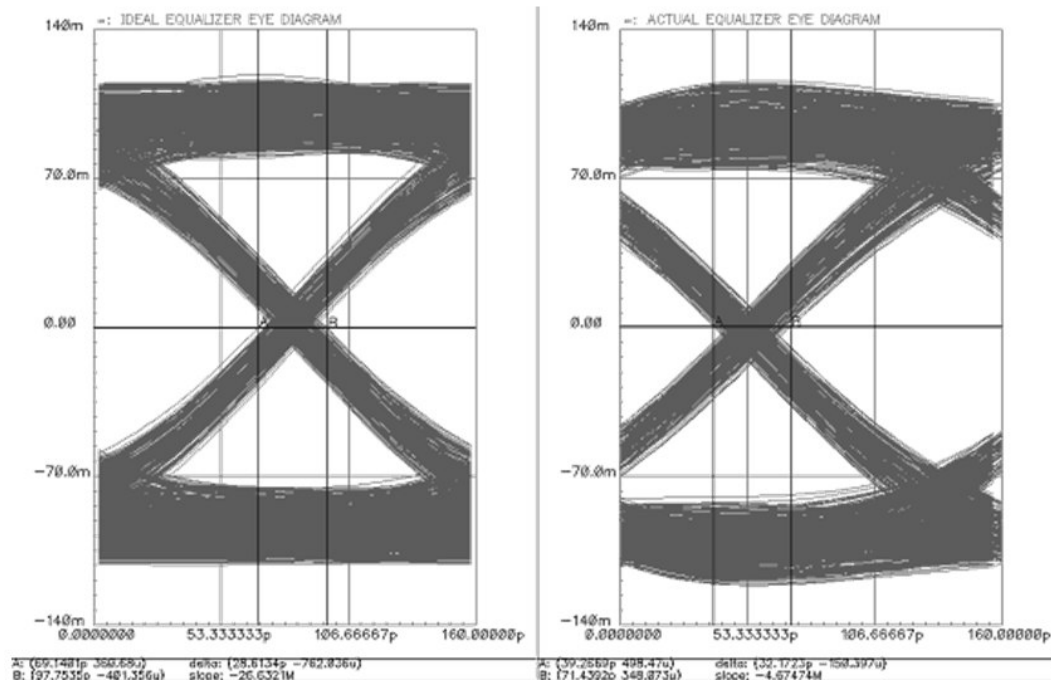
Figure 5.40. Simulation result comparison of setup in Figure 5.36 and 5.38 for 4 tap & x8 FIR equalizer

The jitter measurements on eye diagrams of both actual and ideal waveforms indicate that, the non-idealities inserted by actual MDAC and delay elements result in only 4ps of degradation in jitter result. The jitter using ideal components is measured as 28ps, whereas the one with actual components is 32ps. Current consumption at 3.125Gpbs and 1.2V power supply is 13mA.

Since MATLAB results showed that, including the same time span of T/2 (4xT/8), it is possible to obtain better results using less number of taps but wider symbol spacing. That's why another simulation was run with setting the second and the last coefficients to 0 and using only two taps with symbol spacing of T/4. Coefficients for two tap fractionally spaced equalizer with T/4 tap spacing are obtained from MATLAB simulations and converted to 5 bit digital word, 5th bit being the sign bit. The coefficients and their digital counterparts are listed below:

Table 5.8. Tap coefficients for 2 tap,T/4 FIR equalizer in analog and digital format

	C1	C2	C3	C4
Analog	15	0	-10	0
Digital	11111	10000	01010	10000

Waveforms at the output of FIR filter are shown below:

Figure 5.41. Simulation result comparison of setup in Figure 2.26 and 2.27 for 2 tap & x4 FIR equalizer
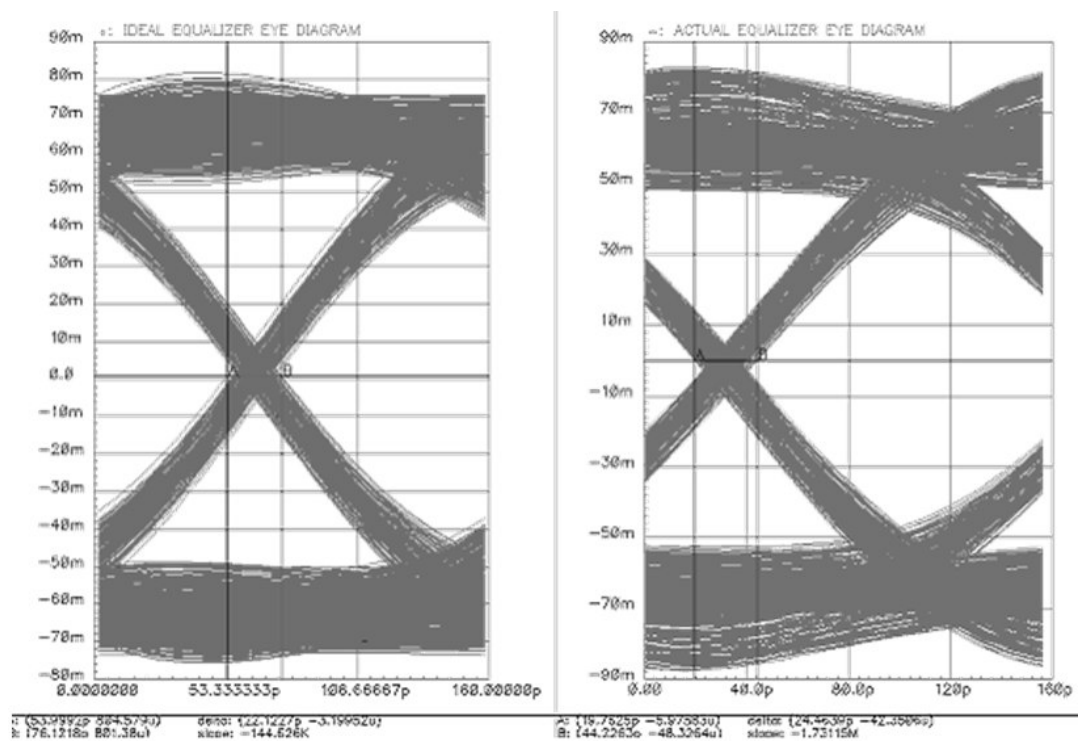
Jitter value obtained by 2 taps with T/4 spacing results in 24ps jitter with the actual setup and 22ps with the ideal setup. It is observed that ideal and actual setups do not result in very distinct jitter values.

6. CONCLUSION AND FUTURE WORK

Understanding the basics of telecommunication transceiver systems, the non-idealities that lead to data deterioration during transmission are investigated in this thesis. Specifications like XAUI, PCI Express, and CEI-6 etc. determine a timing budget for each component of the communication system to make sure reliable data transfer. To comply with these specifications, equalizers are used to reduce jitter on the data path removing the ISI. This way the flip flops sampling the data at the receiver end will have enough time to sample the data with the minimum amount of setup-hold violations.

In this thesis, basic communication system, channel non-idealities that lead to inter-symbol interference and equalizers as a solution to ISI were represented. Equalizers were investigated according to their locations on the data path (RX side or TX side), their implementations (active or passive, discrete or continuous time etc.). Linear transversal equalizers are implemented as a specific example to equalization. MATLAB Simulink tool is used to understand basics of equalization. In order to obtain the coefficients for LTE, different adaptation algorithms are implemented in Simulink after providing brief theoretical derivations of the adaptation algorithms. After extensive simulations, some comparison results regarding to different adaptation algorithms, cost functions and equalizer properties of linear transversal equalizers (such as number of taps, tap spacing etc.) are obtained. Some results to these comparisons are better convergence of RLS algorithm over LMS algorithm, irregular superiority of FSE over SSE with the change in number of taps and tap spacing, and finally smaller error rates when MSE cost function is used instead of CMA counterpart but less complexity since CMA is a blind algorithm offering the possibility of equalization without complexity introduced by training.

Finally, STMicroelectronics CMOS 65nm technology is used to implement transistor level LTE with the coefficients obtained from MATLAB. Spectre tool is used for verification in Cadence environment. Building blocks of the equalizer are also exposed to dense simulations and the best possible equalizer with the less design limitations is selected. The number of taps in CMOS implementation was limited due to linearity

degradation and bandwidth limitation of cascading the continuous time transport delays. Other limitation for number of taps is the bandwidth limitation of MDAC stages. Since the addition of tap weights after being multiplied by tap samples, is done in current mode, all MDAC stages drive the same output resistance to convert the total current into voltage. The more MDAC stages are needed, the more capacitance each stage sees at its output. All these effects limited the number of taps to 4 for this design. The limitation to the delay of each continuous time transport delay cells is due to bandwidth-delay trade off. Increasing the delay of one stage required increase in capacitance and resistance of the components resulting in lower 3dB frequency. That's why, 40ps (T/8) fractionally spaced equalizer with 4 taps is simulated with the data rate of 3.125Gbps. The almost closed eye at the input of the equalizer and 24ps of jitter at the output of the equalizer are observed. The total current consumption is only 13 mA at 3.125Gbps with 1.2V power supply.

This work is differentiated from other equalizer examples by its implementation simplicity, low power consumption, jitter performance, ability to operate at low power supply of 1.2V at quite high data rate of 3.125Gbps. Its implementation simplicity mainly stems from its clock independent continuous time transport delay usage as taps. This way any need for clock data recovery is avoided. Another advantage is using inductance imitation techniques for bandwidth enhancement instead of passive inductor usage. This way, significant area saving is achieved. This equalizer is also one of the rare designs in literature implemented in 65 nanometer CMOS technology.

Further work might involve verifying the system with post layout extraction and fabrication. Even though all the simulation results in this thesis are obtained from the schematic, all the nodes were loaded with estimated parasitic capacitors using the special component "pcapacitor" in "analogLib". This way, surprises after layout are tried to be avoided.

For this thesis, only fixed coefficient equalizer is implemented in Cadence environment. As a further investigation, adaptive algorithms might be implemented using Verilog or VHDL languages and adaptive equalization might be obtained if needed.

REFERENCES

1. Lucky, W. R., "A Survey of the Communication Theory Literature: 1968-1973", *IEEE Trans. on Information Theory*, Vol. 19, No. 6, pp. 725-739, November 1973.

2. Hui, P., H.–J. Park, "High Speed Transceivers", *IEEE ISSCC Digest of Technical Papers*, 3 February-7 February 2008, pp. 96-97, San Francisco, CA, 2008.

3. Le, M. Q., P. J. Hurst and J. P. Keane, "An Adaptive Analog Noise-Predictive Decision-Feedback Equalizer", *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 2, pp. 105-113, February 2002.

4. Le, M. Q., P. J. Hurst and K. C. Dyer, "An Analog DFE for Disk Drives Using A Mixed-Signal Integrator", *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 5, pp. 592-598, May 1999.

5. Hartman, G. P., K. W. Martin, and A. Mclaren, "Continuous Time Adaptive-Analog Coaxial Cable Equalizer in 0.5 um CMOS", *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 1999.*, July 1999, Vol. 2, pp. 97-100, Orlando, FL, 1999.

6. Baker, A. J., "An Adaptive Cable Equalizer for Serial Digital Video Rates to 400 Mb/s", *IEEE ISSCC Digest of Technical Papers*, 08 February-10 February 1996, pp. 174-175, New York, 1996.

7. Shakiba, M.H., "A 2.5 Gb/s Adaptive Cable Equalizer", *IEEE ISSCC Digest of Technical Papers*, 15 February-17 February 1999, pp. 396-397, San Fransisco, CA, 1999.

8. Choi, J. -S., M.-S. Hwang, and D.-K. Jeong, "A 0.18-um CMOS 3.5-Gb/s Continuous-Time Adaptive Cable Equalizer Using Enhanced Low-Frequency Gain

Control Method", *IEEE Journal of Solid-State Circuits*, Vol. 39, No. 3, pp. 419-425, March 2004.

9. Zhang, G. E. and M. M. Green, "A 10 Gb/s BiCMOS Adaptive Cable Equalizer", *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 11, pp. 2132- 2140, November 2005.

10. Gondi, S., J. Lee, D. Takeuchi and B. Razavi, "A 10Gb/s CMOS Adaptive Equalizer for Backplane Applications", *IEEE ISSCC Digest of Technical Papers*, 10 February 2005, pp. 328-329, San Fransisco, CA, 2005.

11. Gondi, S., B. Razavi, "Equalization and Clock and Data Recovery Techniques for 10-Gb/s CMOS Serial-Link Receivers", *IEEE Journal of Solid-State Circuits*, Vol. 42, No. 9, pp. 1999-2011, September 2005.

12. Reynolds, S., et al., "A 7-tap Transverse Analog-FIR Filter in a 0.13um CMOS for Equalization of 10 Gb/s Fiber-Optic Data Systems", *IEEE ISSCC Digest of Technical Papers* , February 2005, pp. 330-331, San Fransisco, CA, 2005.

13. Hazneci, A. and S. P. Voinigescu, "9-Gb/s, 7-Tap Transversal Filter in 0.18μm SiGe BiCMOS for Backplane Equalization", *IEEE Compound Semiconductor Integrated Circuit Symposium*, 24 October- 27 October 2004, pp. 101-104, 2004.

14. Lin, X., S. Saw and J. Liu, "A CMOS 0.25-um Continuous-Time FIR Filter with 125 ps per Tap Delay as a Fractionally Spaced Receiver Equalizer for 1-Gb/s Data", *IEEE Journal of Solid State Circuits*, Vol. 40, No. 3, pp. 593-602, March 2005.

15. Guilar, N. J., F.-K. Lau, P. J. Hurst and S. H. Lewis, "A Passive Switched-Capacitor Finite-Impulse-Response Equalizer", *IEEE Journal of Solid State Circuits*, Vol. 42, No. 2, pp. 400-409, March 2007.

16. Sun, R., J. Park, F. O'Mahony and C. P. Yue, C. P., "A Low-Power, 20-Gb/s Continuous-Time Adaptive Passive Equalizer", *IEEE International Symposium on Circuits and Systems*, 2005, 23 May-26 May 2005, Vol. 2, pp. 920-923, 2005.

17. Lin, X., J. Liu, H. Lee and H. Liu, "A 2.5- to 3.5-Gb/s Adaptive FIR Equalizer with Continuous-Time Wide-Bandwidth Delay Line in 0.25-μm CMOS", *IEEE Journal of Solid State Circuits*, Vol. 41, No. 8, pp. 1908- 1918, August 2006.

18. Fayed, A. A. and M. Ismail, "A Low-Voltage Low-Power CMOS Analog Adaptive Equalizer for UTP-5 Cables", *IEEE Transactions on Circuits and Systems I: Regular Papers, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 55, No. 2, pp. 480-495, March 2008.

19. Li, L. and M. Green, "An 11.75-Gb/s Combined Decision Feedback Equalizer and Clock Data Recovery Circuit in 0.18-μm CMOS", *33rd European Solid State Circuits Conference, 2007. ESSCIRC*, 11 September – 13 September, pp. 508-511, Munich, 2007.

20. Tomita, Y., M. Kibune, J. Ogawa, and W. W. Walker, "A 10-Gb/s Receiver With Series Equalizer and On-Chip ISI Monitor in 0.11-um CMOS", *IEEE Journal of Solid State Circuits*, Vol. 40, No. 4, pp. 986- 993, April 2007.

21. Bulzacchelli, J. F. et al, "A 10-Gb/s 5-Tap DFE/4-Tap FFE Transceiver in 90-nm CMOS Technology", *IEEE Journal of Solid State Circuits*, Vol. 41, No. 12, pp. 2885-2900, December 2007.

22. El-Fattah, A. A. A. et al, "Equalizer Implementation for 10 Gbps Serial Data Link in 90nm CMOS Technology", *International Conference on Microelectronics, 2007*, 29 December-31 December, pp. 453-456, Cairo, 2007.

23. Liu, J., X. Lin, "Equalization in High-Speed Communication Systems", *IEEE Circuits and Systems Magazine*, Vol. 4, No. 2, pp. 4-17, 2004.

24. Intel: *Crosstalk, Overview and Models,* http://download.intel.com/education/higher-ed/signal/ELCT762/class19_Crosstalk_overview.ppt.

25. Heyfitch, V., "Challenges in Transmission Line Modeling at Multi-gigabit Data Rates.", *International Conference on Computational Science*, 06 June-09 June 2004, pp. 1004-1011, Krakow, Poland, 2004.

26. Poulton, L and W. Dally, "Transmitter Equalization for 4Gb/s Signaling". *Proc. Hot Interconnects '96*, 15 August-17 August 1996, pp. 29-39, Stanford University, 1996.

27. Foley, D. and M. Flynn, "A low-power 8-PAM Serial Transceiver in 0.5-μm Digital CMOS", *IEEE Journal of Solid State Circuits*, Vol. 37, No. 3, pp. 310-316, March 2002.

28. Maxim Integrated Products,"Designing a Simple, Small, Wide-Band and Low-Power Equalizer for FR4 Copper Links", *Design Con.* Santa Clara, CA, 2003.

29. Maxim Integrated Products," MAX3800: 3.2 Gbps Adaptive Equalizer and Cable Driver", *Design Con.* Santa Clara, CA, 2001.

30. Wong, C., J. Rudell, G. Uehara, and P. Gray, "A 50 MHz Eight-Tap Adaptive Equalizer for Partial-Response Channels", *IEEE Journal of Solid State Circuits*, Vol. 30, No. 3, pp. 228-234, March 1995.

31. Staszewski, R. M., "A 550-Msample/s 8-Tap FIR Digital Filter for Magnetic Recording Read Channels", *IEEE Journal of Solid State Circuits*, Vol. 35, No. 8, pp. 1205-1210, August 2000.

32. Brown, J.E.C., P. J. Hurst, L. Der, "A 35 Mb/s Mixed-Signal Decision-Feedback Equalizer for Disk Drives in 2-um CMOS", *IEEE Journal of Solid State Circuits*, Vol. 31, No. 9, pp. 1258–1266, September 1996.

33. Kiriaki, S. et al, "A 160-MHz Analog Equalizer for Magnetic Disk Read Channels", *IEEE Journal of Solid State Circuits*, Vol. 32, No. 11, pp. 1839 - 1850, November 1997.

34. Guilar, N. J., P.-K. Lau, P. J. Hurst and S. H. Lewis, "A 200 Ms/s Passive Switched-Capacitor FIR Equalizer Using a Time-Interleaved Topology". *Proceedings of the IEEE Custom Integrated Circuits Conference, 2005*, 18 September – 21 September 2005, pp. 633-636, 2005.

35. Morton, J. M., *Adaptive Equalization for Indoor Wireless Channels*, M. S. Thesis, Virginia Polytechnic Institute and State University, 1998.

36. Proakis, J., *Digital Communications* (4 ed.), McGraw-Hill, New York, 1989.

37. Bellanger, M. G., *Adaptive Digital Filters,* Marcel Dekker Inc., New York, 2001.

38. Mathworks: *Communications Toolbox*, http://www.mathworks.com, 2008.

39. Haykin, S., *Adaptive Filter Theory*, Prentic Hall, 1986.

40. Farhang-Boroujeny, B., *Adaptive Filters Theory and Applications*, John Wiley&Sons Ltd., Chichester, 1998.

41. Tato, L. M., H.C. Miranda, "Simulation of a RLS Adaptive Equalizer using Simulink", *Il Congresı de Usuarios de Matlab*, September 1996, Spain, 1996.

42. Grözing, M., M. Berroth, E. Gerhardt, B. Franz, W. Templ, "High-Speed ADC Building Blocks in 90 nm CMOS" *Fourth Joint Symposium on Opto- and Microelectronic Devices and Circuits (SODC 2006)* 02 September-08 September 2006, Duisburg, 2006.